```
*** ** **                                                                                                  **** ******** *  ****!* * ***
 E  B  L                                                                                                  L N    S N    S  A    S S  S
 R  E  E                                                                                                  I U    E U    O  D    Y E  E
 R  G  V                                                                                                  N M    Q M    U  D    L Q  G
 O  I  E                                                                                                  E B    U B    R  R    L    M
 R  N  L                                                                                                  * E    E E    C  E    A E  E
 S  S  *                                                                                                  * R    N R    E  S    B R  N
 *  *  *                                                                                                  * *    C *    *  S    L R  T
 *  *  *                                                                                                  * *    E *    *  *    E *  *
*** ** **                                                                                                  **** ******** *  ****!* * ***
```

```
*  COMMENT                                                5 / 10 / 72*     1 00001000                          1
*                                                                    *     2 00002000                          1
*                          M I N I / T E X T                         *     3 00003000                          1
*                            M A R K  V.29                           *     4 00004000                          1
*                                                                    *     5 00005000                          1
*GENERAL.                                                            *     6 00006000                          1
*      MINI/TEXT IS A MULTI-USER ALGOL PROGRAM FOR THE B-5700 DESIGNED *     7 00007000                          1
*      TO ALLOW USERS TO CREATE AND MAINTAIN PROGRAM AND DATA FILES ON *     8 00008000                          1
*      THE DISK FROM REMOTE CONSOLES. IN ADDITION, A USER MAY COMPILE  *     9 00009000                          1
*      AND RUN SOURCE LANGUAGE PROGRAMS AND EXECUTE LIBRARY PROGRAMS   *    10 00010000                          1
*      AND HAVE THE RESULTS LISTED ON A REMOTE.                        *    11 00011000                          1
*                                                                    *    12 00012000                          1
*      THESE FEATURES ARE IMPLEMENTED BY PROVIDING FILE MAINTENANCE    *    13 00013000                          1
*      FUNCTIONS, SIMULATED KEYPUNCH CAPABILITIES AND VARIOUS OTHER    *    14 00014000                          1
*      CONTROL STATEMENTS.                                             *    15 00015000                          1
*                                                                    *    16 00016000                          1
*      THE STATEMENTS USED TO COMMUNICATE WITH MINI/TEXT ARE DEFINED   *    17 00017000                          1
*      AND DESCRIBED IN THIS DOCUMENT.                                 *    18 00018000                          1
*                                                                    *    19 00019000                          1
*CONVENTIONS USED IN THE DESCRIPTION OF MINI/TEXT.                    *    20 00020000                          1
*      THE SYNTAX OF THE STATEMENTS USED TO COMMUNICATE WITH MINI/TEXT *    21 00021000                          1
*      IS DESCRIBED THROUGH THE USE OF METALINGUISTIC SYMBOLS. THESE   *    22 00022000                          1
*      SYMBOLS HAVE THE FOLLOWING MEANINGS:                            *    23 00023000                          1
*                                                                    *    24 00024000                          1
*        A. < > LEFT AND RIGHT BROKEN BRACKETS ARE USED TO CONTAIN ONE *    25 00025000                          1
*               OR MORE CHARACTERS REPRESENTING A METALINGUISTIC VARIABLE*   26 00026000                          1
*               WHOSE VALUE IS GIVEN BY A METALINGUISTIC FORMULA.      *    27 00027000                          1
*                                                                    *    28 00028000                          1
*        B. ::= THIS SYMBOL MEANS "IS DEFINED AS," AND SEPARATES THE  *    29 00029000                          1
*               METALINGUISTIC VARIABLE ON THE LEFT OF THE FORMULA FROM *    30 00030000                          1
*               ITS DEFINITION ON THE RIGHT.                           *    31 00031000                          1
*                                                                    *    32 00032000                          1
*        C.  /  THIS SYMBOL MEANS "OR." IT IS USED TO SEPARATE MULTIPLE *    33 00033000                          1
*               DEFINITIONS OF A METALINGUISTIC VARIABLE.              *    34 00034000                          1
*                                                                    *    35 00035000                          1
*        D. [ ] BRACKETS ARE USED TO ENCLOSE METALINGUISTIC VARIABLES  *    36 00036000                          1
*               WHICH ARE DEFINED BY THE MEANING OF THE ENGLISH-LANGUAGE *   37 00037000                          1
*               EXPRESSION CONTAINED WITHIN THE BRACKETS.              *    38 00038000                          1
*                                                                    *    39 00039000                          1
*      ANY MARK OR SYMBOL IN A METALINGUISTIC FORMULA WHICH IS NOT ONE  *
```

```
*       OF THE ABOVE METALINGUISTIC SYMBOLS DENOTES ITSELF, THE          *    40 00040000          1
*       JUXTAPOSITION OF METALINGUISTIC VARIABLES AND/OR SYMBOLS IN A    *    41 00041000          1
*       METALINGUISTIC FORMULA DENOTES JUXTAPOSITION OF THESE ELEMENTS   *    42 00042000          1
*       IN THE CONSTRUCT INDICATED.                                      *    43 00043000          1
*                                                                        *    44 00044000          1
*ENTER STATEMENT.                                                       *    45 00045000          1
*SYNTAX.                                                                *    46 00046000          1
*       THE SYNTAX FOR <ENTER STATEMENT> IS AS FOLLOWS:                  *    47 00047000          1
*                                                                        *    48 00048000          1
*         <ENTER STATEMENT> ::= * <ENTER VERB> <ENTER PARAMETERS>        *    49 00049000          1
*         <ENTER VERB> ::= ENTER / MAKE / CREATE                         *    50 00049100          1
*         <ENTER PARAMETERS> ::= <ENTER TYPE> <FILE NAME>                *    51 00050000          1
*         <ENTER TYPE> ::= PROGRAM / DATA / PATCHES / [EMPTY]            *    52 00051000          1
*         <FILE NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]           *    53 00052000          1
*                         [7 CHARACTER OR LESS SUFFIX]                   *    54 00053000          1
*                                                                        *    55 00054000          1
*SEMANTICS.                                                             *    56 00055000          1
*       THE ENTER STATEMENT ENABLES A REMOTE USER TO ESTABLISH A DATA    *    57 00056000          1
*       FILE, SYMBOLIC PROGRAM OR SYMBOLIC PROGRAM PATCH FILE ON DISK.   *    58 00057000          1
*                                                                        *    59 00058000          1
*       AN ENTER TYPE SPECIFIES THE KIND OF INFORMATION TO BE ENTERED    *    60 00059000          1
*       AND DETERMINES THE BLOCKING SPECIFICATIONS USED WITH THE FILE ON *    61 00060000          1
*       DISK. IF AN ENTER TYPE IS NOT SPECIFIED "PROGRAM" IS ASSUMED.    *    62 00061000          1
*                                                                        *    63 00062000          1
*       INPUT MAY BE FROM EITHER PAPER TAPE OR THE KEYBOARD AND MAY      *    64 00063000          1
*       CONSIST OF AS MANY AS 80 CHARACTERS. THE CARRIAGE RETURN AND     *    65 00064000          1
*       LINE FEED CHARACTERS ARE NOT TRANSMITTED TO MINI/TEXT SO A       *    66 00065000          1
*       USER MAY ALLOW A SINGLE LINE OF TEXT TO EXTEND OVER SEVERAL      *    67 00066000          1
*       PHYSICAL LINES IF HE DESIRES. WHEN ENTERING "PROGRAM" OR PROGRAM *    68 00067000          1
*       "PATCHES" ONLY THE FIRST 72 CHARACTERS OF AN INPUT LINE ARE      *    69 00068000          1
*       PLACED IN THE PROGRAM RECORD.                                    *    70 00069000          1
*                                                                        *    71 00070000          1
*       SEQUENCE NUMBERS, WITH THE EXCEPTION OF ENTERING PATCHES, ARE    *    72 00071000          1
*       AUTOMATICALLY PROVIDED. THESE NUMBERS ARE STORED IN THE          *    73 00072000          1
*       APPROPRIATE PLACE IN PROGRAM STATEMENTS AND BECOME A PERMANENT   *    74 00073000          1
*       PART OF THE PROGRAM FILE. SEQUENCE NUMBERS DO NOT BECOME A PART  *    75 00074000          1
*       OF THE DATA FILE AND SERVE ONLY AS REFERENCE NUMBERS IN THE      *    76 00075000          1
*       EVENT A USER CHOOSES TO CHANGE A DATA RECORD OR TERMINATE THE    *    77 00076000          1
*       PRESENT SESSION AND RESUME FROM THAT POINT, AT A LATER TIME.     *    78 00077000          1
*                                                                        *    79 00078000          1
*       WHEN ENTERING PROGRAM PATCHES THE FIRST CHARACTERS OF A LINE OF  *    80 00079000          1
*       TEXT ARE ASSUMED TO BE A SEQUENCE NUMBER. IF THESE CHARACTERS ARE *    81 00080000          1
*       NUMERIC THEY ARE PLACED IN THE PROPER POSITION WITHIN THE PROGRAM *    82 00081000          1
*       RECORD.  IF THE REQUIRED DIGITS ARE NOT FOUND AN ERROR           *    83 00082000          1
*       MESSAGE IS EMITTED. A COLON MAY OPTIONALLY FOLLOW THE SEQUENCE   *    84 00083000          1
*       NUMBER.                                                          *    85 00084000          1
*                                                                        *    86 00085000          1
*       WHILE OPERATING IN THE ENTER MODE A USER MAY SPECIFY THAT A       *    87 00086000          1
*       SPECIFIC DATA RECORD OR PROGRAM STATEMENT BE MODIFIED THROUGH THE *    88 00087000          1
*       USE OF THE CHANGE STATEMENT. THE "COMPILE","CONTROL","COPY",     *    89 00088000          1
*       "SEQUENCE","LOCK" AND "QUIT" STATEMENTS MAY ALSO BE USED WHILE   *    90 00089000          1
*       IN THE ENTER MODE. IN WHICH CASE THE FILE CURRENTLY BEING CREATED *    91 00090000          1
*       IS MADE A PERMANENT FILE AND ENTERED IN THE DISK DIRECTORY. IN    *    92 00091000          1
*       ADDITION, THE "RENAME", "REMOVE" AND "SEQUENCE SPECIFICATION"    *    93 00092000          1
*       STATEMENTS MAY BE USED WHILE IN THE ENTER MODE.                  *    94 00093000          1
*                                                                        *    95 00094000          1
*RESUME STATEMENT.                                                      *    96 00095000          1
```

```
*SYNTAX.                                                                *     97 00096000  1
*      THE SYNTAX FOR <RESUME STATEMENT> IS AS FOLLOWS:                 *     98 00097000  1
*                                                                       *     99 00098000  1
*        <RESUME STATEMENT> ::= * RESUME <RESUME PARAMETERS>            *    100 00099000  1
*        <RESUME PARAMETERS> ::= <RESUME TYPE> <FILE NAME>              *    101 00100000  1
*                                <RESUMPTION POINT>                     *    102 00101000  1
*        <RESUME TYPE> ::= PROGRAM / DATA / PATCHES / [EMPTY]           *    103 00102000  1
*        <FILE NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]           *    104 00103000  1
*                        [7 CHARACTER OR LESS SUFFIX] / [EMPTY]         *    105 00104000  1
*        <RESUMPTION POINT> ::= <SEQUENCE NUMBER> / END / [EMPTY]       *    106 00105000  1
*        <SEQUENCE NUMBER> ::= [AN INTEGER OF 8 OR LESS DIGITS]         *    107 00106000  1
*                                                                       *    108 00107000  1
*SEMANTICS.                                                             *    109 00108000  1
*      THE RESUME STATEMENT PROVIDES THE USER THE ABILITY TO ESTABLISH  *    110 00109000  1
*      A PROGRAM OR DATA FILE IN MULTIPLE SESSIONS. SPECIFICATIONS      *    111 00110000  1
*      ARE ANALOGOUS TO ENTER PARAMETERS WITH THE EXCEPTION OF THE FILE *    112 00111000  1
*      NAME AND RESUMPTION POINT. IF A FILE NAME IS NOT SPECIFIED THE   *    113 00112000  1
*      FILE NAME GIVEN IN THE LAST ENTER, RESUME, CHANGE, LIST, COPY    *    114 00113000  1
*      OR SEQUENCE STATEMENT IS ASSUMED. IF A SEQUENCE NUMBER IS GIVEN  *    115 00114000  1
*      FOR THE RESUMPTION POINT IT SPECIFIES THE LAST VALID LINE OF TEXT *    116 00115000  1
*      IN THE FILE AND RESUMPTION IS FROM THAT POINT ON. IF THE         *    117 00116000  1
*      RESUMPTION POINT IS EMPTY OR THE WORD "END" RESUMPTION IS FROM   *    118 00116100  1
*      THE END OF THE FILE. IN ANY CASE MINI/TEXT RETYPES THE LAST VALID*    119 00116200  1
*      LINE OF TEXT WHEN THE RESUME STATMENT IS ENTERED.                *    120 00117000  1
*                                                                       *    121 00118000  1
*CHANGE STATEMENT.                                                      *    122 00119000  1
*SYNTAX.                                                                *    123 00120000  1
*      THE SYNTAX FOR <CHANGE STATEMENT> IS AS FOLLOWS:                 *    124 00121000  1
*                                                                       *    125 00122000  1
*        <CHANGE STATEMENT> ::= * <CHANGE PARAMETERS> <                 *    126 00123000  1
*                                 <CHANGE RESPONSE> /                   *    127 00124000  1
*                               * <CHANGE PARAMETERS> :                 *    128 00125000  1
*                                 <CHANGE RESPONSE>                     *    129 00125100  1
*        <CHANGE PARAMETERS> ::= <CHANGE VERB> <CHANGE TYPE> <FILE NAME>*    130 00126000  1
*                                <CHANGE NUMBER> <CHANGE OPTION>        *    131 00127000  1
*                                                                       *    132 00128000  1
*        <CHANGE VERB> ::= CHANGE / FIX / [EMPTY]                       *    133 00129000  1
*        <CHANGE TYPE> ::= PROGRAM / DATA / [EMPTY]                     *    134 00130000  1
*        <FILE NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]           *    135 00131000  1
*                        [7 CHARACTER OR LESS SUFFIX] / [EMPTY]         *    136 00132000  1
*        <CHANGE NUMBER> ::= [AN INTEGER OF 8 OR LESS DIGITS]           *    137 00133000  1
*        <CHANGE OPTION> ::= R / I / D / [EMPTY]                        *    138 00134000  1
*        <CHANGE RESPONSE> ::= <TEXT LINE> /                            *    139 00135000  1
*                              * SEQ <INTEGER> /                        *    140 00136000  1
*                              * OK  /                                  *    141 00137000  1
*                            * <CHANGE OPTION> [SPACES] * <MODIFICATION> **   142 00138000  1
*                                              [SPACES] * <MODIFICATION> **   143 00138100  1
*                                                       <MODIFICATION> **   144 00138200  1
*        <TEXT LINE> ::= [A LINE OF TEXT CONSISTING OF 80 OR LESS       *    145 00139000  1
*                         CHARACTERS]                                   *    146 00140000  1
*        <MODIFICATION> ::= [CORRECTED TEXT]                            *    147 00141000  1
*                                                                       *    148 00142000  1
*SEMANTICS.                                                             *    149 00143000  1
*      THE CHANGE STATEMENT ALLOWS THE REMOTE USER TO MODIFY DATA RECORDS*    150 00144000  1
*      OR PROGRAM STATEMENTS SPECIFIED BY THE CHANGE NUMBER. WHEN THE   *    151 00145000  1
*      CHANGE STATEMENT IS ENTERED MINI/TEXT WILL TYPE THE EXISTING     *    152 00146000  1
*      LINE, PERFORM A CARRIAGE RETURN AND LINE FEED, AND RETYPE THE    *    153 00147000  1
```

check this could be dangerous

```
*       SEQUENCE NUMBER LEAVING THE REMOTE POSITIONED DIRECTLY BELOW THE  *   154 00148000        1
*       FIRST FILLABLE CHARACTER OF THE RECORD. SEVERAL CHANGE RESPONSES  *   155 00149000        1
*       ARE POSSIBLE.                                                     *   156 00012510        1
*                                                                         *   157 00150000        1
*       IF THE USER TYPES A NEW TEXT LINE IT WILL REPLACE THE CURRENT     *   158 00151000        1
*       RECORD WITHOUT CHANGING THE SEQUENCE NUMBER OF THE LINE.          *   159 00152000        1
*                                                                         *   160 00153000        1
*       IF THE USER TYPES "* SEQ" FOLLOWED BY AN INTEGER THE SEQUENCE     *   161 00154000        1
*       NUMBER OF A PROGRAM RECORD WILL BE CHANGED TO THE INDICATED VALUE *   162 00155000        1
*       WITHOUT AFFECTING THE REST OF THE RECORD.                         *   163 00156000        1
*                                                                         *   164 00157000        1
*       IF THE USER TYPES "* OK" THE EXISTING RECORD IS NOT CHANGED.      *   165 00158000        1
*                                                                         *   166 00159000        1
*       THE USER MAY EDIT THE CURRENT TEXT LINE BY REPLACING EXISTING     *   167 00160000        1
*       CHARACTERS WITH OTHERS, BY INSERTING NEW CHARACTERS INTO THE LINE *   168 00160050        1
*       OR BY DELETING CHARACTERS FROM THE LINE. THE <CHANGE RESPONSE> TO *   169 00160100        1
*       PERFORM EDITING CONSISTS OF AN INITIAL ASTERISK FOLLOWED BY       *   170 00160150        1
*       A <CHANGE OPTION> TO INDICATE THE TYPE OF EDITING, A SECOND       *   171 00160200        1
*       ASTERISK TO INDICATE THE POSITION AT WHICH EDITING IS TO OCCUR,   *   172 00160250        1
*       THE DESIRED MODIFICATION AND FINALLY A LEFT ARROW TERMINATING THE *   173 00160300        1
*       EDITING RECORD. A <CHANGE OPTION> OF "R" INDICATES REPLACEMENT    *   174 00160350        1
*       AN "I" INSERTION AND A "D" DELETION. IF THE <CHANGE OPTION> IS    *   175 00160400        1
*       LEFT EMPTY AN "R" IS ASSUMED. EACH OF THE CHANGE OPTIONS ARE      *   176 00160450        1
*       DESCRIBED BELOW AND EXAMPLES GIVEN TO ILLUSTRATE THEIR USE.       *   177 00160500        1
*                                                                         *   178 00160550        1
*       A. REPLACEMENT.                                                   *   179 00160600        1
*          AFTER TYPING THE LEADING ASTERISK AND THE <CHANGE OPTION>      *   180 00160650        1
*          THE USER SHOULD SPACE THE TELETYPE OVER AND TYPE AN ASTERISK   *   181 00160700        1
*          BELOW THE CHARACTER DIRECTLY PRECEDING THE CHARACTERS TO BE     *   182 00160750        1
*          REPLACED. THE DESIRED MODIFICATION SHOULD THEN BE TYPED AND    *   183 00160800        1
*          FOLLOWED BY A LEFT ARROW. THE CHARACTERS BETWEEN THE SECOND    *   184 00160850        1
*          ASTERISK AND THE LEFT ARROW WILL REPLACE THOSE IN THE ORIGINAL *   185 00160900        1
*          RECORD.                                                        *   186 00160950        1
*          EXAMPLES:                                                      *   187 00161000        1
*              *CHANGE A/DEMO 100←                                        *   188 00161050        1
*              0100:01234543210                                           *   189 00161100        1
*              0100:* R  *6789←                                           *   190 00161150        1
*              *100←                                                      *   191 00161200        1
*              0100:01234567890                                           *   192 00161250        1
*              0100:*    *4321←                                           *   193 00161300        1
*                                                                         *   194 00161350        1
*       B. INSERTION.                                                     *   195 00161400        1
*          AFTER TYPING THE LEADING ASTERISK AND THE <CHANGE OPTION>      *   196 00161450        1
*          THE USER SHOULD SPACE THE TELETYPE OVER AND TYPE AN ASTERISK   *   197 00161500        1
*          BELOW THE CHARACTER DIRECTLY PRECEDING THE POSITION AT WHICH   *   198 00161550        1
*          THE INSERTION IS TO OCCUR. THE CHARACTERS TO BE INSERTED SHOULD*   199 00161600        1
*          THEN BE TYPED AND FOLLOWED BY A LEFT ARROW. THE CHARACTERS     *   200 00161650        1
*          BETWEEN THE SECOND ASTERISK AND THE LEFT ARROW WILL BE INSERTED*   201 00161700        1
*          INTO THE EXISTING RECORD. IF, AS A RESULT OF INSERTING         *   202 00161750        1
*          CHARACTERS, THE LENGTH OF THE LINE EXCEEDS THE MAXIMUM ALLOWED *   203 00161800        1
*          RECORD LENGTH THEN THE RIGHT MOST CHARACTERS OF THE LINE       *   204 00161850        1
*          WILL BE LOST. NO INDICATION IS GIVEN TO THE USER IF THIS       *   205 00161900        1
*          OCCURS.                                                        *   206 00161950        1
*          EXAMPLES:                                                      *   207 00162000        1
*              *100←                                                      *   208 00162050        1
*              0100:01234543210                                           *   209 00162100        1
*              0100:* I  *5←                                              *   210 00162150        1
```

```
*                *100←                                                  *   211 00162200   1
*                0100:012345543210                                      *   212 00162250   1
*                0100:*OK←                                              *   213 00162300   1
*                                                                       *   214 00162350   1
*          C. DELETION.                                                 *   215 00162400   1
*             AFTER TYPING THE LEADING ASTERISK AND THE <CHANGE OPTION>  *   216 00162450   1
*             THE USER SHOULD SPACE THE TELETYPE OVER AND TYPE AN ASTERISK*   217 00162500   1
*             BELOW THE CHARACTER DIRECTLY PRECEDING THE CHARACTERS TO BE *   218 00162550   1
*             DELETED. THE TELETYPE SHOULD THEN BE SPACED OR TYPED OVER THE*  219 00162600   1
*             NUMBER OF CHARACTERS TO BE DELETED AND A LEFT ARROW TYPED.  *   220 00162650   1
*             THE NUMBER OF CHARACTERS BETWEEN THE SECOND ASTERISK AND THE*   221 00162700   1
*             LEFT ARROW WILL BE DELETED FROM THE EXISTING RECORD.        *   222 00162750   1
*             EXAMPLES:                                                  *   223 00162800   1
*                *100←                                                  *   224 00162850   1
*                0100:012345543210                                      *   225 00162900   1
*                0100:* D  * ←                                          *   226 00162950   1
*                *100←                                                  *   227 00163000   1
*                0100:01234543210                                       *   228 00163050   1
*                0100:*D *OUT←                                          *   229 00163100   1
*                *100←                                                  *   230 00163150   1
*                0100:01233210                                          *   231 00163200   1
*                0100:*OK←                                              *   232 00163250   1
*                                                                       *   233 00163300   1
*          THE FIRST THREE CHARACTERS OF A RECORD CAN NOT BE EDITED IN THE*   234 00163350   1
*          MANNER DESCRIBED ABOVE. IN ORDER TO EDIT THESE CHARACTERS THE *   235 00163400   1
*          USER SHOULD INCLUDE THE <CHANGE OPTION> IN THE <CHANGE PARAMETERS>*236 00163450   1
*          IF THE FIRST NON-BLANK CHARACTER OF THE <CHANGE RESPONSE> IS  *   237 00163500   1
*          AN ASTERISK IT IS CONSIDERED AS THE SECOND ASTERISK AS        *   238 00163550   1
*          DESCRIBED ABOVE AND INDICATES THE POSITION AT WHICH EDITING   *   239 00163600   1
*          IS TO OCCUR. IF THE FIRST NON-BLANK CHARACTER IS NOT AN ASTERISK*  240 00163650   1
*          THEN EDITING BEGINS AT THE FIRST CHARACTER POSITION OF THE    *   241 00163700   1
*          RECORD.                                                       *   242 00163750   1
*             EXAMPLES:                                                  *   243 00163800   1
*                *100 D ←                                               *   244 00163850   1
*                0100:01233210                                          *   245 00163900   1
*                0100: *2332←                                           *   246 00163950   1
*                *100I←                                                 *   247 00164000   1
*                0100:0110                                              *   248 00164050   1
*                0100:1001←                                             *   249 00164100   1
*                *100←                                                  *   250 00164150   1
*                0100:10010110                                          *   251 00164200   1
*                0100:*OK←                                              *   252 00164250   1
*                                                                       *   253 00165000   1
*          IF THE USER WISHES TO CHANGE A RECORD WITHOUT HAVING IT RETYPED*   254 00166000   1
*          A COLON SHOULD BE TYPED AFTER THE <CHANGE PARAMETERS>. THE    *   255 00167000   1
*          CHARACTER FOLLOWING THE COLON IS CONSIDERED TO BE THE BEGINNING OF* 256 00168000   1
*          THE <CHANGE RESPONSE>.                                        *   257 00169000   1
*                                                                        *   258 00170000   1
*          IF A CHANGE IS ENTERED WHILE IN THE "ENTER" OR "RESUME" MODE  *   259 00171000   1
*          THE USER NEED ONLY SUPPLY THE CHANGE NUMBER.                  *   260 00172000   1
*                                                                        *   261 00173000   1
*          TO EFFECT A CHANGE TO A RECORD AFTER THE FILE HAS BEEN        *   262 00174000   1
*          MADE PERMANENT THE USER MUST ENTER THE TYPE OF CHANGE AND     *   263 00175000   1
*          THE FILE NAME IN ADDITION TO THE CHANGE NUMBER.               *   264 00176000   1
*          IF MULTIPLE CHANGES TO A PERMANENT FILE ARE REQUIRED ONLY THE *   265 00177000   1
*          FIRST SHOULD CONTAIN THE CHANGE TYPE AND FILE NAME.           *   266 00178000   1
*                                                                        *   267 00179000   1
```

```
*COMPILE STATEMENT.                                                    *   268 00180000        1
*SYNTAX.                                                               *   269 00181000        1
*      THE SYNTAX FOR <COMPILE STATEMENT> IS AS FOLLOWS:               *   270 00182000        1
*                                                                      *   271 00183000        1
*        <COMPILE STATEMENT> ::= * COMPILE <COMPILE PARAMETERS>        *   272 00184000        1
*        <COMPILE PARAMETERS> ::= <PROGRAM NAME> /                     *   273 00185000        1
*                                 <PROGRAM NAME> <COMPILE SPECIFICATIONS>*   274 00186000        1
*        <PROGRAM NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]       *   275 00187000        1
*                           [7 CHARACTER OR LESS SUFFIX]               *   276 00188000        1
*        <COMPILE SPECIFICATIONS> ::= <COMPILE OPTION> /               *   277 00189000        1
*                                     <COMPILE SPECIFICATIONS> <COMPILE OPTION>*   278 00190000        1
*        <COMPILE OPTION> ::= <COMPILER> /                             *   279 00191000        1
*                             <PROGRAM DISPOSITION> /                  *   280 00192000        1
*                             <INPUT MEDIA SPECIFIER> /                *   281 00193000        1
*                             <NEW TAPE SPECIFIER> /                   *   282 00194000        1
*                             <COMPILER PRINT FILE DISPOSITION>        *   283 00195000        1
*        <COMPILER> ::= ALGOL / XALGOL / BASIC / FORTRAN / [EMPTY]     *   284 00196000        1
*        <PROGRAM DISPOSITION>::= LIBRARY / GO / SYNTAX / [EMPTY]      *   285 00197000        1
*        <INPUT MEDIA SPECIFIER> ::= TAPE / CARD / CARDS / [EMPTY]     *   286 00198000        1
*        <NEW TAPE SPECIFIER> ::= NEW TAPE / NEWTAPE / [EMPTY]         *   287 00199000        1
*        <COMPILER PRINT FILE DISPOSITION> ::= PRINTER / LIST / [EMPTY]*   288 00200000        1
*                                                                      *   289 00201000        1
*SEMANTICS.                                                            *   290 00202000        1
*      THE COMPILE STATEMENT IS DESIGNED TO AUTOMATICALLY GENERATE     *   291 00203000        1
*      COMPILE AND COMPILER LABEL EQUATION CARDS FROM THE COMPILE      *   292 00204000        1
*      SPECIFICATIONS. ONCE GENERATED THE CONTROL DECK IS SCHEDULED     *   293 00205000        1
*      TO BE RUN.                                                      *   294 00206000        1
*                                                                      *   295 00207000        1
*      THE COMPILER MAY BE EITHER "ALGOL","XALGOL","BASIC" OR "FORTRAN".*   296 00208000        1
*      IF THE COMPILER IS NOT SPECIFIED THEN "ALGOL" IS ASSUMED.       *   297 00209000        1
*                                                                      *   298 00210000        1
*      PROGRAM DISPOSITION MAY BE "LIBRARY", "GO" OR "SYNTAX". AN EMPTY*   299 00211000        1
*      PROGRAM DISPOSITION RESULTS IN A COMPILE TO THE LIBRARY.        *   300 00212000        1
*                                                                      *   301 00213000        1
*      AN INPUT MEDIA SPECIFIER OF "CARD" OR "CARDS" ASSUMES COMPILER  *   302 00214000        1
*      INPUT TO BE A DISK FILE ESTABLISHED BY MINI/TEXT. "TAPE" ASSUMES*   303 00215000        1
*      THE PROGRAM HAS BEEN PREVIOUSLY COMPILED WITH A NEW TAPE BEING  *   304 00216000        1
*      SET UP ON DISK. THE PATCH DECK MUST BE ON DISK, AS CREATED BY   *   305 00217000        1
*      MINI/TEXT. IF THE INPUT MEDIA SPECIFIER IS EMPTY "CARD"         *   306 00218000        1
*      IS ASSUMED.                                                     *   307 00219000        1
*                                                                      *   308 00220000        1
*      A NEW TAPE SPECIFIER CAUSES MINI/TEXT TO REQUEST THE FILE NAME  *   309 00221000        1
*      FOR A NEW TAPE AND GENERATE A LABEL EQUATION CARD TO PUT THE    *   310 00222000        1
*      FILE ON DISK. TO GENERATE A NEW TAPE IT IS ALSO NECESSARY TO    *   311 00223000        1
*      HAVE THE PROPER DOLLAR SIGN CARD IN THE COMPILER CARD FILE ON   *   312 00224000        1
*      DISK.                                                           *   313 00225000        1
*                                                                      *   314 00226000        1
*      THE COMPILER PRINT FILE DISPOSITION SPECIFIES WHETHER THE       *   315 00227000        1
*      PROGRAM LISTING SHOULD BE OUTPUTTED ONTO THE LINE PRINTER OR    *   316 00228000        1
*      DISK FILE. A COMPILER PRINT FILE DISPOSITION OF "PRINTER" CAUSES*   317 00229000        1
*      THE LISTING TO BE DIRECTED TO THE LINE PRINTER. IF THE COMPILER *   318 00230000        1
*      PRINT FILE DISPOSITION IS EMPTY OR THE WORD "LIST", THE COMPILER*   319 00231000        1
*      PRINT FILE IS WRITTEN ON THE DISK. MINI/TEXT WILL INFORM THE USER*   320 00232000        1
*      WHAT IT HAS NAMED HIS PRINTER FILE ON DISK. PROGRAM LISTINGS    *   321 00233000        1
*      OUTPUTTED TO THE DISK MAY BE TRANSMITTED TO THE REMOTE USING THE*   322 00234000        1
*      "LIST STATEMENT".                                               *   323 00235000        1
*                                                                      *   324 00236000        1
```

```
*      IF MINI/TEXT IS IN THE "ENTER", "RESUME" OR "CHANGE" PROGRAM MODE *   325 00237000  1
*      WHEN THE COMPILE STATEMENT IS ENTERED, THE PROGRAM                *   326 00238000  1
*      COMPILED WILL BE THE ONE NAMED IN THE PREVIOUS SPECIFICATIONS.    *   327 00239000  1
*      WHEN NOT IN SUCH A MODE MINI/TEXT WILL REQUEST THE FILE NAME TO   *   328 00240000  1
*      INPUT TO THE COMPILER.                                            *   329 00241000  1
*                                                                        *   330 00242000  1
*CONTROL STATEMENT.                                                      *   331 00243000  1
*SYNTAX.                                                                 *   332 00244000  1
*      THE SYNTAX FOR <CONTROL STATEMENT> IS AS FOLLOWS:                 *   333 00245000  1
*                                                                        *   334 00246000  1
*        <CONTROL STATEMENT> ::= * CONTROL  ← <CONTROL CARD OPTIONS>     *   335 00247000  1
*        <CONTROL CARD OPTIONS> ::= RUN / EXECUTE / COMPILE / FILE /     *   336 00248000  1
*                                   PROCESS / IO / PRIORITY / STACK /CORE*   337 00249000  1
*                                                                        *   338 00250000  1
*SEMANTICS.                                                              *   339 00251000  1
*      THE CONTROL STATEMENT ALLOWS A REMOTE USER TO ESTABLISH           *   340 00252000  1
*      LEGITIMATE CONTROL CARD DECKS IN A PSUEDO CARD READER.            *   341 00253000  1
*      THESE DECKS ARE PROCESSED BY THE MCP USING THE NORMAL             *   342 00254000  1
*      SELECTION ALGORITHM.  ALL CONTROL CARD OPTIONS, EXCLUDING         *   343 00255000  1
*      "CHANGE" AND "REMOVE", ARE ALLOWED. CONTROL CARD INFORMATION      *   344 00256000  1
*      MAY CONSIST OF ANY NUMBER OF LOGICAL RECORDS. A QUESTION MARK     *   345 00257000  1
*      IS INSERTED BY MINI/TEXT INTO THE FIRST CHARACTER OF EACH LOGICAL *   346 00258000  1
*      RECORD. IF A "CHANGE" OR "REMOVE" CONTROL CARD IS ENTERED THE     *   347 00259000  1
*      ENTIRE DECK WILL BE DISCARDED AND THE REMOTE USER WILL BE         *   348 00260000  1
*      NOTIFIED. THE DECK IS ASSUMED TO BE COMPLETED WHENEVER THE USER   *   349 00261000  1
*      ENTERS "END" AS THE FIRST WORD OF A CONTROL CARD. IF THE USER     *   350 00262000  1
*      WISHES TO DISCARD A CONTROL DECK BEFORE IT IS COMPLETED HE MAY    *   351 00263000  1
*      DO SO BY ENTERING "REMOVE" AS THE FIRST WORD OF A CONTROL CARD.   *   352 00264000  1
*                                                                        *   353 00265000  1
*LIST STATEMENT.                                                         *   354 00266000  1
*SYNTAX.                                                                 *   355 00267000  1
*      THE SYNTAX FOR <LIST STATEMENT> IS AS FOLLOWS:                    *   356 00268000  1
*                                                                        *   357 00269000  1
*        <LIST STATEMENT> ::= * LIST <LIST TYPE> <FILE NAME> <LIST RANGE>*   358 00270000  1
*        <LIST TYPE> ::= PROGRAM / DATA / ALGOL / XALGOL / BASIC /       *   359 00273000  1
*                        FORTRAN / PRINTER / PRINT / INFO / [EMPTY]      *   360 00274000  1
*        <FILE NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]            *   361 00275000  1
*                        [7 CHARACTER OR LESS SUFFIX] / [EMPTY]          *   362 00276000  1
*        <LIST RANGE> ::= <INITIAL SEQUENCE NUMBER>                      *   363 00277000  1
*                         <FINAL SEQUENCE NUMBER>                        *   364 00278000  1
*        <INITIAL SEQUENCE NUMBER> ::= FROM <SEQUENCE NUMBER> /          *   365 00279000  1
*                                       <SEQUENCE NUMBER> / [EMPTY]      *   366 00280000  1
*        <FINAL SEQUENCE NUMBER> ::= THRU <SEQUENCE NUMBER> /            *   367 00281000  1
*                                     * <SEQUENCE NUMBER> / [EMPTY]      *   368 00282000  1
*        <SEQUENCE NUMBER> ::= [AN INTEGER OF 8 OR LESS DIGITS]          *   369 00282100  1
*                                                                        *   370 00283000  1
*SEMANTICS.                                                              *   371 00284000  1
*      THE LIST STATEMENT PROVIDES THE REMOTE USER THE ABILITY TO        *   372 00285000  1
*      LIST, ALL OR ANY PORTION, OF A PERMANENT FILE.                    *   373 00286000  1
*                                                                        *   374 00287000  1
*      DATA IS TRANSMITTED IN A FORM ASSOCIATED WITH THE LIST TYPE,      *   375 00288000  1
*      IRREGARDLESS OF LIST TYPE, INFORMATION IS OPTIMIZED TO DECREASE   *   376 00289000  1
*      TYPEWRITER USAGE. ALL BLANKS, FOLLOWING THE LAST NON-BLANK        *   377 00290000  1
*      CHARACTER, ARE DELETED.                                           *   378 00291000  1
*                                                                        *   379 00292000  1
*      A LIST TYPE OF "PROGRAM" CAUSES SOURCE PROGRAM FILES, ON          *   380 00293000  1
*      THE DISK, TO BE LISTED. "DATA" SPECIFIES THAT A DATA FILE IS TO BE*   381 00294000  1
```

```
*       LISTED. "ALGOL","XALGOL","BASIC" AND "FORTRAN" LIST TYPE REFER TO *   382 00295000   1
*       COMPILER PRINT FILES THAT HAVE BEEN LABELEQUATED TO DISK. LABEL   *   383 00296000   1
*       EQUATION INFORMATION MAY BE FURNISHED, AUTOMATICALLY, BY THE      *   384 00297000   1
*       "COMPILE" STATEMENT OR DONE MANUALLY USING THE CONTROL STATEMENT. *   385 00298000   1
*       UPON COMPLETION OF LISTING COMPILER PRINT FILES THE ASSOCIATED    *   386 00299000   1
*       DISK FILE IS AUTOMATICALLY REMOVED. "PRINTER" OR "PRINT" REFER TO *   387 00300000   1
*       A PRINTER FILE ON DISK, BUILT BY AN OBJECT PROGRAM. THIS TYPE OF  *   388 00301000   1
*       FILE IS NEVER REFORMATTED. HOWEVER, TRAILING BLANKS ARE DELETED,  *   389 00302000   1
*       A LIST TYPE OF INFO CAUSES A FILE ENTERED AS DATA TO BE           *   390 00302100   1
*       LISTED WITHOUT SEQUENCE NUMBERS. IF A LIST TYPE IS NOT            *   391 00302200   1
*       SPECIFIED "PROGRAM" IS ASSUMED.                                   *   392 00303000   1
*                                                                         *   393 00304000   1
*       IF A FILE NAME IS NOT SPECIFIED THE CURRENT FILE NAME, AS GIVEN   *   394 00304100   1
*       IN THE LAST ENTER, RESUME, CHANGE, LIST, COPY OR SEQUENCE         *   395 00304200   1
*       STATEMENT IS ASSUMED. IN ADDITION, AFTER A "COMPILE" STATEMENT    *   396 00304300   1
*       WHICH CREATED A COMPILER PRINT FILE ON DISK, A LIST STATEMENT     *   397 00304400   1
*       WILL ASSUME THE NAME ASSIGNED BY MINI/TEXT TO THE PRINTER FILE    *   398 00304500   1
*       IF NO OTHER FILE NAME IS SPECIFIED.                               *   399 00304600   1
*                                                                         *   400 00304700   1
*       A LIST RANGE SPECIFIES THAT ONLY A PORTION OF THE DESIGNATED FILE *   401 00305000   1
*       IS TO BE LISTED.  ONE RECORD OR A COMPLETE RANGE OF RECORDS MAY   *   402 00306000   1
*       BE SPECIFIED.  IN THE ABSENCE OF A LIST RANGE THE ENTIRE FILE     *   403 00307000   1
*       IS TRANSMITTED.                                                   *   404 00308000   1
*                                                                         *   405 00309000   1
*       IF THE USER WISHES TO TERMINATE A TRANSMISSION BEFORE ITS         *   406 00310000   1
*       NORMAL COMPLETION HE MAY DO SO BY DEPRESSING THE BREAK KEY        *   407 00311000   1
*       ON HIS REMOTE CONSOLE.                                            *   408 00312000   1
*                                                                         *   409 00313000   1
*RENAME STATEMENT.                                                        *   410 00314000   1
*SYNTAX.                                                                  *   411 00315000   1
*       THE SYNTAX FOR <RENAME STATEMENT> IS AS FOLLOWS:                  *   412 00316000   1
*                                                                         *   413 00317000   1
*         <RENAME STATEMENT> ::= * RENAME <RENAME LIST>                   *   414 00318000   1
*         <RENAME LIST> ::= <FILE NAME> TO <FILE NAME> /                  *   415 00319000   1
*                           <RENAME LIST> , <FILE NAME> TO <FILE NAME>    *   416 00320000   1
*         <FILE NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]            *   417 00321000   1
*                         [7 CHARACTER OR LESS SUFFIX]                    *   418 00322000   1
*                                                                         *   419 00323000   1
*SEMANTICS.                                                               *   420 00324000   1
*       PERMANENT FILES, ESTABLISHED BY THE USER, ARE RENAMED. FILES NOT  *   421 00325000   1
*       CREATED BY THE USER WILL NOT BE AFFECTED BY THIS STATEMENT.       *   422 00326000   1
*                                                                         *   423 00327000   1
*REMOVE STATEMENT.                                                        *   424 00328000   1
*SYNTAX.                                                                  *   425 00329000   1
*       THE SYNTAX FOR <REMOVE STATEMENT> IS AS FOLLOWS:                  *   426 00330000   1
*                                                                         *   427 00331000   1
*         <REMOVE STATEMENT> ::= * REMOVE <REMOVE LIST>                   *   428 00332000   1
*         <REMOVE LIST> ::= <FILE NAME> / <REMOVE LIST> , <FILE NAME>     *   429 00333000   1
*         <FILE NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]            *   430 00334000   1
*                         [7 CHARACTER OR LESS SUFFIX]                    *   431 00335000   1
*SEMANTICS.                                                               *   432 00336000   1
*       FILES, ESTABLISHED BY THE USER, ARE REMOVED FROM THE DISK         *   433 00337000   1
*       LIBRARY. IF THE FILE WAS NOT CREATED BY THE USER THIS             *   434 00338000   1
*       STATEMENT IS DISCARDED.                                           *   435 00339000   1
*                                                                         *   436 00340000   1
*FIND STATEMENT.                                                          *   437 00340010   1
*SYNTAX.                                                                  *   438 00340020   1
```

```
*      THE SYNTAX FOR <FIND STATEMENT> IS AS FOLLOWS:                          *   439 00340030  1
*                                                                              *   440 00340040  1
*        <FIND STATEMENT> ::= * FIND <FIRST OPTION> <STRING>                   *   441 00340050  1
*                               IN <FILE NAME> <FIND RANGE>                    *   442 00340060  1
*        <FIRST OPTION> ::= FIRST / ALL / [EMPTY]                              *   443 00340070  1
*        <STRING> ::= <IDENTIFIER> / <INTEGER> /                               *   444 00340080  1
*                     <QUOTED STRING> / <OCTAL STRING>                         *   445 00340090  1
*        <IDENTIFIER> ::= <LETTER> / <IDENTIFIER> <LETTER> /                   *   446 00340100  1
*                                    <IDENTIFIER> <DIGIT>                      *   447 00340110  1
*        <LETTER> ::= A/B/C/D/E/F/G/H/I/J/K/L/M/N/O/P/Q/R/S/T/U/V/W/X/Y/Z      *   448 00340120  1
*        <DIGIT> ::= 0/1/2/3/4/5/6/7/8/9                                       *   449 00340130  1
*        <INTEGER> ::= <DIGIT> / <INTEGER> <DIGIT>                             *   450 00340140  1
*        <QUOTED STRING> ::= <DELIMITER> [LEQ 31 CHARACTERS] <DELIMITER>       *   451 00340150  1
*        <DELIMITER> ::= " / #                                                 *   452 00340155  1
*        <OCTAL STRING> ::= 3" <OCTAL NUMBER> "                                *   453 00340160  1
*        <OCTAL NUMBER> ::= <OCTAL DIGIT> / <OCTAL NUMBER> <OCTAL DIGIT>       *   454 00340170  1
*        <OCTAL DIGIT> ::= 0/1/2/3/4/5/6/7                                     *   455 00340180  1
*        <FILE NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]                  *   456 00340190  1
*                        [7 CHARACTER OR LESS SUFFIX] / [EMPTY]                *   457 00340200  1
*        <FIND RANGE> ::= <INITIAL SEQUENCE NUMBER>                            *   458 00340210  1
*                         <FINAL SEQUENCE NUMBER>                              *   459 00340220  1
*        <INITIAL SEQUENCE NUMBER> ::= <NUMBER> / FROM <NUMBER> / [EMPTY]      *   460 00340230  1
*        <FINAL SEQUENCE NUMBER> ::= - <NUMBER> / THRU <NUMBER> / [EMPTY]      *   461 00340240  1
*        <NUMBER> ::= [AN INTEGER OF 8 OR LESS DIGITS]                         *   462 00340250  1
*                                                                              *   463 00340260  1
*SEMANTICS.                                                                    *   464 00340270  1
*      THE FIND STATEMENT GIVES THE USER THE CAPABILITY TO SEARCH A            *   465 00340280  1
*      PROGRAM FILE FOR OCCURANCES OF A SPECIFIED IDENTIFIER, INTEGER          *   466 00340290  1
*      OR SEQUENCE OF CHARACTERS. THE SEQUENCE NUMBER OF EACH RECORD           *   467 00340300  1
*      THAT INCLUDES THE QUANTITY IS TYPED ON THE USERS CONSOLE. IF            *   468 00340310  1
*      THE USER DESIRES TO TERMINATE A FIND STATEMENT BEFORE ITS               *   469 00340320  1
*      NORMAL COMPLETION HE MAY DO SO BY DEPRESSING THE BREAK KEY              *   470 00340330  1
*      ON HIS CONSOLE.                                                         *   471 00340340  1
*                                                                              *   472 00340350  1
*      THE <FIRST OPTION> ALLOWS THE USER TO INDICATE THAT EITHER THE          *   473 00340360  1
*      FIRST OR ALL OCCURANCES OF THE SPECIFIED <STRING> ARE TO BE             *   474 00340370  1
*      SERACHED FOR. IF THE OPTION IS EMPTY THEN "ALL" IS ASSUMED.             *   475 00340380  1
*                                                                              *   476 00340390  1
*      SEVERAL METHODS FOR SPECIFYING THE <STRING> ARE PROVIDED. THEY          *   477 00340400  1
*      ARE DESCRIBED SEPERATELY BELOW. EXAMPLES OF EACH TYPE ARE GIVEN.        *   478 00340410  1
*                                                                              *   479 00340420  1
*      A. THE <IDENTIFIER> TYPE OF <STRING> IS ESSENTIALLY THE SAME AS         *   480 00340430  1
*         AN ALGOL IDENTIFIER WITH THE RESTRICTION THAT ITS MAXIMUM            *   481 00340440  1
*         LENGTH IS 13 CHARACTERS.                                             *   482 00340450  1
*         EXAMPLES:                                                            *   483 00340460  1
*              INTEGER     L101     PROCEDURE     GMAXQQQ.                     *   484 00340470  1
*                                                                              *   485 00340480  1
*      B. AN <INTEGER> IS A SEQUENCE OF UP TO 12 DIGITS SUCH AS,               *   486 00340490  1
*         123456789012.                                                        *   487 00340500  1
*                                                                              *   488 00340510  1
*      C. A <QUOTED STRING> IS A SEQUENCE OF UP TO 31 CHARACTERS               *   489 00340520  1
*         ENCLOSED BY <DELIMITER>S. THE SAME <DELIMITER> MUST BE USED          *   490 00340530  1
*         AT THE BEGINNING AND END OF THE QUOTED STRING. THE FIRST             *   491 00340540  1
*         CHARACTER OF A QUOTED STRING MUST NOT BE A BLANK                     *   492 00340550  1
*         CHARACTER.                                                           *   493 00340560  1
*         EXAMPLES:                                                            *   494 00340570  1
*              "NOTLISTINGORCOPYSEQING"        "#"      #"#                     *   495 00340580  1
```

```
*                #Q="1*0000"#                        "START:=START+1,57"          *   496 00340590   1
*                                                                                *   497 00340600   1
*       D. AN <OCTAL STRING> IS A SEQUENCE OF AN EVEN NUMBER OF OCTAL            *   498 00340610   1
*          DIGITS ENCLOSED BY QUOTES WITH A DIGIT 3 DIRECTLY PRECEDING           *   499 00340620   1
*          THE INITIAL QUOTE. THE NUMBER OF OCTAL DIGITS MUST BE 62 OR           *   500 00340630   1
*          LESS. THE PURPOSE OF THE <OCTAL STRING> TYPE OF <STRING> IS TO        *   501 00340640   1
*          ALLOW THE USER TO SPECIFY CHARACTERS SUCH AS ←, ≤, ≥ ETC.,            *   502 00340650   1
*          WHICH CANNOT BE DIRECTLY ENTERED FROM THE CONSOLE.                    *   503 00340660   1
*          EXAMPLES:                                                             *   504 00340670   1
*               3"37"                                                            *   505 00340680   1
*               3"14"                                                            *   506 00340690   1
*               3"3160434665256063513123312l".                                   *   507 00340700   1
*                                                                                *   508 00340710   1
*       THE <FILE NAME> SPECIFIES THE PROGRAM FILE IN WHICH TO PERFORM           *   509 00340720   1
*       THE SEARCH. IF A FILE NAME IS NOT SPECIFIED THEN THE CURRENT             *   510 00340730   1
*       FILE NAME AS GIVEN IN THE LAST STATEMENT WHICH SPECIFIED A FILE          *   511 00340740   1
*       WILL BE ASSUMED.                                                         *   512 00340750   1
*                                                                                *   513 00340760   1
*       THE <FIND RANGE> ALLOWS THE USER TO SPECIFY THE PORTION OF               *   514 00340770   1
*       THE FILE THAT IS TO BE SEARCHED. IF NO RANGE IS INDICATED THEN           *   515 00340780   1
*       ALL RECORDS IN THE FILE WILL BE SEARCHED.                                *   516 00340790   1
*                                                                                *   517 00340800   1
*REPLACE STATEMENT.                                                              *   518 00340810   1
*SYNTAX.                                                                         *   519 00340820   1
*       THE SYNTAX FOR <REPLACE STATEMENT> IS AS FOLLOWS:                        *   520 00340830   1
*                                                                                *   521 00340840   1
*         <REPLACE STATEMENT> ::=                                                *   522 00340850   1
*                 * REPLACE <FIRST OPTION> <STRING> BY <STRING>                  *   523 00340860   1
*                   IN <FILE NAME> <REPLACE RANGE>                               *   524 00340870   1
*         <FIRST OPTION> ::= FIRST / ALL / [EMPTY]                               *   525 00340880   1
*         <STRING> ::= <IDENTIFIER> / <INTEGER> /                                *   526 00340890   1
*                      <QUOTED STRING> / <OCTAL STRING>                          *   527 00340900   1
*         <IDENTIFIER> ::= <LETTER> / <IDENTIFIER> <LETTER> /                    *   528 00340910   1
*                                     <IDENTIFIER> <DIGIT>                       *   529 00340920   1
*         <LETTER> ::= A/B/C/D/E/F/G/H/I/J/K/L/M/N/O/P/Q/R/S/T/U/V/W/X/Y/Z*     *   530 00340930   1
*         <DIGIT> ::= 0/1/2/3/4/5/6/7/8/9                                        *   531 00340940   1
*         <INTEGER> ::= <DIGIT> / <INTEGER> <DIGIT>                              *   532 00340950   1
*         <QUOTED STRING< ::= <DELIMITER> [LEQ 31 CHARACTERS] <DELIMITER>        *   533 00340960   1
*         <DELIMITER> ::= " / #                                                  *   534 00340965   1
*         <OCTAL STRING> ::= 3" <OCTAL NUMBER> "                                 *   535 00340970   1
*         <OCTAL NUMBER> ::= <OCTAL DIGIT> / <OCTAL NUMBER> <OCTAL DIGIT>        *   536 00340980   1
*         <OCTAL DIGIT> ::= 0/1/2/3/4/5/6/7                                      *   537 00340990   1
*         <FILE NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]                   *   538 00341000   1
*                         [7 CHARACTER OR LESS SUFFIX] / [EMPTY]                 *   539 00341010   1
*         <REPLACE RANGE> ::= LINE <NUMBER> /                                    *   540 00341020   1
*            <INITIAL SEQUENCE NUMBER> <FINAL SEQUENCE NUMBER> / [EMPTY]         *   541 00341030   1
*         <INITIAL SEQUENCE NUMBER> ::= <NUMBER> / FROM <NUMBER> / [EMPTY]*      *   542 00341040   1
*         <FINAL SEQUENCE NUMBER> ::= - <NUMBER> / THRU <NUMBER> / [EMPTY]*      *   543 00341050   1
*         <NUMBER> ::= [AN INTEGER OF 8 OR LESS DIGITS]                          *   544 00341060   1
*                                                                                *   545 00341070   1
*SEMANTICS.                                                                      *   546 00341080   1
*       THE REPLACE STATEMENT GIVES THE USER THE CAPABILITY OF REPLACING         *   547 00341090   1
*       OCCURANCES OF A SPECIFIED IDENTIFIER, INTEGER OR SEQUENCE OF             *   548 00341100   1
*       CHARACTERS IN A PROGRAM FILE WITH ANOTHER STRING. THE CHARACTERS         *   549 00341110   1
*       WITHIN A RECORD IN WHICH REPLACEMENTS OCCUR ARE ADJUSTED TO              *   550 00341120   1
*       ACCOMODATE THE NEW STRING IF ITS LENGTH IS DIFFERENT FROM THAT           *   551 00341130   1
*       OF THE ORIGINAL STRING. IF, DUE TO AN ADJUSTMENT, CHARACTERS             *   552 00341140   1
```

```
*       WOULD BE LOST THE REPLACE STATEMENT IS TERMINATED AND AN          *   553 00341150  1
*       ERROR MESSAGE INCLUDING THE SEQUENCE NUMBER OF THE RECORD         *   554 00341160  1
*       WHICH WOULD OVERFLOW IS EMITTED.                                  *   555 00341170  1
*                                                                         *   556 00341180  1
*       THE <FIRST OPTION> ALLOWS THE USER TO INDICATE THAT EITHER        *   557 00341190  1
*       THE FIRST OR ALL OCCURANCES OF THE SPECIFIED STRING ARE TO BE     *   558 00341200  1
*       REPLACED. IF THE OPTION IS LEFT EMPTY THEN "ALL" IS ASSUMED.      *   559 00341210  1
*                                                                         *   560 00341220  1
*       SEVERAL METHODS OF SPECIFYING THE STRINGS ARE PROVIDED.           *   561 00341230  1
*       THEY ARE DESCRIBED AND EXAMPLES OF EACH TYPE ARE GIVEN IN THE     *   562 00341240  1
*       DESCRIPTION OF THE FIND STATEMENT.                                *   563 00341250  1
*                                                                         *   564 00341260  1
*       THE <FILE NAME> SPECIFIES THE PROGRAM FILE IN WHICH TO PERFORM    *   565 00341270  1
*       THE REPLACEMENTS. IF A FILE NAME IS NOT SPECIFIED THEN THE        *   566 00341280  1
*       CURRENT FILE NAME AS GIVEN IN THE LAST STATEMENT WHICH SPECIFIED  *   567 00341290  1
*       A FILE NAME IS ASSUMED.                                           *   568 00341300  1
*                                                                         *   569 00341310  1
*       THE <REPLACE RANGE> ALLOWS THE USER TO SPECIFY THE PORTION        *   570 00341320  1
*       OF THE FILE IN WHICH REPLACEMENT IS TO OCCUR. IF A "LINE"         *   571 00341330  1
*       NUMBER IS SPECIFIED ONLY THAT RECORD WILL BE AFFECTED. IF NO      *   572 00341340  1
*       <REPLACE RANGE> IS INDICATED REPLACEMENT MAY OCCUR IN ALL         *   573 00341350  1
*       RECORDS OF THE FILE.                                              *   574 00341360  1
*                                                                         *   575 00341370  1
*SEQUENCE STATEMENT.                                                      *   576 00341380  1
*SYNTAX.                                                                  *   577 00342000  1
*       THE SYNTAX FOR <SEQUENCE STATEMENT> IS AS FOLLOWS:                *   578 00343000  1
*                                                                         *   579 00344000  1
*         <SEQUENCE STATEMENT> ::= * <SEQUENCE VERB> <PROGRAM SPECIFIER>  *   580 00345000  1
*                                  <FILE NAME> <SEQUENCE RANGE>           *   581 00346000  1
*                                  <SEQUENCE SPECIFICATIONS>              *   582 00347000  1
*         <SEQUENCE VERB> ::= SEQUENCE / RESEQ                            *   583 00347100  1
*         <PROGRAM SPECIFIER> ::= PROGRAM / [EMPTY]                       *   584 00348000  1
*         <FILE NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]            *   585 00349000  1
*                         [7 CHARACTER OR LESS SUFFIX] / [EMPTY]          *   586 00350000  1
*         <SEQUENCE RANGE> ::= <INITIAL SEQUENCE NUMBER>                  *   587 00351000  1
*                              <FINAL SEQUENCE NUMBER>                    *   588 00352000  1
*         <INITIAL SEQUENCE NUMBER> ::= FROM <SEQUENCE NUMBER> /          *   589 00353000  1
*                                       <SEQUENCE NUMBER> / [EMPTY]       *   590 00354000  1
*         <FINAL SEQUENCE NUMBER> ::= THRU <SEQUENCE NUMBER> /            *   591 00355000  1
*                                     - <SEQUENCE NUMBER> / [EMPTY]       *   592 00355100  1
*         <SEQUENCE NUMBER> ::= [AN INTEGER OF 8 OR LESS DIGITS]          *   593 00355200  1
*         <SEQUENCE SPECIFICATIONS> ::= <BASE SPECIFIER> <STEP SPECIFIER> *   594 00356000  1
*         <BASE SPECIFIER> ::= BASE <NUMBER> / <NUMBER> / [EMPTY]         *   595 00357000  1
*         <STEP SPECIFIER> ::= STEP <NUMBER> / + <NUMBER> / [EMPTY]       *   596 00357100  1
*         <NUMBER> ::= [AN INTEGER OF 8 OR LESS DIGITS]                   *   597 00358000  1
*                                                                         *   598 00359000  1
*SEMANTICS.                                                               *   599 00360000  1
*       THE SEQUENCE STATEMENT ALLOWS THE USER TO RESEQUENCE SOURCE       *   600 00361000  1
*       PROGRAM FILES. IF A SEQUENCE RANGE IS PRESENT ONLY THE SPECIFIED  *   601 00362000  1
*       RANGE OF SEQUENCE NUMBERS (AS THEY APPEAR IN THE ORIGINAL FILE)   *   602 00363000  1
*       WILL BE RESEQUENCED. SEQUENCE SPECIFICATIONS GIVE THE BASE NUMBER *   603 00364000  1
*       AND INCREMENT VALUE TO BE USED IN RESEQUENCING THE FILE. IF A     *   604 00365000  1
*       BASE NUMBER IS NOT SPECIFIED ZERO IS ASSUMED. IF AN INCREMENT     *   605 00366000  1
*       VALUE IS NOT GIVEN A VALUE OF 10 IS USED.                         *   606 00367000  1
*                                                                         *   607 00368000  1
*COPY STATEMENT.                                                          *   608 00369000  1
*SYNTAX.                                                                  *   609 00370000  1
```

```
*       THE SYNTAX FOR <COPY STATEMENT> IS AS FOLLOWS:                          *   610 00371000  1
*                                                                               *   611 00372000  1
*         <COPY STATEMENT> ::= * COPY <PROGRAM SPECIFIER> <SOURCE FILE>         *   612 00373000  1
*                                <COPY RANGE> TO <ADD ON SPECIFIER>             *   613 00374000  1
*                               <DESTINATION FILE> <SEQUENCE SPECIFICATIONS>*   614 00375000  1
*         <PROGRAM SPECIFIER> ::= PROGRAM / [EMPTY]                             *   615 00376000  1
*         <SOURCE FILE> ::= <FILE NAME> / [EMPTY]                               *   616 00376100  1
*         <DESTINATION FILE> ::= <FILE NAME>                                    *   617 00376200  1
*         <FILE NAME> ::= [7 CHARACTER OR LESS PREFIX] [SLASH]                  *   618 00377000  1
*                         [7 CHARACTER OR LESS SUFFIX]                          *   619 00378000  1
*         <COPY RANGE> ::= <INITIAL SEQUENCE NUMBER>                            *   620 00379000  1
*                          <FINAL SEQUENCE NUMBER>                              *   621 00380000  1
*         <INITIAL SEQUENCE NUMBER> ::= FROM <SEQUENCE NUMBER> /                *   622 00381000  1
*                                        <SEQUENCE NUMBER> / [EMPTY]            *   623 00382000  1
*         <FINAL SEQUENCE NUMBER> ::= THRU <SEQUENCE NUMBER> /                  *   624 00382100  1
*                                    - <SEQUENCE NUMBER> / [EMPTY]              *   625 00382200  1
*         <SEQUENCE NUMBER> ::= [AN INTEGER OF 8 OR LESS DIGITS]                *   626 00383000  1
*         <ADD ON SPECIFIER> ::= END / [EMPTY]                                  *   627 00384000  1
*         <SEQUENCE SPECIFICATIONS> ::= <BASE SPECIFIER> <STEP SPECIFIER>       *   628 00385000  1
*         <BASE SPECIFIER> ::= BASE <NUMBER> / <NUMBER> / [EMPTY]               *   629 00386000  1
*         <STEP SPECIFIER> ::= STEP <NUMBER> / + <NUMBER> / [EMPTY]             *   630 00386100  1
*         <NUMBER> ::= [AN INTEGER OF 8 OR LESS DIGITS]                         *   631 00387000  1
*                                                                               *   632 00388000  1
*SEMANTICS.                                                                     *   633 00389000  1
*       THE COPY STATEMENT ALLOWS THE REMOTE USER TO DUPLICATE SOURCE           *   634 00390000  1
*       PROGRAM FILES. IF A COPY RANGE IS PRESENT ONLY THE SPECIFIED            *   635 00391000  1
*       RANGE OF CARD IMAGES (AS SPECIFIED BY THE SEQUENCE NUMBERS IN THE       *   636 00392000  1
*       ORIGINAL FILE) WILL BE COPIED TO THE NEW FILE. IF AN ADD ON             *   637 00393000  1
*       SPECIFIER IS PRESENT IT INDICATES THAT THE FOLLOWING FILE NAME          *   638 00394000  1
*       IS AN EXISTING FILE AND THE CARD IMAGES FROM THE ORIGINAL FILE          *   639 00395000  1
*       ARE TO BE ADDED ON TO THE END OF IT. IF A SEQUENCE SPECIFICATION        *   640 00396000  1
*       IS PRESENT THE NEW FILE WILL BE RESEQUENCED AS IT IS GENERATED.         *   641 00397000  1
*       IF A SEQUENCE SPECIFICATION IS NOT USED THE SEQUENCE NUMBERS OF         *   642 00398000  1
*       THE NEW FILE WILL BE THE SAME AS IN THE ORIGINAL FILE.                  *   643 00399000  1
*                                                                               *   644 00400000  1
*SEQUENCE SPECIFICATION STATEMENT.                                              *   645 00401000  1
*SYNTAX.                                                                        *   646 00402000  1
*       THE SYNTAX FOR <SEQUENCE SPECIFICATION STATEMENT> IS AS FOLLOWS:        *   647 00403000  1
*                                                                               *   648 00404000  1
*         <SEQUENCE SPECIFICATION STATEMENT> ::=                                *   649 00405000  1
*                              * SEQ <SEQUENCE PARAMETERS> /                    *   650 00405100  1
*                              * SEQ <SEQUENCE PARAMETERS> : <TEXT LINE>        *   651 00406000  1
*         <SEQUENCE PARAMETERS> ::= <BASE SPECIFIER> <STEP SPECIFIER>           *   652 00407000  1
*         <BASE SPECIFIER> ::= BASE <NUMBER> / <NUMBER> / [EMPTY]               *   653 00409000  1
*         <STEP SPECIFIER> ::= STEP <NUMBER> / + <NUMBER> / [EMPTY]             *   654 00410000  1
*         <NUMBER> ::= [AN INTEGER OF 8 OR LESS DIGITS]                         *   655 00413000  1
*         <TEXT LINE> ::= [A LINE OF TEXT CONSISTING OF 80 OR LESS              *   656 00413100  1
*                          CHARACTERS]                                          *   657 00413200  1
*                                                                               *   658 00414000  1
*SEMANTICS.                                                                     *   659 00415000  1
*       THE SEQUENCE SPECIFICATION STATEMENT ENABLES A REMOTE USER TO           *   660 00416000  1
*       SPECIFY THE BASE NUMBER AND INCREMENT VALUE                             *   661 00417000  1
*       USED IN CALCULATING THE SEQUENCE NUMBERS ASSIGNED TO PROGRAM            *   662 00418000  1
*       STATEMENTS. IF A SEQUENCE SPECIFICATION STATEMENT IS NOT USED           *   663 00419000  1
*       SEQUENCE NUMBERS BEGINNING WITH ZERO AND STEPPING IN                    *   664 00420000  1
*       INCREMENTS OF TEN ARE ASSUMED. ONCE THE SEQUENCE SPECIFICATIONS         *   665 00421000  1
*       ARE SET THEY WILL BE USED FOR THE REMAINDER OF A SESSION OR             *   666 00422000  1
```

```
*       UNTIL CHANGED BY ANOTHER SEQUENCE SPECIFICATION STATEMENT.
*
*       A TEXT LINE MAY BE ENTERED IN A SEQUENCE SPECIFICATION
*       STATEMENT BY TYPING A COLON FOLLOWED BY THE TEXT LINE
*       AFTER THE SEQUENCE PARAMETERS.
*
*TAB STATEMENT.
*SYNTAX.
*       THE SYNTAX FOR <TAB STATEMENT> IS AS FOLLOWS:
*
*         <TAB STATEMENT> ::= * TAB <TAB PARAMETERS> /
*                             * TAB <TAB PARAMETERS> : <TEXT LINE>
*         <TAB PARAMETERS> ::= <COLUMN NUMBER> / <TAB ADJUSTMENT>
*         <COLUMN NUMBER> ::= [AN INTEGER FROM 1 TO 32]
*         <TAB ADJUSTMENT> ::= + <DISPLACEMENT> / - <DISPLACEMENT>
*         <DISPLACEMENT> ::= [AN INTEGER LESS THAN 32]
*         <TEXT LINE> ::= [A LINE OF TEXT CONSISTING OF 80 OR LESS
*                          CHARACTERS]
*
*SEMANTICS.
*       THE TAB STATEMENT ALLOWS THE USER TO AUTOMATICALLY PRE-POSITION
*       THE PRINT HEAD AT ANY COLUMN. THE TAB STATEMENT HAS AN EFFECT ONLY
*       IN THE "ENTER" OR "RESUME" MODE. A TAB ADJUSTMENT MAY BE USED TO
*       INCREASE OR DECREASE THE <COLUMN NUMBER>. IF THE <COLUMN NUMBER>
*       IS SET OR ADJUSTED TO A VALUE GREATER THAN 32 OR LESS THAN 1 IT
*       WILL BE SET TO 1. A TEXT LINE MAY BE ENTERED IN A TAB STATEMENT BY
*       TYPING A COLON FOLLOWED BY THE TEXT LINE AFTER THE TAB PARAMETERS.
*
*ASTERISK OPTION CONTROL STATEMENT.
*SYNTAX.
*       THE SYNTAX FOR <ASTERISK OPTION CONTROL STATEMENT> IS AS FOLLOWS:
*
*         <ASTERISK OPTION CONTROL STATEMENT> ::= * * /
*                                                 * * : <TEXT LINE>
*         <TEXT LINE> ::= [A LINE OF TEXT CONSISTING OF 80 OR LESS
*                          CHARACTERS]
*
*SEMANTICS.
*       THE ASTERISK OPTION CONTROL STATEMENT PROVIDES THE MEANS FOR
*       SETTING OR RESETTING A PROGRAM OPTION WHICH DETERMINES IF A
*       LEADING ASTERISK IS REQUIRED TO INDICATE A MINI/TEXT COMMAND.
*
*       WHEN SET, ALL COMMANDS -MUST- CONTAIN A LEADING ASTERISK.
*
*       WHEN RESET, THE LEADING ASTERISK IS -OPTIONAL-.
*
*       EACH TIME AN ASTERISK OPTION CONTROL STATEMENT IS ENTERED
*       THE SETTING OF THE OPTION IS CHANGED FROM ITS CURRENT TO ITS
*       OPPOSITE STATE.
*
*       A TEXT LINE MAY BE ENTERED IN AN ASTERISK OPTION CONTROL
*       STATEMENT BY TYPING A COLON FOLLOWED BY THE TEXT LINE
*       AFTER THE CONTROL STATEMENT.
*
*       WHEN A USER INITIATES THE MINI/TEXT PROGRAM THE ASTERISK
*       OPTION IS RESET.
*
```

```
*CHARGE CODE STATEMENT.                                                 *   724 00427025        1
*SYNTAX.                                                                *   725 00427050        1
*       THE SYNTAX FOR <CHARGE CODE STATEMENT> IS AS FOLLOWS:           *   726 00427075        1
*                                                                       *   727 00427100        1
*         <CHARGE CODE STATEMENT> ::= * CHARGE <CHARGE NUMBER> <COMMENT>*   728 00427125        1
*         <CHARGE NUMBER> ::= [A VALID 9 DIGIT JOB ORDER NUMBER]        *   729 00427150        1
*         <COMMENT> ::= [ANY 15 CHARACTERS NOT INCLUDING "," OR "*"]    *   730 00427175        1
*                                                                       *   731 00427200        1
*SEMANTICS.                                                             *   732 00427225        1
*       THE CHARGE CODE STATEMENT ALLOWS THE USER TO SPECIFY OR CHANGE  *   733 00427250        1
*       THE JOB ORDER NUMBER USED BY MINI/TEXT. THE CHARGE NUMBER       *   734 00427275        1
*       SPECIFIED IS USED IN ALL COMPILE STATEMENTS AND IS RECORDED IN  *   735 00427300        1
*       THE LOG WHEN THE USER ENTERS A QUIT STATEMENT. A CHARGE CODE    *   736 00427325        1
*       STATEMENT IS AUTOMATICALLY EXECUTED WHEN A USER RUNS MINI/TEXT. *   737 00427350        1
*       WHEN THE CHARGE CODE STATEMENT IS EXECUTED THE USER IS GIVEN    *   738 00427375        1
*       THREE OPPORTUNITIES TO ENTER A VALID CHARGE NUMBER. IF NO VALID *   739 00427400        1
*       NUMBER IS RECEIVED MINI/TEXT WILL PROCESS A QUIT STATEMENT FOR  *   740 00427425        1
*       THE USER. A 15 CHARACTER COMMENT INCLUDING THE USERS NAME AND   *   741 00427450        1
*       PHONE EXTENSION SHOULD BE ENTERED AFTER THE CHARGE NUMBER.      *   742 00427475        1
*                                                                       *   743 00427500        1
*LOCK STATEMENT.                                                        *   744 00428000        1
*SYNTAX.                                                                *   745 00429000        1
*       THE SYNTAX FOR <LOCK STATEMENT> IS AS FOLLOWS:                  *   746 00430000        1
*                                                                       *   747 00431000        1
*         <LOCK STATEMENT> ::= * <LOCK VERB> <SAVE TIME>                *   748 00432000        1
*         <LOCK VERB> ::= LOCK / SAVE / CLOSE                           *   749 00432050        1
*         <SAVE TIME> ::= [AN INTEGER EQUAL TO 90 OR LESS] / [EMPTY]    *   750 00432100        1
*                                                                       *   751 00433000        1
*SEMANTICS.                                                             *   752 00434000        1
*       THE LOCK STATEMENT CLOSES THE OPEN FILE AND RECORDS THE         *   753 00435000        1
*       CURRENT FILE STATUS IN THE DISK DIRECTORY. THE <SAVE TIME>      *   754 00436000        1
*       ALLOWS THE USER TO SPECIFY THE NUMBER OF DAYS THE FILE IS       *   755 00437000        1
*       TO BE HELD ON DISK AFTER THE DATE OF ITS LAST ACCESS.           *   756 00438000        1
*       IF NO SAVE TIME IS SPECIFIED THEN A DEFAULT                     *   757 00439000        1
*       VALUE OF 30 DAYS IS ASSUMED. THE LOCK STATEMENT WILL CAUSE      *   758 00440000        1
*       MINI/TEXT TO GO OUT OF THE ENTER, RESUME OR CHANGE MODE.        *   759 00440100        1
*       AFTER ENTERING THE LOCK STATEMENT THE USER MAY SPECIFY A NEW     *   760 00440200        1
*       FILE NAME IN SUCCEEDING COMMANDS OR CONTINUE TO USE THE CURRENT *   761 00440300        1
*       NAME BY LEAVING THE <FILE NAME> EMPTY.                          *   762 00440400        1
*                                                                       *   763 00441000        1
*QUIT STATEMENT.                                                        *   764 00442000        1
*SYNTAX.                                                                *   765 00443000        1
*       THE SYNTAX FOR <QUIT STATEMENT> IS AS FOLLOWS:                  *   766 00444000        1
*                                                                       *   767 00445000        1
*         <QUIT STATEMENT> ::= * QUIT / * BYE                           *   768 00446000        1
*                                                                       *   769 00447000        1
*SEMANTICS.                                                             *   770 00448000        1
*       THE QUIT STATEMENT CAUSES MINI/TEXT TO GO TO END OF JOB,        *   771 00449000        1
*       ANY FILES THAT WERE IN THE PROCESS OF BEING CONSTRUCTED         *   772 00450000        1
*       ARE MADE PERMANENT.                                             *   773 00451000        1
*;                                                                      *   774 00452000        1
*COMMENT WRITTEN BY                                                     *   775 00453000        1
*         D H BROWN                                                     *   776 00454000        1
*         U S NAVY MINE DEFENSE LABORATORY                              *   777 00455000        1
*         PANAMA CITY, FLORIDA.                                         *   778 00456000        1
*         MARCH 1969                                                    *   779 00457000        1
*;                                                                      *   780 00458000        1
```

```
                    *BEGIN                                                                  *   781 00459000            1
                    *COMMENT THIS IS A MULTI-USER TEXT EDITING PROGRAM FOR THE BURROUGHS     *   782 00460000            1
                    *        B-5500. IT IS WRITTEN WITH THE SAME PHILOSOPHY AS USED IN       *   783 00461000            1
                    *        MINI/TEXT AND IS CALLED MANY/TEXT. DAVE BROWN. 12/17/68;       *   784 00462000            1
                    *COMMENT:                                                               *   785 00462100            1
                    *   TO ADD ADDITIONAL USERS TO MANY/TEXT 3 CHANGES ARE REQUIRED.        *   786 00462200            1
                    *    (1) AT 464000 AND 464200 THE MAXIMUM AND MAXIMUN-1 NUMBER          *   787 00462300            1
                    *        OF USERS MUST BE INDICATED.                                    *   788 00462400            1
                    *    (2) AT 759000-761000 USER FILES MUST BE INITIALIZED.               *   789 00462500            1
                    *    (3) AT 889000 AND 890000 THE REQUIRED NUMBER OF FILES              *   790 00462600            1
                    *        MUST BE DECLARED AND ADDED TO THE SWITCH FILE.;                *   791 00462700            1
                    *DEFINE MIXMAX = % MAXIMUM NUMBER OF USERS ALLOWED                      *   792 00463000            1
PRT(22) = *LIST, LABEL, OR SEGMENT DESCRIPTOR*
PRT(23) = *OUTER BLOCK DESCRIPTOR*
PRT(24) = *SEGMENT DESCRIPTOR*
                                                                                           START OF SEGMENT **********  2
      1  1      *                4#,                                                     *   793 00464000            2
      1  1      *        MIXM1= % MAXIMUM NUMBER OF USERS LESS 1                         *   794 00464100            2
      1  1      *             3#,                                                        *   795 00464200            2
      1  1      *        BUFW = % WORD SIZE NEEDED FOR LARGEST TERMINAL BUFFER           *   796 00465000            2
      1  1      *              8#,                                                       *   797 00466000            2
      1  1      *        BUFC = % CHARACTER SIZE OF LARGEST BUFFER                       *   798 00467000            2
      1  1      *             56#;                                                       *   799 00468000            2
      1  1      *COMMENT GLOBEL DECLARATIONS FOR MANY/TEXT;                             *   800 00469000            2
      1  1      *INTEGER NUSERS, % TWICE NUMBER OF USERS IN MIX                         *   801 00470000            2
PRT(25) = NUSERS
      1  1      *          USER,  % LAST USER SERVICED                                  *   802 00471000            2
PRT(26) = USER
      1  1      *        DLYS,    % DELAY IN READ TO KEEP FROM BURNING UP PROCESSOR     *   803 00472000            2
PRT(230) = DLYS
      1  1      *        NFUL;  % NUMBER OF FULL INPUT BUFFERS WAITING TO BE SERVICED   *   804 00473000            2
PRT(231) = NFUL
      1  1      *INTEGER RESULT,START; % USED BY SCANNER: START IS NUMBEROF CHARACTERS  *   805 00474000            2
PRT(232) = RESULT
PRT(233) = START
      1  1      *COMMENT     THAT HAVE BEEN SCANNED. MEANING OF RESULT IS GIVE BELOW:    *   806 00475000            2
      1  1      *             RESULT    MEANING                                         *   807 00476000            2
      1  1      *               0         UNDEFINED                                     *   808 00477000            2
      1  1      *               1         NUMBER (INTEGER FOUND)                        *   809 00478000            2
      1  1      *               2         NAME FOUND                                    *   810 00479000            2
      1  1      *               3         SPECIAL CHARACTER FOUND                       *   811 00480000            2
      1  1      *               4         ← INCOUNTERED                          ;      *   812 00481000            2
      1  1      *BOOLEAN DEATH, % INDICATES AN ERROR HAS BEEN MADE                      *   813 00482000            2
PRT(234) = DEATH
      1  1      *        USEFUL,% TRUE WHILE AT LEAST ONE USER IS PRESENT               *   814 00483000            2
PRT(235) = USEFUL
      1  1      *        FILESOK,                                                       *   815 00484000            2
PRT(236) = FILESOK
      1  1      *        GOTSOME, % GETSOME GOTSOME INPUT                               *   816 00485000            2
PRT(237) = GOTSOME
      1  1      *        THRUTYPING, % TRUE WHEN THERE IS NO TYPING TO BE DONE          *   817 00486000            2
PRT(240) = THRUTYPING
      1  1      *        STAR; % TRUE IF USER TYPED LEADING * AND HAVE NOT SEEN GM      *   818 00487000            2
PRT(241) = STAR
      1  1      *ALPHA Q,Q1; % PLACE SCANNER PUTS WHAT IT GETS (ANSW)                   *   819 00488000            2
PRT(242) = Q
PRT(243) = Q1
```

```
      1  1      *REAL T,T1,T2; % GLOBEL TEMPS, 11 BITS FREE IN T1                          *   820 00489000                2
PRT(244) = T
PRT(245) = T1
PRT(246) = T2
      1  1      *DEFINE                                                                    *   821 00489100                2
      1  1      *    CHECKTIME=300#, % APROX TIME BETWEEN   STATUS CHECKS (SEC)           *   822 00489200                2
      1  1      *    DLY=DLYS.[33:15]#, % HOLDS DELAY FOR READS & TIMEOUT TIME            *   823 00489225                2
      1  1      *    DLYT=DLYS.[45:3]#, % DELAY TIME FOR READ                             *   824 00489250                2
      1  1      *    NSEC=DLYS.[18:15]#,%APROX TIME SINCE LAST STATUS CHECK               *   825 00489300                2
      1  1      *    ILTME=T1#, % ELAPSED IDLE TIME FOR PROGRAM     (5)                   *   826 00489400                2
      1  1      *    PRTME=T2.[ 1:23]#, % ELAPSED PROCESSOR TIME FOR PROGRAM (2)          *   827 00489500                2
      1  1      *    IOTME=T2.[24:24]#, % ELAPSED I/O TIME FOR PROGRAM       (3)          *   828 00489600                2
      1  1      *    STOR=10#; % NUMBER OF WORDS PER ROW IN INFO                          *   829 00490000                2
      1  1      *ARRAY INFO[0:MIXM1,0:9]; % 0:9 IS 0:STOR-1                             *   830 00491000                2
PRT(247) = INFO
      1  1      *REAL IZ; % USED TO HOLD INFO[USER,0]                                    *   831 00491100          2:0   2
PRT(250) = IZ
      1  1      *COMMENT INFO CONTAINS INFORMATION ABOUT USERS, THE FOLLOWING DEFINES    *   832 00492000          2:0   2
      1  1      *       INDICATE THE USE OF PARTS OF WORDS IN INFO;                      *   833 00493000          2:0   2
      1  1      *DEFINE                                                                  *   834 00494000          2:0   2
      1  1      *    B=BOOLEAN#,R=REAL#,CF=[33:15]#,FF=[18:15]#,                         *   835 00494100          2:0   2
      1  1      *    TUB[TUB1]=R(B(IZ:=INFO[TUB1,0]) AND B("7≤00000"))#, % TU AND BUFFER *   836 00495000          2:0   2
      1  1      *    STARTOG=IZ.[13:1]#, % TELLS IF LEADING * IS REQUIRED OR OPTIONAL     *   837 00495100          2:0   2
      1  1      *    DEFAULT=0#,  % 0=STARTOG INITIALLY RESET, 1=WOULD BE SET.            *   838 00495200          2:0   2
      1  1      *    NEWBASE = IZ.[41:1]#,     % HAVE NEWBASE FROM SEQ SPECS             *   839 00496000          2:0   2
      1  1      *    FULL[FULL1]=IZ:=INFO[FULL1,0] LSS 0#, % TRUE WHEN MESSAGE RECEIVED   *   840 00497000          2:0   2
      1  1      *    CHANGING=B(IZ.[43:1])#,   % USER IS CHANGING                        *   841 00498000          2:0   2
      1  1      *    LABELEQUATED=B(IZ.[44:1])#, %          USER HAS FILE OPEN          *   842 00499000          2:0   2
      1  1      *    ENTERING=B(IZ.[45:1])#, %            USER IS ENTERING TEXT         *   843 00500000          2:0   2
      1  1      *    PATCHING=B(IZ.[46:1])#, %            USER IS ENTERING PATCHES      *   844 00501000          2:0   2
      1  1      *    PROGRAM=B(IZ)#,                    % USERS FILE IS PROGRAM          *   845 00502000          2:0   2
      1  1      *    ROUTINE  = IZ.[33:4]#,           %IN MULTICALL ROUTINE, E.G. COMPILE*   846 00503000          2:0   2
      1  1      *    PLACE    = IZ.[37:3]#,           % PLACE TO GO IN MULTICALL ROUTINE *   847 00504000          2:0   2
      1  1      *    LISTING=ROUTINE=6#,                                                 *   848 00506000          2:0   2
      1  1      *    COPYSEQING=ROUTINE=5#,   % IN COPY OR RESEQUENCE MODE               *   849 00507000          2:0   2
      1  1      *    RUM      = IZ.[18:15]#,          % CURRENT RECORD NUMBER LSS 32768  *   850 00508000          2:0   2
      1  1      *    FLE    =IZ.[3:6]#,             % THE FILE NUMBER ASSIGNED TO USER  *   851 00509000          2:0   2
      1  1      *    TABAMOUNT = INFO[USER,1].[36:5]#,                                   *   852 00509100          2:0   2
      1  1      *    BUFOUT=INFO[USR,1].[33:3]#,% NUMBER OF BUFFERS OUT, USED IN OUTPUTIT*   853 00510000          2:0   2
      1  1      *    INC      = INFO[USER,1].[18:15]#, % STEP SIZE LSS 32768             *   854 00511000          2:0   2
      1  1      *    NCHRS[NCHRS1]=INFO[NCHRS1,1].[41:7]#, % NO. CHRS USER HAS TYPED    *   855 00512000          2:0   2
      1  1      *    SEQNO    = INFO[USER,2]        #,% CURRENT SEQ NO, ≤ 99999999      *   856 00513000          2:0   2
      1  1      *    MID      = INFO[USER,3]#, % PREFIX OF USERS FILE NAME                *   857 00514000          2:0   2
      1  1      *    FID      = INFO[USER,4]#, % SUFFIX OF USERS FILE NAME                *   858 00515000          2:0   2
      1  1      *    JON = INFO[USER,5]#, % 5,6,7 IS CHARGE CODE                         *   859 00515100          2:0   2
      1  1      *    IAOT = INFO[USER,8].[1:23]#, % USERS I/O TIME                       *   860 00515200          2:0   2
      1  1      *    PRCT = INFO[USER,8].[24:24]#, % USERS PROCESSOR TIME                *   861 00515300          2:0   2
      1  1      *    IDLT = INFO[USER,9].[ 1:23]#, % USERS IDLE TIME                     *   862 00515400          2:0   2
      1  1      *    STRT = INFO[USER,9].[24:24]#, % USERS STARTING TIME                 *   863 00515500          2:0   2
      1  1      *    FILESJUSTFIXED=REAL(FILESOK)=3#, %                                  *   864 00515600          2:0   2
      1  1      *    BUMPIDLE=BUMPTIMES(TRUE)#, % BUMP ALL TIMES FOR ALL USERS           *   865 00515700          2:0   2
      1  1      *    BUMPTIME=BUMPTIMES(FALSE)#, % BUMP PRCT AND IAOT ONLY               *   866 00515800          2:0   2
      1  1      *    COUNT    =       Q.[12:6]#, % NUMBER OF CHARACTERS GOTTEN BY SCANNER*   867 00516000          2:0   2
      1  1      *    ANSW = Q#, % USE PRT INSTEED OF ARRAY FOR ANSW                      *   868 00516100          2:0   2
      1  1      *    NUMBER   = RESULT=101#,  % SCANNER FOUND A NUMBER                   *   869 00517000          2:0   2
      1  1      *    NAME     = RESULT=102#,  % SCANNER FOUND A NAME                     *   870 00518000          2:0   2
      1  1      *    BAND(D,P)=B(D) AND B(P)#, RAND(P,D)=R(BAND(P,D))#,                  *   871 00519000          2:0   2
```

```
      1  1      *    BOR(D,P)=B(D) OR B(P)#, ROR(P,D)=R(BOR(P,D))#;                    *   872 00520000      2:0     2
      1  1      *ARRAY DICT[0:37], % DICTIONARY OF WORDS AND SENTENCES                *   873 00521000      2:0     2
PRT(251) = DICT
      1  1      *      JUNK[0:15], % USED INSTEAD OF LOCAL ARRAYS                     *   874 00522000      4:0     2
PRT(252) = JUNK
      1  1      *      USRIBUF[0:MIXM1 ,0:9], % INPUT BUFFER FOR USERS               *   875 00524000      5:3     2
PRT(253) = USRIBUF
      1  1      *      USROBUF[0:MIXM1 ,0:16];                                       *   876 00525000      7:3     2
PRT(254) = USROBUF
      1  1      *COMMENT USEROBUF IS USED TO DO ALL WRITTING TO REMOTES. THE FIRST WORD *   877 00526000      9:2     2
      1  1      *       OF USEROBUF CONTAINS INFORMATION ABOUT MESSAGE. FOLLOWING      *   878 00527000      9:2     2
      1  1      *       DEFINES INDICATE USE OF THIS WORD;                             *   879 00528000      9:2     2
      1  1      *DEFINE                                                               *   880 00529000      9:2     2
      1  1      *    CHIN = USROBUF[USER,0]#;                                         *   881 00530000      9:2     2
      1  1      *    % NOTE: OTHER DEFINES ARE IN OUTPUTIT                            *   882 00531000      9:2     2
      1  1      *ARRAY FILES[0:MIXM1];                                                *   883 00532000      9:2     2
PRT(255) = FILES
      1  1      *COMMENT THIS ARRAY IS USED TO KEEP INFOMATION ABOUT USER FILES. THE   *   884 00533000     11:1     2
      1  1      *       MEANING OF WORDS IN THIS ARRAY IS DEFINED BELOW;               *   885 00534000     11:1     2
      1  1      *DEFINE                                                               *   886 00535000     11:1     2
      1  1      *      ACCS[I]=FILES[I],[6:6]#,    % SERIAL,UPDATE OR RANDOM         *   887 00535100     11:1     2
      1  1      *      FUSR[FUSR1]=FILES[FUSR1],[18:15]#,% FILE USER,  MIXMAX IF AVAILABL*  888 00536000     11:1     2
      1  1      *      FTYP[FTYP1]=FILES[FTYP1],[33:15]#,% INDICATES BUFFERING ON FILE   *  889 00537000     11:1     2
      1  1      *      BUFN[BUFN1]=FILES[BUFN1],[33:3]#, % NUMBER OF BUFFERS             *  890 00538000     11:1     2
      1  1      *      RECL[RECL1]=FILES[RECL1],[36:6]#, % RECORD LENGTH                 *  891 00539000     11:1     2
      1  1      *      BUFZ[BUFZ1]=FILES[BUFZ1],[42:6]#, % BUFFER SIZE                   *  892 00540000     11:1     2
      1  1      *      PTYP="1#<"#,                      % (1,10,30) PROGRAM FILES       *  893 00541000     11:1     2
      1  1      *      DTYP="1≥0"#,                      % (1,15,0) DATA AND PRINT FILES  *  894 00542000     11:1     2
      1  1      *      CTYP="1≥<"#;                      % (1,15,30) COMPILER PRINT FILES*   895 00543000     11:1     2
      1  1      *ARRAY WORD[0:62], % CONTAINS RESERVED WORDS                          *   896 00544000     11:1     2
PRT(256) = WORD
      1  1      *      CBUF[0:MIXM1,0:10]; % STORAGE BUFFER FOR TEXT TO BE CHANGED    *   897 00545000     13:1     2
PRT(257) = CBUF
      1  1      *DEFINE % W VALUES FOR RESERVED WORDS                                 *   898 00545100     15:0     2
      1  1      *  STARV=RESULT=105#, SPECIAL=RESULT=103#, ARROV=RESULT=104#, W=RESULT#, *  899 00545200     15:0     2
      1  1      *  COLNV=W=22#, PLUSV=W=23#, MNUSV=W=24#,                             *   900 00545300     15:0     2
      1  1      *  OKV  =W=25#, PATV =W=26#, PROGV=W=27#,                             *   901 00545400     15:0     2
      1  1      *  DATAV=W=28#, ALGLV=W=29#, XALGV=W=30#,                             *   902 00545500     15:0     2
      1  1      *  BASCV=W=31#, FORTV=W=32#, GOV  =W=33#,                             *   903 00545600     15:0     2
      1  1      *  SYTXV=W=34#, TAPEV=W=35#, NTAPV=W=36#,                             *   904 00545700     15:0     2
      1  1      *  NEWV =W=37#, PRNTV=W=38#,                                          *   905 00545800     15:0     2
      1  1      *  FROMV=W=40#,                BASEV=W=42#,                           *   906 00545900     15:0     2
      1  1      *                INFOV=W=44#, TOV  =W=45#,                          *   907 00546000     15:0     2
      1  1      *  ENDV =W=46#, FRSTV=W=47#, ALLV =W=48#,                             *   908 00546100     15:0     2
      1  1      *  LINEV=W=49#, SAVEV=W=50#, SEQV =W=51#,                             *   909 00546200     15:0     2
      1  1      *  COMAV=W=52#, BYV  =W=53#, INV  =W=54#,                             *   910 00546300     15:0     2
      1  1      *  SPECV=W=55#,                                                       *   911 00546400     15:0     2
      1  1      *  RV=56#, DV=58#, RVM1=55#, THRUV=MNUSV#, STEPV=PLUSV#,              *   912 00546500     15:0     2
      1  1      *  CHNGV=W= 7#, RMOVV=W=12#, LCSV=W=0#, RPLV=W=15#, TABV=W=13#,       *   913 00546600     15:0     2
      1  1      *  NOTOV=W NEQ 45#, NCOMV=W NEQ 52#, NINV=W NEQ 54#;                  *   914 00546700     15:0     2
      1  1      *DEFINE  % TO USE WITH CREATEFILE                                     *   915 00547000     15:0     2
      1  1      *      PFILE="D0≠<"#, % PROGRAM FILE [20:60] (1,10,30)                *   916 00548000     15:0     2
      1  1      *      DFILE="D0#0"#, % DATA FILE    [20:10] (1,10,0)                 *   917 00549000     15:0     2
      1  1      *      CFILE="#09<"#; % CONTROL FILE [10:9] (1,10,30)                 *   918 00550000     15:0     2
      1  1      *DEFINE GETNUMBER(GETNUMBER1) = IF GET(GETNUMBER1) THEN GO BAD#;      *   919 00551000     15:0     2
      1  1      *DEFINE STEPIT=BEGIN OLST:=START; SCANNER END#;                       *   920 00552000     15:0     2
      1  1      *DEFINE                                                               *   921 00554000     15:0     2
```

```
      1  1      *    RENAMEFILE=FILEMAINTENANCE#,                                  *   922 00555000      15!0     2
      1  1      *    REMOVEFILE=FILEMAINTENANCE#,                                  *   923 00556000      15!0     2
      1  1      *    LISTER=LISTBACK#; %                                           *   924 00557000      15!0     2
      1  1      *ALPHA FILE IN B487 14 (2×MIXMAX,BUFW);                            *   925 00558000      15!0     2
PRT(260) = B487
      1  1      *ALPHA FILE OUT TWX 14 (2×MIXMAX,BUFW);                            *   926 00559000      19!0     2
PRT(261) = TWX
      1  1      *FILE GFILE DISK (1,10,30); % GLOBEL DISK FILE                     *   927 00559100      23!0     2
PRT(262) = GFILE
      1  1      *FILE HELP DISK RANDOM "MANYTEX" "HLFIXER" (1,STOR);               *   928 00560000      26!2     2
PRT(263) = HELP
      1  1      *PROCEDURE WRITEHELP;                                              *   929 00561000      30!0     2
PRT(264) = WRITEHELP
      1  2      *    WRITE(HELP[USER],STOR,INFO[USER,*]);                          *   930 00562000      30!0     2
      1  1      *STREAM PROCEDURE MOVE(FROM,TOO,SKF,SKT,CHRS); VALUE SKF,SKT,CHRS; *   931 00563000      36!2     2
PRT(265) = MOVE
      1  2      *BEGIN                                                             *   932 00564000      36!2     2
      1  2      *COMMENT WILL MOVE CHRS CHARACTERS FROM FROM TO TOO AFTER SKIPPING SKF *   933 00565000      37!0     2
      1  2      *       CHARACTERS IN FROM AND SKT IN TOO, SKF, SKT, CHRS MAY BE GTR 63;*   934 00566000      37!0     2
      1  2      *  SI:=LOC SKF; SI:=SI+6;                                          *   935 00567000      37!0     2
      2  2      *  IF SC NEQ "1" THEN SI:=FROM ELSE                                *   936 00568000      37!2     2
      2  2      *    BEGIN SI:=FROM; SI:=SI+32; SI:=SI+32 END;                     *   937 00569000      38!2     2
      2  2      *  SI:=SI+SKF; FROM:=SI; DI:=TOO; DI:=DI+SKT;                      *   938 00570000      39!1     2
      2  2      *  SI:=LOC SKT; SI:=SI+6;                                          *   939 00571000      40!3     2
      2  2      *  IF SC NEQ "1" THEN ELSE                                         *   940 00572000      41!1     2
      2  2      *    BEGIN DI:=DI+32; DI:=DI+32 END;                               *   941 00573000      41!3     2
      2  2      *  SI:=LOC CHRS; SI:=SI+6;                                         *   942 00573100      42!2     2
      2  2      *  IF SC NEQ "1" THEN SI:=FROM ELSE                                *   943 00573200      43!0     2
      2  2      *    BEGIN SI:=FROM; DS:=32CHR; DS:=32CHR END;                     *   944 00573300      44!0     2
      2  2      *  DS:=CHRS CHR                                                    *   945 00573400      44!3     2
      2  2      *END OF MOVE;                                                      *   946 00574000      45!1     2
      1  1      *PROCEDURE SEND(M1,M2); VALUE M1,M2; INTEGER M1,M2;                *   947 00575000      45!2     2
PRT(266) = SEND
      1  2      *BEGIN                                                             *   948 00576000      45!2     2
      1  2      *  INTEGER CN,CR; DEFINE PCRLF=CR#;                                *   949 00577000      45!2     2
                                                                                   START OF SEGMENT **********     3
STACK(F+2) = CN
STACK(F+3) = CR
      2  3      *  INTEGER STREAM PROCEDURE SENDER(D,M,B,CN); VALUE M,CN;          *   950 00578000               3
PRT(267) = SENDER
      2  4      *  BEGIN LOCAL T;                                                  *   951 00579000               3
      2  4      *    SI:=LOC M; DI:=LOC T; DI:=DI+1; DS:=7 CHR;                    *   952 00580000               3
      3  4      *    SI:=D; T(SI:=SI+32; SI:=SI+32); SI:=SI+M; M:=SI;              *   953 00581000        1!0    3
      3  4      *    DI:=B; DI:=DI+CN; SI:=LOC CN; SI:=SI+6;                       *   954 00582000        3!3    3
      3  4      *    IF SC="0" THEN ELSE BEGIN DI:=DI+32; DI:=DI+32 END; SI:=M;    *   955 00582100        5!0    3
      3  4      *    48(IF SC="#" THEN JUMP OUT; DS:=CHR; TALLY:=TALLY+1);         *   956 00583000        6!2    3
      3  4      *    SENDER:=TALLY; DS:=3LIT",≤≠";                                 *   957 00584000        9!0    3
      3  4      *  END SENDER;                                                     *   958 00585000       10!0    3
      2  3      *         CN:=CHIN;                                                *   959 00586000       11!0    3
      2  3      *  IF M1 NEQ 0 THEN                                                *   960 00587000       12!3    3
      2  3      *    BEGIN                                                         *   961 00588000       13!2    3
      2  3      *      IF M1 LSS 0 THEN BEGIN DEATH:=TRUE; CN:=0; CR:=9 END        *   962 00589000       14!0    3
      4  3      *                  ELSE CR:=4;                                     *   963 00590000       17!2    3
      3  3      *      MOVE(DICT,USROBUF[USER,1],0,CN,CR);                         *   964 00591000       18!3    3
      3  3      *      CN:=CN+CR+                                                  *   965 00592000       21!3    3
      3  3      *          SENDER(DICT,M1,USROBUF[USER,1],CN+CR);                  *   966 00593000       22!2    3
      3  3      *      PCRLF:=3 %DO .CRLF AT END OF LINE                           *   967 00594000       27!1    3
```

```
      3  3     *    END;                                                                  *   968 00595000      27:2     3
      2  3     *  IF M2 LSS 0 THEN PCRLF:=3;                                              *   969 00596000      28:0     3
      2  3     *  IF M2 NEQ 0 THEN                                                        *   970 00597000      30:0     3
      2  3     *    CN:=CN+                                                               *   971 00598000      30:3     3
      2  3     *        SENDER(DICT,M2,USROBUF[USER,1],CN);                               *   972 00599000      31:2     3
      2  3     *  CHIN:=CN+PCRLF; THRUTYPING:=FALSE;                                      *   973 00600000      35:3     3
      2  3     *END SEND MESSAGE;                                                        *   974 00601000      39:0     3
                                                                                                3 IS   42 LONG, NEXT SEG    2
      1  1     *DEFINE ERR=SEND#;                                                        *   975 00602000      45:2     2
      1  1     *PROCEDURE SCANNER;  COMMENT FOR MANY/TEXT. 1/18/69 DAVE BROWN;           *   976 00610000      45:2     2
PRT(270) = SCANNER
      1  2     *  BEGIN                                                                  *   977 00611000      45:2     2
      1  2     *    DEFINE BLNKS=S64#,NCHRS=SKP#,ASP=SKP#;      INTEGER BLANKS;          *   978 00612000      45:2     2
                                                                                          START OF SEGMENT **********    4
ERROR IN COL 29:                          =                                                                          175
SYNTAX ERROR. PLEASE TAKE TO PROGRAMMING ASSISTANT.
STACK(F+2) = BLANKS
   1  2  3     *    STREAM PROCEDURE SCAN(INARY,SKP,RESULT,ANSW,BLANKS); VALUE SKP;      *   979 00613000                4
PRT(271) = SCAN
   1  2  4     *     BEGIN  LOCAL S64;   LABEL DBLNK;                                   *   980 00614000                4
   1  2  4     *       COMMENT SKIP OVER PREVIOUSLY SCANNED CHARACTERS;                  *   981 00615000      1:0       4
   1  2  4     *       SI:=LOC SKP; SI:=SI+6;                                            *   982 00616000      1:0       4
   1  3  4     *       IF SC NEQ "1" THEN SI:=INARY ELSE                                 *   983 00617000      1:2       4
   1  3  4     *         BEGIN SI:=INARY; SI:=SI+32; SI:=SI+32 END; SI:=SI+SKP;          *   984 00618000      2:2       4
   1  3  4     *       DI:=ANSW; DI:=DI+3; ASP:=DI; % STORE ANSW POINTER                 *   985 00618100      3:3       4
ERROR IN COL 33:                                  :                                   00612000                          258
MISSING COLON AFTER LABEL.
   2  3  4     *     DBLNK:                                                            *   986 00619000      5:0       4
   2  3  4     *       63(IF SC≠" " THEN JUMP OUT; SI:=SI+1; TALLY:=TALLY+1);          *   987 00620000      5:0       4
   2  3  4     *          IF SC NEQ " " THEN ELSE % HAVE GEQ 64 BLANKS                 *   988 00621000      7:0       4
   2  3  4     *            BEGIN % PUT 64 IN BLANKS, I.E. NUMBER SCANNED DIV 64       *   989 00622000      7:2       4
   2  4  4     *              DI:=BLANKS; DI:=DI+6; DS:=LIT"1";                        *   990 00623000      7:3       4
   2  4  4     *              SI:=SI+1; TALLY:=0; GO DBLNK                             *   991 00624000      8:3       4
   2  4  4     *            END;                                                       *   992 00625000      9:2       4
   2  3  4     *       BLNKS:=TALLY; % NUMBER OF BLANKS SCANNED MOD 64                 *   993 00626000      9:2       4
   2  3  4     *       DI:=RESULT; DI:=DI+7; TALLY:=0;                                 *   994 00627000      9:3       4
   2  3  4     *       COMMENT HAVE A NON BLANK CHARACTER, SEE WHAT IT IS;              *   995 00628000      10:2      4
   2  3  4     *       IF SC≥"0" THEN                                                  *   996 00629000      10:2      4
   2  3  4     *         BEGIN % HAVE AN INTEGER (1)                                   *   997 00630000      11:0      4
   2  3  4     *          DS:=LIT"N"; DI:=ASP;                                         *   998 00631000      11:0      4
ERROR IN COL 30:                                ASP                                   00618100                          250
ILLEGAL STREAM STATEMENT.
   3  4  4     *           12(DS:=CHR; TALLY:=TALLY+1; IF SC<"0" THEN JUMP OUT);       *   999 00632000      12:0      4
ERROR IN COL 63:                                                                 )      00631000                          119
MISSING SEMICOLON OR END.
   4  4  4     *         END                                                           *  1000 00633000      13:1      4
   4  4  4     *        ELSE                                                           *  1001 00634000      13:1      4
   4  3  4     *       IF SC=ALPHA THEN                                                *  1002 00635000      13:2      4
   4  3  4     *         BEGIN % HAVE AN ALPHA NAME (2)                                *  1003 00636000      14:0      4
   4  3  4     *          DS:=LIT"0"; DI:=ASP;                                         *  1004 00637000      14:0      4
ERROR IN COL 30:                                ASP                                   00632000                          250
ILLEGAL STREAM STATEMENT.
   5  4  4     *           12(IF SC=ALPHA THEN DS:=CHR ELSE JUMP OUT; TALLY:=TALLY+1);  *  1005 00638000      15:0      4
ERROR IN COL 69:                                                                 )      00637000                          119
MISSING SEMICOLON OR END.
   6  4  4     *         END                                                           *  1006 00639000      15:1      4
   6  4  4     *        ELSE                                                           *  1007 00640000      15:1      4
```

```
   6  3  4      *          BEGIN % HAVE A SPECIAL CHARACTER (3), IF "←" THEN (4)                                  *  1008 00641000      15:2     4
   6  4  4      *           IF SC="←" THEN DS:=LIT"Q" ELSE                                                        *  1009 00642000      15:2     4
   6  4  4      *           IF SC="*" THEN DS:=LIT"R" ELSE DS:=LIT"P";                                            *  1010 00642100      16:3     4
   6  4  4      *           DI:=ASP; DS:=CHR; TALLY:=1;                                                           *  1011 00643000      18:2     4
ERROR IN COL 18:                    ASP                                                                         00638000                    250
ILLEGAL STREAM STATEMENT.
   7  4  4      *         END;                                                                                    *  1012 00644000      18:3     4
   7  3  4      *       NCHRS:=TALLY; % NUMBER OF CHARACTERS SCANNED                                              *  1013 00645000      18:3     4
ERROR IN COL 15:              :                                                                                 00643000                    175
SYNTAX ERROR. PLEASE TAKE TO PROGRAMMING ASSISTANT.
   8  3  4      *       COMMENT PUT NUMBER CHARACTERS SCANNED IN OUTPUT ARRAY;                                   *  1014 00646000      18:3     4
   8  3  4      *       SI:=LOC NCHRS; SI:=SI+7; DI:=ANSW; DI:=DI+2; DS:=CHR;                                   *  1015 00647000      18:3     4
ERROR IN COL 22:                       ;                                                                      00645000                    175
SYNTAX ERROR. PLEASE TAKE TO PROGRAMMING ASSISTANT.
   9  3  4      *       COMMENT PUT NUMBER OF BLANKS SCANNED (MOD 64) IN BLANKS;                                 *  1016 00648000      19:1     4
   9  3  4      *       SI:=LOC BLNKS; SI:=SI+7; DI:=BLANKS; DI:=DI+7; DS:=CHR;                                 *  1017 00649000      19:1     4
   9  3  4      *     END OF SCAN;                                                                              *  1018 00650000      20:2     4
ERROR IN COL 18:                                                                                              00647000                    267
LABEL UNDEFINED IN GO TO STATEMENT.
  10  2  3      *     ANSW:=0; RESULT:=104;                                                                     *  1019 00651000      20:3     4
  10  2  3      *     IF START=0 OR STAR THEN                                                                   *  1020 00652000      22:2     4
  10  2  3      *       IF R(STAR)="R" THEN % FIND OR REPLACE                                                   *  1021 00652100      23:3     4
  10  2  3      *         SCAN(JUNK[*],START,RESULT,ANSW,BLANKS)                                                *  1022 00652200      25:0     4
  10  2  3      *        ELSE                                                                                   *  1023 00652300      27:1     4
  10  2  3      *         SCAN(USRIBUF[USER,*],START,RESULT,ANSW,BLANKS);                                       *  1024 00653000      27:3     4
  10  2  3      *     IF START=0 AND STARV THEN STAR:=TRUE;                                                     *  1025 00656000      30:3     4
  10  2  3      *     IF START:=START+BLANKS+COUNT GTR 80 OR ARROV THEN STAR:=FALSE;                             *  1026 00657000      33:3     4
  10  2  3      *  END OF SCANNER;                                                                              *  1027 00658000      38:3     4
                                                                                                                   4 IS   41 LONG, NEXT SEG     2
  10  1  1      *BOOLEAN PROCEDURE SEARCHER(PRESENT,TYPE); VALUE PRESENT,TYPE;                                   *  1028 00659000       45:2     2
PRT(272) = SEARCHER
  10  1  2      *     BOOLEAN PRESENT; INTEGER TYPE;                                                            *  1029 00660000       45:2     2
  10  1  2      *  BEGIN                                                                                        *  1030 00661000       45:2     2
  10  1  2      *    DEFINE SRCH=GFILE#;                                                                        *  1031 00662000       45:2     2
                                                                                                              START OF SEGMENT **********     5
  10  2  3      *    ALPHA M,F; LABEL BAD,SRCHIT; INTEGER OLST;                                                 *  1032 00663000                    5
STACK(F+3) = M
STACK(F+4) = F
STACK(F+5) = OLST
  10  2  3      *    DEFINE % ALSO INDICATES MEANING OF TYPE                                                    *  1033 00664000                    5
  10  2  3      *      A=JUNK#, DATA=TYPE=0#, PROG=TYPE=1#,COMPILER=TYPE=2#,                                    *  1034 00665000                    5
  10  2  3      *      PRINT=TYPE=3#,DOESNTMATTER=TYPE=4#, A0=M#, A3=F#, A4=Q#;                                 *  1035 00666000                    5
  10  2  3      *  STEPIT; SCANNER; % LOOK FOR SEPERATOR IN FILE NAME                                           *  1036 00666100                    5
  10  2  3      *  IF Q≠"1/0000"   AND REAL(PRESENT) LEQ 1 THEN % NOT A FILE NAME                              *  1037 00666200       1:3      5
  10  2  3      *    BEGIN % USE THE FILE NAME IN INFO                                                          *  1038 00666300       3:2      5
  10  2  3      *     M:=MID; F:=FID; START:=OLST; STAR:=TRUE; % BACK UP TO WHAT WAS                            *  1039 00666400       4:0      5
  10  3  3      *     IF M GTR 0 THEN GO SRCHIT;  % TO LET NAMES BE FORGOTTEN                                   *  1040 00666500       9:0      5
  10  3  3      *    END;                                                                                       *  1041 00666600      10:1      5
  10  2  3      *  START:=OLST; % BACK UP TO FILE NAME                                                          *  1042 00666700      10:1      5
  10  2  3      *    M:=F:="       ";   SCANNER;                                                               *  1043 00667000      11:0      5
  10  2  3      *    IF (NAME OR NUMBER) AND COUNT LEQ 7 THEN                                                   *  1044 00668000      12:3      5
  10  2  3      *      MOVE(ANSW,M,3,1,COUNT) ELSE GO BAD;                                                      *  1045 00669000      16:0      5
  10  2  3      *    SCANNER;                                                                                   *  1046 00670000      18:3      5
  10  2  3      *    IF Q="1/0000" THEN SCANNER ELSE GO BAD;                                                    *  1047 00671000      19:1      5
  10  2  3      *    IF (NAME OR NUMBER) AND COUNT LEQ 7 THEN                                                   *  1048 00672000      21:0      5
  10  2  3      *      BEGIN                                                                                    *  1049 00673000      24:1      5
  10  2  3      *        MOVE(ANSW,F,3,1,COUNT);                                                                *  1050 00674000      24:3      5
```

```
 10  3  3     *SRCHIT: FILL SRCH WITH M,F;   SEARCH(SRCH,A[*]);                    *  1051 00675000      27:0     5
 10  3  3     *        A0:=A[0]; A3:=A[3]; A4:=A[4];                              *  1052 00676000      31:1     5

######## -INVALD INDEX:ALGOL  /MINI    = 2, S =  48, A =  92:3, EFF INX IS -2  ########################################

 LABEL  000000000LINE   00176271CC COMPILE MINI/TEXT ALGOL LIBRARY;ALGOL FILE CARD=SYMBOL/MANYTEX DISK E ALGOL  /MINI
```

```
? EXECUTE PATCH/MERGE

          PACKET  1068
          INPUT    341 CARDS FROM CRA
          TIME    1702
          DATE   76271 MONDAY, 09/27/76



        *** MESSAGE OF THE DAY ***

            WELCOME TO B5700 TIME SHARING,
            MAY YOUR DATA BE DEPENDABLE AND YOUR PROGRAMS PROVEN.
               FROM THE STAFF OF THE BIG BAD BURROUGHS.

        *** BURROUGHS B5700 TSMCP MARK XVI.0.69  AND INTRINSICS MARK XVI.0.00 ***

         ? EXECUTE PATCH/MERGE
         ?DATA CARD
                5:PATCH/MERGE= 2 BOJ 1702 02/06/75
                CDA IN CARD:PATCH/MERGE= 2
                PBD1069 OUT 011 LINE:PATCH/MERGE= 2
                DKA OUT SER MASTERF 0000000:PATCH/MERGE= 2
                DKA IN SER PATCH TSSMCP:PATCH/MERGE= 2
                DKA OUT SER PATCHDE 0000000:PATCH/MERGE= 2
                DKA REL PATCH TSSMCP:PATCH/MERGE= 2
                CDA REL CARD:PATCH/MERGE= 2
           ?END,
                PATCHDE/0000000/0000000= 0 SEGS--CREATED 09/27/76 AT 00:21:11:11
                DKA REL PATCHDE 0000000:PATCH/MERGE= 2
                PBD1069 REL 011 LINE 30:PATCH/MERGE= 2
                MASTERF/0000000/0000000= 300 SEGS--CREATED 09/27/76 AT 17:02:47:22
                DKA REL MASTERF 0000000:PATCH/MERGE= 2
                PATCH/MERGE= 2,PST= 24 EOJ
                PKT#1068 REMOVED
```

BURROUGHS B-5700 PATCH/MERGE PROGRAM MARK XVI.0.00 MONDAY, 09/27/76, 5:02 PM.

CARD INPUT IS CARD
PATCHES/TSSMCP IS NOT ON DISK
PATCH /TSSMCP WILL BE MERGED

********** CONFLICTS **********

```
     FORM       = 75#,% SWITCH D(PCC)"FORM"="SPECIAL"        20240000 C  846 CONFLICTED WITH:
FORM        = 76#, % SWITCH D(PCC)"FORM"="SPECIAL"          20240000 C  724    DISCARDED

     FORM       = 75#,% SWITCH D(PCC)"FORM"="SPECIAL"        20240000 C  846 CONFLICTED WITH:
FORM       = 77#,% SWITCH D(PCC)"FORM"="SPECIAL"           20240000 C  603    DISCARDED

  "FORM   ",       75 ,     % SWITCH D(PCC)                41485600 C  846 CONFLICTED WITH:
"FORM   ",       76 ,                                      41485600 C  724    DISCARDED
 "FORM   ",       77,      % SWITCH D(PCC)                41485600 C  603    DISCARDED
```

NUMBER OF ERRORS DETECTED = 0.
PROCESSOR TIME =   23 SECONDS.
I/O TIME =   21 SECONDS.

```
? EXECUTE ESPOL/DISK

          PACKET  1070
          INPUT    496 CARDS FROM ZIP
          TIME    1703
          DATE   76271 MONDAY, 09/27/76



        *** MESSAGE OF THE DAY ***

            WELCOME TO B5700 TIME SHARING.
            MAY YOUR DATA BE DEPENDABLE AND YOUR PROGRAMS PROVEN.
               FROM THE STAFF OF THE BIG BAD BURROUGHS.

        *** BURROUGHS B5700 TSMCP MARK XVI.0.69  AND INTRINSICS MARK XVI.0.00 ***

         ? EXECUTE ESPOL/DISK
             ? STACK=1024
             ? CORE= 12000
             ? FILE TAPE= SYMBOL/TSSMCP DISK SERIAL
             ? FILE DISK= TSS/XMCP
             ? FILE STUFF= TSSMCP/XSTUFF
             ? FILE LINE= LINE PRINT OR BACK UP
        ? DATA CARD
              4:ESPOL/DISK= 3 BOJ 1703 06/22/76
              DKA OUT SER CODE 0000000:ESPOL/DISK= 3
              CDB IN CARD:ESPOL/DISK= 3
              DKA IN SER SYMBOL TSSMCP:ESPOL/DISK= 3
              DKA OUT SER TSSMCP XSTUFF:ESPOL/DISK= 3
              DKA OUT SER CODISK 0000000:ESPOL/DISK= 3
             3TI
              TIME FOR ESPOL/DISK= 3 IS 15:17 IN 21:01
              TSSMCP/XSTUFF REMOVED
              TSSMCP/XSTUFF/0000000= 150 SEGS--CREATED 09/15/76 AT 15:37:11:44
              DKA LOK TSSMCP XSTUFF:ESPOL/DISK= 3
              CDB REL CARD:ESPOL/DISK= 3
        ?END.
              DKA REL SYMBOL TSSMCP:ESPOL/DISK= 3
              DKA OUT SER TSS XMCP:ESPOL/DISK= 3
              DKA LOK TSS XMCP:ESPOL/DISK= 3
              CODISK/0000000/0000000= 1800 SEGS--CREATED 09/27/76 AT 17:03:38:53
              DKA REL CODISK 0000000:ESPOL/DISK= 3
              CODE/0000000/0000000= 0 SEGS--CREATED 09/27/76 AT 17:03:28:25
              DKA REL CODE 0000000:ESPOL/DISK= 3
              ESPOL/DISK= 3,PST= 19:05 EOJ
              PKT#1070 REMOVED
```

DAN ROSS

B5700 TEXT EDITOR

R/C

REFERENCE

MANUAL

BY

RON BRODY

12/22/71 VERSION 94

INTRODUCTION

R/C PERMITS A USER AT A REMOTE TELETYPEWRITER TO CREATE AND MAINTAIN SOURCE OR DATA FILES ON THE B5500 SYSTEM DISK. FILE TYPES CREATED AND MAINTAINED WITH R/C ARE: ALGOL, XALGOL, COBOL, FORTRAN, BASIC, AND DATA. THESE FILES REPRESENT NORMAL (80 CHARACTER-PER-CARD) PUNCHED CARD DECKS EXCEPT THAT THEY ARE STORED ON THE DISK. EACH RECORD CAN BE THOUGHT OF AS ONE CARD.

R/C ALLOWS THE USER TO:

* CREATE A VARIETY OF FILES
* RESEQUENCE FILES
* PRINT OR PUNCH FILES
* DELETE OR INSERT RECORDS INTO A FILE
* MODIFY RECORDS WITHIN A FILE
* SCAN A FILE FOR THE OCCURRENCE OF A STRING (OPTIONALLY REPLACING IT WITH ANOTHER STRING)
* REMOVE FILES
* COMPILE FILES
* PERFORM MANY OTHER FUNCTIONS

THE TELETYPEWRITER (IN CONJUNCTION WITH THE PROGRAM R/C) CAN BE CONSIDERED A KEYPUNCH EXTENSION WHICH ELIMINATES PUNCHED CARDS. IT OFFERS GREAT FLEXIBILITY IN FILE HANDLING.

R/C HAS TWO MAJOR RESTRICTIONS:

WHILE THE SEQUENCE NUMBER 99999999 IS PERMITTED FOR CONVENIENCE, THE MAXIMUM SEQUENCE NUMBER IS 2097151 (2*21-1).

THE MAXIMUM NUMBER OF RECORDS PERMITTED IN A FILE IS 8191.

IN THE DISCUSSION OF CERTAIN R/C VERBS, SOME OF THE ELEMENTS OF THE SYNTAX ARE GIVEN AS <M>, <N>, <I>, OR <J>. IN EACH CASE, THESE ELEMENTS REPRESENT INTEGER VALUES WHICH MUST BE PROVIDED BY THE USER. THE VALUE MAY BE A RECORD NUMBER FOR SOME VERBS, THE SEQUENCE NUMBER FOR OTHERS, OR AN INCREMENT AMOUNT. THE BRACKETED CONSTRUCT IS ONLY A FORM OF NOTATION USED TO REPRESENT AN INTEGER PARAMETER.

PROGRAM OPERATION AND RECORD SEQUENCING

DURING CREATION AND FILE MAINTENANCE, R/C AUTOMATICALLY ADVANCES (BY THE CURRENT VALUE OF THE "INCREMENT") THE SEQUENCE NUMBER OF EACH RECORD THAT IS INPUT. THE USER MAY SET THIS INCREMENT TO ANY DESIRED VALUE THROUGH USE OF THE "*INC" VERB (SEE BELOW). THE INCREMENT VALUE IS INITIALIZED TO 100 WHEN THE USER FIRST RUNS R/C.

THE INITIAL SEQUENCE NUMBER IS SET TO THE INCREMENT WHEN A NEW FILE IS OPENED BY A USER. FOR EXAMPLE, IF THE CURRENT INCREMENT WERE 100, THE FIRST SEQUENCE NUMBER IN THE FILE WOULD BE "100:". THIS INITIAL SEQUENCE NUMBER MAY BE CHANGED BY THE USER THROUGH THE USE OF THE CONSTRUCT "*<N>", (SEE BELOW).

AFTER EACH RECORD IS TYPED INTO THE FILE, THE SEQUENCE NUMBER OF THE NEXT RECORD IN THE FILE IS SET TO THE LAST SEQUENCE NUMBER PLUS THE INCREMENT. THIS SEQUENCE NUMBER IS NEXT TYPED ON THE TELETYPEWRITER.

IF THE FILE TYPE IS NOT "COBOL", THE SEQUENCE NUMBER IS FOLLOWED BY A COLON. IF A RECORD ALREADY EXISTS WITH THIS SEQUENCE NUMBER, LEADING ZEROS ARE TYPED AS A WARNING; OTHERWISE LEADING ZEROS ARE SUPPRESSED. AFTER THE SEQUENCE NUMBER IS TYPED OUT, THE USER MAY THEN ENTER THE DESIRED CONTENTS FOR THAT SEQUENCE NUMBER OR MAY ENTER A VERB TO PERFORM SOME OTHER FUNCTION.

ALTERNATIVELY, THE USER MAY SET THE SEQUENCE TO SOME OTHER VALUE THROUGH THE USE OF THE CONSTRUCT "*<N>", WHERE <N> IS THE DESIRED SEQUENCE NUMBER. NOTE THAT LEADING ZEROS ARE ACCEPTED BUT NOT NECESSARY WITH THIS CONSTRUCT.

BY USE OF THE "*<N>" CONSTRUCT AND THE "*INC" VERB TO SET THE SEQUENCE NUMBER INCREMENT, THE USER MAY SET UP HIS OWN NUMBERING SEQUENCE THROUGHOUT HIS FILE.

THE EXAMPLES ILLUSTRATE THE SEQUENCE NUMBER OF THE RECORD TYPED ON THE LEFT SIDE OF THE PAGE AS IT APPEARS ON THE TELETYPEWRITER AND THE MANNER IN WHICH THESE RECORDS ARE INCREMENTED. ALTHOUGH ALGOL, XALGOL, BASIC AND FORTRAN FILES ACTUALLY CARRY THE SEQUENCE NUMBER IN CHARACTER POSITIONS 73-80 OF THE RECORD, R/C TYPES THE NUMBER ON THE LEFT MARGIN OF THE TELETYPEWRITER.

PROGRAM EXECUTION

INITIAL REMOTE TERMINAL OPERATIONS

FOR LOGGING-IN TO A TELETYPEWRITER, PRESS THE "ORIG" BUTTON, WAIT FOR A DIAL TONE FROM THE SPEAKER, AND DIAL THE COMPUTER NUMBER. THE B5500 RESPONDS WITH THE MESSAGE:

BURROUGHS B-5500: <TT>/<BB>

(THE STATION NUMBER IS <TT>/<BB> WHERE <TT> IS THE TERMINAL NUMBER AND <BB> IS THE BUFFER NUMBER.)

YOU MAY TYPE:

? LI: <USERCODE>/<AUTHENTICATION CODE>←

THIS LOG-IN MESSAGE MAY HAVE BEEN PRECEDED BY A "?BO←" MESSAGE WHICH WOULD HAVE BLACKED OUT THE LINE ON WHICH THE LOG-IN MESSAGE WAS TYPED.

THE B5500 VALIDATES THE <USERCODE> AND <AUTHENTICATION-CODE> AND RESPONDS BY TYPING OUT THE STATION NUMBER AND THE TIME OF DAY OF THE LOG-IN.

TO CONNECT A REMOTE TERMINAL TO R/C, ENTER:

?? RUN R/C;END.←

THE B5500 RESPONDS BY EITHER TYPING OUT A "BOJ" (BEGINNING OF JOB) MESSAGE, A "SCHEDULED" MESSAGE, OR A "RUNNING" MESSAGE. A "BOJ" MESSAGE INDICATES THAT R/C WAS NOT PREVIOUSLY RUNNING BUT HAS NOW BEEN ENTERED INTO THE MIX AND IS READY FOR USE. A "SCHEDULED" MESSAGE INDICATES THAT R/C WAS NOT PREVIOUSLY RUNNING AND IS SCHEDULED. IN THIS CASE, R/C IS NOT BROUGHT INTO THE MIX UNTIL OTHER SYSTEM USERS COMPLETE THEIR WORK. THE "RUNNING" MESSAGE INDICATES R/C IS ALREADY IN THE MIX.

WITH R/C IN THE MIX, IT AUTOMATICALLY SEARCHES OUT AND LOCKS ONTO REMOTE TERMINALS WHICH HAVE REQUESTED CONNECTION (BY "RUN R/C"). AS SOON AS YOUR TERMINAL IS LOCKED, R/C TYPES ONE OF THE FOLLOWING MESSAGE SEQUENCES, ACCORDING TO THE MANNER IN WHICH R/C ENDED DURING YOUR LAST R/C RUN (FIRST-TIME USERS ARE CONSIDERED TO HAVE CAUSED NORMAL LAST ENTRIES):

INITIAL MESSAGE AFTER NORMAL TERMINATION OF LAST RUN:

```
<R/C VERSION NUMBER>
HELLO <USERCODE>
:
```

INITIAL MESSAGE AFTER "* END X" OR ABNORMAL LAST RUN TERMINATION:

IF A FILE WAS OPEN

```
<R/C VERSION NUMBER>
<FILE NAME OF OPEN FILE>
HELLO+<USERCODE>
<SEQUENCE NUMBER IN THE OPEN FILE>:
```

OR IF NO FILE IS OPEN

```
<R/C VERSION NUMBER>
HELLO+<USERCODE>
:
```

IF A MESSAGE HAS BEEN SENT FROM ANOTHER USER TO YOUR USERCODE (SEE THE MAIL VERB), "MAIL %" IS TYPED INSTEAD OF "HELLO".

EXAMPLES:

```
VERSION #
HELLO  BLUM
:
```

THIS IS THE NORMAL INITIAL SEQUENCE FROM R/C.

```
VERSION #
```

```
HELLO+ WILNER
:
```

THIS SEQUENCE INDICATES THAT R/C REMEMBERS THE USERS STATE FROM THE PREVIOUS RUN. THE STATE INCLUDES THE INCREMENT VALUE, TAB AMOUNT, SAVE FACTOR, PERCENT ON-OFF, AND VERB REPLACEMENTS.

```
VERSION #
FILE/NAME
HELLO+ SHARPE
00050600:
```

THIS SEQUENCE INDICATES THAT USER HAS THE FILE "FILE/NAME" OPEN AND IS AT THE RECORD WITH SEQUENCE NUMBER 50600. THE LEADING ZEROS INDICATE THAT A RECORD ALREADY EXISTS WITH THAT NUMBER.

FINAL REMOTE TERMINAL OPERATIONS

THE USER SHOULD TERMINATE HIS USE OF R/C WITH THE "* END" COMMAND (SEE BELOW). R/C WILL RESPOND WITH "GOOD BYE."

AFTER HE HAS ENDED R/C, IF THE USER IS DONE WITH THE REMOTE TERMINAL HE SHOULD LOG-OUT BY ENTERING: "?LO←"

R/C INPUT

INPUT TO R/C IS EITHER COMMANDS OR RECORDS.

COMMANDS ARE INDICATED BY THE CHARACTER "*" IN THE FIRST INPUT POSITION. (IF THERE IS NO OPEN FILE, THE "*" IS OPTIONAL.) ALL OTHER INPUT IS CONSIDERED RECORDS TO BE PLACED IN THE OPEN FILE. THE FORMAT FOR A COMMAND IS THE "*" FOLLOWED BY AN R/C VERB AND, IF NEEDED, ITS PARAMETERS. THE VERB AND ITS PARAMETERS MUST BE SEPARATED BY A DELIMITER.

A DELIMITER IS EITHER A SPACE OR ANY SPECIAL CHARACTER EXCEPT A ";", A """, A ".", A "(", OR A "[". MULTIPLE DELIMITERS ARE TREATED AS A SINGLE DELIMITER.

A COMMAND MAY BE FOLLOWED BY ANOTHER INPUT (EITHER A COMMAND OR A RECORD) IF IT IS TERMINATED BY A ";". AN ERROR IN A COMMAND OF A MULTIPLE INPUT INHIBITS THE PROCESSING OF THE REST OF THAT INPUT.

EXAMPLES:

```
100:* INC 50←
    THIS IS AN EXAMPLE OF ONE OF THE INPUT COMMANDS.

100:BEGIN←
    THIS IS AN EXAMPLE OF PLACING A RECORD AT SEQUENCE
NUMBER 100.

100:* INC 3;* RESEQ←
    THIS IS AN EXAMPLE OF MULTIPLE COMMANDS.

100:*35; THIS RECORD GOES AT 35←
    THIS IS AN EXAMPLE OF A COMMAND FOLLOWED BY A RECORD OF
INPUT.

100:* OPEN A/B DATA;* PRINT FOR ME;*CLOSE←
    THIS IS ANOTHER EXAMPLE OF MULTIPLE COMMANDS. NOTE
THAT THE "*" MUST APPEAR IN THE NEXT CHARACTER POSITION
FOLLOWING THE SEMICOLON OR THE REMAINDER OF THE RECORD IS
TREATED AS DATA.
```

AN INPUT LINE IS SENT TO THE COMPUTER BY TYPING THE CHARACTER "←". TYPING ERRORS CAN BE CORRECTED, BY BACKSPACING AND LINE ERASING, BEFORE A MESSAGE IS SENT. THE BACKSPACE CHARACTER IS THE APOSTROPHE (SHIFT 7) AND THE LINE ERASE CHARACTER IS THE UP-ARROW (SHIFT N). ALL THE FOLLOWING LINES OF INPUT ARE EQUIVALENT (NOTE THE UNDERLINED CHARACTERS REPRESENT USE OF THE SHIFT):

```
100:THIS IS IT←

100:THIS IS NOT, BUTNTHIS IS IT←
```

```
100:THE7IS IS IT←
    -
100:THESE777IS IS IT←
       ---
```

IF, AFTER BACKSPACING AND LINE ERASING, THE INPUT LINE CONTAINS MORE THAN 240 CHARACTERS, THE INPUT IS DISCARDED WITH AN "INPUT OVERFLW" ERROR MESSAGE. DATA RECORDS ARE ALSO DISCARDED (WITH THE ERROR MESSAGE) IF THEY ARE TOO LARGE FOR THE FILE. (I. E. GTR 66 FOR COBOL FILES; GTR 80 FOR DATA FILES; AND GTR 72 FOR ALL OTHER FILES)

THERE ARE TWO CLASSES OF REQUESTS TO R/C: LONG AND SHORT. LONG OPERATIONS ARE THOSE THAT USUALLY ARE SLOW TO EXECUTE AND ARE CHARACTERIZED BY THE "WAIT..." MESSAGE. ALL OTHER REQUESTS ARE CLASSIFIED AS SHORT OPERATIONS. LONG OPERATIONS ARE SOMETIMES QUEUED BEFORE THE "WAIT..." MESSAGE, TO BE EXECUTED ONE AT A TIME. SHORT REQUESTS ARE DONE AS THEY ARE RECEIVED. THE USERS IN THE LONG REQUEST QUEUE (AND THE USER PERFORMING A LONG OPERATION IF IT IS NOT TYPING ON THE REMOTE) PERIODICALLY RECEIVE A FEW "RUBOUT" CHARACTERS OF REASSURANCE. R/C IGNORES ANY INPUT SENT BY USERS IN THE QUEUE OR BY THE USER WHOSE LONG OPERATION IS BEING PROCESSED.

IF A USER PRODUCES NO INPUT FOR FIVE MINUTES, HE IS SENT THE MESSAGE "LOOK ALIVE". IF HE DOES NOT RESPOND WITHIN ANOTHER FIVE MINUTE PERIOD, R/C PROCESSES A "* END DS" FOR THAT USER.

R/C OUTPUT

OUTPUT TO THE TELETYPEWRITER OF THE SPECIAL CHARACTERS ←, ≠, <, ≤, >, AND ≥ IS REPLACED BY A "$" CHARACTER IN ORDER THAT THEY DO NOT EVOKE TELETYPWRITER CONTROL FUNCTIONS WITH WHICH THEY ARE ASSOCIATED. (THESE INCLUDE LINE-FEED, CARRIAGE-RETURN, MESSAGE-END, AND PAPER-TAPE-ON.)

WHEN THE "BREAK" KEY IS DEPRESSED DURING OUTPUT, THE OUTPUT IS TERMINATED WITH THE MESSAGE "BREAK".

R/C FILES

ALL FILES CREATED BY R/C ARE PERMANENT DISK FILES. THE SAVE FACTOR IS NORMALLY 7 DAYS, BUT IT MAY BE CHANGED BY THE SAVE VERB (SEE BELOW).

FILE TYPES

R/C ENABLES THE USER TO CREATE AND MAINTAIN ALGOL, COBOL, FORTRAN, XALGOL, BASIC, AND DATA FILES. THESE FILES HAVE 80 CHARACTER-LONG RECORDS, (ONE CARD IMAGE).

XALGOL, BASIC, ALGOL AND FORTRAN FILES CONTAIN EIGHT DIGIT SEQUENCE NUMBERS LOCATED IN THE POSITIONS 73-80 OF THE CARD IMAGE.

COBOL FILES CONTAIN SIX-DIGIT SEQUENCE NUMBERS, PLACED IN POSITIONS 1-6 OF THE RECORD.

DATA FILES ARE NOT PHYSICALLY SEQUENCED ALTHOUGH R/C MAINTAINS AN INTERNAL, EIGHT-DIGIT NUMBER FOR EACH RECORD.

FILE NAMES

FILE NAMES MUST BE SUPPLIED TO R/C. THE FORM OF A NAME IS <FILE PREFIX> / <FILE SUFFIX>. THROUGHOUT THIS DOCUMENT, <FILE-NAME> IS USED TO SPECIFY A FILE AND SHOULD BE IN THE FORM ABOVE. THE <FILE PREFIX> AND THE <FILE SUFFIX> MAY EACH BE NO LONGER THAN SEVEN CHARACTERS.

EXAMPLES:

```
A/B
GRIMY/GULCH
ZAP/1
16JAN/SUFFIX
0000000/DISK
```

RECORD REFERENCING

RECORDS IN THE OPEN FILE (SEE OPEN BELOW) ARE REFERRED TO BY THEIR SEQUENCE NUMBER. "DATA" FILES ARE IMPLICITLY SEQUENCED BY THE VALUE OF THE INCREMENT WHEN THEY ARE OPENED.

AN ALTERNATE METHOD OF REFERENCING RECORDS IN THE OPEN FILE IS RELATIVE SEQUENCE NUMBERS. A RELATIVE SEQUENCE NUMBER IS AN INTEGER PRECEEDED BY A + OR - SIGN. IT MAY BE USED ANYPLACE A SEQUENCE NUMBER IS USED. IT IS TRANSLATED TO A SEQUENCE NUMBER BY MOVING FORWARD OR BACKWARD THE INDICATED NUMBER OF RECORDS AND USING THE SEQUENCE NUMBER OF THAT RECORD.

RECORDS IN A NON-OPEN FILE (EXTERNAL FILE) ARE REFERRED TO BY THEIR RELATIVE POSITION WITHIN THE FILE. THE FIRST RECORD IS 1, THE SECOND 2, ETC. ANY SEQUENCING THAT MAY BE ON THE RECORDS IS IGNORED.

FILE-HANDLING VERBS

THIS SECTION DESCRIBES VERBS THAT HANDLE FILES AS A WHOLE, RATHER THAN RECORDS WITHIN A FILE. HOWEVER, A FEW VERBS HAVE OPTIONS IN R/C SYNTAX THAT PERMIT ACCESS TO RECORDS WITHIN THE FILE. THE COMPLETE SYNTAX IS GIVEN FOR EACH VERB AS WELL AS A DISCUSSION AND EXAMPLES OF ITS USE. AN ASTERISK ("*") MUST ALWAYS BE THE FIRST CHARACTER IN THE INPUT STRING WHEN A COMMAND IS ENTERED. IF THIS IS NOT FOLLOWED, AN EXISTING RECORD MAY BE OVERWRITTEN BY THE COMMAND ITSELF.

FILE OPENING AND CREATION (OPEN)

```
* OPEN <FILE-NAME> <FILE-TYPE> NEW

* OPEN <FILE-NAME> <FILE-TYPE> OLD

* OPEN <FILE-NAME> <FILE-TYPE>
```

THE "* OPEN " VERB ATTACHES THE USER TO THE DISK FILE <FILE-NAME>. THE <FILE-TYPE> MUST BE EITHER "ALGOL", "COBOL", "FORTRAN", "XALGOL", "BASIC", OR "DATA". IF THE <OPEN TYPE> IS "NEW", A NEW DISK FILE IS CREATED. IF THE <OPEN TYPE> IS "OLD", THE DISK FILE <FILE-NAME> IS OPENED AND RESEQUENCED BY THE CURRENT VALUE OF THE INCREMENT. IF THE <OPEN TYPE> IS NEITHER "NEW" NOR "OLD" THE DISK FILE <FILE-NAME> IS OPENED AND IT IS READ TO DETERMINE ITS SEQUENCE NUMBERS. THIS LATTER FORM IS SLOWER THAN THE OPEN "OLD".

EXAMPLES:

```
:* OPEN PROGRAM/SOURCE ALGOL NEW←
   100:
```

THIS CREATES A NEW DISK FILE CALLED PROGRAM/SOURCE.

```
63500:* OPEN ANOTHER/PROG DATA OLD←
 4500:
```

THIS OPENS THE FILE ANOTHER/PROG SEQUENCING IT BY THE CURRENT INCREMENT VALUE. NOTE THAT THE FILE THAT WAS OPEN IS FIRST CLOSED BEFORE THE NEXT FILE IS OPENED.

```
:* OPEN YET/ANOTHER COBOL←
WAIT...
READ ONLY FILE.
  7504
```

THIS OPENS THE FILE YET/ANOTHER USING THE SEQUENCE NUMBERS WITHIN THE FILE. THE MESSAGE "READ ONLY FILE"

INDICATES THAT THE USER IS FORBIDDEN (BY THE FILE SECURITY SYSTEM) TO MODIFY THE FILE.

ERRORS:

DUP FILE: <FILE-NAME>

A FILE, <FILE-NAME>, ALREADY EXISTS AND THE USER IS TRYING TO CREATE A FILE WITH THAT NAME WITH AN "* OPEN ... NEW".

NO FILE: <FILE-NAME>

THE USER IS TRYING TO OPEN A FILE, <FILE-NAME>, AND IT DOES NOT EXIST ON DISK.

BAD FILE: <FILE-NAME>

THE FILE <FILE-NAME> WHICH THE USER IS TRYING TO OPEN IS NOT BLOCKED CORRECTLY. THE CORRECT BLOCKING IS 10-WORD RECORDS WITH A MULTIPLE OF 3 RECORDS PER BLOCK.

INV USER: <FILE-NAME>

THE USER IS TRYING TO OPEN A FILE TO WHICH HE HAS NO ACCESS. IF THE USER HAS EITHER SECONDARY OR TERTIARY ACCESS, THE MESSAGE: "READ ONLY FILE" IS TYPED.

FILE TOO LONG

THE USER IS TRYING TO OPEN A FILE WITH MORE THAN 8191 RECORDS.

SEQ OVERFLOW

THE FILE THE USER IS OPENING CAUSES THE SEQUENCE COUNTER TO EXCEED 2,097,151. THE FILE IS OPENED, BUT THE USER SHOULD RESEQUENCE IT.

SEQ ERR=<N>

THE FILE CONTAINS A RECORD WHOSE SEQUENCE NUMBER <N> IS LESS THAN THE SEQUENCE NUMBER OF THE PRECEEDING RECORD. THE FILE IS NOT OPENED.

BAD FILE TYPE: <FILE-TYPE>

<FILE-TYPE> IS NOT "ALGOL", "COBOL", "FORTRAN", "XALGOL", "BASIC", OR "DATA".

FILE CLOSING (CLOSE)

FILES ARE CLOSED BY USE OF THE FOLLOWING CONSTRUCT:

```
* CLOSE
```

THIS VERB DETATCHES THE OPEN FILE, FROM R/C.

EXAMPLES:

```
5600:* CLOSE←
:
```

THIS IS AN EXAMPLE OF CLOSING A FILE THAT IS IN THE CORRECT ORDER.

```
450:* RESEQ 100←
9000:* CLOSE←
WAIT...
:
```

THIS IS AN EXAMPLE OF CLOSING A FILE THAT IS NOT IN ORDER.

ERROR:

NO FILE OPEN: CLOSE

THERE IS NO OPEN FILE TO CLOSE.

LISTINGS ON THE TELETYPEWRITER (LIST)

TO LIST FILES OR ANY OF THEIR SEPARATE RECORDS, THE FOLLOWING CONSTRUCTS APPLY:

* LIST

* LIST <FILE-NAME>

* LIST <FILE-NAME> NO

* LIST <FILE-NAME> <M>

* LIST <FILE-NAME> <M> <N>

* LIST <M>

* LIST <M> <N>

THE "LIST" VERB CAUSES AN ENTIRE FILE OR PORTIONS OF A FILE TO BE LISTED ON THE TELETYPEWRITER. LISTING MAY BE DISCONTINUED BY PRESSING THE BREAK KEY ON THE TELETYPEWRITER.

THE FIRST FORM LISTS THE OPEN FILE.

THE SECOND FORM LISTS THE FILE <FILE-NAME>.

THE THIRD FORM LISTS THE FILE <FILE-NAME>, WITHOUT THE RECORD NUMBERS.

THE FOURTH FORM LISTS <FILE-NAME> FROM THE <M>TH RECORD TO THE END.

THE FIFTH FORM LISTS <FILE-NAME> FROM THE <M>TH TO THE <N>TH RECORDS.

THE SIXTH FORM LISTS SEQUENCE NUMBER <M> OF THE OPEN FILE.

THE LAST FORM LISTS SEQUENCE NUMBERS <M> THROUGH <N> OF THE OPEN FILE.

EXAMPLES:

```
        500:* LIST←
        100:BEGIN
        200: INTEGER I, J, K ;
        300: REAL X, Y, Z ;
        400: ARRAY A [0 : 9] ;
        500:

       5500:* LIST 8900,+3←
       8950:      I := I + 5 ;
       9125:      GO TO NEXT ;
       9300: HELP:
       9400:

        300* LIST 60←
      000060     MOVE A TO B.
      000070

      :* LIST SOME/FILE←
           1:BEGIN
           2; INTEGER I, J, K ;
           3: REAL X, Y, Z ;
           4: ARRAY A [0 : 9] ;
           5:       A [I] := X ;
           6:END.
      :

        500:*LIST SOME/FILE NO←
      BEGIN
       INTEGER I, J, K ;
       REAL X, Y, Z ;
       ARRAY A [0 : 9] ;
          A [I] := X ;
      END.
        500:
```

```
:* LIST LIBRARY/FILE 2,4←
      2:PROCEDURE READDATA 567,653
      3:PROCEDURE WRITEDATA 654,789
      4:PROCEDURE DATA 790,808
:

:* LIST SOME/FILE 5←
      5:     A [I] := X ;
      6:END.
:

:* LIST SOME/FILE 200,500←
USE RECORD #S.
:
```

THE LAST EXAMPLE ILLUSTRATES THE COMMON ERROR OF REFERENCING AN EXTERNAL FILE WITH SEQUENCE NUMBERS INSTEAD OF RECORD NUMBERS.

COMPRESSED FILE LISTINGS (QUICK)

A COMPRESSED FILE LISTING MAY BE OBTAINED FROM R/C BY USE OF THE CONSTRUCTS:

```
* QUICK
- -----

* QUICK <FILE-NAME>
- ----- -----------

* QUICK <FILE-NAME> NO
- ----- ----------- --

* QUICK <FILE-NAME>  <M>
- ----- -----------  ---

* QUICK <FILE-NAME> <M> <N>
- ----- ----------- --- ---

* QUICK <M>
- ----- ---

* QUICK <M> <N>
- ----- --- ---
```

THE "* QUICK" VERB LISTS ON THE TELETYPEWRITER DELETING ALL CONTIGUOUS BLANKS EXCEPT THE FIRST. THE FILE IS NOT AFFECTED BY THE VERB.

EXAMPLE:

```
 4500:* LIST 4300,4400←
 4300:          FOR I :=     A STEP -1 UNTIL 0   DO
 4400:            X [I]   :=   SIN  (Y) ;
 4500:* QUICK -2, + 1←
 4300: FOR I := A STEP -1 UNTIL 0 DO
 4400: X [I] := SIN (Y) ;
 4500:
```

FILE REMOVAL (REMOVE)

TO REMOVE A FILE USE THE FOLLOWING CONSTRUCT:

* REMOVE <FILE-NAME>

THE REMOVE VERB REMOVES THE FILE <FILE-NAME> FROM DISK.

* REMOVE LISTING

REMOVES LINE/<USERCODE>, THE LISTING FILE FROM THE LAST COMPILATION.

EXAMPLES:

```
 3200:* REMOVE A/B←
 3200:

  546:* REMOVE ANOTHER/FILE←
NO FILE: ANOTHER/FILE
  546:

:* OPEN EXAMPLE/X COBOL OLD←
46500* REMOVE EXAMPLE/X←
:
```

LINE-PRINTER FILE REPRODUCTION (PRINT)

THE PRINT VERB:

*PRINT <A> <B>

PRINTS THE OPENED FILE ON THE LINE PRINTER (LABELED <A> <B>). IF "<B>" IS "DOUBLE" THEN DOUBLE SPACING IS USED.

* PRINT <A> <B> <M>

AS ABOVE, STARTING WITH SEQUENCE NUMBER <M>.

* PRINT <A> <B> <M>, <N>

AS ABOVE, STOPPING WITH SEQUENCE NUMBER <N>.

EXAMPLES:

```
:* OPEN TEST/CASE DATA;* PRINT TC DOUBLE;* CLOSE.
:
```

THIS EXAMPLE ILLUSTRATES AN INSTANCE WHERE A SEQUENCED FILE SHOULD BE TREATED AS DATA TO SHORTEN THE OPERATION. IF THE FILE WAS OPENED "ALGOL OLD" IT WOULD HAVE BEEN RESEQUENCED BY THE CURRENT VALUE OF THE INCREMENT AND THEN WHEN IT WAS CLOSED IT WOULD HAVE BEEN RECOPIED. IF IT WAS OPENED "ALGOL" IT WOULD HAVE BEEN READ TO DETERMINE ITS SEQUENCE NUMBERS. EITHER WAY WOULD HAVE MADE THE WHOLE OPERATION MUCH SLOWER THAN OPENING THE FILE "DATA".

```
   8700:* PRINT FOR USER.
WAIT...
   8700:
```

PUNCHED-CARD FILE REPRODUCTION (PUNCH)

TO PUNCH A FILE:

* PUNCH <A> <B>

PUNCHES A CARD DECK (LABELED <A> <B>) OF THE OPENED FILE.

* PUNCH <A> <B> <M>

AS ABOVE, STARTING WITH SEQUENCE NUMBER <M>.

* PUNCH <A> <B> <M>, <N>

AS ABOVE, STOPPING WITH SEQUENCE NUMBER <N>.

EXAMPLE:

```
7600:* PUNCH A B 100,+10←
 WAIT...
7600:
```

FILE COMPILATION (COMPILE)

FILES MAY BE COMPILED TO THE B5500 LIBRARY BY THE FOLLOWING CONSTRUCT:

```
*COMPILE <FILE-NAME>
```

THIS VERB INITIATES THE COMPILATION OF THE OPEN FILE TO LIBRARY USING THE COMPILER INDICATED IN THE OPEN STATEMENT. THE OBJECT CODE IS NAMED <FILE-NAME>. THE LISTING OUTPUT OF THE COMPILATION IS EQUATED TO "LINE/<USERCODE>" ON DISK. THE "* LISTING" VERB MAY BE USED TO LIST THE SYNTAX ERRORS.

```
* COMPILE <FILE-NAME> <COMPILER>
```

COMPILES THE OPEN FILE USING THE SPECIFIED COMPILER.

EXAMPLES:

```
    5700:* COMPILE OBJECT/CODE←
WAIT...
:

     479:* COMPILE TEST/OBJECT EZTRAN←
QUEUED( 1).WAIT...
:
```

IN THE LAST EXAMPLE, THE "EZTRAN" COMPILER (EZTRAN/DISK) WILL BE USED. IF THE FILE IS NOT IN ORDER, IT WILL BE REORDERED. SINCE THIS IS A LONG OPERATION THE USER GETS A "WAIT" MESSAGE. THE "QUEUED" MESSAGE INDICATES THAT ANOTHER USERS LONG OPERATION IS BEING PROCESSED AND THAT THIS LONG OPERATION IS QUEUED UNTIL THE OTHER IS DONE. THE "1" INDICATES THAT THIS IS THE FIRST REQUEST IN THE QUEUE.

OUTPUT OF THE COMPILATION (LISTING)

THE LISTING FILE OF THE COMPILER IS EQUATED TO LINE/<USERCODE> ON DISK. THE FILE MAY BE ACCESSED BY THE USE OF THE LISTING VERB:

```
* LISTING <FILE-TYPE> <S>, <L>, <U>
```

LISTS THE SEQUENCE NUMBERS RELATED TO SEGMENT <S> FROM RELATIVE ADDRESS <L> TO RELATIVE ADDRESS <U>. <FILE-TYPE> IS ALGOL, XALGOL, BASIC, COBOL, OR FORTRAN AND INDICATES WHICH COMPILER CREATED THE LISTING FILE "LINE/<USERCODE>". (THIS FILE IS AUTOMATICALLY GENERATED BY THE COMPILE VERB).

```
* LISTING <FILE-TYPE> ERRORS
```

LISTS THE SYNTAX ERRORS OF YOUR LAST COMPILATION.

```
* LISTING
```

PRINTS THE LINE FILE OF YOUR LAST COMPILATION ON THE PRINTER.

EXAMPLES:

```
:* LISTING ALGOL 5, 25, 35←
WAIT... SEGMENT =      5:
    4300: REL. ADDR. =   26.
    4400: REL. ADDR. =   29.
    9200: REL. ADDR. =   32.
    9300: REL. ADDR. =   35.
:* LISTING ALGOL ERRORS←
WAIT...
    7800:ERROR 100   I.
:

    8900:* LISTING←
    8900:
```

FILE ZIPPED AS AN "EXECUTE" DECK (ZIP)

TO IMPLEMENT THE B5500 ZIP FUNCTION THROUGH R/C, USE THE FOLLOWING CONSTRUCT:

```
* ZIP
- ---
```

THIS CONSTRUCT ZIPS THE OPENED FILE AFTER IT LINKS ALL THE CONTROL CARDS AS INDICATED BY "?". SEE THE ALGOL REFERENCE MANUAL FOR A DESCRIPTION OF THE "ZIP WITH FILE-ID" STATEMENT.

```
* ZIP <FILE-NAME>
- --- -----------
```

COPIES THE OPENED FILE CREATING <FILE-NAME> AND ZIPS <FILE-NAME>. NOTE THIS "ZIP" CONSTRUCT DOES NOT DESTROY THE OPEN FILE, AS DOES THE FIRST FORM.

EXAMPLES:

```
:COLUMN ON "@";COLUMN 73←
:OPEN MAKE/MANUAL DATA NEW;%%EXECUTE XREF/JONES.←
     200:%%FILE DISK = TEACHER/0000094.←
     300:%% DATA CARD.←
     400:$ DISK SIX DOCONLY DOCUMENT FINAL←
     500:@99999999←
     600:%% END.←
     700:* ZIP TEMP/NAME←
WAIT...
     700:
```

THIS EXAMPLE ILLUSTRATES HOW TO CREATE A CONTROL DECK AND INITIATE ITS EXECUTION. THE DECK WAS SAVED (UNDER THE NAME "MAKE/MANUAL") SINCE THE ZIP CONSTRUCT INCLUDED THE DUMMY FILE "TEMP/NAME". (NOTE THAT THE ABOVE DECK WILL CREATE AN R/C USERS MANUAL.)

```
:* OPEN MAKE/MANUAL DATA;* ZIP T/N;* CLOSE←
WAIT...
:
```

RECORD HANDLING VERBS

RECORD COPYING (DITTO)

RECORDS MAY BE COPIED FROM ONE PLACE TO ANOTHER WITHIN A FILE BY THE CONSTRUCT:

```
* DITTO <M>
```

COPIES CARD IMAGE <M> AS THE NEXT RECORD.

```
* DITTO <M>, <N>
```

COPIES THE CARD IMAGES <M> TO <N> AS THE NEXT RECORDS.

```
* DITTO OVERITE ON

* DITTO OVERITE OFF

* DITTO OVERITE
```

IF THE DITTO OVERITE IS OFF AND AN EXISTING RECORD IS ABOUT TO BE OVERWRITTEN, THE DITTO TERMINATES WITH AN "OVERITE OFF" MESSAGE. (IT IS INITIALLY OFF.) THE ABOVE COMMANDS ARE USED TO SET THE OPTION AND TO PRINT ITS CURRENT SETTING. ITS SETTING MAY BE REVERSED FOR ONE COMMAND BY PREFIXING THE DITTO WITH A ". (E.G. *"DITTO 10,50.)

```
* DITTO <M> MOVE

* DITTO <M>, <N> MOVE
```

THE MOVE OPTION, MOVES RECORDS <M> THRU <N> TO THE

CURRENT LOCATION. THIS OPTION IS VERY FAST, BUT OVERITING IS NOT ALLOWED.

EXAMPLES:

```
500:* DITTO 200,300←
700:* LIST←
100:ONE
200:TWO
300:THREE
400:FOUR
500:TWO
600:THREE
700:* DITTO 200,300 MOVE;*LIST←
100:ONE
400:FOUR
500:TWO
600:THREE
700:TWO
800:THREE
900:
```

COPYING EXTERNAL FILES (COPY)

WHOLE OR PARTIAL EXTERNAL FILES MAY BE COPIED INTO THE CURRENTLY OPENED FILE BY THE CONSTRUCTS:

* COPY <FILE-NAME>

* COPY <FILE-NAME> <M>

* COPY <FILE-NAME> <M>, <N>

* COPY <FILE-NAME> MERGE

THE COPY VERB COPIES RECORDS FROM ANOTHER FILE (<FILE-NAME>).

THE FIRST FORM COPIES THE WHOLE FILE. THE SECOND FORM COPIES THE <M>-TH RECORD (WHERE THE FIRST RECORD OF <FILE-NAME> IS 1, THE SECOND RECORD IS 2, THE THIRD IS 3, ETC.). THE THIRD FORM COPIES THE <M>-TH THROUGH THE <N>-TH RECORDS. THE LAST FORM USES THE SEQUENCE NUMBERS OF THE RECORDS OF <FILE-NAME> TO DETERMINE THEIR POSITION IN THE OPEN FILE.

* COPY OVERITE ON

* COPY OVERITE OFF

* COPY OVERITE

IF THE COPY OVERITE IS OFF AND AN EXISTING RECORD IS ABOUT TO BE OVERWRITTEN, THE COPY TERMINATES WITH AN "OVERITE OFF" MESSAGE. (IT IS INITIALLY OFF.) THE ABOVE

COMMANDS ARE USED TO SET THE OPTION AND TO PRINT ITS CURRENT SETTING. ITS SETTING MAY BE REVERSED FOR ONE COMMAND BY PREFIXING THE COPY WITH A ". (E.G. *"COPY A/B 10,50.)

EXAMPLES:

```
:* OPEN SOURCE/MARK7 ALGOL NEW←
     100:* COPY SOURCE/MARK6←
WAIT...
    8900:

     500:* COPY LIBRARY/FILE 345, 368←
WAIT...
    2800:

:QUICK PATCH/FILE←
       1:A 00000050
       2:B 00000150
       3:C 00000200
       4:X 00000175
       5:Y 00000250
       6:Z 00000250
:OPEN MY/SOURCE ALGOL OLD←
     400:* COPY OVERITE ON;*COPY PATCH/FILE MERGE←
     400:* LIST←
      50:A
     100:1
     150:B
     175:X
     200:C
     250:Z
     300:3
     400:
```

INTRA-RECORD EDITING (INLINE)

RECORDS MAY BE EDITED BY USE OF THE "* INLINE" CONSTRUCTS DESCRIBED IN THE NEXT PARAGRAPHS.

```
* INLINE <M>
```

```
* INLINE
```

```
* INLINE <M> <EDIT CHR>
```

```
* INLINE <EDIT CHR>
```

THIS SETS UP LINE <M> FOR INLINE EDITING. IF THE SEQUENCE NUMBER <M> IS NOT INCLUDED WITH THE COMMAND, THE PREVIOUS RECORD IS USED AND THE INITIAL PRINTING OF IT IS SUPPRESSED.

```
* INLINE ECHO ON
```

```
* INLINE ECHO OFF
```

```
* INLINE ECHO
```

THE MODIFIED RECORD WILL BE TYPED ON THE TELETYPE IF THE INLINE ECHO IS ON. (IT IS INITIALLY ON.) THE ABOVE COMMANDS ARE USED TO SET THE OPTION AND TO PRINT ITS CURRENT SETTING. ITS SETTING MAY BE REVERSED FOR ONE COMMAND BY PREFIXING THE INLINE WITH A -. (E.G. * -INLINE +3.)

TO MODIFY A PORTION OF A RECORD (CARD IMAGE) USE THE "* INLINE" VERB. R/C PRINTS THE RECORD NUMBER AND THE LINE, THEN GIVES A CARRIAGE RETURN AND LINE FEED. IT NEXT SPACES THE PRINT

BALL (ON THE NEW LINE) DIRECTLY BELOW THE FIRST CHARACTER POSITION OF THE OLD LINE, ABOVE. THE USER SPACES THE PRINT BALL TO THE PROPER POSITION AND ACCOMPLISHES THE FOLLOWING ACTIONS:

TO INSERT A STRING, TYPE THE LETTER "I" FOLLOWED BY THE STRING AND A "←".

TO DELETE A STRING, USE THE LETTER "D" FOLLOWED BY SPACES UNDER THE CHARACTERS TO BE DELETED AND THEN A "←".

TO REPLACE A STRING, USE THE LETTER "R" FOLLOWED BY THE NEW STRING OF THE SAME LENGTH AND A "←".

IF THE MODIFICATION IS TO BE DONE IN THE FIRST CHARACTER, THEN INCLUDE THE "I", "R", OR "D" WITH THE INLINE COMMAND.

EXAMPLES:

TO INSERT CHARACTERS INTO A RECORD:
-- ------ ---------- ---- - -------

```
    300:*INLINE 30500←
  30500:ABCDEFGHIJK
00030500:  I12345←
  30500:ABC12345DEFGHIJK
```

TO DELETE CHARACTERS FROM A RECORD:
-- ------ ---------- ---- - -------

```
   7700:*INLINE 67700←
  67700:ABCDEFGHIJKLMNOP
00067700: D     ←
  67700:ABHIJKLMNOP
```

TO REPLACE CHARACTERS IN A RECORD:
-- ------- ---------- -- - -------

```
    600:* INLINE 551←
    551:ABCDEFGHIJKLMNOPQRST
00000551:R12345←
    551:A12345GHIJKLMNOPQRST
```

TO MODIFY A RECORD BEGINNING WITH ITS FIRST CHARACTER:
-- ------ - ------ --------- ---- --- ----- ----------

```
00006700:*INLINE 70800 D←
  70800:ABCDEFGHIJKLM
00070800:    ←
  70800:DEFGHIJKLM
  70900:* INLINE D; ←
  70800:EFGHIJKLM
```

THE "R" AND "I" FORMS OF THIS LATTER CONSTRUCT OPERATE IN A SIMILAR FASHION.

STRINGS

R/C SCANS FOR THE OCCURRENCE OF A STRING BY USE OF THE "* SCAN" AND "* CHANGE" VERBS. IN THIS APPLICATION THE FOLLOWING DEFINITION OF "STRING" APPLIES:

```
<STRING> ::= <LEFT> <TEXT> <RIGHT> <COLUMN RANGE>
        <LEFT> ::=  "/ ( / [ / .
        <RIGHT> ::= "/ ) / ] / .
        <TEXT> ::= ≤ANY STRING OF CHARACTERS
                   EXCLUDING THE <RIGHT>  CORRESPONDING
                   TO THE <LEFT> CHARACTER USED.≥
        <COLUMN RANGE> ::= ≤EMPTY≥ /
                   @<INTEGER> / @<INTEGER>-<INTEGER>
```

IF THE STRING IS DELIMITED BY PERIODS, AN ADDED CHECK WILL BE MADE BEFORE DETERMINING A MATCHED STRING. THIS CHECK INSURES THAT THE MATCHED STRING WILL BE IMMEDIATELY PRECEEDED AND FOLLOWED BY A NON-ALPHANUMERIC CHARACTER. HENCE THIS FORM SHOULD BE USED FOR IDENTIFIERS.

IF THE <COLUMN RANGE> IS SPECIFIED, A MATCH WILL BE FOUND ONLY IF THE FIRST CHARACTER APPEARS WITHIN THE RANGE.

EXAMPLES:

```
"FD"@8
.LIST.
"M := IMAGE [5] ;"@15-25
( WORD [I] := "SCAN" ;)
["]
""
```

CHANGING THE OCCURRENCE OF A STRING (CHANGE)

ENTIRE STRINGS MAY BE CHANGED IN A FILE BY USE OF THE "*CHANGE" VERB.

* CHANGE <STRING> TO <STRING>

SCANS THE CURRENT RECORD REPLACING EVERY OCCURRENCE OF THE FIRST STRING WITH THE SECOND. IF NO "<STRING> TO <STRING>" APPEARS THE PREVIOUSLY USED STRINGS WILL BE USED AGAIN.

* CHANGE <M> <STRING> TO <STRING>

AS ABOVE, FOR THE RECORD WITH SEQUENCE NUMBER <M>.

* CHANGE <M>, <N> <STRING> TO <STRING>

AS ABOVE, FOR ALL RECORDS FROM SEQUENCE NUMBER <M> TO <N>.

* CHANGE ECHO ON

* CHANGE ECHO OFF

* CHANGE ECHO

THE MODIFIED RECORDS WILL BE TYPED ON THE TELETYPE IF THE CHANGE ECHO IS ON. (IT IS INITIALLY OFF.) THE ABOVE COMMANDS ARE USED TO SET THE OPTION AND TO PRINT ITS CURRENT VALUE. ITS SETTING MAY BE REVERSED FOR A COMMAND BY PREFIXING THE CHANGE WITH A -. (E.G. *-CHANGE "X" TO "Z".)

EXAMPLES:

```
00004200:* CHANGE "REMOTE/CARD" TO "R/C";*CHANGE 5300←
00005300:

   8700:* -CHANGE -1 ,TWX,@25-30 TO "TELETYPWRITER"←
   8600: THE OUPUT IS TYPED ON THE TELETYPWRITER IF
00008600:

    450:* CHANGE 232, 448 "IMAGE [I]" TO (Z [J])←
WAIT...
00000448:

  47000:* CHANGE 2200+6 " "@1 "   "←
00002800:* CHANGE +1+3 "  "@1 ""←
00003200:
```

SCANNING FOR OCCURRENCE OF A STRING (SCAN)

* SCAN <STRING>

SCANS THE FILE FROM THE CURRENT SEQUENCE NUMBER TO THE END OF THE FILE OR UNTIL THE FIRST OCCURRENCE OF <STRING>. IF THE STRING IS FOUND THE RECORD CONTAINING IT IS TYPED ON THE TELETYPEWRITER. IF <STRING> IS ACTUALLY TWO <STRING>S THE SCAN STOPS AT THE FIRST OCCURRENCE OF EITHER <STRING>. IF NO <STRING> APPEARS WITH THE COMMAND, IT IS ASSUMED TO BE THE SAME <STRING> AS IN THE LAST "* SCAN" OR "* CHANGE" COMMAND. IF THE SCAN VERB IS PREFIXED BY A -, THE SCAN WILL BE FOR RECORDS NOT CONTAINING THE <STRING>.

* SCAN <M> <STRING>

AS ABOVE, EXCEPT THAT THE SCAN BEGINS AT SEQUENCE NUMBER <M>.

* SCAN <M>, <N> <STRING>

AS ABOVE, EXCEPT THAT THE SCANNING STOPS AFTER SEQUENCE NUMBER <N>.

* SCAN <FILE-NAME> <STRING>

THE FILE <FILE-NAME> IS SCANNED FOR <STRING>. EVERY RECORD IN WHICH <STRING> OCCURS IS TYPED ON THE TELETYPEWRITER.

* SCAN <FILE-NAME> <M> <STRING>

AS ABOVE, EXCEPT THE SCAN BEGINS AT RECORD NUMBER <M>.

```
* SCAN <FILE-NAME>  <M>, <N> <STRING>
```

AS ABOVE, EXCEPT THAT THE SCAN STOPS AFTER THE <N>-TH RECORD.

EXAMPLES:

```
:* SCAN LIBRARY/FILE "TABLE OF CONTENTS"@3←
WAIT...
     1:% TABLE OF CONTENTS ... READDATA
    98:% TABLE OF CONTENTS ... READDATA
    99:% TABLE OF CONTENTS ... WRITEDATA
   273:% TABLE OF CONTENTS ... WRITEDATA
   274:% TABLE OF CONTNETS ... DATE
   285:% TABLE OF CONTENTS ... DATE
:

  8700:* SCAN 1 ,PRINTER, ,LINE,←
WAIT...
   100: FILE PRINTER(1 (2, 15) ;
   101:* SCAN←
WAIT...
  2400:          WRITE (LINE, F, LISTA) ;
  2401:* SCAN←
WAIT...
EOF NO STRING
  8700:
```

RECORD FORMATING (EDIT)

RECORDS MAY BE FORMATED BY THE FOLLOWING CONSTRUCTS:

* EDIT <M>, <N>: <F>

EDITS EACH RECORD FROM SEQUENCE NUMBER <M> TO <N> ACCORDING TO THE FORMAT RECORD WITH SEQUENCE NUMBER <F>.

THE RESULT OF THE EDIT IS THAT EACH RECORD BECOMES IDENTICAL WITH THE FORMAT RECORD EXCEPT WHERE THERE IS A "#" OR "@" CHARACTER IN FORMAT RECORD. EACH "@" IS REPLACED BY THE NEXT CHARACTER OF THE INPUT RECORD. EACH "#" SKIPS THE NEXT CHARACTER OF THE INPUT RECORD.

* EDIT ECHO ON

* EDIT ECHO OFF

* EDIT ECHO

THE MODIFIED RECORDS WILL BE TYPED ON THE TELETYPE IF THE EDIT ECHO IS ON. (IT IS INITIALLY OFF.) THE ABOVE COMMANDS ARE USED TO SET THE ECHO AND TO PRINT ITS CURRENT SETTING. IT SETTING MAY BE REVERSED FOR A COMMAND BY PREFIXING THE EDIT WITH A -. (E.G. *- EDIT 200,800:2.)

EXAMPLE:

```
100:1234567890←
200:ABCDEFGHIJK←
300:  @@@...@@. THIS IS THE WAY THE LINE ENDS...@##@@@←
400:* -EDIT 100, 200:300←
100:  123...45. THIS IS THE WAY THE LINE ENDS...690
200:  ABC...DE. THIS IS THE WAY THE LINE ENDS...FIJK
```

RESEQUENCING RECORD NUMBERS (RESEQ)

FILE SEQUENCE NUMBERS MAY BE MODIFIED BY THE FOLLOWING CONSTRUCTS:

* RESEQ

RESEQUENCES THE FILE BY THE CURRENT INCREMENT COUNTER.

* RESEQ <I>

SETS THE INCREMENT COUNTER TO <I> AND RESEQUENCES.

* RESEQ <M>, <N>

RESEQUENCES THE SEQUENCE NUMBERS <M> THRU <N> BY THE CURRENT INCREMENT COUNTER. THE FIRST SEQUENCE NUMBER IS <M>.

* RESEQ <M>, <N>, <K>

AS ABOVE, EXCEPT THAT THE FIRST SEQUENCE NUMBER IS <K>.

* RESEQ <M>, <N>, <K>, <I>

AS ABOVE, EXCEPT THE INCREMENT IS SET TO <I> FIRST.

EXAMPLES:

```
   895:* RESEQ←
   323:* RESEQ 100←
 32300:
```

DELETION OF RECORDS (DELETE)

RECORDS MAY BE DELETED FROM A FILE BY THE FOLLOWING CONSTRUCTS:

* DELETE <M>

DELETES CARD IMAGE <M>.

* DELETE <M>, <N>

DELETES CARD IMAGES <M> THROUGH <N>.

* DELETE

DELETES THE CURRENT CARD.

EXAMPLES:

```
    3400:* DELETE 3300←
    3300:

    2100:* DELETE 1500, * 3;* LIST 1400,2000←
    1400:      BEGIN
    1900:        GO TO ERROR ;
    2000:      END ;
    2100:

    2100:* 2000←
    2000:      END ;
00002000:*DELETE←
    2000:
```

OPERATIONAL COMMANDS

SET THE INCREMENT COUNTER (INC)

THE INCREMENT COUNTER MAY BE SET BY THE CONSTRUCT "* INC".

```
* INC <N>
```

THIS CONSTRUCT SETS THE INCREMENT COUNTER TO <N>.

```
* INC
```

THIS CONSTRUCT PRINTS THE CURRENT VALUE.

EXAMPLES:

```
100:BEGIN←
200: INTEGER I, J, K ;←
300:* INC 3←
300: BOOLEAN B ;←
303: REAL X, Y, Z ;←
306:
```

NON-TELETYPEWRITER CHARACTERS (PERCENT)

SPECIAL CHARACTERS ON THE B5500 BUT NOT AVAILABLE ON THE TELETYPEWRITER KEYBOARD (INCLUDING THE QUESTION MARK "?") MAY BE INSERTED INTO R/C MAINTAINED FILES BY THE "* PERCENT" CONSTRUCT.

```
* PERCENT ON
* PERCENT OFF
* PERCENT
```

THE "* PERCENT" CONSTRUCTS ALLOW A USER TO INPUT THE SPECIAL RESERVED CHARACTERS (I.E. LEFT-ARROW, NEQ, LSS, GEQ, GTR, LEQ, AND ?). THE ABOVE COMMANDS ARE USED TO SET THE OPTION AND TO PRINT ITS CURRENT SETTING. WHEN THE PERCENT OPTION IS ON, INPUT FROM THE TELETYPEWRITER IS SCANNED FOR SPECIAL CHARACTER PAIRS. IF A PAIR IS FOUND IT IS REPLACED BY THE SINGLE CHARACTER AS FOLLOWS:

CHARACTER PAIR	INTERNAL CHARACTER
%-	← (LEFT ARROW)
%=	≠ (NEQ)
%(	< (LSS)
%)	> (GTR)
%[	≤ (LEQ)
%]	≥ (GEQ)
%%	? (QUESTION MARK)
%*	* (STAR) (1ST CHR ONLY)

A "%" FOLLOWED BY ANY OTHER CHARACTER REMAINS AS IS. WHEN THE PERCENT OPTION IS ON, THE TRANSLATION TAKES PLACE IMMEDIATELY. HENCE ONLY ONE BACKSPACE (SHIFT 7) IS NEEDED.

EXAMPLES:

```
100:* PERCENT OFF←
```

```
100:%% COMPILE ... %=  %(   %A←
200:* PERCENT ON←
200:%% COMPILE ... %=  %(   %A←
300:* LIST←
100:%% COMPILE ... %=  %(   %A
200:? COMPILE ... ≠  <   %A
300:
```

NOTE THAT ON THE TELETYPWRITER THE CHARACTERS "≠" AND "<" WOULD HAVE BEEN TYPED AS A "$".

R/C VERB RENAMING (REPLACE)

R/C VERBS MAY BE REPLACED BY OTHER WORDS THROUGH THE CONSTRUCT "* REPLACE".

* REPLACE <A>:<B>

THE REPLACE VERB RENAMES THE VERB <A> WITH THE WORD <B>.

EXAMPLES:

```
:* REPLACE LIST:TYPE←
:* REPLACE INLINE:IN←
:* REPLACE REPLACE:R←
:* R CHANGE:C←
:
```

AUTOMATIC FIRST CHARACTER PLACEMENT (TAB)

R/C USERS MAY INITIALLY (AUTOMATICALLY) PRE-POSITION THE PRINT BALL AT ANY COLUMN BY USE OF THE "* TAB" CONSTRUCT.

```
* TAB <N>
```

SETS UP AUTOMATIC INDENTING TO POSITION <N>, WHERE 1 LEQ <N> LEQ 54.

```
* TAB +<N>
```

```
* TAB -<N>
```

ADJUSTS THE TAB AMOUNT BY THE VALUE OF <N>.

```
* TAB ON
```

```
* TAB OFF
```

```
* TAB
```

THE TAB OPTION PERMITS THE USER (BY SETTING THE OPTION OFF) TO SURPRESS THE TAB SPACING ON THE TELETYPEWRITER. THE AUTOMATIC INDENTING STILL APPLIES TO THE RECORDS CREATED. THE LAST CONSTRUCT PRINTS BOTH THE SETTING OF THE OPTION AND THE TAB VALUE.

EXAMPLE:

```
4500:* TAB 5←
4500:   BEGIN←
```

```
4600:     * TAB+2←
4600:       K := K + DELTAK ;←
4700:       * TAB - 2;END ;←
4800:     * TAB 1←
4800:
```

SETTING THE SAVE FACTOR (SAVE)

THE SAVE FACTOR OF A DISK FILE IS THE NUMBER OF DAYS AFTER ITS LAST ACCESS BEFORE IT WILL BE CONSIDERED EXPIRED. THE SAVE FACTOR FOR A NEW USER IS SET AT 7 DAYS. IT MAY BE CHANGED BY:

* SAVE <N>

SETS THE SAVE FACTOR AT <N> DAYS.

* SAVE

PRINTS THE CURRENT VALUE OF THE SAVE FACTOR.

EXAMPLE:

```
:* SAVE 10←
:
```

TABULATING (COLUMN)

```
* COLUMN <OPTIONAL STRING>

* COLUMN ON <OPTIONAL STRING>

* COLUMN OFF <OPTIONAL STRING>

* COLUMN <STP1> <STP2> <STP3> <STP4> <OPTL STRING>
```

THE COLUMN VERB IS USED TO SET UP COLUMN POSITIONS AND INDICATE WHAT CHARACTER IS THE COLUMN CHARACTER. THE FIRST FORM PRINTS THE CURRENT SETTINGS. THE SECOND AND THIRD FORMS ARE USED TO TURN THE OPTION ON OR OFF. THE FOURTH FORM SETS UP FROM 1 TO 4 COLUMN POSITIONS. ANY FORM MAY SET THE COLUMN CHARACTER.

WHEN THE OPTION IS ON, EACH OCCURENCE OF THE COLUMN CHARACTER IN AN INPUT RECORD CAUSES BLANKS TO BE INSERTED TO THE NEXT HIGHER COLUMN POSITION.

EXAMPLE:

```
:COLUMN ON "#";* COLUMN 10 20 25;COLUMN←
COLUMN  ON  # 10 20 25
:OPEN COLUMN/SHOW ALGOL NEW←
   100:ABC#123#XYZ#IJK#789←
   200:* LIST←
   100:ABC       123         XYZ  IJK#789
   200:1234567890123456789012345678 90←
   300:
```

AUXILIARY COMMANDS

MESSAGES TO OTHERS (MAIL)

MESSAGES MAY BE RECEIVED FROM, AND SENT TO, OTHER USERS OF R/C BY THE "* MAIL" VERB DESCRIBED BELOW.

* MAIL

THIS LISTS THE MAIL SENT TO YOU, AND REMOVES IT FROM YOUR "MAIL-BOX".

* MAIL TO <USERCODE>: MESSAGE

THIS ADDS YOUR MESSAGE TO THE <USERCODE>-S MAIL.

EXAMPLES:

```
:* MAIL←
PLEASE SEE ME WHEN YOU GET A CHANCE-BALDWIN
THERE WILL NOT BE ANY REMOTE TIME 10/24/70-SYSTEM
:* MAIL TO BALDWIN:MY OFFICE- 10AM- FRIDAY←
```

HOW TO DETACH YOURSELF FROM R/C (END)

THREE METHODS ARE AVAILABLE TO THE USER WHEN HE WISHES TO MAKE AN EXIT FROM R/C:

```
* END
- ---

* END <X>
- --- ---

* END  DS
- ---  --
```

THE FIRST CONSTRUCT DETACHES THE USER FROM R/C.

THE SECOND CONSTRUCT CAUSES R/C TO REMEMBER, FROM ONE RUN TO THE NEXT, THE VALUES OF THE OPTIONS AND THE EFFECTS OF THE "* REPLACE" VERB. THE "* END" CONSTRUCT DOES NOT GENERATE THIS LASTING EFFECT. <X> DENOTES ANY ALPHANUMERIC CHARACTER STRING.

THE THIRD CONSTRUCT ALLOWS A USER TO END WITHOUT CLOSING THE OPEN FILE. R/C REMEMBERS THE FILE STRUCTURE FOR THE NEXT TIME. HOWEVER THE FILE SHOULD NOT BE ACCESSED BY ANY OTHER PROGRAM UNTIL IT IS CLOSED USING R/C. THIS FORM CAN BE USED WHEN THE FILE EDITING IS SPLIT OVER MORE THAN ONE RUNNING OF R/C.

EXAMPLES:

```
:* END←
 GOOD BYE

     634:* END X←
 WAIT...
 GOOD BYE

     892:* END  DS←
 GOOD BYE
```

LEARNING ABOUT R/C ON THE TELETYPWRITER (TEACH)

THE TEACH VERB ALLOWS A USER AT THE TELETYPWRITER TO LEARN ABOUT THE R/C VERBS AND SPECIFICALLY WHAT VERBS HE MAY USE.

* TEACH

THIS LISTS THE VERBS THAT YOU ARE PERMITTED TO USE. THE LIST REFLECTS THE USE OF REPLACE ON ANY OF THE VERBS.

* TEACH <VERB>

THIS LISTS THE SYNTAX AND SEMANTICS OF THE RESERVED WORD <VERB>.

EXAMPLES:

```
:* TEACH SAVE←
 * SAVE $N$
SETS THE SAVE FACTOR AT $N$ DAYS.
 * SAVE
PRINTS THE CURRENT VALUE OF THE SAVE FACTOR.
:

 121970:* TEACH←
THE VALID REQUESTS ARE:
EXECUTE  DITTO   COPY     IN       ZIP      C        EDIT
SAVE     RESEQ   PUNCH    PRINT    DELETE   CLOSE    COMPILE
COLUMN   SCAN    LISTING  INC      TAB      PERCENT  QUICK
LIST     OPEN    MAIL     TEACH    REMOVE   REPLACE  END
FOR SYNTAX OF A VERB (E.G. TAB), INPUT:
 * TEACH VERB. (E.G. * TEACH TAB)
 121970:
```

NOTE THAT THE USE OF THE REPLACE VERB (FOR INLINE AND CHANGE) IS REFLECTED BY THE TEACH VERB.

MACRO VERB
----- ----

MACRO (EXECUTE)

THE EXECUTE VERB GIVES R/C USERS A SIMPLE MACRO CAPABILITY.

* EXECUTE <SOURCE> <OPTIONAL PARAMETER LIST>

RECORDS ARE READ FROM THE <SOURCE> AND TREATED AS IF THEY CAME FROM THE TELETYPE. HENCE, THEY MAY CONTAIN COMMANDS. THE <OPTIONAL PARAMETER LIST>, IF PRESENT, CONTAINS FROM 0 TO 5 PARAMETERS ENCLOSED WITHIN PARENTHESIS AND MAY BE FOLLOWED BY A REPEAT FACTOR. EACH PARAMETER MAY BE A SEQUENCE NUMBER, A RELATIVE SEQUENCE NUMBER, A RECORD NUMBER, A FILE PREFIX, A FILE SUFFIX, OR A VERB. WITHIN THE <SOURCE> THE PARAMETERS ARE REFERRED TO BY #1, #2, ..., #5. THE REPEAT FACTOR IS AN INTEGER FROM 0 TO 999 AND INDICATES THE NUMBER OF TIMES TO PERFORM THE EXECUTE. IF IT IS NOT SPECIFIED, IT IS ASSUMED TO BE 1.

* EXECUTE ECHO ON

* EXECUTE ECHO OFF

* EXECUTE ECHO

IF THE EXECUTE ECHO IS ON, EACH RECORD WILL BE TYPED ON THE TELETYPE BEFORE IT IS EXECUTED. IT IS INITIALLY OFF. IT MAY BE REVERSED FOR ONE COMMAND BY PREFIXING THE COMMAND BY A -.

* EXECUTE LIBRARY = <NAME>

* EXECUTE LIBRARY

THE EXECUTE LIBRARY FACILITY ALLOWS THE USER TO

ABBREVIATE:

* EXECUTE <FILE PREFIX>/<MACRO LIBRARY> (P1,P2,P3,P4)

WITH:

* <FILE PREFIX> P1,P2,P3,P4

THE <MACRO LIBRARY> IS "MACRO" AND ANY OTHER NAME DESIGNATED THRU THE * EXECUTE LIBRARY = <NAME> COMMAND.

EXAMPLES:

```
:* OPEN CREATE/MACRO DATA NEW←
    100:%* OPEN #1/#2 #3 NEW←
    200:* CLOSE←
:* CREATE MOVE/MACRO DATA←
    100:%* DITTO #1,#2 MOVE←
    200:*CLOSE←
:

5700:* TEACH CC←
    1:* SCAN + 1 "$$"@1
    2:* DELETE -1
    3:? COMPILE SOME/JOB FORTRAN.
    4:? DATA
    5:FILE  6=FILE6, UNIT = PRINTER
5700:*1;* INC 10;*EXECUTE CC/MACRO ()@2←
WAIT...
    2200:$$ JOB
    2900:$$ IBJOB
    2840:
```

APPENDIX A
R/C SOURCE TAPE

R/C RESIDES ON A SYSTEM DUMP TAPE CONTAINING THE FILES

1. RCSY94/RON
2. TEACHER/0000094
3. XREF/JONES

THE FILES SHOULD BE LOADED USING THE CONTROL DECK:

```
? LOAD FROM RC RCSY94/RON,TEACHER/0000094,XREF/JONES.
? END.
```

RCSY94/RON IS THE SOURCE FILE FOR THE PROGRAM R/C. IT SHOULD BE COMPILED WITH:

```
? COMPILE R/C WITH ALGOL FOR LIBRARY.
? ALGOL FILE TAPE = RCSY94/RON SERIAL.
? STACK = 256.
? CORE = 2800. = 2500 + 100 × MAXUSERS.
? PRIORITY = 1.
? DATA
$ TAPE
                                                    99999999
? END.
```

WHEN PATCHES ARE ISSUED THEY SHOULD BE PLACED AFTER THE "$ TAPE" CARD AND R/C SHOULD BE RECOMPILED.

THE MAXIMUM NUMBER OF USERS HANDLED BY R/C IS DEFINED TO BE 3 IN RCSY94/RON. TO INCREASE THE NUMBER, A PATCH CARD MUST BE ADDED. AS AN EXAMPLE, THE FOLLOWING CARD SHOULD BE ADDED TO INCREASE THE MAXIMUM NUMBER OF USERS TO 5.

```
   DEFINE MAXUSERS = 5#, MAXUSER =4# ;                    00002500
```

THE OUTPUT BUFFERS ARE DEFINED TO BE 56 CHARACTERS IN RCSY94/RON. THEY MAY BE SET AT 28 CHARACTERS USING THE FOLLOWING PATCH CARDS:

```
   DEFINE CHRSPERBUFFER = 28#,     % OR 56                00002600
   WORDSPERBUFFER = 5#,          % OR 8                    00002700
   WRDSPERBUFFER = 4# ;          % OR 7                    00002800
```

TEACHER/0000094 IS AN AUXILIARY FILE USED BY R/C. IT IS ALSO THE USERS REFERENCE MANUAL. THE PROGRAM XREF/JONES IS PROVIDED FOR GENERATING THE REFERENCE MANUAL AND FOR UPDATING THE TEACHER FILE. (NOTE, ONLY THE OBJECT CODE FOR XREF/JONES IS INCLUDED. THE SOURCE CAN BE OBTAINED THROUGH THE CUBE LIBRARY ON REQUEST.)

THE FOLLOWING DECK WILL PRODUCE A USERS REFERENCE MANUAL.

```
? EXECUTE XREF/JONES.
? FILE DISK = TEACHER/0000094.
? DATA CARD.
$ DISK SIX DOCONLY DOCUMENT FINAL
                                                        99999999
? END.
```

PATCHES TO R/C AND THE TEACHER FILE WILL BE ISSUED THRU CUBE. RATHER THAN PATCH DECKS, NEW SYMBOLIC FILES WILL BE INCLUDED ON THE CUBE TAPE. THE SEQUENCE NUMBERS WILL REMAIN THE SAME EXCEPT IN THE PATCHED AREA.

APPENDIX B

R / C USER-S GUIDE

```
* CHANGE <STRING-1> TO <STRING-2>
* CHANGE <SEQ NUMBER> <STRING-1> TO <STRING-2>
* CHANGE <START SEQ NUMBER> <END SEQ NUMBER> <STRING-1> TO <STRING-2>
* CHANGE ECHO ON
* CHANGE ECHO OFF
* CHANGE ECHO
* -CHANGE <STRING-1> TO <STRING-2>
* -CHANGE <SEQ NUMBER> <STRING-1> TO <STRING-2>
* -CHANGE <START SEQ NUMBER> <END SEQ NUMBER> <STRING-1> TO <STRING-2>
* CLOSE
* COLUMN <OPTIONAL STRING>
* COLUMN ON <OPTIONAL STRING>
* COLUMN OFF <OPTIONAL STRING>
* COLUMN <STOP-1> <STOP-2> <STOP-3> <STOP-4> <OPTIOANL STRING>
* COMPILE <OBJECT FILE-NAME>
* COMPILE <OBJECT FILE-NAME> <COMPILER>
* COPY <FILE-NAME>
* COPY <FILE-NAME> <START RECORD NUMBER>
* COPY <FILE-NAME> <START RECORD NUMBER> <END RECORD NUMBER>
* COPY $(FILE-NAME> MERGE
* COPY OVERITE ON
* COPY OVERITE OFF
* COPY OVERITE
* -COPY <FILE-NAME>
* -COPY <FILE-NAME> <START RECORD NUMBER>
* -COPY <FILE-NAME> <START RECORD NUMBER> <END RECORD NUMBER>
* -COPY <FILE-NAME> MERGE
* DELETE <SEQ NUMBER>
* DELETE <START SEQ NUMBER> <END SEQ NUMBER>
* DELETE
* DITTO <SEQ NUMBER>
* DITTO <START SEQ NUMBER> <END SEQ NUMBER>
* DITTO OVERITE ON
* DITTO OVERITE OFF
* DITTO OVERITE
* -DITTO <SEQ NUMBER>
* -DITTO <START SEQ NUMBER> <END SEQ NUMBER>
* DITTO <SEQ NUMBER> MOVE
* DITTO <START SEQ NUMBER> <END SEQ NUMBER> MOVE
* EDIT <START SEQ NUMBER> <END SEQ NUMBER>:<FORMAT SEQ NUMBER>
* EDIT ECHO ON
* EDIT ECHO OFF
* EDIT ECHO
* -EDIT <START SEQ NUMBER> <END SEQ NUMBER>:<FORMAT SEQ NUMBER>
* END
* END <X>
* END DS
* EXECUTE <SOURCE> <OPTIONAL PARAMETER LIST>
*-EXECUTE <SOURCE> <OPTIONAL PARAMETER LIST>
* EXECUTE ECHO ON
* EXECUTE ECHO OFF
* EXECUTE ECHO
* EXECUTE LIBRARY = <NAME>
```

```
* EXECUTE LIBRARY
* INC <INCREMENT>
* INC
* INLINE <SEQ NUMBER>
* INLINE <SEQ NUMBER> I
* INLINE <SEQ NUMBER> R
* INLINE <SEQ NUMBER> D
* INLINE
* INLINE I
* INLINE R
* INLINE D
* INLINE ECHO ON
* INLINE ECHO OFF
* INLINE ECHO
* -INLINE <SEQ NUMBER>
* -INLINE <SEQ NUMBER> I
* -INLINE <SEQ NUMBER> R
* -INLINE <SEQ NUMBER> D
* -INLINE
* -INLINE I
* -INLINE R
* -INLINE D
* LIST
* LIST <FILE-NAME>
* LIST <FILE-NAME> NO
* LIST <FILE-NAME> <START RECORD NUMBER>
* LIST <FILE-NAME> <START RECORD NUMBER> <END RECORD NUMBER>
* LIST <SEQ NUMBER>
* LIST <START SEQ NUMBER> <END SEQ NUMBER>
* LISTING <FILE-TYPE> <SEGMENT> <LOW REL ADDR> <HIGH REL ADDR>
* LISTING <FILE-TYPE> ERRORS
* LISTING
* MAIL
* MAIL TO <USER-CODE>: <MESSAGE>
* OPEN <FILE-NAME> <FILE-TYPE> NEW
* OPEN <FILE-NAME> <FILE-TYPE> OLD
* OPEN <FILE-NAME> <FILE-TYPE>
* PERCENT ON
* PERCENT OFF
* PERCENT
* PRINT <IDENT-1> <IDENT-2>
* PRINT <IDENT-1> DOUBLE
* PRINT <IDENT-1> <IDENT-2> <START SEQ NUMBER>
* PRINT <IDENT-1> DOUBLE <START SEQ NUMBER>
* PRINT <IDENT-1> <IDENT-2> <START SEQ NUMBER> <END SEQ NUMBER>
* PRINT <IDENT-1> DOUBLE <START SEQ NUMBER> <END SEQ NUMBER>
* PUNCH <IDENT-1> <IDENT-2>
* PUNCH <IDENT-1> <IDENT-2> <START SEQ NUMBER>
* PUNCH <IDENT-1> <IDENT-2> <START SEQ NUMBER> <END SEQ NUMBER>
* QUICK
* QUICK <FILE-NAME>
* QUICK <FILE-NAME> NO
* QUICK <FILE-NAME> <START RECORD NUMBER>
```

```
* QUICK <FILE-NAME> <START RECORD NUMBER> <END RECORD NUMBER>
* QUICK <SEQ NUMBER>
* QUICK <START SEQ NUMBER> <END SEQ NUMBER>
* REMOVE <FILE-NAME>
* REPLACE <OLD VERB> : <NEW VERB>
* RESEQ
* RESEQ <INCREMENT>
* RESEQ <START SEQ NUMBER> <END SEQ NUMBER>
* RESEQ <START SEQ NUMBER> <END SEQ NUMBER> <NEW START SEQ NUMBER>
* RESEQ <START SEQ NUMBER> <END SEQ NUMBER> <NEW .. NUMBER> <INCREMENT>
* SAVE <SAVE-FACTOR IN DAYS>
* SAVE
* SCAN <STRING>
* SCAN <START SEQ NUMBER> <STRING>
* SCAN <START SEQ NUMBER> <END SEQ NUMBER> <STRING>
* SCAN <FILE-NAME> <STRING>
* SCAN <FILE-NAME> <START RECORD NUMBER> <STRING>
* SCAN <FILE-NAME> <START RECORD NUMBER> <END RECORD NUMBER> <STRING>
* SCAN
* SCAN <START SEQ NUMBER>
* SCAN <START SEQ NUMBER> <END SEQ NUMBER>
* SCAN <FILE-NAME>
* SCAN <FILE-NAME> <START RECORD NUMBER>
* SCAN <FILE-NAME> <START RECORD NUMBER> <END RECORD NUMBER>
*-SCAN <STRING>
*-SCAN <START SEQ NUMBER> <STRING>
*-SCAN <START SEQ NUMBER> <END SEQ NUMBER> <STRING>
*-SCAN <FILE-NAME> <STRING>
*-SCAN <FILE-NAME> <START RECORD NUMBER> <STRING>
*-SCAN <FILE-NAME> <START RECORD NUMBER> <END RECORD NUMBER> <STRING>
*-SCAN
*-SCAN <START SEQ NUMBER>
*-SCAN <START SEQ NUMBER> <END SEQ NUMBER>
*-SCAN <FILE-NAME>
*-SCAN <FILE-NAME> <START RECORD NUMBER>
*-SCAN <FILE-NAME> <START RECORD NUMBER> <END RECORD NUMBER>
* TAB <COLUMN NUMBER>
* TAB + <OFF-SET NUMBER>
* TAB - <OFF-SET NUMBER>
* TAB ON
* TAB OFF
* TAB
* TEACH
* TEACH <VERB>
* ZIP
* ZIP <SCRATCH FILE-NAME>
```

DAN ROSS

B5700 TEXT EDITOR R/C REFERENCE MANUAL

TABLE OF CONTENTS

INTRODUCTION . . . . . PAGE 1
PROGRAM OPERATION AND RECORD SEQUENCING . . . . . PAGE 2
PROGRAM EXECUTION . . . . . PAGE 3
INITIAL REMOTE TERMINAL OPERATIONS . . . . . PAGE 3
FINAL REMOTE TERMINAL OPERATIONS . . . . . PAGE 5
R/C INPUT . . . . . PAGE 5
R/C OUTPUT . . . . . PAGE 7
R/C FILES . . . . . PAGE 8
FILE TYPES . . . . . PAGE 8
FILE NAMES . . . . . PAGE 8
RECORD REFERENCING . . . . . PAGE 9
FILE-HANDLING VERBS . . . . . PAGE 10
FILE OPENING AND CREATION (OPEN) . . . . . PAGE 11
FILE CLOSING (CLOSE) . . . . . PAGE 14
LISTINGS ON THE TELETYPEWRITER (LIST) . . . . . PAGE 15
COMPRESSED FILE LISTINGS (QUICK) . . . . . PAGE 18
FILE REMOVAL (REMOVE) . . . . . PAGE 19
LINE-PRINTER FILE REPRODUCTION (PRINT) . . . . . PAGE 20
PUNCHED-CARD FILE REPRODUCTION (PUNCH) . . . . . PAGE 21
FILE COMPILATION (COMPILE) . . . . . PAGE 22
OUTPUT OF THE COMPILATION (LISTING) . . . . . PAGE 23
FILE ZIPPED AS AN "EXECUTE" DECK (ZIP) . . . . . PAGE 24
RECORD HANDLING VERBS . . . . . PAGE 25
RECORD COPYING (DITTO) . . . . . PAGE 26
COPYING EXTERNAL FILES (COPY) . . . . . PAGE 28
INTRA-RECORD EDITING (INLINE) . . . . . PAGE 30
STRINGS . . . . . PAGE 33
CHANGING THE OCCURRENCE OF A STRING (CHANGE) . . . . . PAGE 34
SCANNING FOR OCCURRENCE OF A STRING (SCAN) . . . . . PAGE 36
RECORD FORMATING (EDIT) . . . . . PAGE 38
RESEQUENCING RECORD NUMBERS (RESEQ) . . . . . PAGE 39
DELETION OF RECORDS (DELETE) . . . . . PAGE 40
OPERATIONAL COMMANDS . . . . . PAGE 41
SET THE INCREMENT COUNTER (INC) . . . . . PAGE 42
NON-TELETYPEWRITER CHARACTERS (PERCENT) . . . . . PAGE 43
R/C VERB RENAMING (REPLACE) . . . . . PAGE 45
AUTOMATIC FIRST CHARACTER PLACEMENT (TAB) . . . . . PAGE 46
SETTING THE SAVE FACTOR (SAVE) . . . . . PAGE 48
TABULATING (COLUMN) . . . . . PAGE 49
AUXILIARY COMMANDS . . . . . PAGE 50
MESSAGES TO OTHERS (MAIL) . . . . . PAGE 51
HOW TO DETACH YOURSELF FROM R/C (END) . . . . . PAGE 52
LEARNING ABOUT R/C ON THE TELETYPWRITER (TEACH) . . . . . PAGE 53
MACRO VERB . . . . . PAGE 54
MACRO (EXECUTE) . . . . . PAGE 55
APPENDIX A . . . . . A- 1
R/C SOURCE TAPE . . . . . A- 1
APPENDIX B . . . . . B- 1
R / C USER-S GUIDE . . . . . B- 1

DAN ROSS

B5700
TEXT EDITOR

R/C

(Compiler Listing)

```
?COMPILE R/C ALGOL LIBRARY

            PACKET     41
            INPUT       9 CARDS FROM CRA
            TIME     1548
            DATE    76194 MONDAY, 07/12/76


          *** MESSAGE OF THE DAY ***

              WELCOME TO B5700 TIME SHARING,
              MAY YOUR DATA BE DEPENDABLE AND YOUR PROGRAMS PROVEN.
                 FROM THE STAFF OF THE BIG BAD BURROUGHS.

          *** BURROUGHS B5700 TSMCP MARK XVI.0.00  AND INTRINSICS MARK XVI.0.00 ***

           ?COMPILE R/C ALGOL LIBRARY
               ?ALGOL STACK= 1024
               ?ALGOL FILE TAPE= RCSY94/Z100006 DISK SERIAL
               ? STACK= 256
               ? CORE= 3300
           ? DATA CARD
                    4:ALGOL/R= 2 BOJ 1548 06/23/76
                    CDA IN CARD DA:ALGOL/R= 2
                    DKA IN SER RCSY94 Z100006:ALGOL/R= 2
                    PBD0042 OUT 011 LINE:ALGOL/R= 2
                    DKA OUT RDM R C:ALGOL/R= 2
                    R/C REMOVED
                    R/C/0000000= 360 SEGS--CREATED 07/02/76 AT 00:11:41:57
                    DKA LOK R C:ALGOL/R= 2
                    DKA REL RCSY94 Z100006:ALGOL/R= 2
                    PBD0042 REL 011 LINE 4214:ALGOL/R= 2
                    CDA REL CARD DA:ALGOL/R= 2
?END.
                    ALGOL/R= 2 PS= 2240 ECL
                    PKT#0041 REMOVED
```

LEXL EDIT06

B2500

DAN BOSS

R /C
==========

```
                R / C  -- A MULTI USER REMOTE/CARD,                                  00000500 T  0000
  WRITTEN BY RON BRODY;  BURROUGHS CORP.; PAOLI, PA. 215-NI4-4700 X219               00001000 T  0000
BEGIN                                                                               00001500 T  0000
                                                                        START OF SEGMENT ********** 2
 DEFINE VERSION = 94# ;  % NOVEMBER 18, 1971,                                       00002000 T  0000
   DEFINE MAXUSERS = 8#, MAXUSER = 7#;                                              00002500 P  0000
 DEFINE CHRSPERBUFFER = 56#,        % OR 28                                         00002600 T  0000
   WORDSPERBUFFER = 8#,             % OR 5                                          00002700 T  0000
   WDSPERBUFFER = 7# ;              % OR 4                                          00002800 T  0000
 % * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00003000 T  0000
 ALPHA FILE IN TWXINPUT 14 (MAXUSER + MAXUSER, 8) ;                                 00003500 T  0000
 ALPHA FILE OUT TWXOUTPUT 14 (MAXUSERS, WORDSPERBUFFER) ;                           00004000 T  0004
 DEFINE TWXOUT = TWXOUTPUT (STATIONI, 0)# ;                                         00008500 T  0007
 ARRAY PRETANK [0 : 3],                                                             00009500 T  0007
   BUFFERS [0 : MAXUSERS, 0 : 44] ;                                                 00010000 T  0009
 DEFINE BUFFER [BUFFER1] = BUFFERS [USER, BUFFER1]#,                                00010200 T  0011
   BLOC = BUFFER [29]#,                                                             00010300 T  0011
   BUFF [BUFF1] = BUFFERS [MAXUSERS, BUFF1]# ;                                      00010400 T  0011
 INTEGER ARRAY READYQ [0 : MAXUSERS] ;                                              00011000 T  0011
 DEFINE RATTLEINDEX = READYQ [MAXUSERS]# ;                                          00011500 T  0013
 INTEGER USER,                                                                      00012000 T  0013
   USER32,                                                                          00012200 T  0013
   CLOCK,                                                                           00013000 T  0013
   READYQTOP,                                                                       00013500 T  0013
   NEXTCLOCK,                                                                       00014500 T  0013
   TINK,                                                                            00015000 T  0013
   BIGBIRD ;                                                                        00016000 T  0013
 BOOLEAN GLOBALBOOL ;                                                               00016500 T  0013
 DEFINE                                                                             00017000 T  0013
   TANKEDOUTPUT = GLOBALBOOL . [47 : 1]#,                                           00017010 T  0013
   OUTPUTREADY = (GLOBALBOOL)#,                                                     00017020 T  0013
   Q = GLOBALBOOL . [46 : 1]#,                                                      00017100 T  0013
   LOCKED = GLOBALBOOL . [45 : 1]#,                                                 00017200 T  0013
   XLOCKED = GLOBALBOOL . [44 : 1]#,                                                00017210 T  0013
   YLOCKED = GLOBALBOOL . [43 : 1]#,                                                00017220 T  0013
   QINPUT = GLOBALBOOL . [42 : 1]#,                                                 00017300 T  0013
   ERRTOG = GLOBALBOOL . [1 : 1]# ;                                                 00017500 T  0013
 ARRAY INPUT [0 : 14] ;                                                             00018000 T  0013
 DEFINE TO = INPUT [10]#,                                                           00018100 T  0014
   T1 = INPUT [11]#,                                                                00018200 T  0014
   TN = INPUT [12]#,                                                                00018300 T  0014
   FREEHEAD = INPUT [13]#,                                                          00018400 T  0014
   MAXFREEHEAD = INPUT [14]# ;                                                      00018500 T  0014
 DEFINE CHRS = BUFFER [30]#,                                                        00019000 T  0014
   NCHRS = BUFFER [31]#,                                                            00019100 T  0014
   USERCODEI = BUFFER [32]#,                                                        00019500 T  0014
   STATIONI = BUFFER [33]#,                                                         00020000 T  0014
```

```
  BREAKI = BUFFER [34]#,                                                                00020500 T  0014
  ABNORMALEND = BUFFER [35]#,                                                           00020600 T  0014
  INREADYQ = BUFFER [36]#,                                                              00020700 T  0014
  FIRSTCHANCE = BUFFER [37]#,                                                           00020710 T  0014
  ILFORI = BUFFER [38]#,                                                                00020800 T  0014
  TRANSLATEI = BUFFER [39]#,                                                            00020900 T  0014
  HEADI = BUFFER [40]#,                                                                 00021000 T  0014
  TIMEI = BUFFER [41]#,                                                                 00021100 T  0014
  TAILI = BUFFER [42]#,                                                                 00021500 T  0014
  SLOTI = BUFFER [43]#,                                                                 00022000 T  0014
  BLOCK = BUFFER [44]#,                                                                 00022100 T  0014
  COUNTI = BUFFER [0]# ;                                                                00022500 T  0014
ALPHA ARRAY RECORD [0 : 9] ;                                                            00023000 T  0014
REAL ARRAY LINKLISTS [0 : 32 × MAXUSERS + 1, 0 : 255] ;                                 00023500 T  0016
DEFINE TIMEX = TIME (1)#,                                                               00023600 T  0020
  FIRST = LINKLISTS [USER32, 0]#,                                                       00023800 T  0020
  LAST = LINKLISTS [USER32, 1]#,                                                        00023900 T  0020
  LEFTSIDE = [35 : 5]#,                                                                 00024000 T  0020
  RIGHTSIDE = [40 : 8]#,                                                                00024500 T  0020
  LL [LL1] =                                                                            00025000 T  0020
    LINKLISTS [(TINK := LL1),LEFTSIDE + USER32, TINK,RIGHTSIDE]#,                       00025500 T  0020
  S = [1 : 21]#,                                                                        00026000 T  0020
  SF = 1 : 27 : 21#,                                                                    00026500 T  0020
  F = [22 : 13]#,                                                                       00027000 T  0020
  FF = 22 : 35 : 13#,                                                                   00027500 T  0020
  T = [35 : 13]#,                                                                       00028000 T  0020
  TF = 35 : 35 : 13#,                                                                   00028500 T  0020
  INFINITY = 2097151#,  %MAXIMUM SEQUENCE NUMBER = 2*21-1,                              00029000 T  0020
  FINITY = 2097150#,                                                                    00029010 T  0020
  MAXFILELENGTH = 8191# ;% = 2*13-1,                                                    00029500 T  0020
DEFINE MODIFY (MODIFY1) =                                                               00029700 T  0020
    MODIFIED := MODIFIED OR TWO ((MODIFY1),LEFTSIDE)# ;                                 00029800 T  0020
DEFINE WAITFLAG = BOOL , [47 : 1]#,  WAITING = (BOOL)#,                                 00030500 T  0020
  INLINETOG = BOOL , [46 : 1]#,                                                         00031000 T  0020
  EXTRALFCR = BOOL , [45 : 1]#,                                                         00031500 T  0020
  EXECUTEECHO = BOOL , [44 : 1]#,                                                       00032000 T  0020
  TRANSLATING = BOOL , [43 : 1]#,      % INITIALLY ON                                   00032500 T  0020
  XECHO = BOOL , [42 : 1]#,                                                             00033000 T  0020
  NUM1 = BOOL , [36 : 2]#,                                                              00035000 T  0020
  NUM2 = BOOL , [34 : 2]#,                                                              00035500 T  0020
  NUM3 = BOOL , [32 : 2]#,                                                              00036000 T  0020
  NUM4 = BOOL , [30 : 2]#,                                                              00036500 T  0020
  EMPTY1 = BOOL , [36 : 1]#,                                                            00037500 T  0020
  EMPTY2 = BOOL , [34 : 1]#,                                                            00038000 T  0020
  EMPTY3 = BOOL , [32 : 1]#,                                                            00038500 T  0020
  EMPTY4 = BOOL , [30 : 1]#,                                                            00039000 T  0020
  NOSTAR = BOOL , [29 : 1]#,                                                            00039500 T  0020
  MOREINPUT = BOOL , [23 : 1]#,                                                         00042500 T  0020
  NOTFIRSTINPUT = BOOL , [22 : 1]#,                                                     00043000 T  0020
  INLINEECHO = BOOL , [21 : 1]#,       % INITIALLY ON                                   00043010 T  0020
  CHANGEECHO = BOOL , [20 : 1]#,                                                        00043020 T  0020
  EDITECHO = BOOL , [19 : 1]#,                                                          00043030 T  0020
  COPYCLOBBER = BOOL , [18 : 1]#,                                                       00043040 T  0020
  DITTOCLOBBER = BOOL , [17 : 1]#,                                                      00043050 T  0020
  TEMPTOG = BOOL , [16 : 1]#,                                                           00043060 T  0020
  TABON = BOOL , [15 : 1]#,            % INITIALLY ON                                   00043070 T  0020
  COLUMNS = BOOL , [12 : 1]#,                                                           00043100 T  0020
```

```
  INORDER = BOOL , [1 : 1]#,                                                                    00043500 T  0020
  INITIALBOOL = BOOLEAN ("44000+")# ;                                                           00043600 T  0020
ARRAY CONTROLS [0 : 90] ;                                                                       00043700 T  0020
DEFINE VN                    =      CONTROLS [89]#,                                             00043800 T  0022
  STRINGI                    =      CONTROLS [88]#,                                             00043900 T  0022
  STRINGID                   =      CONTROLS [87]#,                                             00044000 T  0022
  STRINGILEFT                =      CONTROLS [86]#,                                             00044100 T  0022
  STRINGIREPEAT              =      CONTROLS [85]#,                                             00044200 T  0022
  STRINGJ                    =      CONTROLS [84]#,                                             00044300 T  0022
  STRINGJD                   =      CONTROLS [83]#,                                             00044400 T  0022
  STRINGJLEFT                =      CONTROLS [82]#,                                             00044500 T  0022
  STRINGJREPEAT              =      CONTROLS [81]#,                                             00044600 T  0022
  CHARACTER                  =      CONTROLS [80]#,                                             00044700 T  0022
  MAXCOLSTOP                 =      CONTROLS [79]#,                                             00044800 T  0022
  COLSTOPS                   =      CONTROLS [78]#,                                             00044900 T  0022
  COLSTOP4                   =      CONTROLS [77]#,                                             00045000 T  0022
  COLSTOP3                   =      CONTROLS [76]#,                                             00045100 T  0022
  COLSTOP2                   =      CONTROLS [75]#,                                             00045200 T  0022
  COLSTOP1                   =      CONTROLS [74]#,                                             00045300 T  0022
  COLSTOP [COLSTOP1]         =      CONTROLS [73 + COLSTOP1]#,                                  00045400 T  0022
  RELATIVENUMBER             =      CONTROLS [73]#,                                             00045500 T  0022
  STRING                     =      CONTROLS [30]# ; % = CONTROLS [37]                          00046600 T  0022
REAL PARAMETER0,             %      CONTROLS [38]                                               00046610 T  0022
  PARAMETER1,                %      CONTROLS [39]                                               00046620 T  0022
  PARAMETER2,                %      CONTROLS [40]                                               00046630 T  0022
  PARAMETER3,                %      CONTROLS [41]                                               00046640 T  0022
  PARAMETER4,                %      CONTROLS [42]                                               00046650 T  0022
  USERCODE,                  %      CONTROLS [43]                                               00046700 T  0022
  STATION,                   %      CONTROLS [44]                                               00046800 T  0022
  PREFIX,                    %      CONTROLS [45]                                               00046900 T  0022
  SUFFIX,                    %      CONTROLS [46]                                               00047000 T  0022
  MACROLIBRARY ;             %      CONTROLS [47]                                               00047100 T  0022
BOOLEAN MODIFIED ;           %      CONTROLS [48]                                               00047200 T  0022
INTEGER FILEINFO,            %      CONTROLS [49]                                               00047300 T  0022
  TABAMOUNT,                 %      CONTROLS [50]                                               00047400 T  0022
  FILEACCESS,                %      CONTROLS [51]                                               00047500 T  0022
  SAVEFACTOR,                %      CONTROLS [52]                                               00047600 T  0022
  PREWHERE,                  %      CONTROLS [53]                                               00047700 T  0022
  XDEX,                      %      CONTROLS [54]                                               00047800 T  0022
  N,                         %      CONTROLS [55]                                               00047900 T  0022
  AT,                        %      CONTROLS [56]                                               00048000 T  0022
  D,                         %      CONTROLS [57]                                               00048100 T  0022
  M,                         %      CONTROLS [58]                                               00048200 T  0022
  INC,                       %      CONTROLS [59]                                               00048300 T  0022
  I,                         %      CONTROLS [60]                                               00048400 T  0022
  RESETN ;                   %      CONTROLS [61]                                               00048500 T  0022
BOOLEAN BOOL ;               %      CONTROLS [62]                                               00048800 T  0022
DEFINE COBOLFILE = BOOLEAN (FILEINFO)#,                                                         00048820 T  0022
  DATAFILE = FILEINFO = DATA#,                                                                  00048830 T  0022
  ALGOLFILE = FILEINFO GEQ ALGOL#,                                                              00048840 T  0022
  COMPILER = FILEINFO#,                                                                         00048850 T  0022
  LENGTH = (IF ALGOLFILE THEN 72 ELSE IF COBOLFILE THEN 66 ELSE 80)#,                           00048860 T  0022
  HALFLENGTH=(IF ALGOLFILE THEN 36 ELSE IF COBOLFILE THEN 33 ELSE 40)#,                         00048870 T  0022
  FULLLENGTH = (IF DATAFILE THEN 80 ELSE 72)#,                                                  00048880 T  0022
  HALFFULLLENGTH = (IF DATAFILE THEN 40 ELSE 36)#,                                              00048890 T  0022
  COBOL = 1#,                                                                                   00049600 T  0022
  DATA = 2#,                                                                                    00049610 T  0022
```

```
  ALGOL = 4#,                                                                    00049620 T  0022
  XALGOL = 6#,                                                                   00049630 T  0022
  FORTRAN = 8#,                                                                  00049640 T  0022
  BASIC = 10#,                                                                   00049650 T  0022
  FILEOPEN = FILEACCESS GTR 0#,                                                  00050710 T  0022
  FILECLOSED = FILEACCESS LEQ 0#,                                                00050720 T  0022
  READONLYFILE = FILEACCESS = 2#,                                                00050730 T  0022
  READWRITEFILE = FILEACCESS GEQ 3# ;                                            00050740 T  0022
SAVE ARRAY IMAGE [0 : 29] ;                                                      00058000 T  0022
DEFINE RSWDM = 27#,                                                              00058500 T  0024
  RSWD [RSWD1] = CONTROLS [RSWD1]#,                                              00061500 T  0024
  RWTEACH = RSWD [24]# ;                                                         00062000 T  0024
FILE DISC DISK SERIAL (2, 10, 30) ;                                              00064000 T  0024
FILE LIBRARY DISK SERIAL (2, 10, 30) ;                                           00065500 T  0027
FILE R1 DISK SERIAL "R/C" "#1" (1, 90) ;                                         00069500 T  0031
FILE R2 DISK SERIAL "R/C" "#2" (1, 256) ;                                        00070000 T  0034
FILE IO DISK RANDOM [20:150] (1, 30) ;                                           00070500 T  0038
ARRAY ZIPPY [0 : MAX (29, MAXUSERS + MAXUSERS + 1)] ;                            00071500 T  0041
FORMAT ZIPPER ("CC COMPILE ", A1, A6, "/", A1, A6, " WITH ",                     00072000 T  0047
                                                                  START OF SEGMENT **********    3
    A1, A6, " LIBRARY; ALGOL FILE CARD=", A1, A6, "/", A1, A6,                   00072500 T  0047
    " SERIAL; ALGOL FILE LINE=LINE/", A1, A6, " SERIAL; END,"),                  00073000 T  0047
  EOJ ("≤≠GOOD BYE≤≠≠≠←"),                                                       00073600 T  0047
  NOROOM (X8, "SORRY, FULL UP.≤≠BYE≤≠←"),                                        00073700 T  0047
  USERUN (X8, "USE:≤≠?? RUN ...←"),                                              00078600 T  0047
  STAR ("*", X79),                                                               00079000 T  0047
  DATE (X6, A1, A6, "/", A1, A6, " LISTED AT", I3, ":", I2, " ON ",              00079300 T  0047
        A6, "DAY ", 0, " BY ", A1, A6, X62),                                     00079400 T  0047
  WAITF ("WAIT...←"),                                                            00085000 T  0047
  RATTLE (X8, "<<<←"),                                                           00087000 T  0047
  TEACH1 ("≤≠THE VALID VERBS ARE:←"),                                            00087500 T  0047
  TEACH2 (7 (A1, A6, X2)),                                                       00088000 T  0047
  TEACH3 ("FOR SYNTAX OF A VERB (E.G. TAB): INPUT: * TEACH",                     00088500 T  0047
          " VERB. (E.G. * TEACH TAB) "),                                         00089000 T  0047
  BROKEN (X8, "≤≠BREAK≤≠←") ;                                                    00092000 T  0047
                                                                  3 IS  128 LONG, NEXT SEG    2
DEFINE ONOFF (ONOFF1) = (IF ONOFF1 THEN " ON    " ELSE " OFF   ")# ;             00097100 T  0047
DEFINE XMAX = 5# ;                                                               00099600 T  0047
ARRAY XARRAY [0:MAXUSER, 0:XMAX × 13 - 1] ;                                      00099700 T  0047
DEFINE                                                                           00099800 T  0051
 XSUB = XDEX × 13#,                                                              00099810 T  0051
 XPARAMETERS [XPARAMETERS1] = XARRAY [USER, XSUB + XPARAMETERS1]#,                00099900 T  0051
 XSTART = XARRAY [USER, XSUB + 5]#,                                              00100000 T  0051
 XLAST = XARRAY [USER, XSUB + 6]#,                                               00100100 T  0051
 XN = XARRAY [USER, XSUB + 7]#,                                                  00100200 T  0051
 XREPEAT = XARRAY [USER, XSUB + 8]#,                                             00100300 T  0051
 XPREFIX = XARRAY [USER, XSUB + 9]#,                                             00100400 T  0051
 XSUFFIX = XARRAY [USER, XSUB + 10]#,                                            00100500 T  0051
 XFILETYPE = XARRAY [USER, XSUB + 11]#,                                          00100600 T  0051
 XNCHRS = XARRAY [USER, XSUB + 12]# ;                                            00100700 T  0051
PROCEDURE PROGRAM ; FORWARD ;                                                    00101000 T  0051
ALPHA PROCEDURE OCTDECIMAL (N, M, F) ;                                           00101100 T  0051
VALUE N, F ;                                                                     00101200 T  0051
INTEGER N, M, F ;                                                                00101300 T  0051
BEGIN                                                                            00101400 T  0051
ALPHA STREAM PROCEDURE OCTDECX (N, F, Q, T) ;                                    00101500 T  0051
                                                                  START OF SEGMENT **********    4
```

```
VALUE F, Q, T ;                                                                 00101600 T  0000
   BEGIN                                                                        00102500 T  0000
LABEL EXIT ;                                                                    00102600 T  0000
     DI := LOC OCTDECX ;                                                        00103000 T  0000
     SI := N ;                                                                  00103100 T  0000
     T (Q (DS := F OCT ; JUMP OUT 2 TO EXIT) ;                                  00103200 T  0000
        SKIP F DB ; DS := SET ; JUMP OUT TO EXIT) ;                             00103500 T  0004
     Q (F (SI := SI + 2 ; DS := 2 CHR ; DS := LIT "/" ; DS := 2 CHR ;           00103600 T  0005
          DS := LIT "/" ; DS := 2 CHR ; JUMP OUT 2 TO EXIT) ;                   00103700 T  0008
        DI := DI + 7 ; DS := CHR ; JUMP OUT TO EXIT) ;                          00103800 T  0010
     DS := 8 DEC ;                                                              00104000 T  0011
     F (DI := DI - 7 ; DS := 6 FILL) ;                                          00104100 T  0012
EXIT:                                                                           00104200 T  0013
   END OCTDECX ;                                                                00104300 T  0013
   IF F LEQ 1 THEN                                                              00104400 T  0015
   BEGIN                                                                        00104500 T  0015
     N := N ;                                                                   00104600 T  0016
     OCTDECIMAL := OCTDECX (N, F, 0, 0) ;                                       00104700 T  0017
   END ELSE IF F = 2 THEN                                                       00104800 T  0019
     OCTDECIMAL := OCTDECX (M, 0, 1, 0)                                         00104900 T  0020
   ELSE IF F = 3 THEN                                                           00104910 T  0022
     OCTDECIMAL := OCTDECX (N, 1, 1, 0)                                         00104920 T  0025
   ELSE IF F = 4 THEN                                                           00104930 T  0027
     OCTDECIMAL := OCTDECX (M, N, 1, 1)                                         00104940 T  0029
   ELSE                                                                         00105000 T  0031
     OCTDECIMAL := OCTDECX (M, N := 47 - N, 0, 1) ;                             00105100 T  0032
END OCTDECIMAL ;                                                                00105200 T  0036
                                                                        4 IS   39 LONG, NEXT SEG    2

DEFINE OCTDEC (OCTDEC1) = OCTDECIMAL (OCTDEC1, M, 0)#,                          00105300 T  0051
  OCTDEX (OCTDEX1) = OCTDECIMAL (OCTDEX1, M, 1)#,                               00105400 T  0051
  FIRSTCHAR (FIRSTCHAR1) = OCTDECIMAL (0, FIRSTCHAR1, 2)#,                      00105500 T  0051
  MMDDYY = OCTDECIMAL (TIME (5), M, 3)#,                                        00105600 T  0051
  DEC (DEC1, DEC2) = OCTDECIMAL (DEC2, DEC1, 4)#,                               00105700 T  0051
  TWO (TWO1) = BOOLEAN (OCTDECIMAL (TWO1, M, 5))# ;                             00105800 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *      00120500 T  0051
DEFINE SEQUENCE =                                                               00121000 T  0051
   IF ALGOLFILE THEN                                                            00121500 T  0051
     IMAGE [9] := OCTDEC (IF N = FINITY THEN 99999999 ELSE N)                   00122000 T  0051
   ELSE IF COBOLFILE THEN                                                       00122500 T  0051
   BEGIN                                                                        00123000 T  0051
     IMAGE [0].[1:35] := OCTDEC (N) ;                                           00124000 T  0051
     IMAGE [9] := SUFFIX & ","[1:43:5] ;                                        00124500 T  0051
   END# ;                                                                       00125500 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *      00126000 T  0051
PROCEDURE STATIONFIX (STATION, I) ;                                             00126500 T  0051
VALUE STATION, I ;                                                              00126600 T  0051
REAL STATION ;                                                                  00126700 T  0051
INTEGER I ;                                                                     00126800 T  0051
BEGIN                                                                           00126900 T  0051
REAL X ;                                                                        00127000 T  0051
                                                                  START OF SEGMENT **********    5
   IF I LEQ 4 THEN                                                              00127100 T  0000
     X := STATUS (STATION, I)                                                   00127200 T  0000
   ELSE IF I LEQ 6 THEN                                                         00127300 T  0003
     RELEASE (STATION)                                                          00127400 T  0005
   ELSE IF I LEQ 8 THEN                                                         00127500 T  0006
   BEGIN                                                                        00127600 T  0010
```

```
      SEEK (TWXINPUT (STATION)) ;                                          00127625 T  0011
      X := STATUS (STATION, 0) ;                                           00127650 T  0014
    END                                                                    00127675 T  0017
    ELSE IF I = 9 THEN                                                     00127700 T  0017
    BEGIN                                                                  00127800 T  0018
      WRITE (TWXOUTPUT (STATION), NOROOM) ;                                00127900 T  0019
      RELEASE (STATION) ;                                                  00128000 T  0023
    END                                                                    00128100 T  0026
    ELSE IF I = 10 THEN                                                    00128200 T  0026
    BEGIN                                                                  00128300 T  0027
      IF BOOLEAN (STATUS (STATION, 0)).[30:1] OR                           00128400 T  0027
         USERCODEI NEQ STATUS (STATION) THEN                               00128500 T  0031
        ABNORMALEND := 1 ;                                                 00128600 T  0035
    END ;                                                                  00128700 T  0037
END STATIONFIX ;                                                           00128800 T  0037
                                                                  5 IS   40 LONG, NEXT SEG    2
DEFINE CHARGE (CHARGE1) = STATIONFIX (CHARGE1, 0)#,                        00128900 T  0051
 FREEFILE (FREEFILE1) = STATIONFIX (FREEFILE1, 3)#,                        00129000 T  0051
 UNFREEFILE (UNFREEFILE1) = STATIONFIX (UNFREEFILE1, 4)#,                  00129100 T  0051
 FORGET (FORGET1) = STATIONFIX (FORGET1, 5)#,                              00129200 T  0051
 DETACH = STATIONFIX (STATION, 6)#,                                        00129300 T  0051
 ATTACH = STATIONFIX (STATION, 7)#,                                        00129400 T  0051
 REATTACH = STATIONFIX (STATION, 8)#,                                      00129500 T  0051
 NOMOREROOM = STATIONFIX (STATION, 9)#,                                    00129600 T  0051
 CHECK (CHECK1) = STATIONFIX (CHECK1, 10)# ;                               00129700 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   00129800 T  0051
PROCEDURE OUTPUT ;                                                         00130000 T  0051
  BEGIN                                                                    00130200 T  0051
STREAM PROCEDURE MOVE (S, D, W, C) ; VALUE W, C ;                          00130400 T  0051
                                                  START OF SEGMENT **********    6
BEGIN                                                                      00130600 T  0000
   SI := S ;                                                               00130800 T  0000
   DI := D ;                                                               00131000 T  0000
   DS := W WDS ;                                                           00131200 T  0000
   DS := C CHR ;                                                           00131400 T  0001
END MOVE ;                                                                 00131600 T  0001
INTEGER USER,                                                              00131800 T  0001
  T,                                                                       00132200 T  0001
  SPOT ;                                                                   00132400 T  0001
BOOLEAN X ;                                                                00132600 T  0001
LABEL FAKEOUT,                                                             00132800 T  0001
  NEXT ;                                                                   00133000 T  0001
DEFINE A = INPUT# ;                                                        00133200 T  0001
     CHARGE (0) ;                                                          00133400 T  0001
     TANKEDOUTPUT := FALSE ;                                               00133600 T  0003
     A [WDSPERBUFFER] := "+    " ;                                         00133800 T  0004
     FOR USER := 0 STEP 1 UNTIL BIGBIRD DO                                 00134000 T  0006
     BEGIN                                                                 00134200 T  0008
       IF COUNTI GEQ 0 THEN                                                00134600 T  0008
       BEGIN                                                               00134800 T  0009
         IF TIMEI - TIMEX GTR 180 THEN                                     00135000 T  0010
           GO TO FAKEOUT ;                                                 00135200 T  0013
         SPOT := HEADI ;                                                   00135400 T  0013
         IF REAL ((X := BOOLEAN (STATUS (STATIONI, 0)).[22:9])             00135600 T  0015
                  AND BOOLEAN ("6A")) NEQ 0 THEN                           00135800 T  0019
         BEGIN                                                             00136000 T  0021
           IF X.[39:1] THEN % BUSY                                         00136200 T  0021
```

```
      BEGIN                                                                    00136400 T  0022
        T := 15 ;                                                              00136600 T  0022
        GO TO FAKEOUT ;                                                        00136800 T  0023
      END ;                                                                    00137000 T  0024
      IF REAL (X AND BOOLEAN (10)) NEQ 0 AND NOT X THEN                        00137200 T  0024
        WRITE (TWXOUT, BROKEN) ; % CLEAR WRITE READY                           00137400 T  0026
      IF SPOT GEQ 0 THEN                                                       00137600 T  0031
      BEGIN                                                                    00137800 T  0032
        A [0] := FREEHEAD ;                                                    00138000 T  0032
        WRITE (IO [TAILI], 1, A [*]) ;                                         00138200 T  0034
        FREEHEAD := SPOT ;                                                     00138400 T  0040
      END ;                                                                    00138600 T  0041
      COUNTI := XDEX := -1 ;                                                   00138800 T  0041
      TIMEI := 0 ;                                                             00139000 T  0044
      BREAKI := 1 ;                                                            00139200 T  0046
      MOREINPUT := FALSE ;                                                     00139400 T  0048
      GO TO NEXT ;                                                             00139600 T  0049
    END ;                                                                      00139800 T  0050
    IF SPOT GEQ 0 THEN                                                         00140000 T  0050
    BEGIN                                                                      00140200 T  0051
      READ (IO [SPOT], 30, BUFF [*]) ;                                         00140400 T  0051
      MOVE (BUFF [BLOCK], A [1], 0, CHRSPERBUFFER) ;                           00140600 T  0057
      WRITE (TWXOUT, WORDSPERBUFFER, A [*]) [FAKEOUT] ;                        00140800 T  0061
      T := CHRSPERBUFFER ;                                                     00141000 T  0069
      IF BLOCK := BLOCK + WDSPERBUFFER GEQ 29 THEN                             00141200 T  0070
      BEGIN                                                                    00141400 T  0074
        BLOCK := 1 ;                                                           00141600 T  0074
        A [0] := FREEHEAD ;                                                    00141800 T  0076
        WRITE (IO [SPOT], 1, A [*]) ;                                          00142000 T  0078
        FREEHEAD := SPOT ;                                                     00142200 T  0083
        HEADI := BUFF [0] ;                                                    00142400 T  0084
      END ;                                                                    00142600 T  0087
    END ELSE                                                                   00142800 T  0087
    BEGIN                                                                      00143000 T  0087
      MOVE (BUFFER [1], A [1], 0, CHRSPERBUFFER) ;                             00143200 T  0087
      WRITE (TWXOUT, WORDSPERBUFFER, A [*]) [FAKEOUT] ;                        00143400 T  0090
      IF BLOC := BLOC - WDSPERBUFFER LSS 1 THEN                                00143600 T  0098
      BEGIN                                                                    00143800 T  0102
        COUNTI := -1 ;                                                         00144000 T  0103
        IF ABNORMALEND GEQ 2 THEN                                              00144200 T  0105
          ABNORMALEND := ABNORMALEND + 1 ;                                     00144400 T  0107
        GO TO NEXT ;                                                           00144600 T  0111
      END ;                                                                    00144800 T  0111
      T := CHRSPERBUFFER ;                                                     00145000 T  0111
      MOVE (BUFFER[WORDSPERBUFFER],BUFFER[1],29-WORDSPERBUFFER,0);             00145200 T  0112
    END ;                                                                      00145400 T  0116
FAKEOUT:                                                                       00145600 T  0116
    IF TIMEI:=MAX(TIMEI,TIMEX)+T*6 LSS TN OR NOT OUTPUTREADY THEN              00145800 T  0117
    BEGIN                                                                      00146000 T  0125
      TN := TIMEI ;                                                            00146200 T  0125
      TANKEDOUTPUT := TRUE ;                                                   00146400 T  0128
    END ;                                                                      00146600 T  0129
NEXT:                                                                          00146800 T  0129
  END ;                                                                        00147000 T  0130
END ;                                                                          00147200 T  0130
IF OUTPUTREADY THEN                                                            00147400 T  0132
  NEXTCLOCK := CLOCK - TO * (TN - TIMEX - 90) / 150                            00147600 T  0132
```

```
      ELSE
        NEXTCLOCK := -99 ;
      CHARGE (STATION) ;
    END OUTPUT ;

% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
PROCEDURE WRITETWX ;
  BEGIN
INTEGER STREAM PROCEDURE COUNT (S) ;

BEGIN
   SI := S ;
   28 (IF SC = "←" THEN
         JUMP OUT ;
       TALLY := TALLY + 1 ;
       SI := SI + 1) ;
   S := SI ;
   DI := S ;
   DS := LIT "←" ;
   COUNT := TALLY ;
END COUNT ;
STREAM PROCEDURE MOVE (S, D, SKPS, SKPD, N) ;
VALUE SKPS, SKPD, N ;
BEGIN
   SI := S ;
   DI := D ;
   SI := SI + SKPS ;
   DI := DI + SKPD ;
   DS := N CHR ;
END MOVE ;
INTEGER C, J, K ;
DEFINE A = PRETANK# ;
LABEL NOSKIP,
  SKIP ;
     IF BOOLEAN (ILFCRI) THEN
     BEGIN
       ILFCRI := 0 ;
       IF FIRSTCHAR (A [0]) = "≤" THEN
         J := 2 ;
     END ;
     IF C := COUNT (A) - J NEQ 0 AND NOT BOOLEAN (BREAKI) THEN
     BEGIN
       IF K := COUNTI LSS 0 THEN
       BEGIN
         BUFFER [4] := "←    " ;
         MOVE (A [0], BUFFER [1], J, 0, 28) ;
         IF TIMEI - TIMEX GEQ 180 THEN
           GO TO NOSKIP ;
         WRITE (TWXOUT, 5, BUFFER [*]) [NOSKIP : NOSKIP] ;
         TIMEI := MAX (TIMEI, TIMEX) + C × 6 ;
         GO TO SKIP ;
NOSKIP:
         COUNTI := C ;
         BLOCK := BLOC := 1 ;
         HEADI := -1 ;
         IF TIMEI LSS TN OR NOT OUTPUTREADY THEN
         BEGIN
```

```
00147800 T  0136
00148000 T  0137
00148200 T  0138
00148400 T  0139
     6 IS  145 LONG, NEXT SEG     2
00148600 T  0051
00148800 T  0051
00149200 T  0051
00149400 T  0051
START OF SEGMENT **********     7
00149600 T  0000
00149800 T  0000
00150000 T  0000
00150200 T  0001
00150400 T  0001
00150600 T  0001
00150800 T  0003
00151000 T  0003
00151200 T  0003
00151400 T  0004
00151600 T  0004
00151800 T  0005
00152000 T  0005
00152200 T  0005
00152400 T  0006
00152600 T  0006
00152800 T  0006
00153000 T  0007
00153200 T  0007
00153400 T  0008
00153600 T  0008
00153700 T  0008
00153800 T  0008
00154000 T  0008
00154200 T  0008
00154400 T  0010
00154600 T  0010
00154800 T  0012
00155000 T  0014
00155200 T  0016
00155400 T  0016
00155600 T  0020
00155800 T  0021
00156000 T  0023
00156200 T  0024
00156400 T  0026
00156600 T  0029
00156800 T  0031
00157000 T  0032
00157200 T  0041
00157400 T  0048
00157600 T  0050
00157800 T  0050
00158000 T  0052
00158200 T  0055
00158400 T  0058
00158600 T  0060
```

```
    NEXTCLOCK := CLOCK - TO × ((TN:=TIMEI)-TIMEX-120) / 150 ;                00158800 T  0061
    TANKEDOUTPUT := TRUE ;                                                  00159000 T  0067
  END ;                                                                     00159200 T  0069
  GO TO SKIP ;                                                              00159400 T  0069
END ;                                                                       00159600 T  0069
IF K LSS CHRSPERBUFFER THEN                                                 00159800 T  0069
BEGIN                                                                       00160000 T  0070
  MOVE (A, BUFFER [BLOC], J, K, CHRSPERBUFFER - K) ;                        00160200 T  0070
  J := J + CHRSPERBUFFER - K ;                                              00160400 T  0075
  IF COUNTI := K := K + C LSS CHRSPERBUFFER THEN                            00160600 T  0077
    GO TO SKIP ;                                                            00160800 T  0080
  C := K - CHRSPERBUFFER ;                                                  00161000 T  0081
END ;                                                                       00161200 T  0082
IF BLOC := BLOC + WDSPERBUFFER GEQ 29 THEN                                  00161400 T  0082
BEGIN                                                                       00161600 T  0086
  BLOC := 1 ;                                                               00161800 T  0086
  IF FREEHEAD NEQ MAXFREEHEAD THEN                                          00162000 T  0088
  BEGIN                                                                     00162200 T  0090
    READ (IO [FREEHEAD], 1, BUFFER [*]) ;                                   00162400 T  0090
    K := BUFFER [0] ;                                                       00162600 T  0096
  END ELSE                                                                  00162800 T  0098
    K := MAXFREEHEAD := MAXFREEHEAD + 1 ;                                   00163000 T  0098
  BUFFER [0] := -1 ;                                                        00163200 T  0101
  WRITE (IO [FREEHEAD], 30, BUFFER [*]) ;                                   00163400 T  0103
  IF HEADI GEQ 0 THEN                                                       00163600 T  0109
  BEGIN                                                                     00163800 T  0110
    READ (IO [TAILI], 30, BUFFER [*]) ;                                     00164000 T  0111
    BUFFER [0] := FREEHEAD ;                                                00164200 T  0118
    WRITE (IO [TAILI], 30, BUFFER [*]) ;                                    00164400 T  0120
  END ELSE                                                                  00164600 T  0126
    HEADI := FREEHEAD ;                                                     00164800 T  0126
  TAILI := FREEHEAD ;                                                       00165000 T  0129
  FREEHEAD := K ;                                                           00165200 T  0131
END ;                                                                       00165400 T  0132
MOVE (A, BUFFER [BLOC], J, 0, 29) ;                                         00165600 T  0132
COUNTI := C ;                                                               00165800 T  0136
END ;                                                                       00166000 T  0138
SKIP:                                                                       00166200 T  0138
  END WRITETWX ;                                                            00166400 T  0139
                                                                  7 IS  144 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00170000 T  0051
DEFINE ITSOLD (ITSOLD1) = BOOLEAN (KOUNT (ITSOLD1, 0, 0))#,                 00170010 T  0051
  LOC (LOC1) = KOUNT (LOC1, 1, 0)# ;                                        00170020 T  0051
INTEGER PROCEDURE KOUNT (N, M, KK) ;                                        00170030 T  0051
VALUE N, M, KK ;                                                            00170040 T  0051
INTEGER N, M, KK ;                                                          00170050 T  0051
  BEGIN                                                                     00170060 T  0051
INTEGER K ;                                                                 00170070 T  0051
                                                          START OF SEGMENT **********    8
REAL L ;                                                                    00170080 T  0000
    WHILE N LSS (L := LL [AT]).S DO                                         00170090 T  0000
      AT := L.F ;                                                           00170100 T  0005
    WHILE N GTR (L := LL [AT]).S DO                                         00170110 T  0007
      AT := L.T ;                                                           00170120 T  0012
    IF KK NEQ 0 THEN                                                        00170130 T  0014
    BEGIN                                                                   00170140 T  0014
      IF M = INFINITY THEN M := M - 1 ;                                     00170150 T  0015
```

```
      WHILE M GEQ (L := LL [AT]).S AND K := K + 1 NEQ KK DO                    00170160 T  0017
        AT := L.T ;                                                            00170170 T  0025
      KOUNT := K - REAL (M LSS L.S) ;                                          00170180 T  0028
    END ELSE                                                                   00170190 T  0030
    IF BOOLEAN (M) THEN                                                        00170200 T  0030
      KOUNT := AT                                                              00170210 T  0031
    ELSE                                                                       00170220 T  0031
      KOUNT := REAL (N = L.S) ;                                                00170230 T  0032
  END KOUNT ;                                                                  00170240 T  0034
                                                                    8 IS   37 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00170500 T  0051
DEFINE                                                                         00175000 T  0051
  WRITESEQUENCE = WRITEALINE (0)#,                                             00175100 T  0051
  WRITELFCR = WRITEALINE (1)#,                                                 00175200 T  0051
  WRITESEQ = WRITEALINE (2)#,                                                  00175300 T  0051
  WRITEQUEUED = WRITEALINE (5)#,                                               00175600 T  0051
  WRITESEGMENT = WRITEALINE (6)#,                                              00175700 T  0051
  WRITERELADDR = WRITEALINE (7)# ;                                             00175800 T  0051
PROCEDURE WRITEALINE (K) ;                                                     00175900 T  0051
VALUE K ;                                                                      00176000 T  0051
INTEGER K ;                                                                    00176100 T  0051
  BEGIN                                                                        00176200 T  0051
STREAM PROCEDURE FORM (PRETANK, N, K, LFCR, COLON, F) ;                        00176300 T  0051
                                                                  START OF SEGMENT **********    8
VALUE N, K, LFCR, COLON, F ;                                                   00176400 T  0000
BEGIN                                                                          00176500 T  0000
LABEL EXIT ;                                                                   00176600 T  0000
  DI := PRETANK ;                                                              00176700 T  0000
  LFCR (DS := 2 LIT "≤≠" ;                                                     00176800 T  0000
        K (SI := LOC N ;                                                       00176900 T  0001
           DS := K DEC ;                                                       00177000 T  0003
           F (PRETANK := DI ;                                                  00177100 T  0003
              DI := DI - K ;                                                   00177200 T  0004
              DS := K FILL ;                                                   00177300 T  0005
              DI := PRETANK) ;                                                 00177400 T  0005
           JUMP OUT) ;                                                         00177500 T  0006
        COLON (DS := LIT ":") ;                                                00177600 T  0007
        JUMP OUT TO EXIT) ;                                                    00177700 T  0008
  COLON (SI := LOC N ;                                                         00177800 T  0009
         F (DS := 7 LIT "QUEUED(" ;                                            00177900 T  0010
            DS := 2 DEC ;                                                      00178000 T  0013
            DS := LIT ")" ;                                                    00178100 T  0013
            JUMP OUT 2 TO EXIT) ;                                              00178200 T  0013
         K (DS := 9 LIT "REL ADDR=" ;                                          00178300 T  0014
            DS := 4 DEC ;                                                      00178400 T  0017
            JUMP OUT 2 TO EXIT) ;                                              00178500 T  0017
         DS := 8 LIT "SEGMENT=" ;                                              00178600 T  0018
         DS := 4 DEC ;                                                         00178700 T  0019
         JUMP OUT TO EXIT) ;                                                   00178900 T  0020
  F (N (DS := LIT " ") ; JUMP OUT TO EXIT) ;                                   00179000 T  0020
  DS := LIT ">" ;                                                              00179100 T  0024
EXIT:                                                                          00179200 T  0024
  DS := LIT "←" ;                                                              00179300 T  0024
END FORM ;                                                                     00179400 T  0025
DEFINE XON = FORM (PRETANK, 0, 0, 0, 0, 0)#,                                   00179500 T  0025
  TABIT = FORM (PRETANK, I, 0, 0, 0, 1)#,                                      00179600 T  0025
  LFCR = FORM (PRETANK, 0, 0, 1, 0, 0)#,                                       00179700 T  0025
```

```
COLON = FORM (PRETANK, 0, 0, 1, 1, 0)#,                                                00179800 T  0025
SEQ = FORM (PRETANK, IF N = INFINITY THEN 99999999 ELSE N,                             00179900 T  0025
            IF COBOLFILE THEN 6 ELSE 8, 1,                                             00180000 T  0025
            IF COBOLFILE THEN 0 ELSE 1, 1)#,                                           00180100 T  0025
OLDSEQ = FORM (PRETANK, IF N = INFINITY THEN 99999999 ELSE N,                          00180200 T  0025
               IF COBOLFILE THEN 6 ELSE 8, 1,                                          00180300 T  0025
               IF COBOLFILE THEN 0 ELSE 1, 1-REAL(ITSOLD (N)))#,                       00180400 T  0025
QUEFORM = FORM (PRETANK, READYQTOP, 0, 0, 1, 1)#,                                      00180500 T  0025
SEGMENT = FORM (PRETANK, PARAMETER2, 0, 0, 1, 0)#,                                     00180600 T  0025
RELADDR = FORM (PRETANK, PARAMETER3, 1, 0, 1, 0)#,                                     00180700 T  0025
TWX (TWX1) = BEGIN TWX1 ; WRITETWX ; END# ;                                            00180800 T  0025
   IF K = 0 THEN                                                                       00181000 T  0025
   BEGIN                                                                               00181100 T  0026
     IF FILEOPEN THEN                                                                  00181200 T  0027
     BEGIN                                                                             00181300 T  0028
       TWX (OLDSEQ) ;                                                                  00181400 T  0028
       IF INLINETOG AND EXTRALFCR THEN                                                 00181500 T  0037
         TWX (LFCR) ;                                                                  00181600 T  0039
       IF TABON AND TABAMOUNT NEQ 0 THEN                                               00181700 T  0042
       BEGIN                                                                           00181800 T  0044
         IF I := TABAMOUNT GTR 27 THEN                                                 00181900 T  0045
         BEGIN                                                                         00182000 T  0046
           I := I - 27 ;                                                               00182100 T  0046
           TWX (TABIT) ;                                                               00182200 T  0048
           I := 27 ;                                                                   00182300 T  0050
         END ;                                                                         00182400 T  0051
         TWX (TABIT) ;                                                                 00182500 T  0051
       END ;                                                                           00182600 T  0054
       END                                                                             00182700 T  0054
       ELSE                                                                            00182800 T  0054
         TWX (COLON) ;                                                                 00182900 T  0054
       IF XDEX LSS 0 AND NOT ERRTOG THEN                                               00183000 T  0059
         TWX (XON)                                                                     00183100 T  0061
       ELSE                                                                            00183200 T  0061
         ERRTOG := FALSE ;                                                             00183300 T  0065
     END                                                                               00183400 T  0067
     ELSE IF K = 1 THEN                                                                00183500 T  0067
     BEGIN                                                                             00183600 T  0068
       TWX (LFCR) ;                                                                    00183700 T  0069
       ILFCRI := 1 ;                                                                   00183800 T  0071
     END                                                                               00183900 T  0073
     ELSE                                                                              00184000 T  0073
       TWX (IF K=2 THEN SEQ ELSE IF K=5 THEN QUEFORM ELSE                              00184100 T  0073
            IF K=6 THEN SEGMENT ELSE IF K=7 THEN RELADDR) ;                            00184200 T  0073
   END WRITEALINE ;                                                                    00184300 T  0098
                                                                                 9 IS   99 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *      00190500 T  0051
PROCEDURE WRITEROW (ROW, Q, F) ;                                                       00191000 T  0051
VALUE Q,                                                                               00191500 T  0051
  F ;                                                                                  00192000 T  0051
BOOLEAN Q ;                                                                            00192500 T  0051
INTEGER F ;                                                                            00193000 T  0051
ARRAY ROW [0] ;                                                                        00193500 T  0051
   BEGIN                                                                               00193600 T  0051
STREAM PROCEDURE MOVE (S, D, SKPS, N) ;                                                00193700 T  0051
                                                                          START OF SEGMENT **********   10
VALUE SKPS, N ;                                                                        00193800 T  0000
```

```
BEGIN                                                                                00193900 T  0000
   SI := S ;                                                                         00194000 T  0000
   DI := D ;                                                                         00194100 T  0000
   SI := SI + SKPS ;                                                                 00194200 T  0000
   DS := N CHR ;                                                                     00194300 T  0001
END MOVE ;                                                                           00194400 T  0001
STREAM PROCEDURE BLANKOUTSPECIALCHARACTERS (S, D, N, K) ;                            00195000 T  0001
VALUE N,                                                                             00195500 T  0001
  K ;                                                                                00195600 T  0001
   BEGIN                                                                             00196000 T  0001
     DI := LOC N ; DS := 6 LIT "≠>><≥≤←" ;                                           00197500 T  0002
     DI := D ;                                                                       00198000 T  0003
     DS := 8 LIT "        " ;                                                        00198500 T  0003
     SI := D ;                                                                       00199000 T  0004
     DS := 9 WDS ;                                                                   00199500 T  0005
     SI := S ;                                                                       00200000 T  0005
     DI := D ;                                                                       00200500 T  0005
     2 (K (IF SC = " " THEN                                                          00201000 T  0005
     BEGIN                                                                           00201500 T  0007
       N (SI := SI - 1 ;                                                             00202000 T  0007
       IF SC = " " THEN                                                              00202500 T  0009
         DI := DI - 1 ;                                                              00203000 T  0009
       SI := SI + 1) ;                                                               00203500 T  0010
       DS := CHR ;                                                                   00204000 T  0010
     END ELSE                                                                        00204500 T  0010
     IF SC = ALPHA THEN                                                              00205000 T  0011
       DS := CHR                                                                     00205500 T  0011
     ELSE                                                                            00206000 T  0011
     BEGIN                                                                           00206500 T  0012
       D := DI ;                                                                     00207000 T  0012
       DI := LOC N ;                                                                 00207500 T  0012
       6 (IF SC = DC THEN JUMP OUT ; SI := SI - 1) ;                                 00208000 T  0012
       DI := D ;                                                                     00208500 T  0015
       IF TOGGLE THEN                                                                00209000 T  0015
         DS := 1 LIT "$"                                                             00209500 T  0015
       ELSE                                                                          00210000 T  0016
         DS := CHR ;                                                                 00210500 T  0016
     END)) ;                                                                         00211000 T  0016
   END BLANKOUTSPECIALCHARACTERS ;                                                   00219000 T  0017
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  00219500 T  0017
BOOLEAN STREAM PROCEDURE ALLBLANK (S, SKP, N) ;                                      00220000 T  0017
VALUE SKP,                                                                           00220500 T  0017
  N ;                                                                                00221000 T  0017
   BEGIN                                                                             00221500 T  0017
LABEL GRPMKIT ;                                                                      00222000 T  0018
     SI := S ;                                                                       00222500 T  0018
     SI := SI + SKP ;                                                                00223000 T  0018
     SI := SI + N ;                                                                  00223500 T  0018
     N (SI := SI - 1 ;                                                               00224000 T  0019
     IF SC NEQ " " THEN                                                              00224500 T  0020
       JUMP OUT TO GRPMKIT) ;                                                        00225000 T  0021
     TALLY := 1 ;                                                                    00225500 T  0021
     SI := SI - 1 ;                                                                  00226000 T  0022
GRPMKIT:                                                                             00226500 T  0022
     SI := SI + 1 ;                                                                  00227000 T  0022
     N := SI ;                                                                       00227500 T  0023
     DI := N ;                                                                       00228000 T  0023
```

```
      DS := 1 LIT "+" ;                                                                00228500 T  0023
      ALLBLANK := TALLY ;                                                              00229000 T  0024
    END ALLBLANK ;                                                                     00229500 T  0024
BOOLEAN DATUM ;                                                                        00229600 T  0025
DEFINE FILEINFO = F# ;                                                                 00229700 T  0025
INTEGER Z ;                                                                            00229800 T  0025
      BLANKOUTSPECIALCHARACTERS (ROW, INPUT, Q, HALFFULLLENGTH) ;                      00230000 T  0025
      IF DATAFILE THEN                                                                 00230100 T  0030
      BEGIN                                                                            00230200 T  0030
        MOVE (INPUT [9], ZIPPY [15], 0, 8) ;                                           00230300 T  0031
        DATUM := NOT ALLBLANK (ZIPPY [15], 0, 8) ;                                     00230400 T  0033
      END ;                                                                            00230500 T  0036
      EXTRALFCR := NOT (COBOLFILE OR Q:=ALLBLANK (INPUT [Z:=7], 7, 9)) ;               00231000 T  0036
      IF EXTRALFCR OR COBOLFILE THEN                                                   00231500 T  0039
        WRITELFCR ;                                                                    00232000 T  0042
      IF Q THEN                                                                        00232500 T  0043
        IF Q := ALLBLANK (INPUT [7], 0, 7) THEN                                        00233000 T  0044
          IF Q := ALLBLANK (INPUT [Z:=3], 4, 28) THEN                                  00233500 T  0047
            Q := ALLBLANK (INPUT [Z:=0], 0, 28) ;                                      00234000 T  0050
      IF NOT Q THEN                                                                    00234500 T  0054
        FOR F := 0 STEP 3 UNTIL Z DO                                                   00235000 T  0054
        BEGIN                                                                          00235500 T  0056
          MOVE (INPUT [F], PRETANK [0], 4 × F DIV 3, 28) ;                             00236000 T  0056
          WRITETWX ;                                                                   00236500 T  0059
        END ;                                                                          00237000 T  0059
      IF DATUM THEN                                                                    00237050 T  0061
      BEGIN                                                                            00237100 T  0062
        WRITELFCR ;                                                                    00237200 T  0062
        MOVE (ZIPPY [15], PRETANK, 0, 9) ;                                             00237300 T  0063
        WRITETWX ;                                                                     00237400 T  0065
      END ;                                                                            00237450 T  0065
      WRITELFCR ;                                                                      00237460 T  0065
    END WRITEROW ;                                                                     00237500 T  0066
                                                                          10 IS   69 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00237510 T  0051
PROCEDURE ERRORX (K, A, B) ;                                                           00237512 T  0051
VALUE K,                                                                               00237514 T  0051
  A,                                                                                   00237516 T  0051
  B ;                                                                                  00237518 T  0051
INTEGER K ;                                                                            00237520 T  0051
REAL A,                                                                                00237522 T  0051
  B ;                                                                                  00237524 T  0051
  BEGIN                                                                                00237526 T  0051
STREAM PROCEDURE CRUNCH (S, K, A, B) ; VALUE K, A, B ;                                 00237528 T  0051
                                                                        START OF SEGMENT **********   11
BEGIN                                                                                  00237530 T  0000
LABEL E0, E1, E2, E3, E4, E5, E6, FILENAME, CRUNCH, DEBLANK ;                          00237532 T  0000
    SI := LOC A ;                                                                      00237534 T  0000
    SI := SI + 1 ;                                                                     00237536 T  0000
    DI := S ;                                                                          00237538 T  0000
    DS := 2 LIT "≤≠" ;                                                                 00237540 T  0000
    CI := CI + K ;                                                                     00237542 T  0001
    GO TO E0 ;                                                                         00237544 T  0001
    GO TO E1 ;                                                                         00237546 T  0002
    GO TO E2 ;                                                                         00237548 T  0002
    GO TO E3 ;                                                                         00237550 T  0002
    GO TO E4 ;                                                                         00237552 T  0002
```

```
      GO TO E5 ;                                                    00237554 T  0003
      GO TO E6 ;                                                    00237556 T  0003
      GO TO E0 ;                                                    00237558 T  0003
      GO TO E0 ;                                                    00237560 T  0003
   E1:                                                              00237562 T  0004
      DS := 10 LIT "INV USER: " ;                                   00237564 T  0004
      GO TO E6 ;                                                    00237566 T  0005
   E2:                                                              00237568 T  0005
      DS := 2 LIT "NO" ;                                            00237570 T  0005
      GO TO FILENAME ;                                              00237572 T  0006
   E3:                                                              00237574 T  0006
      DS := 3 LIT "BAD" ;                                           00237576 T  0006
      GO TO FILENAME ;                                              00237578 T  0007
   E5:                                                              00237580 T  0008
      DS := 8 LIT "NO FILE " ;                                      00237582 T  0008
   E0:                                                              00237584 T  0009
      DS := 7 CHR ;                                                 00237586 T  0009
      SI := SI + 1 ;                                                00237588 T  0010
      DS := 7 CHR ;                                                 00237590 T  0010
      GO TO CRUNCH ;                                                00237592 T  0010
   E4:                                                              00237594 T  0011
      DS := 3 LIT "DUP" ;                                           00237596 T  0011
   FILENAME:                                                        00237598 T  0011
      DS := 7 LIT " FILE: " ;                                       00237600 T  0011
   E6:                                                              00237602 T  0013
      DS := 7 CHR ;                                                 00237604 T  0013
      DS := LIT "/" ;                                               00237606 T  0014
      SI := SI + 1 ;                                                00237608 T  0014
      DS := 7 CHR ;                                                 00237610 T  0015
   CRUNCH:                                                          00237612 T  0015
      DS := LIT "←" ;                                               00237614 T  0015
      SI := S ;                                                     00237616 T  0016
      DI := S ;                                                     00237618 T  0016
      28 (IF SC = " " THEN                                          00237620 T  0017
        BEGIN                                                       00237622 T  0017
   DEBLANK:                                                         00237624 T  0017
          SI := SI + 1 ;                                            00237626 T  0018
          IF SC = " " THEN                                          00237628 T  0019
            GO TO DEBLANK ;                                         00237630 T  0019
          IF SC = ALPHA THEN                                        00237632 T  0020
            DS := 1 LIT " " ;                                       00237634 T  0020
        END ELSE                                                    00237636 T  0021
          DS := CHR) ;                                              00237638 T  0021
   END CRUNCH ;                                                     00237640 T  0021
     IF A = "#000000" THEN A := "      " ;                         00237642 T  0022
     IF B = "#000000" THEN B := "      " ;                         00237644 T  0024
     CRUNCH (PRETANK, K, A, B) ;                                    00237646 T  0026
     WRITETWX ;                                                     00237648 T  0027
     IF K LEQ 6 THEN                                                00237650 T  0028
     BEGIN                                                          00237652 T  0029
       ERRTOG := TRUE ;                                             00237654 T  0029
       MOREINPUT := FALSE ;                                         00237656 T  0031
       NOSTAR := FALSE ;                                            00237658 T  0033
       XDEX := -1 ;                                                 00237660 T  0034
     END ELSE IF K = 8 THEN                                         00237662 T  0035
       ILFCRI := 1 ;                                                00237664 T  0039
   END ERRORX ;                                                     00237666 T  0042
```

```
                                                                              11 IS    43 LONG, NEXT SEG     2
DEFINE ERROR (ERROR1, ERROR2, ERROR3, ERROR4) =                                  00237668 T  0051
   BEGIN                                                                         00237670 T  0051
     ERRORX (ERROR2, ERROR3, ERROR4) ;                                           00237672 T  0051
     GO TO ERROR1 ;                                                              00237674 T  0051
   END ERROR#,                                                                   00237676 T  0051
 SHOW (SHOW1, SHOW2) = ERRORX (8, SHOW1, SHOW2)# ;                               00237678 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *                00237700 T  0051
BOOLEAN PROCEDURE FILECHECK (B) ;                                                00237740 T  0051
VALUE B ;                                                                        00237750 T  0051
BOOLEAN B ;                                                                      00237760 T  0051
BEGIN                                                                            00237770 T  0051
LABEL NEXT ;                                                                     00237780 T  0051
                                                                            START OF SEGMENT **********    12
   IF B THEN                                                                     00237790 T  0000
   BEGIN                                                                         00237800 T  0000
     IF FILECLOSED THEN                                                          00237810 T  0000
       ERROR (NEXT, 5, " OPEN: ", PARAMETER0) ;                                  00237820 T  0001
     IF B.[46:1] AND READONLYFILE THEN                                           00237830 T  0005
       ERROR (NEXT, 0, "READ ON", "LY FILE") ;                                   00237840 T  0006
   END ELSE                                                                      00237850 T  0011
     IF FILEOPEN THEN                                                            00237860 T  0011
     BEGIN                                                                       00237870 T  0012
       SEARCH (DISC, INPUT [*]) ;                                                00237880 T  0012
       IF INPUT [0] LSS FILEACCESS THEN                                          00237890 T  0014
       BEGIN                                                                     00237900 T  0015
         CHARGE (STATION) ;                                                      00237905 T  0016
         CLOSE (DISC) ;                                                          00237910 T  0017
         FILL DISC WITH PREFIX, SUFFIX ;                                         00237920 T  0019
         SEARCH (DISC, INPUT [*]) ;                                              00237930 T  0022
         IF INPUT [0] LSS FILEACCESS THEN                                        00237940 T  0024
         BEGIN                                                                   00237950 T  0025
           FILEACCESS := 0 ;                                                     00237960 T  0026
           INORDER := TRUE ;                                                     00237970 T  0027
           ERROR (NEXT, 1 + REAL (INPUT [0] LSS 0), PREFIX, SUFFIX) ;            00237980 T  0028
         END ;                                                                   00238000 T  0031
       END ;                                                                     00238010 T  0031
     END ;                                                                       00238020 T  0031
   IF FALSE THEN                                                                 00238030 T  0031
NEXT:                                                                            00238040 T  0032
     FILECHECK := TRUE ;                                                         00238050 T  0033
END FILECHECK ;                                                                  00238060 T  0033
                                                                              12 IS    37 LONG, NEXT SEG     2
DEFINE OPENCHECK = IF FILECHECK (TRUE) THEN GO TO NEXT#,                         00238070 T  0051
  READONLYCHECK = IF FILECHECK (BOOLEAN (3)) THEN GO TO NEXT#,                   00238080 T  0051
  SECURITYCHECK = IF FILECHECK (FALSE) THEN GO TO NEXT# ;                        00238090 T  0051
PROCEDURE STATE (S) ;                                                            00238100 T  0051
VALUE S ;                                                                        00238200 T  0051
BOOLEAN S ;                                                                      00238300 T  0051
   BEGIN                                                                         00238400 T  0051
STREAM PROCEDURE STUFFSTATE (N, RECORD, P0, C) ;                                 00238500 T  0051
                                                                            START OF SEGMENT **********    13
VALUE N ;                                                                        00238600 T  0000
   BEGIN                                                                         00239000 T  0000
LABEL EXIT ;                                                                     00239100 T  0000
     N (DI := C ;                                                                00239200 T  0000
        SI := P0 ;                                                               00239300 T  0001
```

```
          DS := 25 WDS ;                                                    00239400 T  0001
          SI := RECORD ;                                                    00239600 T  0001
          DS := 10 WDS ;                                                    00239700 T  0002
          JUMP OUT TO EXIT) ;                                               00240400 T  0002
        SI := C ;                                                           00240500 T  0003
        DI := PO ;                                                          00240600 T  0003
        DS := 25 WDS ;                                                      00240700 T  0003
        DI := RECORD ;                                                      00240900 T  0003
        DS := 10 WDS ;                                                      00241000 T  0004
EXIT:                                                                       00241700 T  0004
    END STUFFSTATE ;                                                        00246000 T  0004
INTEGER I, K ;                                                              00247500 T  0005
        CLOSE (DISC) ;                                                      00248000 T  0005
        K := IF S.[46:1] THEN SLOTI ELSE 46 ;                               00248500 T  0007
        IF S THEN                                                           00248600 T  0011
        BEGIN                                                               00248700 T  0011
          STUFFSTATE (1, RECORD, PARAMETERO, CONTROLS [38]) ;               00250000 T  0012
          WRITE (R1 [K], 90, CONTROLS [*]) ;                                00250600 T  0014
          IF S.[46:1] AND FILEOPEN AND REAL (MODIFIED) NEQ 0 THEN           00251000 T  0019
          BEGIN                                                             00251500 T  0022
            K := D.LEFTSIDE ;                                               00252000 T  0022
            FOR I := 0 STEP 1 UNTIL K DO                                    00252500 T  0023
            BEGIN                                                           00253000 T  0025
              IF MODIFIED THEN                                              00253100 T  0025
                WRITE (R2 [32×SLOTI + I], 256, LINKLISTS [USER32+I, *]) ;   00253200 T  0025
              MODIFIED := MODIFIED.[16:31] ;                                00253500 T  0033
            END ;                                                           00254000 T  0034
            MODIFIED := FALSE ;                                             00254100 T  0037
          END ;                                                             00254200 T  0037
        END SAVESTATE ELSE                                                  00254500 T  0037
        BEGIN                                                               00255500 T  0037
          READ (R1 [K], 90, CONTROLS [*]) ;                                 00256500 T  0038
          STUFFSTATE (0, RECORD, PARAMETERO, CONTROLS [38]) ;               00257500 T  0043
          FILL DISC WITH PREFIX, SUFFIX ;                                   00258500 T  0045
          IF S.[46:1] THEN                                                  00259000 T  0049
            MODIFIED := FALSE ;                                             00259500 T  0050
          USER32 := USER × 32 ;                                             00260000 T  0051
        END RESTORESTATE ;                                                  00260500 T  0052
        PREWHERE := -1 ;                                                    00261000 T  0052
    END STATE ;                                                             00262000 T  0053
                                                                  13 IS   56 LONG, NEXT SEG    2
DEFINE SAVESTATE = STATE (BOOLEAN (3))#,                                    00262500 T  0051
  RESTORESTATE = STATE (BOOLEAN (2))#,                                      00263000 T  0051
  UNSWAPSTATE = STATE (FALSE)#,                                             00263500 T  0051
  SWAPSTATE = STATE (TRUE)# ;                                               00264000 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00264500 T  0051
DEFINE WAIT (WAIT1, WAIT2) =                                                00265000 T  0051
  BEGIN                                                                     00265100 T  0051
    IF NOT WAITING THEN                                                     00265150 T  0051
      IF WAITX (WAIT1, WAIT2) THEN                                          00265200 T  0051
        GO TO NEXT ;                                                        00265250 T  0051
  END# ;                                                                    00265300 T  0051
BOOLEAN PROCEDURE WAITX (TOCKS, FORCED) ;                                   00265500 T  0051
VALUE TOCKS,                                                                00265600 T  0051
  FORCED ;                                                                  00265700 T  0051
INTEGER TOCKS ;                                                             00265800 T  0051
BOOLEAN FORCED ;                                                            00265900 T  0051
```

```
   BEGIN                                                                                00266000 T  0051
DEFINE SEGMENT =# ;                                                                     00266100 T  0051
                                                                                  START OF SEGMENT **********   14
     IF TOCKS GEQ CLOCK OR FORCED THEN                                                  00266200 T  0000
       IF Q THEN                                                                        00266300 T  0001
       BEGIN                                                                            00266400 T  0002
         READYQ [READYQTOP := READYQTOP + 1] := USER ;                                  00266500 T  0003
         INREADYQ := 1 ;                                                                00266600 T  0005
         WRITEQUEUED ;                                                                  00266700 T  0007
         N := RESETN ;                                                                  00266900 T  0008
         IF NOTFIRSTINPUT THEN                                                          00267000 T  0008
           SAVESTATE ;                                                                  00267100 T  0009
         STATION := 0 ;                                                                 00267200 T  0010
         WAITX := TRUE ;                                                                00267300 T  0011
       END ELSE                                                                         00267400 T  0012
       BEGIN                                                                            00267500 T  0012
         IF FORCED.[46:1] THEN                                                          00267600 T  0012
         BEGIN                                                                          00267700 T  0013
           WAITX := BOOLEAN (USER := READYQ [1]) ;                                      00267800 T  0014
           CHARGE (STATIONI) ;                                                          00267900 T  0015
           INREADYQ := 0 ;                                                              00268000 T  0017
           FOR I := 2 STEP 1 UNTIL READYQTOP DO                                         00268100 T  0019
             READYQ [I - 1] := READYQ [I] ;                                             00268200 T  0021
           READYQTOP := READYQTOP - 1 ;                                                 00268300 T  0025
           RESTORESTATE ;                                                               00268400 T  0026
           READ (IO [USER], 30, IMAGE [*]) ;                                            00268410 T  0027
         END ;                                                                          00268500 T  0032
           WRITE (PRETANK [*], WAITF) ;                                                 00268600 T  0032
           WRITETWX ;                                                                   00268700 T  0036
           WAITFLAG := TRUE ;                                                           00268800 T  0037
           READYQ [0] := USER ;                                                         00268900 T  0039
       END ;                                                                            00269300 T  0040
   END WAITX ;                                                                          00269400 T  0040
                                                                              14 IS   43 LONG, NEXT SEG     2
DEFINE INTERRUPT (INTERRUPT1) = INTERUPT (INTERRUPT1, 0, 0)#,                           00282000 T  0051
  INTERUPT (INTERUPT1, INTERUPT2, INTERUPT3) =                                          00282100 T  0051
   BEGIN                                                                                00282500 T  0051
     IF CLOCK := CLOCK = INTERUPT1 LEQ NEXTCLOCK THEN                                   00283000 T  0051
       OUTPUT ;                                                                         00283100 T  0051
     IF CLOCK LEQ 0 THEN                                                                00283200 T  0051
       IF INTERRUPTS (INTERUPT2, INTERUPT3) THEN                                        00283500 T  0051
         GO TO NEXT ;                                                                   00284000 T  0051
   END# ;                                                                               00284500 T  0051
BOOLEAN PROCEDURE INTERRUPTS (LIB, LOC) ;                                               00285000 T  0051
VALUE LIB, LOC ;                                                                        00285100 T  0051
INTEGER LIB, LOC ;                                                                      00285200 T  0051
BEGIN                                                                                   00285500 T  0051
LABEL NEWBIRD, NONE, NEXT ;                                                             00286000 T  0051
                                                                                  START OF SEGMENT **********   15
   TO := CLOCK := MAX (50, TO * 150 / (-T1 + T1 := TIMEX)) ;                            00286500 T  0000
   IF WAITING THEN                                                                      00287000 T  0007
   BEGIN                                                                                00287500 T  0008
     INPUT [5] := 0 & "+"[1:43:5] ;                                                     00288000 T  0008
     READ (TWXINPUT (0, 0), 8, INPUT [*]) [NONE] ;                                      00288500 T  0010
     QINPUT := TRUE ;                                                                   00289000 T  0017
NEWBIRD:                                                                                00289500 T  0019
     SWAPSTATE ;                                                                        00290000 T  0020
```

```
      CLOSE (LIBRARY) ;                                                                    00290100 T  0020
      CHARGE (STATION := 0) ;                                                              00290500 T  0022
      INREADYQ := 3 ;                                                                      00291000 T  0024
      Q := TRUE ;                                                                          00291500 T  0026
      PROGRAM ;                                                                            00292000 T  0027
      Q := FALSE ;                                                                         00292500 T  0028
      USER := READYQ [0] ;                                                                 00293000 T  0030
      CHARGE (STATIONI) ;                                                                  00295500 T  0031
      INREADYQ := 0 ;                                                                      00296000 T  0033
      UNSWAPSTATE ;                                                                        00296500 T  0035
      SECURITYCHECK ;                                                                      00297000 T  0035
      IF LIB NEQ 0 THEN                                                                    00297100 T  0037
      BEGIN                                                                                00297200 T  0038
        FILL LIBRARY WITH IF BOOLEAN (LIB) THEN PARAMETER1 ELSE PREFIX,                    00297300 T  0038
          IF BOOLEAN (LIB) THEN PARAMETER2 ELSE SUFFIX ;                                   00297400 T  0041
        READ SEEK (LIBRARY [LOC]) ;                                                        00297500 T  0045
      END ;                                                                                00297600 T  0050
NONE:                                                                                      00297700 T  0050
      IF RATTLEINDEX := RATTLEINDEX + 1 = 5 THEN                                           00298000 T  0051
      BEGIN                                                                                00298500 T  0053
        FOR TINK := 0 STEP 1 UNTIL READYQTOP DO                                            00299000 T  0054
        BEGIN                                                                              00299500 T  0055
          USER := READYQ [TINK] ;                                                          00300000 T  0055
          IF COUNTI LSS 0 THEN                                                             00300500 T  0056
            IF REAL (BOOLEAN (STATUS (STATIONI, 0)).[22:9] AND                             00301000 T  0057
                     BOOLEAN ("6C")) = 0 THEN                                              00301100 T  0062
              WRITE (TWXOUT, RATTLE) ;                                                     00301500 T  0063
        END ;                                                                              00302000 T  0068
        USER := READYQ [RATTLEINDEX := 0] ;                                                00302500 T  0071
        CHARGE (STATION) ;                                                                 00302700 T  0073
        IF 2 × BIGBIRD + 2 LSS STATUS (ZIPPY [*]) THEN                                     00303000 T  0074
          GO TO NEWBIRD ;                                                                  00303500 T  0078
        IF FALSE THEN                                                                      00304000 T  0078
NEXT:                                                                                      00304500 T  0079
          INTERRUPTS := TRUE ;                                                             00305000 T  0080
      END ;                                                                                00305500 T  0080
    END ;                                                                                  00306000 T  0080
    CLOCK := TO ;                                                                          00306100 T  0080
    T1 := TIMEX ;                                                                          00306200 T  0081
    IF OUTPUTREADY THEN                                                                    00306300 T  0083
      NEXTCLOCK := CLOCK - TO × (TN - T1 - 90) / 150 ;                                     00306400 T  0083
END INTERRUPTS ;                                                                           00307000 T  0088
                                                                                  15 IS   93 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *  00318000 T  0051
INTEGER PROCEDURE XFILE (P, S, FS) ;                                                       00318100 T  0051
VALUE P, S, FS ;                                                                           00318110 T  0051
REAL P, S, FS ;                                                                            00318120 T  0051
BEGIN                                                                                      00318130 T  0051
DEFINE SEGMENT = # ;                                                                       00318140 T  0051
                                                                          START OF SEGMENT **********   16
    IF P = 12 THEN                                                                         00318150 T  0000
    BEGIN                                                                                  00318160 T  0000
      IF NUM1 THEN                                                                         00318170 T  0001
      BEGIN                                                                                00318180 T  0002
        NUM1 := FALSE ;                                                                    00318190 T  0002
        P := PARAMETER1 := OCTDEC (PARAMETER1) ;                                           00318200 T  0004
      END ELSE                                                                             00318210 T  0006
```

```
      P := PARAMETER1 ;                                                                00318220 T  0006
    IF NUM2 THEN                                                                       00318230 T  0007
    BEGIN                                                                              00318240 T  0008
      NUM2 := FALSE ;                                                                  00318250 T  0009
      S := PARAMETER2 := OCTDEC (PARAMETER2) ;                                         00318260 T  0010
    END ELSE                                                                           00318270 T  0013
      S := PARAMETER2 ;                                                                00318280 T  0013
  END ;                                                                                00318290 T  0014
  FILL LIBRARY WITH P, S ;                                                             00318300 T  0014
  SEARCH (LIBRARY, INPUT [*]) ;                                                        00318310 T  0018
  IF XFILE := INPUT [0] LSS FS THEN                                                    00318320 T  0020
    ERRORX (1 + REAL (INPUT [0] LSS 0), P, S) ;                                        00318330 T  0021
END XFILE ;                                                                            00318350 T  0024
                                                                                    16 IS   27 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   00318360 T  0051
PROCEDURE READIN ;                                                                     00319000 T  0051
  BEGIN                                                                                00319100 T  0051
BOOLEAN PROCEDURE MORE ;                                                               00319210 T  0051
                                                                                  START OF SEGMENT **********   17
BEGIN                                                                                  00319220 T  0000
LABEL NEXT,                                                                            00319230 T  0000
                                                                                  START OF SEGMENT **********   18
  EXIT ;                                                                               00319240 T  0000
INTEGER STREAM PROCEDURE TRAILINGBLANKS (S, N) ;                                       00319250 T  0000
VALUE N ;                                                                              00319260 T  0000
BEGIN                                                                                  00319270 T  0000
LABEL DONE ;                                                                           00319280 T  0000
  SI := S ;                                                                            00319290 T  0000
  SI := SI + 7 ;                                                                       00319300 T  0000
  S := TALLY ;                                                                         00319310 T  0000
  DI := S ;                                                                            00319320 T  0000
  2 (N (IF SC NEQ " " THEN JUMP OUT 2 TO DONE ;                                        00319330 T  0001
        SI := SI - 1 ;                                                                 00319340 T  0003
        DI := DI + 8)) ;                                                               00319350 T  0003
DONE:                                                                                  00319360 T  0004
  TRAILINGBLANKS := DI ;                                                               00319370 T  0004
END TRAILINGBLANKS ;                                                                   00319380 T  0005
INTEGER XSUB ;                                                                         00319390 T  0006
DEFINE FILEINFO = XFILETYPE# ;                                                         00319400 T  0006
  IF FILEOPEN THEN                                                                     00319410 T  0006
  BEGIN                                                                                00319420 T  0007
    IF N GTR FINITY THEN                                                               00319430 T  0008
      IF N := LL [LAST.F].S + INC GTR FINITY THEN                                      00319440 T  0009
      BEGIN                                                                            00319450 T  0016
        N := FINITY ;                                                                  00319460 T  0016
        ERROR (NEXT, 0, "SEQ. OV", "ER=FLOW") ;                                        00319470 T  0017
      END ;                                                                            00319480 T  0023
    IF N LEQ 0 THEN                                                                    00319490 T  0023
      N := 1 ;                                                                         00319500 T  0023
  END ;                                                                                00319510 T  0025
  IF MOREINPUT THEN                                                                    00319520 T  0025
  BEGIN                                                                                00319530 T  0025
    READ (ID [USER + MAXUSERS], 30, IMAGE [*]) ;                                       00319540 T  0026
    CHRS := NCHRS ;                                                                    00319545 T  0032
    GO TO EXIT ;                                                                       00319550 T  0035
  END ;                                                                                00319560 T  0035
  IF XDEX LSS 0 THEN                                                                   00319570 T  0035
```

```
    BEGIN                                                                                00319580 T  0036
NEXT:                                                                                    00319590 T  0037
      IF NOT NOSTAR THEN                                                                 00319600 T  0037
        WRITESEQUENCE ;                                                                  00319610 T  0038
      CHRS := 0 ;                                                                        00319620 T  0039
      SAVESTATE ;                                                                        00319630 T  0041
    END ELSE                                                                             00319640 T  0042
    BEGIN                                                                                00319650 T  0042
      XSUB := XDEX * 13 ;                                                                00319660 T  0042
      WHILE XN := XN + 1 GTR XLAST DO                                                    00319670 T  0043
        IF XREPEAT := XREPEAT - 1 GTR 0 THEN                                             00319680 T  0051
          XN := XSTART                                                                   00319690 T  0056
        ELSE                                                                             00319700 T  0059
        BEGIN                                                                            00319710 T  0060
          IF XSUFFIX = "#MACRO#" THEN                                                    00319720 T  0061
          BEGIN                                                                          00319730 T  0063
            IF XFILE (XPREFIX, XSUFFIX, 7) LSS 7 THEN                                    00319740 T  0063
              GO TO NEXT ;                                                               00319750 T  0068
            READ (LIBRARY) ;                                                             00319760 T  0069
            DETACH ;                                                                     00319770 T  0073
            CLOSE (LIBRARY, PURGE) ;                                                     00319780 T  0074
            REATTACH ;                                                                   00319790 T  0075
          END ;                                                                          00319800 T  0076
          IF BOOLEAN (XNCHRS,[1:1]) THEN                                                 00319810 T  0076
          BEGIN                                                                          00319820 T  0079
            READ (IO [2*MAXUSERS+XMAX*USER+XDEX], 30, IMAGE [*]) ;                       00319830 T  0079
            CHRS := ABS (XNCHRS) ;                                                       00319840 T  0087
            XDEX := XDEX - 1 ;                                                           00319850 T  0090
            GO TO EXIT ;                                                                 00319860 T  0092
          END ;                                                                          00319870 T  0094
          IF XDEX := XDEX - 1 LSS 0 THEN                                                 00319880 T  0094
            GO TO NEXT ;                                                                 00319890 T  0095
          XSUB := XDEX * 13 ;                                                            00319900 T  0096
        END ;                                                                            00319910 T  0097
      IF XFILE (XPREFIX, XSUFFIX, 2) LSS 2 THEN                                          00319920 T  0098
        GO TO NEXT ;                                                                     00319930 T  0102
      IF XECHO THEN                                                                      00319940 T  0103
        WRITESEQUENCE ;                                                                  00319950 T  0104
      SAVESTATE ;                                                                        00319960 T  0105
      INTERRUPT (3) ;                                                                    00319970 T  0106
      READ (LIBRARY [XN - 1], 10, IMAGE [*]) ;                                           00319980 T  0111
      CLOSE (LIBRARY) ;                                                                  00319990 T  0119
      CHRS := (I := FULLLENGTH) -                                                        00320000 T  0121
          TRAILINGBLANKS (IMAGE [I,[41:4]-1], I,[41:6]) ;                                00320010 T  0126
      IF XECHO THEN                                                                      00320020 T  0130
        WRITEROW (IMAGE [*], FALSE, XFILETYPE) ;                                         00320030 T  0131
EXIT:                                                                                    00320040 T  0135
      MORE := TRUE ;                                                                     00320050 T  0135
    END ;                                                                                00320060 T  0135
END MORE ;                                                                               00320070 T  0135
                                                                  18 IS  139 LONG, NEXT SEG   17
BOOLEAN STREAM PROCEDURE LINEEDIT (S, D, C, CHRS, P, OVER80, EIGHTY1) ;                  00321000 T  0000
VALUE C,                                                                                 00321100 T  0000
    P,                                                                                   00321200 T  0000
    OVER80,                                                                              00321300 T  0000
    EIGHTY1 ;                                                                            00321400 T  0000
      BEGIN                                                                              00321500 T  0000
```

```
LOCAL T,                                                                00321600 T  0000
   PERCENT1, PERCENT ;                                                  00321700 T  0000
LABEL AROUND, NEXT ;                                                    00321800 T  0000
      P (DI := LOC PERCENT ; DS := 14 LIT "%?-+=≠(<)>[≤]≥") ;            00321900 T  0000
      SI := LOC C ;                                                     00322000 T  0003
      DI := LOC T ;                                                     00322100 T  0003
      SI := SI + 6 ;                                                    00322200 T  0004
      DI := DI + 7 ;                                                    00322300 T  0004
      DS := CHR ;                                                       00322400 T  0004
      SI := S ;                                                         00322500 T  0004
      DI := D ;                                                         00322600 T  0005
      T (DI := DI + 32 ;  DI := DI + 32) ;                              00322700 T  0005
      DI := DI + C ;                                                    00322800 T  0007
      56(IF SC = "+" THEN                                               00322900 T  0007
           GO TO AROUND ;                                               00323000 T  0008
         IF SC = "≥" THEN% DISCONNECT OR EXCLAMATION                    00323010 T  0008
         BEGIN                                                          00323020 T  0009
           TALLY := 1 ;                                                 00323030 T  0009
           GO TO AROUND ;                                               00323040 T  0009
         END ;                                                          00323050 T  0010
         IF SC = "≠" THEN% LINE ERASE                                   00323100 T  0010
         BEGIN                                                          00323200 T  0010
           C := TALLY ;                                                 00323300 T  0010
           OVER80 := TALLY ;                                            00323400 T  0011
           DI := D ;                                                    00323500 T  0011
           GO TO AROUND ;                                               00323600 T  0011
         END ;                                                          00323700 T  0012
         IF SC = "≤" THEN% BACKSPACE                                    00323800 T  0012
         BEGIN                                                          00323900 T  0012
           S := SI ;                                                    00324000 T  0012
           T := DI ;                                                    00324100 T  0013
           SI := LOC C ;                                                00324200 T  0013
           DI := LOC LINEEDIT ;                                         00324300 T  0013
           IF 8 SC NEQ DC THEN                                          00324400 T  0014
           BEGIN                                                        00324500 T  0014
             OVER80 (SI := SI - 8 ;                                     00324600 T  0014
                     DI := LOC EIGHTY1 ;                                00324700 T  0016
                     IF 8 SC = DC THEN                                  00324800 T  0016
                       OVER80 := TALLY) ;                               00324900 T  0017
             SI := C ;                                                  00325000 T  0017
             SI := SI - 8 ;                                             00325100 T  0017
             C := SI ;                                                  00325200 T  0018
             DI := T ;                                                  00325300 T  0018
             DI := DI - 1 ;                                             00325400 T  0018
           END ELSE                                                     00325500 T  0018
             DI := T ;                                                  00325600 T  0019
           SI := S ;                                                    00325700 T  0019
AROUND:                                                                 00325800 T  0019
         END ELSE                                                       00325900 T  0019
         BEGIN                                                          00326500 T  0020
           S := SI ;                                                    00326600 T  0020
           OVER80 (DI := DI + 1 ;                                       00326700 T  0020
                   SI := C ;                                            00326900 T  0021
                   SI := SI + 8 ;                                       00327000 T  0022
                   C := SI ;                                            00327100 T  0022
                   SI := S ;                                            00327200 T  0022
                   JUMP OUT TO AROUND) ;                                00327300 T  0022
```

```
T := DI ;                                                                00327500 T  0023
P (DI := S ;                                                             00327600 T  0023
   SI := T ;                                                             00327700 T  0025
   SI := SI - 1 ;                                                        00327800 T  0025
   IF SC = "%" THEN                                                      00327900 T  0025
   BEGIN                                                                 00328000 T  0026
     SI := LOC PERCENT ;                                                 00328100 T  0026
     7 (IF SC = DC THEN                                                  00328200 T  0026
        BEGIN                                                            00328300 T  0027
          DI := T ;                                                      00328400 T  0027
          DI := DI - 1 ;                                                 00328500 T  0028
          DS := CHR ;                                                    00328600 T  0028
          SI := S ;                                                      00328700 T  0028
          JUMP OUT 2 TO AROUND ;                                         00328900 T  0029
        END ;                                                            00329000 T  0029
        SI := SI + 1 ;                                                   00329100 T  0029
        DI := DI - 1) ;                                                  00329200 T  0030
   END) ;                                                                00329300 T  0030
SI := C ;                                                                00330600 T  0030
SI := SI + 8 ;                                                           00330700 T  0031
C := SI ;                                                                00330800 T  0031
SI := LOC C ;                                                            00330900 T  0031
DI := LOC EIGHTY1 ;                                                      00331000 T  0031
IF 8 SC = DC THEN                                                        00331100 T  0032
BEGIN                                                                    00331200 T  0032
  TALLY := 1 ;                                                           00331300 T  0032
                  OVER80 := TALLY ;                                      00331400 T  0033
                  TALLY := 0 ;                                           00331500 T  0033
                END ;                                                    00331600 T  0033
                SI := S ;                                                00331700 T  0033
                DI := T ;                                                00331800 T  0034
                IF TOGGLE THEN                                           00331900 T  0034
                  DI := DI + 1                                           00332000 T  0034
                ELSE BEGIN                                               00332100 T  0034
                  DS := CHR ;                                            00332200 T  0035
                  SI := SI - 1 ;                                         00332300 T  0035
                END ;                                                    00332400 T  0035
                GO TO NEXT ;                                             00332500 T  0035
              END ;                                                      00332550 T  0035
              IF SC = "←" THEN JUMP OUT ;                                00332560 T  0035
              IF SC = "≥" THEN JUMP OUT ;                                00332570 T  0036
NEXT:                                                                    00332580 T  0037
              SI := SI + 1) ;                                            00332600 T  0037
        SI := LOC C ;                                                    00332700 T  0039
        DI := CHRS ;                                                     00332800 T  0039
        DS := WDS ;                                                      00332900 T  0039
        LINEEDIT := TALLY ;                                              00333100 T  0039
END LINEEDIT ;                                                           00333200 T  0040
BOOLEAN PROCEDURE FINALANALYSIS ;                                        00333210 T  0041
BEGIN                                                                    00333220 T  0041
STREAM PROCEDURE MOVE (S, D, SKPS, SKPD, N) ;                            00333230 T  0041
                                                                START OF SEGMENT **********   19
VALUE SKPS, SKPD, N ;                                                    00333240 T  0000
   BEGIN                                                                 00333250 T  0000
LOCAL T ;                                                                00333260 T  0000
     SI := LOC N ;                                                       00333270 T  0000
     DI := LOC T ;                                                       00333280 T  0000
```

```
      SI := SI + 6 ;                                                    00333290 T  0000
      DI := DI + 7 ;                                                    00333300 T  0000
      DS := CHR ;                                                       00333310 T  0001
      SI := S ;                                                         00333320 T  0001
      DI := D ;                                                         00333330 T  0001
      SI := SI + SKPS ;                                                 00333340 T  0001
      DI := DI + SKPD ;                                                 00333350 T  0002
      T (DS := 32 CHR ;  DS := 32 CHR) ;                                00333360 T  0002
      DS := N CHR ;                                                     00333370 T  0004
    END MOVE ;                                                          00333380 T  0005
INTEGER STREAM PROCEDURE HUNT (S, D, C, N) ;                            00333390 T  0005
VALUE C,                                                                00333400 T  0005
  N ;                                                                   00333500 T  0005
     BEGIN                                                              00333600 T  0005
LABEL AGAIN,                                                            00333700 T  0006
  XIT ;                                                                 00333800 T  0006
       SI := D ;                                                        00333900 T  0006
       DI := D ;                                                        00334000 T  0006
       DS := 8 LIT " " ;                                                00334100 T  0006
       DS := 9 WDS ;                                                    00334200 T  0007
       D := TALLY ;                                                     00334300 T  0008
       DI := LOC D ;                                                    00334400 T  0008
       SI := LOC C ;                                                    00334500 T  0008
       SI := SI + 7 ;                                                   00334600 T  0008
       DS := CHR ;                                                      00334700 T  0009
AGAIN:                                                                  00334800 T  0009
       SI := LOC N ;                                                    00334900 T  0009
       SI := SI + 1 ;                                                   00335000 T  0010
       IF 7 SC = DC THEN                                                00335100 T  0010
         GO TO XIT ;                                                    00335200 T  0011
       SI := N ;                                                        00335300 T  0011
       SI := SI - 8 ;                                                   00335400 T  0011
       N := SI ;                                                        00335500 T  0011
       SI := S ;                                                        00335600 T  0012
       DI := LOC D ;                                                    00335700 T  0012
       IF SC = DC THEN                                                  00335800 T  0012
         GO TO XIT ;                                                    00335900 T  0013
       S := SI ;                                                        00336000 T  0013
       SI := HUNT ;                                                     00336100 T  0013
       SI := SI + 8 ;                                                   00336200 T  0013
       HUNT := SI ;                                                     00336300 T  0014
       GO TO AGAIN ;                                                    00336400 T  0014
XIT:                                                                    00336500 T  0014
     END HUNT ;                                                         00336600 T  0014
BOOLEAN STREAM PROCEDURE MORE (IMAGE, INPUT, C, CHRS) ;                 00344000 T  0016
VALUE C ;                                                               00344010 T  0016
     BEGIN                                                              00344500 T  0016
LOCAL QUOTES,                                                           00345000 T  0016
  ENDQUOTE,                                                             00345500 T  0016
  ZERO,                                                                 00345510 T  0016
  TEMP ;                                                                00346000 T  0016
LABEL NOTHINGYET,                                                       00346500 T  0016
  BUMP,                                                                 00347000 T  0016
  FOUNDQUOTE,                                                           00347500 T  0016
  FOUNDSEMICOLAN,                                                       00348000 T  0016
  LOOP,                                                                 00348500 T  0016
  XIT,                                                                  00349000 T  0016
```

```
  EXIT ;                                                                00349100 T  0016
       SI := IMAGE ;                                                    00349500 T  0016
       DI := LOC QUOTES ;                                               00350000 T  0016
       DS := 2 LIT """" ;                                               00350500 T  0016
       DS := 6 LIT ",.()[]" ;                                           00351000 T  0017
       DI := LOC ENDQUOTE ;                                             00351100 T  0018
       DS := 2 LIT ";;" ;                                               00351200 T  0018
LOOP:                                                                   00351300 T  0018
       IMAGE := SI ;                                                    00351310 T  0018
       SI := LOC C ;                                                    00351330 T  0019
       DI := LOC ZERO ;                                                 00351340 T  0019
       IF 8 SC = DC THEN                                                00351350 T  0019
         GO TO XIT ;                                                    00351360 T  0020
       SI := C ;                                                        00351370 T  0020
       SI := SI - 8 ;                                                   00351380 T  0020
       C := SI ;                                                        00351390 T  0021
       SI := IMAGE ;                                                    00351400 T  0021
       CI := CI + MORE ;                                                00351500 T  0021
       GO TO NOTHINGYET ;                                               00352000 T  0022
       GO TO LOOP ;                                                     00352500 T  0022
       GO TO FOUNDQUOTE ;                                               00353000 T  0022
NOTHINGYET:                                                             00353500 T  0022
       IF SC = ALPHA THEN                                               00354000 T  0022
         GO TO BUMP ;                                                   00354500 T  0023
       IF SC = " " THEN                                                 00355000 T  0023
         GO TO BUMP ;                                                   00355500 T  0024
       DI := LOC QUOTES ;                                               00356000 T  0024
       4 (IF SC = DC THEN                                               00356500 T  0024
       BEGIN                                                            00359000 T  0025
         TEMP := SI ;                                                   00359500 T  0025
         ENDQUOTE := DI ;                                               00360000 T  0026
         DI := LOC ENDQUOTE ;                                           00360500 T  0026
         SI := ENDQUOTE ;                                               00361000 T  0026
         DS := 1 CHR ;                                                  00361500 T  0027
         TALLY := 2 ;                                                   00362000 T  0027
         MORE := TALLY ;                                                00362500 T  0027
         SI := TEMP ;                                                   00363000 T  0027
         JUMP OUT TO LOOP ;                                             00363500 T  0028
       END ;                                                            00364000 T  0028
       SI := SI - 1 ;                                                   00364100 T  0028
       DI := DI + 1) ;                                                  00364200 T  0028
       IF SC = ";" THEN                                                 00365500 T  0029
         GO TO FOUNDSEMICOLAN ;                                         00369000 T  0029
BUMP:                                                                   00371500 T  0030
       SI := SI + 1 ;                                                   00372000 T  0030
       GO TO LOOP ;                                                     00372500 T  0030
FOUNDQUOTE:                                                             00374500 T  0030
       DI := LOC ENDQUOTE ;                                             00375000 T  0030
       IF SC = DC THEN                                                  00375500 T  0031
       BEGIN                                                            00376000 T  0031
         DI := DI - 1 ;                                                 00376100 T  0031
         DS := LIT ";" ;                                                00376200 T  0032
         TALLY := 0 ;                                                   00376500 T  0033
         MORE := TALLY ;                                                00377000 T  0033
       END ;                                                            00377500 T  0033
       GO TO LOOP ;                                                     00378000 T  0033
XIT:                                                                    00378500 T  0033
```

```
            SI := LOC ENDQUOTE ;                                                    00378600 T  0033
            DI := IMAGE ;                                                           00378700 T  0034
            DS := 2 CHR ;                                                           00378800 T  0034
            GO TO EXIT ;                                                            00378900 T  0034
FOUNDSEMICOLAN:                                                                     00378910 T  0035
            TALLY := 1 ;                                                            00378920 T  0035
            MORE := TALLY ;                                                         00378930 T  0035
            SI := LOC C ;                                                           00378940 T  0035
            DI := CHRS ;                                                            00378950 T  0035
            DS := WDS ;                                                             00378960 T  0036
            SI := LOC C ;                                                           00378970 T  0036
            DI := LOC TEMP ;                                                        00378980 T  0036
            SI := SI + 6 ;                                                          00378990 T  0036
            DI := DI + 7 ;                                                          00379000 T  0037
            DS := CHR ;                                                             00379010 T  0037
            SI := IMAGE ;                                                           00379020 T  0037
            SI := SI + 1 ;                                                          00379030 T  0037
            DI := INPUT ;                                                           00379040 T  0038
            TEMP (DS := 32 CHR ; DS := 32 CHR) ;                                    00379050 T  0038
            DS := C CHR ;                                                           00379060 T  0040
EXIT:                                                                               00379070 T  0040
        END MORE ;                                                                  00379500 T  0040
INTEGER STREAM PROCEDURE FIX (IM, TAB, C, Z, P, Q) ;                                00380000 T  0042
VALUE TAB,                                                                          00380500 T  0042
  C,                                                                                00381000 T  0042
  P,                                                                                00381100 T  0042
  Q ;                                                                               00381200 T  0042
    BEGIN                                                                           00381500 T  0042
LOCAL T ;                                                                           00382000 T  0042
        P (SI := IM ;                                                               00382100 T  0042
           IF SC = "%" THEN                                                         00382110 T  0043
           BEGIN                                                                    00382120 T  0043
             SI := SI + 1 ;                                                         00382130 T  0043
             IF SC = "*" THEN                                                       00382140 T  0044
             BEGIN                                                                  00382150 T  0045
               SI := C ;                                                            00382180 T  0045
               SI := SI - 8 ;                                                       00382190 T  0045
               C := SI ;                                                            00382200 T  0046
               TALLY := 1 ;                                                         00382210 T  0046
               FIX := TALLY ;                                                       00382220 T  0046
             END ;                                                                  00382230 T  0046
           END) ;                                                                   00382240 T  0046
        SI := Z ;                                                                   00382500 T  0047
        DI := Z ;                                                                   00383000 T  0047
        DS := 8 LIT "        " ;                                                    00383500 T  0047
        DS := 9 WDS ;                                                               00384000 T  0048
        SI := LOC C ;                                                               00384100 T  0049
        DI := LOC T ;                                                               00384200 T  0049
        SI := SI + 6 ;                                                              00384300 T  0049
        DI := DI + 7 ;                                                              00384400 T  0049
        DS := CHR ;                                                                 00384500 T  0050
        SI := IM ;                                                                  00384600 T  0050
        SI := SI + FIX ;                                                            00384650 T  0050
        DI := Z ;                                                                   00384700 T  0051
        DI := DI + TAB ;                                                            00384800 T  0051
        T (DS := 32 CHR ;  DS := 32 CHR) ;                                          00384900 T  0051
        DS := C CHR ;                                                               00385000 T  0053
```

```
      SI := Z ;                                                                  00389000 T  0054
      DI := IM ;                                                                 00389500 T  0054
      DS := 10 WDS ;                                                             00390000 T  0054
      Q (DI := IM ; DS := 1 LIT "0") ;                                           00390100 T  0054
    END FIX ;                                                                    00390500 T  0056
INTEGER C,                                                                       00390505 T  0057
  H,                                                                             00390508 T  0057
  K ;                                                                            00390510 T  0057
LABEL ERR, NEXT ;                                                                00390520 T  0057
    NOSTAR := (FIRSTCHAR (IMAGE [0]) NEQ "*" OR H := CHRS = 0)                   00390540 T  0057
              AND READWRITEFILE ;                                                00390550 T  0062
    NOTFIRSTINPUT := MOREINPUT ;                                                 00390560 T  0065
    IF NOSTAR THEN                                                               00390570 T  0067
    BEGIN                                                                        00390580 T  0068
      I := IF COBOLFILE THEN 6 ELSE 0 ;                                          00390590 T  0068
      IF XDEX GEQ 0 THEN IF BOOLEAN (XFILETYPE) THEN I := 0 ;                    00390595 T  0070
      MOREINPUT := FALSE ;                                                       00390600 T  0075
      IF H + TABAMOUNT GTR LENGTH THEN                                           00390610 T  0077
        GO TO ERR ;                                                              00390630 T  0082
      H := H + TABAMOUNT + I - FIX (IMAGE, TABAMOUNT + I, H,                     00390640 T  0082
                ZIPPY, TRANSLATING AND H GEQ 2, I - 6) ;                         00390650 T  0086
      IF COLUMNS THEN                                                            00390660 T  0090
      BEGIN                                                                      00390670 T  0091
        FOR K := 1 STEP 1 UNTIL COLSTOPS DO                                      00390680 T  0091
          IF I := MIN (H, MAXCOLSTOP) NEQ                                        00390690 T  0095
             C := HUNT (IMAGE, ZIPPY, CHARACTER, I) THEN                         00390700 T  0098
          BEGIN                                                                  00390710 T  0102
            WHILE C GEQ I := COLSTOP [K] DO                                      00390720 T  0102
              K := K + 1 ;                                                       00390730 T  0105
            I := I - 1 ;                                                         00390760 T  0107
            MOVE (IMAGE, ZIPPY, 0, 0, C) ;                                       00390770 T  0108
            IF H := H + I - (C := C + 1) GTR FULLLENGTH THEN                     00390780 T  0111
            BEGIN                                                                00390785 T  0116
ERR:                                                                             00390790 T  0116
              FINALANALYSIS := TRUE ;                                            00390795 T  0117
              ERROR (NEXT, 0, "INPUT  ", "OVERFLW") ;                            00390800 T  0117
            END ;                                                                00390805 T  0122
            MOVE (IMAGE [C,[41:4]], ZIPPY [I,[41:4]], C,[45:3],                  00390810 T  0122
              I,[45:3], H - I) ;                                                 00390820 T  0125
            MOVE (ZIPPY, IMAGE, 0, 0, 80) ;                                      00390840 T  0127
          END ELSE                                                               00390850 T  0129
            K := 5 ;                                                             00390860 T  0129
      END ;                                                                      00390870 T  0131
      CHRS := H ;                                                                00390875 T  0131
      IF XDEX LSS 0 AND NOT INLINETOG AND N := N+INC LSS INFINITY THEN            00390880 T  0133
        WRITESEQUENCE ;                                                          00390910 T  0136
      N := N - INC ;                                                             00390920 T  0138
    END                                                                          00391110 T  0139
    ELSE                                                                         00391120 T  0139
    BEGIN                                                                        00391130 T  0139
      IF H GTR 240 THEN                                                          00391140 T  0142
        GO TO ERR ;                                                              00391150 T  0142
      INLINETOG := FALSE ;                                                       00391160 T  0143
      MOREINPUT := MORE (IMAGE, ZIPPY, H, NCHRS) ;                               00391170 T  0145
      IF MOREINPUT THEN                                                          00391175 T  0151
        WRITE (IO [USER + MAXUSERS], 30, ZIPPY [*]) ;                            00391180 T  0151
    END ;                                                                        00391190 T  0157
```

```
NEXT:
END FINALANALYSIS ;

INTEGER C,
  LASTUSER ;
REAL X ;
LABEL AGAIN,
  INPUTFULL,
  EXIT,
  NEXT,
  ESCAPE ;
INTEGER PROCEDURE READTWX ;
BEGIN
LABEL NONE, TROUBLE, EXIT ;

REAL TIMEOUT, X ;
   INPUT [5] := 0 & "←"[1:43:5] ;
   IF NOT Q THEN
     TIMEOUT := IF OUTPUTREADY THEN MAX(0,MIN(15,(TN-TIMEX-60)/60))
                ELSE 15 ;
   READ (TWXINPUT (0, TIMEOUT), 8, INPUT [*]) [NONE:TROUBLE] ;
   GO TO EXIT ;
NONE:
   IF Q THEN
   BEGIN
     USER := MAXUSERS ;
     READTWX := 1 ;                    % ESCAPE
     GO TO EXIT ;
END ;
IF OUTPUTREADY THEN
  OUTPUT ;
T1 := TIMEX ;
FOR USER := 0 STEP 1 UNTIL BIGBIRD DO
BEGIN
  CHECK (STATION1) ;
  IF BOOLEAN (ABNORMALEND) THEN
  BEGIN
    READTWX := 1 ;
    GO TO EXIT ;
  END ;
  IF X := (T1 - TIME1)/1000 LSS 0 THEN
    X := X + 5184 ;
  IF X GTR 15 AND X LSS 100 THEN
  BEGIN
    IF X LSS 18 THEN
      FIRSTCHANCE := 0
    ELSE IF X GEQ 36 THEN
    BEGIN
      WRITE (PRETANK [*], EOJ) ;
      WRITETWX ;
      ABNORMALEND := READTWX := 1 ;
      GO TO EXIT ;
    END ELSE IF FIRSTCHANCE = 0 THEN
    BEGIN
      FIRSTCHANCE := 1 ;
      X := TIME1 ;
      ERRORX (7, "LOOK    ", "ALIVE, ") ;
```

```
                                                00391210 T  0157
                                                00391230 T  0158
                                      19 IS  161 LONG, NEXT SEG   17
                                                00391240 T  0041
                                                00391250 T  0041
                                                00391260 T  0041
                                                00392000 T  0041
                                                00392500 T  0041
                                                00394500 T  0041
                                                00394600 T  0041
                                                00394700 T  0041
                                                00394800 T  0041
                                                00394900 T  0041
                                                00395000 T  0041
                                             START OF SEGMENT **********   20
                                                00395100 T  0000
                                                00395200 T  0000
                                                00395300 T  0002
                                                00395400 T  0003
                                                00395500 T  0009
                                                00395600 T  0012
                                                00395700 T  0020
                                                00395800 T  0021
                                                00395900 T  0022
                                                00396000 T  0022
                                                00396100 T  0023
                                                00396200 T  0024
                                                00396300 T  0024
                                                00396400 T  0025
                                                00396500 T  0025
                                                00396600 T  0025
                                                00396700 T  0026
                                                00396800 T  0028
                                                00396900 T  0029
                                                00397100 T  0029
                                                00397200 T  0031
                                                00397300 T  0032
                                                00397400 T  0032
                                                00397500 T  0033
                                                00397600 T  0034
                                                00397700 T  0034
                                                00397800 T  0037
                                                00397900 T  0039
                                                00398000 T  0041
                                                00398100 T  0041
                                                00398200 T  0042
                                                00398300 T  0044
                                                00398400 T  0047
                                                00398500 T  0048
                                                00398600 T  0052
                                                00398700 T  0052
                                                00398800 T  0055
                                                00398900 T  0055
                                                00399000 T  0058
                                                00399100 T  0058
                                                00399200 T  0060
                                                00399300 T  0062
```

```
          TIMEI := X ;
        END ;
      END ;
    END ;
    READTWX := 2 ;                          % AGAIN
    GO TO EXIT ;
TROUBLE:
    READ (TWXINPUT (0, 0), 1, INPUT [*]) ;
    INPUT [1] := "≥" ;
EXIT:
END READTWX ;

PROCEDURE INITIALIZE ;
  BEGIN
MONITOR INTOVR, FLAG ;

INTEGER I,
  C ;
REAL U ;
BOOLEAN OLDUSER ;
DEFINE DIRCTRY = CONTROLS# ;
LABEL OLD,
  FAULT,
  NEW,
  MAILCALL,
  NEXT ;
    USER := BIGBIRD := BIGBIRD + 1 ;
    ATTACH ;
    STATIONI := STATION ;
    IF USERCODEI := USERCODE  = -1 THEN
      USERCODE := OCTDEX (100×STATION.[9:4]+STATION.[14:4]) ;
    COUNTI := -1 ;
    ILFCRI := 1 ;
    ERRORX (7, "VERSION", OCTDEX (VERSION)) ;
FAULT:
    READ (R1 [45], 90, DIRCTRY [*]) ;
    IF OLDUSER THEN
    BEGIN
      OLDUSER := FALSE ;
      I := C + C ;
      ERROR (OLD, 0, "BACKUP ", "ERROR. ") ;
    END ;
    C := 200 ;
    FOR I := 0 STEP 2 WHILE U := DIRCTRY [I] NEQ 12 DO
      IF USERCODE = U THEN
      BEGIN
        OLDUSER := TRUE ;
        IF STATION = DIRCTRY [I + 1] THEN
          GO TO OLD ;
        C := I ;
      END ELSE
      IF U = 0 AND NOT OLDUSER THEN
        C := I ;
    IF C NEQ 200 THEN
      I := C
    ELSE IF I LEQ 88 THEN
      DIRCTRY [I + 2] := 12
```

```
00399400 T  0063
00399500 T  0065
00399600 T  0065
00399700 T  0065
00399800 T  0067
00399900 T  0068
00400000 T  0071
00400100 T  0071
00400200 T  0077
00400300 T  0078
00400400 T  0079
20 IS   84 LONG, NEXT SEG   17
00406000 T  0041
00407000 T  0041
00407500 T  0041
START OF SEGMENT **********   21
00408000 T  0002
00408100 T  0002
00408200 T  0002
00408500 T  0002
00408600 T  0002
00409000 T  0002
00409500 T  0002
00410000 T  0002
00410500 T  0002
00410600 T  0002
00420000 T  0002
00421000 T  0004
00421500 T  0005
00422500 T  0007
00423000 T  0010
00426000 T  0014
00426100 T  0016
00427900 T  0018
00427910 T  0020
00427920 T  0021
00427930 T  0026
00427940 T  0026
00427950 T  0027
00427960 T  0028
00427970 T  0029
00427980 T  0034
00428000 T  0034
00428100 T  0034
00428200 T  0039
00428300 T  0040
00428400 T  0040
00428500 T  0041
00428600 T  0043
00428700 T  0043
00428800 T  0044
00428900 T  0044
00429000 T  0046
00429100 T  0048
00429300 T  0048
00432500 T  0049
00433500 T  0051
```

```
      ELSE                                                                    00434500 T  0052
        WHILE DIRCTRY [I := I - 2] LSS 0 DO ;                                 00435000 T  0053
OLD:                                                                          00436500 T  0057
      C := SLOTI := I / 2 ;                                                   00437000 T  0057
      DIRCTRY [I] := - USERCODE ;                                             00437500 T  0060
      DIRCTRY [I + 1] := STATION ;                                            00438000 T  0061
      WRITE (R1 [45], 90, DIRCTRY [*]) ;                                      00438500 T  0063
      IF NOT OLDUSER THEN                                                     00438600 T  0068
        GO TO NEW ;                                                           00438700 T  0068
      INTOVR := FAULT ;                                                       00439500 T  0069
      FLAG := FAULT ;                                                         00440500 T  0070
      RESTORESTATE ;                                                          00441000 T  0072
      STATION := STATIONI ;                                                   00441500 T  0073
      IF VN LSS 94 OR VN GTR VERSION THEN                                     00441800 T  0075
        GO TO FAULT ;                                                         00441900 T  0077
      IF FILECLOSED THEN                                                      00443000 T  0078
        GO TO MAILCALL ;                                                      00444500 T  0078
      IF D GTR MAXFILELENGTH THEN                                             00445100 T  0079
        GO TO FAULT ;                                                         00445200 T  0080
      READ SEEK (R2 [32 × C]) ;                                               00445300 T  0080
      SECURITYCHECK ;                                                         00447000 T  0086
      IF INPUT [5] + 2 LSS D OR INPUT [3] NEQ 10 THEN                         00450000 T  0088
        ERROR (MAILCALL, 3, PREFIX, SUFFIX) ;                                 00452500 T  0090
      AT := D.LEFTSIDE ;                                                      00453000 T  0094
      FOR I := 0 STEP 1 UNTIL AT DO                                           00453100 T  0095
        READ (R2, 256, LINKLISTS [USER32 + I, *]) [FAULT] ;                   00453200 T  0096
      AT := 0 ;                                                               00453500 T  0104
      FOR I := 1 STEP 1 UNTIL D DO                                            00454000 T  0105
      BEGIN                                                                   00454100 T  0106
        IF AT NEQ LL [AT := LL [AT] . T] . F THEN                             00454400 T  0106
          I := D                                                              00454500 T  0114
        ELSE IF AT = 1 THEN                                                   00454600 T  0114
          ERROR (NEXT, 6, PREFIX, SUFFIX) ;                                   00455000 T  0116
      END ;                                                                   00455100 T  0119
      ERROR (MAILCALL, 7, "LINKLIS", "T ERROR") ;                             00456500 T  0121
NEW:                                                                          00458500 T  0125
      WRITE (R2 [32 × C + 31], 1, IMAGE [*]) ;                                00459000 T  0125
      LOCK (R2) ;                                                             00459500 T  0131
      USER32 := USER × 32 ;                                                   00460000 T  0132
      BOOL := INITIALBOOL ;                                                   00461000 T  0134
      INC := 100 ;                                                            00462500 T  0134
      MACROLIBRARY := "MACRO  " ;                                             00462900 T  0135
      CHARACTER := "#" ;                                                      00463000 T  0136
      SAVEFACTOR := 7 ;                                                       00463500 T  0137
      COLSTOPS := STRINGI := 0 ;                                              00465000 T  0138
      FILL RSWD [*] WITH "EXECUTE", "DITTO  ", "COPY   ", "INLINE ",          00466000 T  0140
                                                                  START OF SEGMENT **********   22
        "ZIP    ", "CHANGE ", "EDIT   ", "SAVE   ", "RESEQ  ",                00466500 T  0142
        "PUNCH  ", "PRINT  ", "DELETE ", "CLOSE  ", "COMPILE",                00467000 T  0142
        "COLUMN ", "SCAN   ", "LISTING", "INC    ", "TAB    ",                00467500 T  0142
        "PERCENT", "QUICK  ", "LIST   ", "OPEN   ", "MAIL   ",                00468000 T  0142
        "TEACH  ", "REMOVE ", "REPLACE", "END    " ;                          00468500 T  0142
                                                                  22 IS   28 LONG, NEXT SEG   21
MAILCALL:                                                                     00469000 T  0142
      FILEACCESS := 0 ;                                                       00469100 T  0143
      INORDER := TRUE ;                                                       00469200 T  0143
NEXT:                                                                         00469300 T  0145
```

```
      TRANSLATEI := REAL (TRANSLATING) ;                                                  00472100 T  0146
      VN := VERSION ;                                                                    00472200 T  0148
      ERRORX (0, (IF XFILE ("MAIL % ", USERCODE, -1) = 7 THEN "MAIL % "                  00489500 T  0149
        ELSE "HELLO  ") & REAL (NOT OLDUSER)[42:47:1], USERCODE) ;                       00490000 T  0153
    END INITIALIZE ;                                                                     00490500 T  0155
                                                                          21 IS  166 LONG, NEXT SEG   1?
      LASTUSER := MAXUSERS ;                                                             00490600 T  0041
      IF QINPUT THEN                                                                     00491000 T  0041
      BEGIN                                                                              00491100 T  0042
        QINPUT := FALSE ;                                                                00491200 T  0043
        GO TO INPUTFULL ;                                                                00491300 T  0044
      END ;                                                                              00491400 T  0045
      IF STATION NEQ 0 THEN                                                              00493200 T  0045
      BEGIN                                                                              00493300 T  0046
        LASTUSER := USER ;                                                               00493400 T  0046
NEXT:                                                                                    00493410 T  0047
        IF MORE THEN                                                                     00493500 T  0048
          GO TO EXIT ;                                                                   00494000 T  0048
      END ;                                                                              00494400 T  0049
      IF NOT Q AND READYQTOP GTR 0 THEN                                                  00494600 T  0049
      BEGIN                                                                              00494700 T  0051
        LASTUSER := REAL (WAITX (0, BOOLEAN (3))) ;                                      00494800 T  0051
        SECURITYCHECK ;                                                                  00494900 T  0053
        GO TO EXIT ;                                                                     00495000 T  0054
      END ;                                                                              00495100 T  0055
AGAIN:                                                                                   00495200 T  0055
      CHARGE (0) ;                                                                       00495300 T  0056
      IF 2 * BIGBIRD LSS C := STATUS (ZIPPY [*]) = 2 THEN                                00495500 T  0057
      BEGIN                                                                              00495600 T  0061
        LASTUSER := BIGBIRD + 1 ;                                                        00495650 T  0062
        FOR X := 0 STEP 2 UNTIL C DO                                                     00495700 T  0063
        BEGIN                                                                            00495800 T  0065
          STATION := 0 & ZIPPY [X] [9:9:9] ;                                             00495900 T  0065
          FOR USER := 0 STEP 1 UNTIL BIGBIRD DO                                          00496000 T  0067
            IF STATION = STATION1 THEN                                                   00496100 T  0068
              USER := MAXUSERS ;                                                         00496200 T  0069
          IF USER LEQ MAXUSERS THEN                                                      00496300 T  0073
          BEGIN                                                                          00496400 T  0074
            IF BIGBIRD LSS MAXUSER THEN                                                  00496500 T  0074
            BEGIN                                                                        00496600 T  0075
              USERCODE := ZIPPY [X + 1] ;                                                00496700 T  0075
              INITIALIZE ;                                                               00496800 T  0077
              GO TO NEXT ;                                                               00496900 T  0077
            END ;                                                                        00497000 T  0078
            NOMOREROOM ;                                                                 00497100 T  0078
          END ;                                                                          00497300 T  0079
        END ;                                                                            00497400 T  0079
      END ;                                                                              00497500 T  0081
      IF X := READTWX NEQ 0 THEN                                                         00497600 T  0081
      BEGIN                                                                              00497700 T  0083
        IF X = 2 THEN                                                                    00497800 T  0083
          GO TO AGAIN ;                                                                  00497900 T  0084
        GO TO ESCAPE ;                                                                   00498100 T  0084
      END ;                                                                              00498300 T  0085
INPUTFULL:                                                                               00506000 T  0085
      X := INPUT [0] ;                                                                   00506100 T  0086
      USER := 0 ;                                                                        00506500 T  0087
```

```
WHILE STATIONI NEQ 0 & X[9:9:9] DO                                  00507000 T  0087
  IF USER := USER + 1 GTR BIGBIRD THEN                              00507500 T  0091
    GO TO AGAIN ;                                                   00507600 T  0093
CHARGE (X) ;                                                        00508000 T  0094
IF C := CHRS NEQ 0 THEN                                             00508500 T  0095
  READ (IO [USER], 30, IMAGE [*]) ;                                 00509000 T  0097
BREAKI := 0 ;                                                       00511000 T  0103
IF LINEEDIT (INPUT [1], IMAGE, C, C,                                00512000 T  0105
             TRANSLATEI, C GTR 240, 241) THEN                       00512100 T  0108
  ERROR (AGAIN, 7, "DEL≤≠← ", CHRS := 0) ;                          00512300 T  0110
IF BOOLEAN (X.[25:1]) THEN                                          00512500 T  0116
BEGIN                                                               00512600 T  0116
  IF FIRSTCHAR (INPUT [5]) = "←" THEN                               00513000 T  0117
    C := C - 4 ;                                                    00513500 T  0119
  CHRS := C ;                                                       00513700 T  0121
  WRITE (IO [USER], 30, IMAGE [*]) ;                                00514000 T  0123
  GO TO AGAIN ;                                                     00520000 T  0128
END ;                                                               00520500 T  0128
IF BOOLEAN (INREADYQ) THEN                                          00520600 T  0128
  ERROR (AGAIN, 7, "PLEASE ", "WAIT...") ;                          00520700 T  0129
WRITELFCR ;                                                         00520800 T  0134
CHRS := C ;                                                         00520810 T  0134
CLOCK := TO ;                                                       00520850 T  0136
T1 := TIMEX ;                                                       00520900 T  0137
IF OUTPUTREADY THEN                                                 00520950 T  0139
  NEXTCLOCK := CLOCK + TO × (TN - T1 - 90) / 150 ;                  00520960 T  0139
IF LASTUSER NEQ LASTUSER := USER THEN                               00522000 T  0144
  RESTORESTATE ;                                                    00522500 T  0145
SECURITYCHECK ;                                                     00522600 T  0146
WAITFLAG := FALSE ;                                                 00522700 T  0148
EXIT:                                                               00532000 T  0150
     IF FINALANALYSIS THEN                                          00532500 T  0150
       GO TO NEXT ;                                                 00533500 T  0150
     IF OUTPUTREADY THEN                                            00533600 T  0151
       IF TN - 60 LEQ TIMEX THEN                                    00533700 T  0151
         OUTPUT ;                                                   00533800 T  0154
ESCAPE:                                                             00534500 T  0155
   END READIN ;                                                     00546500 T  0155
                                                          17 IS  160 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   00547000 T  0051
DEFINE RDISC (RDISC1, RDISC2) =                                     00547100 T  0051
    IF RDISCX (RDISC1, RDISC2) THEN GO TO NEXT# ;                   00547200 T  0051
BOOLEAN PROCEDURE RDISCX (WHERE, IMAGE) ;                           00547500 T  0051
VALUE WHERE ;                                                       00548000 T  0051
INTEGER WHERE ;                                                     00548500 T  0051
ARRAY IMAGE [0] ;                                                   00549000 T  0051
   BEGIN                                                            00549500 T  0051
LABEL EOF ;                                                         00549600 T  0051
                                                        START OF SEGMENT **********  23
STREAM PROCEDURE ZOT (D) ;                                          00549710 T  0000
BEGIN                                                               00549720 T  0000
   DI := D ;                                                        00549730 T  0000
   DS := RESET ;                                                    00549740 T  0000
END ZOT ;                                                           00549750 T  0000
     IF PREWHERE + 1 NEQ PREWHERE := ABS (WHERE) - 2 THEN           00550000 T  0000
       READ SEEK (DISC [PREWHERE]) ;                                00550500 T  0003
     READ (DISC, 10, IMAGE [*]) [EOF] ;                             00551000 T  0009
```

```
      IF COBOLFILE THEN                                                    00551010 T  0014
        ZOT (IMAGE) ;                                                      00551020 T  0015
      IF WHERE LSS 0 THEN                                                  00551030 T  0016
        SEQUENCE ;                                                         00551040 T  0017
      IF FALSE THEN                                                        00551100 T  0032
      BEGIN                                                                00551150 T  0032
EOF:                                                                       00551200 T  0033
        ERRORX (5, "AT SEQ#", OCTDEX (N)) ;                                00551250 T  0034
        RDISCX := TRUE ;                                                   00551350 T  0036
        PREWHERE := -2 ;                                                   00551400 T  0037
      END ;                                                                00551450 T  0038
    END RDISC ;                                                            00551500 T  0038
                                                                        23 IS   44 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   00552000 T  0051
DEFINE WRITEAT =                                                           00552500 T  0051
    IF WRITEATX (QUICK, -N, RECORD) THEN                                   00552800 T  0051
      GO TO NEXT# ;                                                        00552900 T  0051
BOOLEAN PROCEDURE WRITEATX (QUICK, NN, RECORD) ;                           00553000 T  0051
VALUE QUICK, NN ;                                                          00553100 T  0051
BOOLEAN QUICK ;                                                            00553200 T  0051
INTEGER NN ;                                                               00553300 T  0051
ARRAY RECORD [0] ;                                                         00553400 T  0051
    BEGIN                                                                  00554000 T  0051
LABEL NEXT ;                                                               00554100 T  0051
                                                                    START OF SEGMENT **********   24
      N := ABS (NN) ;                                                      00555500 T  0000
      IF NOT COBOLFILE THEN                                                00556000 T  0001
        WRITESEQ ;                                                         00556500 T  0001
      IF NN LSS 0 THEN                                                     00556600 T  0002
        RDISC (AT, RECORD) ;                                               00557000 T  0003
      IF COBOLFILE THEN                                                    00557500 T  0006
        RECORD [0].[1:35] := OCTDEC (N) ;                                  00558000 T  0006
      WRITEROW (RECORD, QUICK, FILEINFO) ;                                 00558500 T  0010
      IF BOOLEAN (BREAKI) THEN                                             00559600 T  0012
NEXT:                                                                      00559700 T  0013
        WRITEATX := TRUE ;                                                 00559800 T  0014
    END WRITEAT ;                                                          00560000 T  0014
                                                                        24 IS   18 LONG, NEXT SEG    2
DEFINE WRITEME (WRITEME1, WRITEME2) =                                      00560100 T  0051
    IF WRITEATX (QUICK, WRITEME1, WRITEME2) THEN                           00560200 T  0051
      GO TO NEXT# ;                                                        00560300 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   00560500 T  0051
BOOLEAN PROCEDURE TOGGLE (OLDVALUE, I) ;                                   00561000 T  0051
VALUE OLDVALUE,                                                            00561500 T  0051
  I ;                                                                      00562000 T  0051
BOOLEAN OLDVALUE ;                                                         00562500 T  0051
REAL I ;                                                                   00563000 T  0051
BEGIN                                                                      00563500 T  0051
LABEL NEXT ;                                                               00564000 T  0051
                                                                    START OF SEGMENT **********   25
    IF I = 3 THEN                                                          00564100 T  0000
    BEGIN                                                                  00564110 T  0000
      IF REAL (OLDVALUE) = "ALGOL  " THEN                                  00564120 T  0001
        TOGGLE := BOOLEAN (ALGOL)                                          00564130 T  0002
      ELSE IF REAL (OLDVALUE) = "XALGOL " THEN                             00564140 T  0002
        TOGGLE := BOOLEAN (XALGOL)                                         00564150 T  0005
      ELSE IF REAL (OLDVALUE) = "DATA   " THEN                             00564160 T  0006
```

```
      TOGGLE := BOOLEAN (DATA)                                                    00564170 T  0009
    ELSE IF REAL (OLDVALUE) = "FORTRAN" THEN                                      00564180 T  0010
      TOGGLE := BOOLEAN (FORTRAN)                                                 00564190 T  0013
    ELSE IF REAL (OLDVALUE) = "COBOL   " THEN                                     00564200 T  0014
      TOGGLE := BOOLEAN (COBOL)                                                   00564210 T  0017
    ELSE IF REAL (OLDVALUE) = "BASIC   " THEN                                     00564220 T  0018
      TOGGLE := BOOLEAN (BASIC) ;                                                 00564230 T  0021
    GO TO NEXT ;                                                                  00564240 T  0023
  END ;                                                                           00564250 T  0025
  IF (IF I = 1 THEN EMPTY1 ELSE EMPTY2) THEN                                      00564500 T  0025
    ERROR (NEXT, 7, PARAMETER0, ONOFF (TOGGLE := OLDVALUE)) ;                     00565000 T  0028
  I := IF I = 1 THEN PARAMETER1 ELSE PARAMETER2 ;                                 00565100 T  0035
  IF NOT (TOGGLE := I = "ON      ") THEN                                          00565500 T  0037
    IF I NEQ "OFF     " THEN                                                      00566000 T  0039
      ERRORX (0, "MISSING", " ON/OFF") ;                                          00566500 T  0040
NEXT:                                                                             00567000 T  0042
END TOGGLE ;                                                                      00567500 T  0043
                                                                    25 IS   50 LONG, NEXT SEG    2
DEFINE FILETYPE (FILETYPE1) = REAL (TOGGLE (BOOLEAN (FILETYPE1), 3))# ;           00567600 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00568000 T  0051
BOOLEAN PROCEDURE VERIFAX (XEROX, DD) ;                                           00680000 T  0051
VALUE XEROX, DD ;                                                                 00680500 T  0051
INTEGER XEROX,                                                                    00681000 T  0051
  DD ;                                                                            00681100 T  0051
  BEGIN                                                                           00681500 T  0051
DEFINE                                                                            00681600 T  0051
                                                                  START OF SEGMENT **********   26
  PRINTING = XEROX = 2#,                                                          00681700 T  0000
  PUNCHING = XEROX = 4#,                                                          00681800 T  0000
  ZIPPING = XEROX = 8# ;                                                          00681900 T  0000
FILE COPY DISK SERIAL [20:DD] (2, 10, 150, SAVE SAVEFACTOR) ;                     00682000 T  0000
BOOLEAN B ;                                                                       00683500 T  0006
REAL L ;                                                                          00684000 T  0006
LABEL NEXT ;                                                                      00684500 T  0006
    XLOCKED := TRUE ;                                                             00687000 T  0006
    IF BOOLEAN (XEROX) THEN                                                       00687500 T  0008
    BEGIN                                                                         00688000 T  0008
      FILL COPY WITH PREFIX, SUFFIX, *, *, *, 12 ;                                00689000 T  0008
      L := FIRST ;                                                                00691000 T  0012
      WHILE AT := L.T NEQ 1 DO                                                    00692000 T  0014
      BEGIN                                                                       00692500 T  0016
        N := (L := LL [AT]).S ;                                                   00693000 T  0016
        RDISC (=AT, ZIPPY) ;                                                      00693500 T  0021
        WRITE (COPY, 10, ZIPPY [*]) ;                                             00695500 T  0023
        INTERRUPT (1) ;                                                           00696000 T  0027
      END ;                                                                       00696500 T  0033
      READ (DISC [0]) ;                                                           00697500 T  0035
      DETACH ;                                                                    00698000 T  0040
      CLOSE (DISC, PURGE) ;                                                       00698500 T  0041
      LOCK (COPY) ;                                                               00699500 T  0043
      REATTACH ;                                                                  00700000 T  0044
      INORDER := TRUE ;                                                           00702500 T  0045
      FILEACCESS := 0 ;                                                           00705500 T  0047
      SAVESTATE ;                                                                 00706000 T  0048
    END XEROX                                                                     00706500 T  0049
    ELSE                                                                          00707000 T  0049
    BEGIN                                                                         00708000 T  0049
```

```
FILL COPY WITH PARAMETER1, PARAMETER2, *, *, *,                    00709000 T  0049
  IF PRINTING THEN 15 ELSE IF PUNCHING THEN 22 ELSE 12 ;           00709500 T  0052
IF PRINTING THEN                                                   00713000 T  0057
BEGIN                                                              00713500 T  0057
  WRITE (ZIPPY [*], DATE, PREFIX,[6:6], PREFIX,                    00714000 T  0058
    SUFFIX,[6:6], SUFFIX, (L := TIME (1)) DIV 216000,              00714500 T  0067
    L DIV 3600 MOD 60, TIME (6), MMDDYY,                           00715000 T  0076
    USERCODE,[6:6], USERCODE) ;                                    00715500 T  0086
  DETACH ;                                                         00716000 T  0092
  WRITE (COPY [DBL], 17, ZIPPY [*]) ;                              00716500 T  0093
  REATTACH ;                                                       00719000 T  0097
END ;                                                              00720000 T  0098
L := N ;                                                           00720500 T  0098
DD := M := 0 ;                                                     00721000 T  0099
B := PRINTING AND PARAMETER2 = "DOUBLE " ;                         00721100 T  0100
WHILE N := LL [DD := LL [DD],T],S LEQ PARAMETER4 DO                00721500 T  0102
  IF PARAMETER3 LEQ N THEN                                         00722000 T  0112
  BEGIN                                                            00723000 T  0113
    RDISC (DD & (REAL (NOT ZIPPING))[1:47:1], ZIPPY) ;             00723500 T  0113
    IF PRINTING THEN                                               00724000 T  0117
      ZIPPY [14] := OCTDEX (M := M + 1) & "#"[1:43:5] ;            00724500 T  0118
    IF B THEN                                                      00726000 T  0122
      WRITE (COPY [DBL], 17, ZIPPY [*])                            00726500 T  0123
    ELSE WRITE (COPY, 17, ZIPPY [*]) ;                             00727000 T  0126
    INTERRUPT (1) ;                                                00727500 T  0134
  END                                                              00728000 T  0140
    ELSE M := M + 1 ;                                              00728500 T  0140
  IF ZIPPING THEN                                                  00729000 T  0142
    ZIP WITH COPY ;                                                00729500 T  0143
  LOCK (COPY) ;                                                    00730000 T  0144
  N := L ;                                                         00730500 T  0146
END THERMOFAX ;                                                    00731000 T  0147
IF FALSE THEN                                                      00731100 T  0147
NEXT:                                                              00731200 T  0147
  VERIFAX := TRUE ;                                                00731300 T  0148
XLOCKED := FALSE ;                                                 00731400 T  0148
END VERIFAX ;                                                      00731500 T  0150
                                                          26 IS  155 LONG, NEXT SEG    2
DEFINE THERMOFAX (THERMOFAX1, THERMOFAX2) =                        00731600 T  0051
  BEGIN                                                            00731650 T  0051
    WAIT (KOUNT (PARAMETER3, PARAMETER4, CLOCK), XLOCKED) ;        00731700 T  0051
    IF VERIFAX (THERMOFAX1, THERMOFAX2) THEN                       00731750 T  0051
      GO TO NEXT ;                                                 00731800 T  0051
  END#,                                                            00731850 T  0051
CREATEFILE (CREATEFILE1) =                                         00731900 T  0051
  BEGIN                                                            00731950 T  0051
    LIBRARY.AREAS := 20 ;                                          00732000 T  0051
    LIBRARY.AREASIZE := CREATEFILE1 ;                              00732010 T  0051
    LIBRARY.SAVE := SAVEFACTOR ;                                   00732020 T  0051
    WRITE (LIBRARY, 10, RECORD [*]) ;                              00732030 T  0051
    LOCK (LIBRARY) ;                                               00732040 T  0051
    LIBRARY.AREASIZE := 0 ;                                        00732050 T  0051
    LIBRARY.AREAS := 0 ;                                           00732060 T  0051
  END#,                                                            00732100 T  0051
CLOSEMYFILE =                                                      00732150 T  0051
  BEGIN                                                            00732200 T  0051
    IF NOT INORDER THEN                                            00732250 T  0051
```

```
      BEGIN                                                                    00732300 T  0051
        WAIT (KOUNT (1, FINITY, CLOCK), XLOCKED) ;                             00732350 T  0051
        IF VERIFAX (17, (D + 14) DIV 15 × 15) THEN                             00732400 T  0051
          GO TO NEXT ;                                                         00732450 T  0051
      END ELSE                                                                 00732500 T  0051
      BEGIN                                                                    00732550 T  0051
        FILEACCESS := 0 ;                                                      00732600 T  0051
        CLOSE (DISC) ;                                                         00732650 T  0051
      END ;                                                                    00732700 T  0051
    END# ;                                                                     00732750 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   00733000 T  0051
DEFINE WDISC = IF WDISCX (IMAGE) THEN GO TO NEXT# ;                            00733500 T  0051
BOOLEAN PROCEDURE WDISCX (IMAGE) ;                                             00734000 T  0051
ARRAY IMAGE [0] ;                                                              00734500 T  0051
   BEGIN                                                                       00735000 T  0051
REAL L ;                                                                       00735100 T  0051
                                                                  START OF SEGMENT **********   27
LABEL EOT,                                                                     00735500 T  0000
  NEXT ;                                                                       00735600 T  0000
      WHILE N GTR (L := LL [AT]).S DO                                          00743500 T  0000
        AT := L.T ;                                                            00744000 T  0005
      WHILE N LSS (L := LL [AT]).S DO                                          00744500 T  0007
        AT := L.F ;                                                            00745000 T  0012
      IF N NEQ L.S THEN                                                        00745500 T  0014
      BEGIN                                                                    00746000 T  0015
        IF D GEQ MAXFILELENGTH THEN                                            00746500 T  0015
          ERROR (NEXT, 0, "FILE TO", " LONG. ") ;                              00747000 T  0016
        IF PREWHERE NEQ PREWHERE := D - 2 THEN                                 00747100 T  0022
          READ SEEK (DISC [PREWHERE + 1]) ;                                    00747200 T  0023
        L := LL [D := D + 1] := (L.T) & N [SF] & AT [FF] ;                      00747500 T  0030
        MODIFY (D) ;                                                           00748000 T  0038
        LL [AT] . T := D ;                                                     00748500 T  0041
        MODIFY (AT) ;                                                          00748600 T  0046
        AT := L.T ;                                                            00749000 T  0049
        IF AT NEQ 1 THEN                                                       00749500 T  0050
          INORDER := FALSE ;                                                   00750000 T  0051
        LL [AT] . F := D ;                                                     00750500 T  0053
        MODIFY (AT) ;                                                          00750600 T  0058
        AT := D ;                                                              00751000 T  0061
      END ;                                                                    00751500 T  0062
      SEQUENCE ;                                                               00752000 T  0062
      IF PREWHERE + 1 NEQ PREWHERE := AT - 2 THEN                              00752500 T  0076
        WRITE (DISC [PREWHERE], 10, IMAGE [*])                                 00753000 T  0078
      ELSE WRITE (DISC, 10, IMAGE [*]) [EOT] ;                                 00753500 T  0082
      N := N + INC ;                                                           00753510 T  0091
      IF FALSE THEN                                                            00753600 T  0092
      BEGIN                                                                    00753610 T  0092
EOT:                                                                           00753620 T  0093
        LL [L.F] . T := AT := L.T ;                                            00753630 T  0094
        LL [AT] . F := L.F ;                                                   00753640 T  0100
        D := D - 1 ;                                                           00753650 T  0106
        INORDER := FALSE ;                                                     00753660 T  0107
        SHOW ("FILE   ", "FULL.  ") ;                                          00753670 T  0109
        ERRORX (0, "PLEASE ", "REOPEN.") ;                                     00753690 T  0110
NEXT:                                                                          00753700 T  0112
        WDISCX := TRUE ;                                                       00753800 T  0112
      END ;                                                                    00753900 T  0112
```

```
   END WDISC ;                                                          00754000 T  0112
                                                              27 IS  122 LONG, NEXT SEG    2
%  * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     00754500 T  0051
INTEGER PROCEDURE GETPARAMETERS (N) ; VALUE N ; INTEGER N ;             00754600 T  0051
     BEGIN                                                              00754650 T  0051
INTEGER STREAM PROCEDURE STAR (S, D, E) ; VALUE E ;                     00754700 T  0051
                                                          START OF SEGMENT **********   28
   BEGIN                                                                00754750 T  0000
LOCAL N,                                                                00754800 T  0000
  PLUS,                                                                 00754850 T  0000
  MINUS,                                                                00754900 T  0000
  CROSSHATCH,                                                           00754950 T  0000
  K ;                                                                   00755000 T  0000
LABEL DEBLANK,                                                          00755050 T  0000
  NALPHA,                                                               00755100 T  0000
  BLANK,                                                                00755150 T  0000
  NUMALPHA,                                                             00755200 T  0000
  GETREPEAT ;                                                           00755250 T  0000
     SI := S ;                                                          00755300 T  0000
     SI := SI - 1 ;                                                     00755350 T  0000
     DI := D ;                                                          00755400 T  0000
     5 (DS := 8 LIT "+#000000") ;                                       00755450 T  0000
     DI := D ;                                                          00755500 T  0002
     E (IF SC = "(" THEN JUMP OUT ;                                     00755550 T  0002
        IF SC = "@" THEN JUMP OUT TO GETREPEAT ;                        00755600 T  0004
        IF SC = ";" THEN JUMP OUT TO GETREPEAT ;                        00755650 T  0005
        SI := SI + 1) ;                                                 00755700 T  0006
     5 (TALLY := 0 ;                                                    00755750 T  0008
     K := TALLY ;                                                       00755800 T  0008
     PLUS := TALLY ;                                                    00755850 T  0008
     MINUS := TALLY ;                                                   00755900 T  0009
     CROSSHATCH := TALLY ;                                              00755950 T  0009
     TALLY := 1 ;                                                       00756000 T  0009
DEBLANK:                                                                00756050 T  0009
     SI := SI + 1 ;                                                     00756100 T  0009
     IF SC = " " THEN                                                   00756150 T  0010
       GO TO DEBLANK ;                                                  00756200 T  0010
     IF SC = ALPHA THEN                                                 00756250 T  0011
       TALLY := 0                                                       00756300 T  0011
     ELSE                                                               00756350 T  0011
     BEGIN                                                              00756400 T  0012
       IF SC = ";" THEN                                                 00756450 T  0012
         JUMP OUT TO GETREPEAT ;                                        00756500 T  0012
       IF SC = """ THEN                                                 00756550 T  0013
         JUMP OUT TO GETREPEAT ;                                        00756600 T  0013
       IF SC = "(" THEN                                                 00756650 T  0014
         JUMP OUT TO GETREPEAT ;                                        00756700 T  0014
       IF SC = "[" THEN                                                 00756750 T  0015
         JUMP OUT TO GETREPEAT ;                                        00756800 T  0015
       IF SC = "," THEN                                                 00756850 T  0016
         JUMP OUT TO GETREPEAT ;                                        00756900 T  0016
       IF SC = "@" THEN                                                 00756950 T  0017
         JUMP OUT TO GETREPEAT ;                                        00757000 T  0017
       IF SC = "/" THEN                                                 00757050 T  0018
         K := TALLY                                                     00757100 T  0018
       ELSE IF SC = "+" THEN                                            00757150 T  0018
         PLUS := TALLY                                                  00757200 T  0019
```

```
            ELSE IF SC = "#" THEN                                    00757250 T  0019
              CROSSHATCH := TALLY                                    00757300 T  0020
            ELSE IF SC = "=" THEN                                    00757350 T  0020
              MINUS := TALLY ;                                       00757400 T  0021
            GO TO DEBLANK ;                                          00757450 T  0021
          END ;                                                      00757500 T  0022
          IF SC GEQ "0" THEN                                         00757550 T  0022
          BEGIN                                                      00757600 T  0022
            K (JUMP OUT TO NALPHA) ;                                 00757650 T  0022
            K := SI ;                                                00757700 T  0024
            8 (IF SC LSS "0" THEN                                    00757750 T  0025
              JUMP OUT ;                                             00757800 T  0025
            TALLY := TALLY + 1 ;                                     00757850 T  0026
            SI := SI + 1) ;                                          00757900 T  0026
            N := TALLY ;                                             00757950 T  0027
            IF TOGGLE THEN                                           00758000 T  0027
            BEGIN                                                    00758050 T  0027
              IF SC = ALPHA THEN                                     00758100 T  0027
                GO TO NUMALPHA ;                                     00758150 T  0028
BLANK:                                                               00758200 T  0028
              IF SC = " " THEN                                       00758250 T  0028
              BEGIN                                                  00758300 T  0029
                SI := SI + 1 ;                                       00758350 T  0029
                GO TO BLANK ;                                        00758400 T  0030
              END ;                                                  00758450 T  0030
              IF SC = "/" THEN                                       00758500 T  0030
              BEGIN                                                  00758550 T  0031
NUMALPHA:                                                            00758600 T  0031
                SI := K ;                                            00758650 T  0031
                GO TO NALPHA ;                                       00758700 T  0032
              END ;                                                  00758750 T  0032
            END ;                                                    00758800 T  0032
            SI := K ;                                                00758850 T  0032
            DS := N OCT ;                                            00758900 T  0032
          END                                                        00758950 T  0033
          ELSE                                                       00759000 T  0033
          BEGIN                                                      00759050 T  0033
NALPHA:                                                              00759100 T  0033
            DS := 1 LIT "+" ;                                        00759150 T  0033
            7 (IF SC = ALPHA THEN                                    00759200 T  0034
              DS := 1 CHR                                            00759250 T  0035
            ELSE DS := 1 LIT " ") ;                                  00759300 T  0035
          END ;                                                      00759350 T  0036
          DI := DI - 8 ;                                             00759400 T  0036
          SKIP 2 DB ;                                                00759450 T  0036
          DS := PLUS SET ;                                           00759500 T  0037
          DI := DI - 1 ;                                             00759550 T  0037
          SKIP 3 DB ;                                                00759600 T  0037
          DS := MINUS SET ;                                          00759650 T  0038
          DI := DI - 1 ;                                             00759700 T  0038
          SKIP 4 DB ;                                                00759750 T  0038
          DS := CROSSHATCH SET ;                                     00759800 T  0039
          DI := DI + 7 ;                                             00759850 T  0039
          SI := SI - 1) ;                                            00759900 T  0039
GETREPEAT:                                                           00759910 T  0040
          E (IF SC = ")" THEN JUMP OUT ;                             00759950 T  0040
             IF SC = ";" THEN JUMP OUT ;                             00760000 T  0043
```

```
            IF SC = "@" THEN JUMP OUT ;                                                    00760050 T  0044
            SI := SI + 1) ;                                                                00760100 T  0045
         E (DI := LOC STAR ;                                                               00760200 T  0046
            DS := 8 LIT "00000001" ;                                                       00760250 T  0047
            DI := DI - 8 ;                                                                 00760300 T  0048
            10 (IF SC = ";" THEN JUMP OUT ;                                                00760350 T  0048
                IF SC GEQ "0" THEN                                                         00760400 T  0050
                BEGIN                                                                      00760450 T  0050
                  TALLY := 1 ;                                                             00760500 T  0050
                  3 (SI := SI + 1 ;                                                        00760550 T  0051
                     IF SC LSS "0" THEN JUMP OUT ;                                         00760600 T  0051
                     TALLY := TALLY + 1) ;                                                 00760650 T  0052
                  K := TALLY ;                                                             00760700 T  0054
                  SI := SI - K ;                                                           00760750 T  0054
                  DS := K OCT ;                                                            00760800 T  0054
                  JUMP OUT ;                                                               00760850 T  0055
                END ;                                                                      00760900 T  0055
                SI := SI + 1) ;                                                            00760950 T  0055
            JUMP OUT) ;                                                                    00761000 T  0057
END STAR ;                                                                                 00761050 T  0058
DEFINE XSUB = (XDEX + 1) x 13# ;                                                           00761100 T  0059
   IF N = 0 THEN                                                                           00761150 T  0059
     GETPARAMETERS := STAR (IMAGE, PARAMETER0, 0)                                          00761200 T  0059
   ELSE                                                                                    00761250 T  0063
     GETPARAMETERS := STAR (IMAGE, XPARAMETERS [0], 63) ;                                  00761300 T  0064
END GETPARAMETERS ;                                                                        00761350 T  0071
                                                                                  28 IS   74 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *        00761400 T  0051
INTEGER PROCEDURE VERB ;                                                                   00763900 T  0051
BEGIN                                                                                      00764000 T  0051
BOOLEAN PROCEDURE NUMBER (N, C) ;                                                          00764100 T  0051
                                                                                  START OF SEGMENT **********  29
INTEGER N ;                                                                                00764300 T  0000
REAL C ;                                                                                   00764400 T  0000
 BEGIN                                                                                     00764500 T  0000
INTEGER XDEXX ;                                                                            00764510 T  0000
                                                                                  START OF SEGMENT **********  30
LABEL ZERO ;                                                                               00764520 T  0000
  IF XDEX GEQ 0 THEN                                                                       00764530 T  0000
  BEGIN                                                                                    00764540 T  0000
    XDEXX := XDEX + 1 ;                                                                    00764550 T  0001
    WHILE BOOLEAN (C.[4:1]) AND XDEXX := XDEXX - 1 GEQ 0 DO                                00764560 T  0002
      C := XARRAY [USER, XDEXX*13 + ABS (C&0[1:44:4]-1) MOD 5] ;                           00764570 T  0006
  END ;                                                                                    00764580 T  0012
  C.[4:1] := 0 ;                                                                           00764600 T  0012
  IF NUMBER := (NOT BOOLEAN (C.[1:1])) & (C = -"#000000")[46:47:1] THEN                    00764610 T  0014
  BEGIN                                                                                    00764620 T  0017
   IF C . [2:2] NEQ 0 AND FILEOPEN THEN                                                    00764700 T  0018
   BEGIN                                                                                   00764800 T  0020
     C . [1:3] := C . [3:3] ;                                                              00764900 T  0020
     IF C = 0 THEN                                                                         00765100 T  0023
     BEGIN                                                                                 00765200 T  0024
       C := N ;                                                                            00765300 T  0024
       GO TO ZERO ;                                                                        00765400 T  0025
     END ;                                                                                 00765500 T  0028
     IF NOT (ITSOLD (N) OR BOOLEAN (C . [1:1])) THEN                                       00765600 T  0028
       C := C - 1 ;                                                                        00765700 T  0030
```

```
      FOR N := 1 - C STEP 1 UNTIL 0 DO                                                00765800 T  0032
        IF AT := LL [AT] . T = 1 THEN                                                 00765900 T  0037
        BEGIN                                                                         00766000 T  0042
          N := 0 ;                                                                    00766100 T  0042
          AT := LAST . F ;                                                            00766200 T  0043
        END ;                                                                         00766300 T  0046
      FOR N := C + 1 STEP 1 UNTIL 0 DO                                                00766400 T  0046
        IF AT := LL [AT] . F = 0 THEN                                                 00766500 T  0051
        BEGIN                                                                         00766600 T  0056
          N := 0 ;                                                                    00766700 T  0056
          AT := FIRST . T ;                                                           00766800 T  0057
        END ;                                                                         00766900 T  0060
      C := LL [AT] . S ;                                                              00767000 T  0060
    END ELSE C.[2:2] := 0 ;                                                           00767100 T  0065
ZERO:                                                                                 00767200 T  0067
    C := MIN (FINITY, MAX (1, N := C)) ;                                              00767300 T  0068
  END ELSE                                                                            00767310 T  0074
    C.[1:3] := 0 ;                                                                    00767320 T  0074
 END NUMBER ;                                                                         00767400 T  0078
                                                                          30 IS   81 LONG, NEXT SEG   29
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     00767500 T  0000
INTEGER STREAM PROCEDURE INLINEEDIT (S, D, T, C, N, BIDR, INITIAL) ;                  00767600 T  0000
VALUE INITIAL,                                                                        00767700 T  0000
  C,                                                                                  00767800 T  0000
  N,                                                                                  00767900 T  0000
  BIDR ;                                                                              00768000 T  0000
   BEGIN                                                                              00768100 T  0000
LABEL SEARCH,                                                                         00768200 T  0000
  INSERT,                                                                             00768300 T  0000
  DELETE,                                                                             00768400 T  0000
  REPLACE,                                                                            00768500 T  0000
  WRAPUP,                                                                             00768600 T  0000
  LOOP,                                                                               00768700 T  0000
  ERROR1,                                                                             00768710 T  0000
  HERE,                                                                               00768720 T  0000
  THERE,                                                                              00768730 T  0000
  IDR,                                                                                00768800 T  0000
  XIT ;                                                                               00768900 T  0000
     BIDR (SI := S ; SI := SI + 6 ; S := SI ;                                         00769000 T  0000
           SI := D ; SI := SI + 6 ; D := SI ;                                         00769100 T  0001
           DI := T ; DS := 6 LIT "0" ; T := DI ;                                      00769200 T  0002
           SI := C ; SI := SI - 48 ; C := SI) ;                                       00769300 T  0004
     DI := LOC BIDR ;                                                                 00769400 T  0005
     DS := 4 LIT " IDR" ;                                                             00769500 T  0005
     DI := T ;                                                                        00769600 T  0006
     SI := T ;                                                                        00769700 T  0006
     DS := 8 LIT " " ;                                                                00769800 T  0006
     DS := 9 WDS ;                                                                    00769900 T  0007
     2 (N (CI := CI + INITIAL ;                                                       00770400 T  0008
     GO TO SEARCH ;                                                                   00770500 T  0009
     GO TO IDR ;                                                                      00770600 T  0010
     GO TO IDR ;                                                                      00770700 T  0010
     GO TO IDR ;                                                                      00770800 T  0010
     GO TO WRAPUP ;                                                                   00770900 T  0010
SEARCH:                                                                               00771000 T  0011
     SI := LOC C ;                                                                    00771100 T  0011
     SI := SI + 6 ;                                                                   00771200 T  0011
```

```
        DI := LOC N ;                                    00771300 T  0011
        IF 2 SC = DC THEN                                00771400 T  0011
          GO TO ERROR1 ;                                 00771500 T  0012
        SI := C ;                                        00771900 T  0012
        SI := SI - 8 ;                                   00772000 T  0012
        C := SI ;                                        00772100 T  0013
        SI := D ;                                        00772200 T  0013
        DI := T ;                                        00772300 T  0013
        DS := 1 CHR ;                                    00772400 T  0013
        D := SI ;                                        00772500 T  0014
        T := DI ;                                        00772600 T  0014
        SI := S ;                                        00772700 T  0014
        DI := LOC BIDR ;                                 00772800 T  0014
        4 (IF SC = DC THEN                               00772900 T  0015
          JUMP OUT ;                                     00773000 T  0015
        SI := SI - 1 ;                                   00773100 T  0016
        TALLY := TALLY + 1) ;                            00773200 T  0016
        IF TOGGLE THEN                                   00773300 T  0017
        ELSE                                             00773400 T  0017
        BEGIN                                            00773500 T  0017
ERROR1:                                                  00773510 T  0017
          TALLY := 1 ;                                   00773600 T  0017
          JUMP OUT 2 TO HERE ;                           00773700 T  0018
        END ;                                            00773800 T  0019
        INITIAL := TALLY ;                               00773900 T  0019
        TALLY := 0 ;                                     00774000 T  0019
        S := SI ;                                        00774100 T  0019
        GO TO LOOP ;                                     00774200 T  0019
IDR:                                                     00774300 T  0020
        SI := LOC C ;                                    00774400 T  0020
        SI := SI + 6 ;                                   00774500 T  0020
        DI := LOC N ;                                    00774600 T  0020
        IF 2 SC = DC THEN                                00774700 T  0020
        BEGIN                                            00774800 T  0021
          SI := D ;                                      00774900 T  0021
          DI := T ;                                      00775000 T  0022
          TALLY := 4 ;                                   00775100 T  0022
          INITIAL := TALLY ;                             00775200 T  0022
WRAPUP:                                                  00775300 T  0022
          DS := 1 CHR ;                                  00775400 T  0022
          GO TO LOOP ;                                   00775500 T  0023
        END ;                                            00775600 T  0023
        SI := C ;                                        00775700 T  0023
        SI := SI - 8 ;                                   00775800 T  0023
        C := SI ;                                        00775900 T  0024
        SI := S ;                                        00776000 T  0024
        CI := CI + INITIAL ;                             00776100 T  0024
        GO TO WRAPUP ;                                   00776200 T  0025
        GO TO INSERT ;                                   00776300 T  0025
        GO TO DELETE ;                                   00776400 T  0025
        GO TO REPLACE ;                                  00776500 T  0025
INSERT:                                                  00776600 T  0026
        DI := T ;                                        00776700 T  0026
        DS := 1 CHR ;                                    00776800 T  0026
        S := SI ;                                        00776900 T  0026
        T := DI ;                                        00777000 T  0026
        DI := INLINEEDIT ;                               00777100 T  0027
```

```
        DI := DI + 8 ;                                                  00777200 T  0027
        INLINEEDIT := DI ;                                              00777300 T  0027
        GO TO LOOP ;                                                    00777400 T  0027
DELETE:                                                                 00777500 T  0028
        DI := D ;                                                       00777600 T  0028
        DI := DI + 1 ;                                                  00777700 T  0028
        D := DI ;                                                       00777800 T  0028
        SI := SI + 1 ;                                                  00777900 T  0028
        S := SI ;                                                       00778000 T  0029
        GO TO LOOP ;                                                    00778100 T  0029
REPLACE:                                                                00778200 T  0029
        DI := T ;                                                       00778300 T  0029
        DS := 1 CHR ;                                                   00778400 T  0030
        S := SI ;                                                       00778500 T  0030
        T := DI ;                                                       00778600 T  0030
        SI := D ;                                                       00778700 T  0031
        SI := SI + 1 ;                                                  00778800 T  0031
        D := SI ;                                                       00778900 T  0031
LOOP:                                                                   00779000 T  0031
       )) ;                                                             00779100 T  0031
        GO TO THERE ;                                                   00779110 T  0033
HERE:                                                                   00779120 T  0033
        GO TO XIT ;                                                     00779130 T  0033
THERE:                                                                  00779140 T  0034
        TALLY := 0 ;                                                    00779200 T  0034
        S := SI ;                                                       00779300 T  0035
        SI := LOC INLINEEDIT ;                                          00779400 T  0035
        DI := LOC BIDR ;                                                00779500 T  0035
        SI := SI + 6 ;                                                  00779600 T  0036
        DI := DI + 7 ;                                                  00779700 T  0036
        DS := 1 CHR ;                                                   00779800 T  0036
        SI := S ;                                                       00779900 T  0036
        BIDR (2 (32 (IF SC NEQ " " THEN                                 00780000 T  0037
                     BEGIN                                              00780100 T  0039
                       TALLY := 2 ;                                     00780200 T  0039
                       JUMP OUT 3 TO XIT ;                              00780300 T  0039
                     END ;                                              00780400 T  0040
                     SI := SI + 1))) ;                                  00780500 T  0040
        INLINEEDIT (IF SC NEQ " " THEN                                  00780600 T  0041
                    BEGIN                                               00780700 T  0043
                      TALLY := 2 ;                                      00780800 T  0043
                      JUMP OUT 1 TO XIT ;                               00780900 T  0044
                    END ;                                               00781000 T  0044
                    SI := SI + 1) ;                                     00781100 T  0044
XIT:                                                                    00781200 T  0045
     INLINEEDIT := TALLY ;                                              00781300 T  0045
   END INLINE ;                                                         00781400 T  0045
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   00781500 T  0046
LABEL NEXT,                                                             00786000 T  0046
  VERBEXIT ;                                                            00786100 T  0046
DEFINE QUICK = FALSE# ;                                                 00786110 T  0046
NEXT:                                                                   00806500 T  0046
       READIN ;                                                         00807000 T  0047
       IF BOOLEAN (ABNORMALEND) OR USER = MAXUSERS THEN                 00807500 T  0047
       BEGIN                                                            00808000 T  0049
         VERB := RSWDM + REAL (USER = MAXUSERS) ;                       00809000 T  0050
         GO TO VERBEXIT ;                                               00809500 T  0052
```

```
END ;                                                                    00810000 T  0052
IF INLINETOG THEN                                                        00811000 T  0052
BEGIN                                                                    00811500 T  0053
  INLINETOG := FALSE ;                                                   00811600 T  0053
  IF M := INLINEEDIT (IMAGE, RECORD, ZIPPY, CHRS,                        00813000 T  0055
    HALFLENGTH, FILEINFO = COBOL, M) = 0 THEN                            00813500 T  0059
  BEGIN                                                                  00814000 T  0065
    IF INLINEECHO EQV TEMPTOG THEN                                       00814100 T  0065
      WRITEME (N, ZIPPY) ;                                               00814200 T  0067
    IF WDISCX (ZIPPY) THEN ;                                             00814500 T  0070
    NOSTAR := FALSE ;                                                    00815000 T  0072
    GO TO NEXT ;                                                         00815200 T  0074
  END ;                                                                  00815500 T  0074
  IF M = 2 THEN                                                          00816000 T  0074
    ERROR (NEXT, 0, PARAMETER0, " OVRFLW") ;                             00816100 T  0075
  ERROR (NEXT, 0, "NEEDS I", ",R OR D") ;                                00816500 T  0079
END ;                                                                    00817500 T  0083
IF NOSTAR THEN                                                           00818000 T  0083
BEGIN                                                                    00818500 T  0083
  IF WDISCX (IMAGE) THEN ;                                               00822500 T  0084
  GO TO NEXT ;                                                           00826000 T  0086
END ;                                                                    00826500 T  0086
WRITE (IO [USER], 30, IMAGE [*]) ;                                       00827000 T  0086
I := GETPARAMETERS (0) ;                                                 00828000 T  0091
TEMPTOG := PARAMETER0.[2:2] = 0 ;                                        00828100 T  0092
IF NUMBER (N, PARAMETER0) THEN                                           00837000 T  0095
BEGIN                                                                    00837100 T  0096
  IF FILECLOSED THEN                                                     00837500 T  0097
    ERROR (NEXT, 5, " OPEN: ", OCTDEX (PARAMETER0)) ;                    00838000 T  0097
  IF NOT MOREINPUT AND ITSOLD (N := PARAMETER0) THEN                     00838500 T  0102
    WRITEAT ;                                                            00838600 T  0105
  GO TO NEXT ;                                                           00839000 T  0108
END ;                                                                    00839500 T  0108
M := RESETN := N ;                                                       00839700 T  0108
FOR I := 0 STEP 1 UNTIL RSWDM DO                                         00840000 T  0110
  IF PARAMETER0 = RSWD [I] THEN                                          00840500 T  0111
  BEGIN                                                                  00840600 T  0112
    RELATIVENUMBER := PARAMETER1 ;                                       00840605 T  0112
    NUM1 := NUMBER (M, PARAMETER1) ;                                     00840610 T  0113
    NUM2 := NUMBER (M, PARAMETER2) ;                                     00840620 T  0116
    NUM3 := NUMBER (M, PARAMETER3) ;                                     00840630 T  0118
    NUM4 := NUMBER (M, PARAMETER4) ;                                     00840640 T  0121
    VERB := I ;                                                          00840700 T  0123
    GO TO VERBEXIT ;                                                     00841000 T  0124
  END ;                                                                  00841010 T  0125
IF I := XFILE (PARAMETER0, MACROLIBRARY, -1) LSS 2                       00841100 T  0127
  AND MACROLIBRARY NEQ "MACRO  " THEN                                    00841200 T  0129
  I := XFILE (PARAMETER0, "MACRO  ", -1) ;                               00841220 T  0130
IF I LSS 2 OR INPUT [3] NEQ 10 THEN                                      00841300 T  0133
BEGIN                                                                    00841320 T  0135
  SHOW (PARAMETER0, " INVALI") ;                                         00841360 T  0135
  ERROR (NEXT, 0, "D:*    ", RWTEACH) ;                                00841400 T  0137
END ;                                                                    00841500 T  0142
VERBEXIT:                                                                00844500 T  0142
    END ;                                                                00845000 T  0142
                                                    29 IS  145 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00846000 T  0051
```

```
DEFINE QUICKLIST = LISTIT (1)#,                                                 00850000 T  0051
 SCAN = LISTIT (2)#,                                                            00850100 T  0051
 CHANGE = LISTIT (4)#,                                                          00850200 T  0051
 EDIT = LISTIT (8)# ;                                                           00850300 T  0051
PROCEDURE LISTIT (LISTTYPE) ; VALUE LISTTYPE ; INTEGER LISTTYPE ;               00850400 T  0051
     BEGIN                                                                      00850500 T  0051
LABEL NEXT ;                                                                    00850600 T  0051
                                                                    START OF SEGMENT **********   31
DEFINE QUICK = BOOLEAN (LISTTYPE) AND TRUE#,                                    00850700 T  0000
  SCANTOG = LISTTYPE = 2#,                                                      00850800 T  0000
  CHANGETOG = LISTTYPE = 4#,                                                    00850900 T  0000
  EDITTOG = LISTTYPE = 8#,                                                      00851000 T  0000
  POSTING = LISTTYPE GEQ 16# ;                                                  00851100 T  0000
BOOLEAN PROCEDURE STRINGFOUND ;                                                 00851110 T  0000
BEGIN                                                                           00851120 T  0000
BOOLEAN STREAM PROCEDURE PRESENT (S, R, I, SR, T, ID, K) ;                      00851200 T  0000
                                                                    START OF SEGMENT **********   32
VALUE I,                                                                        00851300 T  0000
  SR,                                                                           00851400 T  0000
  ID,                                                                           00851500 T  0000
  K,                                                                            00851600 T  0000
  T ;                                                                           00851700 T  0000
      BEGIN                                                                     00851800 T  0000
LABEL XIT ;                                                                     00851900 T  0000
        SI := S ;                                                               00852000 T  0000
        SI := SI + K ;                                                          00852100 T  0000
        S := SI ;                                                               00852200 T  0000
        SI := LOC SR ;                                                          00852300 T  0001
        DI := LOC K ;                                                           00852400 T  0001
        SI := SI + 6 ;                                                          00852500 T  0001
        DI := DI + 7 ;                                                          00852600 T  0001
        DS := CHR ;                                                             00852700 T  0002
        DI := R ;                                                               00852800 T  0002
        K (DI := DI + 32 ; DI := DI + 32) ;                                     00852900 T  0002
        DI := DI + SR ;                                                         00853000 T  0004
        R := DI ;                                                               00853100 T  0004
        TALLY := 1 ;                                                            00853200 T  0005
        SI := LOC T ;                                                           00853300 T  0005
        DI := LOC K ;                                                           00853400 T  0005
        SI := SI + 6 ;                                                          00853500 T  0005
        DI := DI + 7 ;                                                          00853600 T  0006
        DS := 1 CHR ;                                                           00853700 T  0006
        DI := R ;                                                               00853800 T  0006
        K (2 (32 (                                                              00853900 T  0006
        SI := S ;                                                               00854000 T  0008
        IF I SC = DC THEN                                                       00854100 T  0008
        BEGIN                                                                   00854200 T  0009
          ID (JUMP OUT 4 TO XIT) ;                                              00854300 T  0009
          R := DI ;                                                             00854400 T  0012
          SI := R ;                                                             00854500 T  0012
          IF SC = ALPHA THEN ELSE                                               00854600 T  0012
          BEGIN                                                                 00854700 T  0013
            SI := SI - I ;                                                      00854800 T  0013
            SI := SI - 1 ;                                                      00854900 T  0014
            IF SC = ALPHA THEN ELSE                                             00855000 T  0014
              JUMP OUT 3 TO XIT ;                                               00855100 T  0014
          END ;                                                                 00855200 T  0016
```

```
            END ;                                                                    00855300 T  0016
            DI := DI - I ;                                                           00855400 T  0016
            DI := DI + 1))) ;                                                        00855500 T  0016
            T (                                                                      00855600 T  0017
            SI := S ;                                                                00855700 T  0018
            IF I SC = DC THEN                                                        00855800 T  0018
            BEGIN                                                                    00855900 T  0019
              ID (JUMP OUT 2 TO XIT) ;                                               00856000 T  0019
              R := DI ;                                                              00856100 T  0022
              SI := R ;                                                              00856200 T  0022
              IF SC = ALPHA THEN ELSE                                                00856300 T  0022
              BEGIN                                                                  00856400 T  0023
                SI := SI - I ;                                                       00856500 T  0023
                SI := SI - 1 ;                                                       00856600 T  0023
                IF SC = ALPHA THEN ELSE                                              00856700 T  0024
                  JUMP OUT TO XIT ;                                                  00856800 T  0024
              END ;                                                                  00856900 T  0025
            END ;                                                                    00857000 T  0025
            DI := DI - I ;                                                           00857100 T  0025
            DI := DI + 1) ;                                                          00857200 T  0025
            TALLY := 0 ;                                                             00857300 T  0026
XIT:                                                                                 00857400 T  0026
            PRESENT := TALLY ;                                                       00857500 T  0026
          END PRESENT ;                                                              00857600 T  0027
      IF PRESENT (STRING, ZIPPY, STRINGI, STRINGILEFT, STRINGIREPEAT,                00857610 T  0028
                  1-STRINGID, 0) EQV TEMPTOG THEN                                    00857620 T  0032
        STRINGFOUND := TRUE                                                          00857630 T  0034
      ELSE IF STRINGJ NEQ 0 THEN                                                     00857640 T  0035
        STRINGFOUND :=                                                               00857650 T  0037
          PRESENT (STRING, ZIPPY, STRINGJ, STRINGJLEFT, STRINGJREPEAT,               00857660 T  0038
                   1-STRINGJD, STRINGI) EQV TEMPTOG ;                                00857670 T  0041
    END STRINGFOUND ;                                                                00857680 T  0044
                                                                          32 IS   47 LONG, NEXT SEG   31
    DEFINE GETSTRINGS = IF ISOLATESTRINGS (LISTTYPE) THEN GO TO NEXT# ;              00857700 T  0000
    BOOLEAN PROCEDURE ISOLATESTRINGS (LISTTYPE) ;                                    00857800 T  0000
    VALUE LISTTYPE ;                                                                 00857900 T  0000
    INTEGER LISTTYPE ;                                                               00858000 T  0000
    BEGIN                                                                            00858100 T  0000
    STREAM PROCEDURE ISOLATE (S, D, L1, L2) ;                                        00858200 T  0000
                                                                        START OF SEGMENT **********   33
          BEGIN                                                                      00858300 T  0000
    LOCAL STOPCHR,                                                                   00858400 T  0000
      DX,                                                                            00858500 T  0000
      QUOTES ;                                                                       00858600 T  0000
    LABEL OK,                                                                        00858700 T  0000
      NOSTRING,                                                                      00858800 T  0000
      STRING,                                                                        00858900 T  0000
      JUMPOUT,                                                                       00858910 T  0000
      NO,                                                                            00859000 T  0000
      NEXTNO ;                                                                       00859100 T  0000
            TALLY := 63 ;                                                            00859200 T  0000
            STOPCHR := TALLY ;                                                       00859300 T  0000
            DI := LOC QUOTES ;                                                       00859400 T  0000
            DS := 2 LIT """" ;                                                       00859500 T  0000
            DS := 6 LIT ",.()[]" ;                                                   00859600 T  0001
            2 (SI := S ;                                                             00859700 T  0002
            63 (SI := SI + 1 ;                                                       00859800 T  0002
```

```
                IF SC = ALPHA THEN                                                  00859900 T  0003
                ELSE IF SC NEQ " " THEN                                             00860000 T  0003
                BEGIN                                                               00860100 T  0004
                  DI := LOC QUOTES ;                                                00860200 T  0004
                  4 (IF SC = DC THEN JUMP OUT 2 TO OK ;                             00860300 T  0005
                     SI := SI - 1 ;                                                 00860400 T  0006
                     DI := DI + 1) ;                                                00860500 T  0007
                  IF SC = ";" THEN JUMP OUT ;                                       00860600 T  0007
                END) ;                                                              00860700 T  0008
              GO TO NOSTRING ;                                                      00860800 T  0009
OK:                                                                                 00861200 T  0009
              DX := DI ;                                                            00861300 T  0009
              SI := SI - 1 ;                                                        00861400 T  0010
              IF SC = "," THEN                                                      00861500 T  0010
              BEGIN                                                                 00861600 T  0011
                DI := L1 ;                                                          00861700 T  0011
                DS := LIT "+" ;                                                     00861800 T  0011
              END ;                                                                 00861900 T  0012
              SI := SI + 1 ;                                                        00862000 T  0012
              TALLY := 0 ;                                                          00862100 T  0012
              STOPCHR (DI := DX ;                                                   00862200 T  0012
                       IF SC = DC THEN                                              00862300 T  0014
                         JUMP OUT 1 TO STRING ;                                     00862400 T  0014
                       SI := SI - 1 ;                                               00862500 T  0015
                       DI := D ;                                                    00862600 T  0015
                       DS := 1 CHR ;                                                00862700 T  0015
                       D := DI ;                                                    00862800 T  0015
                       TALLY := TALLY + 1) ;                                        00862900 T  0016
NOSTRING:                                                                           00863000 T  0016
              DI := L1 ;                                                            00863010 T  0016
              DS := 8 LIT "00000010" ;                                              00863020 T  0017
              GO TO JUMPOUT ;                                                       00863030 T  0018
STRING:                                                                             00863100 T  0018
              DI := L1 ;                                                            00863200 T  0018
              DI := DI + 2 ;                                                        00863300 T  0019
              2 (DX := DI ;                                                         00863400 T  0019
                 10  (IF SC GEQ "0" THEN                                          00863500 T  0020
                      BEGIN                                                         00863600 T  0020
                        DI := DX ;                                                  00863700 T  0020
                        DS := LIT "0" ;                                             00863800 T  0021
                        DS := CHR ;                                                 00863900 T  0022
                        IF SC GEQ "0" THEN                                          00864000 T  0022
                        BEGIN                                                       00864100 T  0022
                          SI := SI - 1 ;                                            00864200 T  0022
                          DI := DI - 2 ;                                            00864300 T  0023
                          DS := 2 CHR ;                                             00864400 T  0023
                        END ;                                                       00864500 T  0024
                        JUMP OUT 1 TO NEXTNO ;                                      00864600 T  0024
                      END ;                                                         00864700 T  0024
                      IF SC = ALPHA THEN                                            00864800 T  0024
                      ELSE IF SC NEQ " " THEN                                       00864900 T  0025
                      BEGIN                                                         00865000 T  0025
                        IF SC = ";" THEN JUMP OUT 2 TO NO ;                         00865100 T  0025
                        DI := LOC QUOTES ;                                          00865200 T  0027
                        4 (IF SC = DC THEN                                          00865300 T  0027
                           BEGIN                                                    00865400 T  0028
                             SI := SI - 1 ;                                         00865500 T  0028
```

```
                        JUMP OUT 3 TO NO ;                                  00865600 T  0029
                      END ;                                                 00865700 T  0030
                      SI := SI - 1 ;                                        00865800 T  0030
                      DI := DI + 1) ;                                       00865900 T  0030
                  END ;                                                     00866000 T  0031
                  SI := SI + 1) ;                                           00866100 T  0031
              JUMP OUT TO NO ;                                              00866200 T  0031
NEXTNO:                                                                     00866300 T  0032
           ) ;                                                              00866400 T  0032
           GO TO NO ;                                                       00866410 T  0032
JUMPOUT:                                                                    00866420 T  0032
           JUMP OUT ;                                                       00866430 T  0032
NO:                                                                         00866500 T  0033
           SI := SI - 1 ;                                                   00866600 T  0033
           S := SI ;                                                        00866700 T  0034
           DI := L1 ;                                                       00866800 T  0034
           L1 := TALLY ;                                                    00866900 T  0034
           TALLY := STOPCHR ;                                               00867000 T  0035
           L1 (TALLY := TALLY + 63) ;                                       00867100 T  0035
           STOPCHR := TALLY ;                                               00867200 T  0037
           SI := LOC L1 ;                                                   00867300 T  0037
           DI := DI + 7 ;                                                   00867400 T  0037
           SI := SI + 7 ;                                                   00867500 T  0038
           DS := 1 CHR ;                                                    00867600 T  0038
           DI := L2 ;                                                       00867700 T  0038
           L1 := DI) ;                                                      00867800 T  0038
       END ISOLATE ;                                                        00867900 T  0040
LABEL NEXT ;                                                                00868000 T  0040
INTEGER PROCEDURE DEFINESTRING (I, LEFT, RIGHT) ;                           00868010 T  0040
VALUE LEFT, RIGHT ; INTEGER I, LEFT, RIGHT ;                                00868020 T  0040
BEGIN                                                                       00868030 T  0040
       IF LEFT := 10×I.[12:6] + I.[18:6] = 99 THEN                          00868060 T  0040
       BEGIN                                                                00868070 T  0044
         LEFT := 1 ;                                                        00868080 T  0044
         RIGHT := 80 ;                                                      00868090 T  0045
       END ELSE                                                             00868100 T  0046
       IF RIGHT := 10×I.[24:6] + I.[30:6] = 99 THEN                         00868110 T  0046
         RIGHT := LEFT ;                                                    00868120 T  0050
       I := FULLLENGTH + 1 - STRINGI ;                                      00868130 T  0051
       LEFT := MIN (MAX (LEFT, IF COBOLFILE THEN 6 ELSE 1), I) ;            00868140 T  0055
       RIGHT := MIN (MAX (LEFT,RIGHT), I) ;                                 00868150 T  0062
       DEFINESTRING := LEFT - 1 ;                                           00868160 T  0067
       I := RIGHT - LEFT + 1 ;                                              00868170 T  0068
END DEFINESTRING ;                                                          00868190 T  0070
   IF NOT SCANTOG THEN                                                      00868200 T  0073
   BEGIN                                                                    00868300 T  0074
     IF PARAMETER1 = "ECHO    " THEN                                        00868400 T  0075
     BEGIN                                                                  00868500 T  0076
       IF CHANGETOG THEN                                                    00868600 T  0076
         CHANGEECHO := TOGGLE (CHANGEECHO, 2)                               00868700 T  0077
       ELSE                                                                 00868800 T  0079
         EDITECHO := TOGGLE (EDITECHO, 2) ;                                 00868900 T  0080
       GO TO NEXT ;                                                         00869000 T  0086
     END ;                                                                  00869100 T  0086
     READONLYCHECK ;                                                        00869200 T  0086
   END ;                                                                    00869300 T  0088
   IF EDITTOG THEN                                                          00869400 T  0088
```

```
BEGIN                                                                    00869500 T  0088
  IF NOT (NUM1 AND NUM2 AND NUM3) THEN                                   00869600 T  0089
    ERROR (NEXT, 0, PARAMETER0, " ERROR.") ;                             00869700 T  0092
  IF NOT ITSOLD (N := PARAMETER3) THEN                                   00869800 T  0096
    ERROR (NEXT, 0, "MISSING", " FORMAT") ;                              00869900 T  0098
  RDISC (AT, RECORD) ;                                                   00870000 T  0103
  IF COBOLFILE THEN                                                      00870100 T  0105
    RECORD [0].[1:35] := "@@@@@@" ;                                      00870200 T  0105
END ELSE                                                                 00870300 T  0108
BEGIN                                                                    00870400 T  0108
  I := M := 0 & "9999" [12:24:24] ;                                      00870500 T  0110
  ISOLATE (IMAGE, STRING, I, M) ;                                        00870600 T  0112
  IF I NEQ 64 THEN                                                       00870700 T  0115
  BEGIN                                                                  00870800 T  0115
    RELATIVENUMBER := FILEINFO ;                                         00870900 T  0116
    IF SCANTOG THEN                                                      00871000 T  0117
      IF NOT (EMPTY1 OR (NUM1 AND (NUM2 OR EMPTY2))) THEN                00871100 T  0118
        FILEINFO := DATA ;                                               00871200 T  0122
    STRINGI := I.[41:7] ;                                                00871300 T  0124
    STRINGID := REAL (I LSS 0) ;                                         00871400 T  0125
    STRINGILEFT := DEFINESTRING (I, 0, 0) ;                              00871500 T  0127
    STRINGIREPEAT := I ;                                                 00871600 T  0129
    IF M NEQ 64 THEN                                                     00871700 T  0131
    BEGIN                                                                00871800 T  0131
      STRINGJ := M.[41:7] ;                                              00871900 T  0132
      STRINGJD := REAL (M LSS 0) ;                                       00872000 T  0134
      STRINGJLEFT := DEFINESTRING (M, 0, 0) ;                            00872100 T  0135
      STRINGJREPEAT := M ;                                               00872200 T  0138
    END ELSE                                                             00872300 T  0139
      STRINGJ := 64 ;                                                    00872400 T  0139
    FILEINFO := RELATIVENUMBER ;                                         00872500 T  0142
  END ;                                                                  00874100 T  0143
  IF STRINGI = 0 OR (CHANGETOG AND STRINGJ = 64) THEN                    00874200 T  0143
    ERROR (NEXT, 0, "MISSING", " STRING") ;                              00874300 T  0146
  IF STRINGJ = 64 THEN                                                   00874400 T  0151
    STRINGJ := 0 ;                                                       00874500 T  0152
END ;                                                                    00874600 T  0153
IF FALSE THEN                                                            00874700 T  0153
NEXT:                                                                    00874800 T  0154
  ISOLATESTRINGS := TRUE ;                                               00874900 T  0155
END ISOLATESTRINGS ;                                                     00875000 T  0155

                                                                  33 IS  159 LONG, NEXT SEG   31
PROCEDURE EXTERNALFILE (LISTTYPE) ;                                      00875100 T  0000
VALUE LISTTYPE ; INTEGER LISTTYPE ;                                      00875200 T  0000
        BEGIN                                                            00875300 T  0000
FILE RO DISK SERIAL (2, INPUT [3], INPUT [4]) ;                          00875400 T  0000

                                                          START OF SEGMENT **********   34
LABEL MORE,                                                              00875600 T  0004
  EOF,                                                                   00875700 T  0004
  NEXT ;                                                                 00875800 T  0004
BOOLEAN POSTED,                                                          00875900 T  0004
  B ;                                                                    00876000 T  0004
          LOCKED := TRUE ;                                               00876100 T  0004
          RESETN := N ;                                                  00876300 T  0005
          FILL RO WITH INPUT [1], INPUT [2], *, *, *,                    00876400 T  0006
            12 + REAL (POSTED := PARAMETER1 = "MAIL % ") ;               00876500 T  0009
          N := 0 ;                                                       00876600 T  0012
```

```
                M := INPUT [5] + 1 ;                                                                00876700 T  0012
                B := POSTING ;                                                                      00876800 T  0014
                IF NUM3 THEN                                                                        00876900 T  0015
                BEGIN                                                                               00877000 T  0016
                  IF N := PARAMETER3 - 1 GEQ M THEN                                                 00877100 T  0016
                    ERROR (NEXT, 0, "USE REC", "ORD #S.") ;                                         00877110 T  0018
                  READ SEEK (RO [N]) ;                                                              00877120 T  0024
                  IF NUM4 THEN                                                                      00877200 T  0029
                    M := PARAMETER4 ;                                                               00877300 T  0030
                END                                                                                 00877400 T  0031
                ELSE IF NOT EMPTY3 THEN                                                             00877500 T  0031
                  B := TRUE ;                                                                       00877600 T  0033
                I := IF POSTING THEN ALGOL ELSE DATA ;                                              00877700 T  0034
                WRITE (ZIPPY [*], STAR) ;                                                           00877750 T  0037
MORE:                                                                                               00877800 T  0041
                INTERRUPT (1) ;                                                                     00877900 T  0041
                IF N := N + 1 GTR M THEN                                                            00878000 T  0046
                  GO TO EOF ;                                                                       00878100 T  0048
                READ (RO, 10, ZIPPY [*]) [EOF] ;                                                    00878200 T  0049
                IF SCANTOG THEN                                                                     00878300 T  0054
                  IF NOT STRINGFOUND THEN                                                           00878400 T  0055
                    GO TO MORE ;                                                                    00878500 T  0056
                IF B THEN                                                                           00879100 T  0056
                BEGIN                                                                               00879200 T  0056
                  IF POSTING AND FIRSTCHAR (ZIPPY [0]) = "*" THEN                                   00879300 T  0057
                    GO TO MORE ;                                                                    00879400 T  0060
                  WRITELFCR ;                                                                       00879500 T  0061
                END ELSE WRITESEQ ;                                                                 00879600 T  0061
                WRITEROW (ZIPPY, QUICK, I) ;                                                        00879700 T  0063
                IF POSTED THEN                                                                      00879800 T  0065
                  WRITE (RO, STAR) ;                                                                00879900 T  0065
                IF BREAKI = 0 THEN                                                                  00880000 T  0069
                BEGIN                                                                               00880100 T  0070
                  GO TO MORE ;                                                                      00880200 T  0071
EOF:                                                                                                00880300 T  0071
                  IF POSTED THEN                                                                    00880400 T  0072
                  BEGIN                                                                             00880500 T  0072
                    DETACH ;                                                                        00880600 T  0072
                    CLOSE (RO, PURGE) ;                                                             00880700 T  0073
                    REATTACH ;                                                                      00880800 T  0075
                  END ;                                                                             00880900 T  0076
                END ;                                                                               00880910 T  0076
NEXT:                                                                                               00881000 T  0076
                N := RESETN ;                                                                       00881100 T  0077
                LOCKED := FALSE ;                                                                   00881200 T  0077
              END EXTERNALFILE ;                                                                    00881300 T  0079

                                                                                          34 IS   84 LONG, NEXT SEG   31
PROCEDURE SPECIAL (LISTTYPE, ECHO) ;                                                                00881400 T  0000
VALUE LISTTYPE, ECHO ; INTEGER LISTTYPE ; BOOLEAN ECHO ;                                            00881500 T  0000
              BEGIN                                                                                 00881600 T  0000
LABEL                                                                                               00881700 T  0000

                                                                                  START OF SEGMENT **********   35
  REWRITE,                                                                                          00881800 T  0000
  OVERFLOW,                                                                                         00881850 T  0000
  NEXT ;                                                                                            00881900 T  0000
DEFINE QUICK = FALSE# ;                                                                             00882100 T  0000
INTEGER STREAM PROCEDURE CHANGED (S,D,I,J,STRING,SS,T,T1,SR,M,N,ID) ;                               00882200 T  0000
```

```
VALUE I,                                                                00882300 T  0000
  J,                                                                    00882400 T  0000
  SS,                                                                   00882500 T  0000
  T,                                                                    00882600 T  0000
  T1,                                                                   00882700 T  0000
  SR,                                                                   00882800 T  0000
  ID,                                                                   00882900 T  0000
  M,                                                                    00883000 T  0000
  N ;                                                                   00883100 T  0000
      BEGIN                                                             00883200 T  0000
LOCAL K,                                                                00883300 T  0000
  TOTAL ;                                                               00883400 T  0000
LABEL AROUND,                                                           00883500 T  0000
  XIT,                                                                  00883600 T  0000
  NO,                                                                   00883700 T  0000
  UNDERFLOW,                                                            00883800 T  0000
  HERE,                                                                 00883900 T  0000
  THERE,                                                                00883910 T  0000
  EXIT ;                                                                00883920 T  0000
        DI := D ;                                                       00884000 T  0000
        DS := 8 LIT "        " ;                                        00884100 T  0000
        SI := D ;                                                       00884200 T  0001
        DS := 9 WDS ;                                                   00884300 T  0001
        SI := LOC SS ;                                                  00884400 T  0002
        DI := LOC K ;                                                   00884500 T  0002
        SI := SI + 6 ;                                                  00884600 T  0002
        DI := DI + 7 ;                                                  00884700 T  0002
        DS := CHR ;                                                     00884800 T  0003
        SI := S ;                                                       00884900 T  0003
        DI := D ;                                                       00885000 T  0003
        K (DS := 32 CHR ; DS := 32 CHR) ;                               00885100 T  0003
        DS := SS CHR ;                                                  00885200 T  0005
        S := SI ;                                                       00885300 T  0006
        D := DI ;                                                       00885400 T  0006
        K := TALLY ;                                                    00885500 T  0006
        2 (T (K (DS := N CHR ;                                          00885600 T  0006
                 TALLY := K ;                                           00885700 T  0009
                 JUMP OUT TO HERE) ;                                    00886000 T  0010
              DI := S ;                                                 00886100 T  0011
              SI := STRING ;                                            00886200 T  0011
              IF I SC NEQ DC THEN                                       00886300 T  0011
              BEGIN                                                     00886400 T  0012
NO:                                                                     00886500 T  0012
                SI := S ;                                               00886600 T  0012
                DI := D ;                                               00886700 T  0013
                DS := CHR ;                                             00886800 T  0013
                S := SI ;                                               00886900 T  0013
                D := DI ;                                               00887000 T  0014
                SI := SR ;                                              00887100 T  0014
                SI := SI - 8 ;                                          00887200 T  0014
                SR := SI ;                                              00887300 T  0014
                TALLY := 1 ;                                            00887310 T  0015
                GO TO HERE ;                                            00887400 T  0015
              END ;                                                     00887500 T  0015
              ID (SS := DI ;                                            00887600 T  0015
                  SI := SS ;                                            00887700 T  0016
                  IF SC = ALPHA THEN                                    00887800 T  0017
```

```
                    JUMP OUT TO NO ;                                                00887900 T  0017
                  SI := SI - I ;                                                    00888000 T  0018
                  SI := SI - 1 ;                                                    00888100 T  0018
                  IF SC = ALPHA THEN                                                00888200 T  0018
                    JUMP OUT TO NO ;                                                00888300 T  0019
                  SI := STRING ;                                                    00888400 T  0019
                  SI := SI + I) ;                                                   00888500 T  0020
             TALLY := 1 ;                                                           00888600 T  0020
             CHANGED := TALLY ;                                                     00888700 T  0021
             S := DI ;                                                              00888800 T  0021
             DI := D ;                                                              00888900 T  0021
             GO TO THERE ;                                                          00888910 T  0021
HERE:                                                                               00888920 T  0022
             GO TO AROUND ;                                                         00888930 T  0022
THERE:                                                                              00888940 T  0022
             N (DI := DI + J ;                                                      00889000 T  0022
                D := DI ;                                                           00889100 T  0024
                DI := TOTAL ;                                                       00889200 T  0024
                8 (DI := DI + J ;                                                   00889300 T  0025
                   DI := DI - I) ;                                                  00889400 T  0025
                TOTAL := DI ;                                                       00889500 T  0026
                DI := SR ;                                                          00889600 T  0026
                8 (DI := DI - J) ;                                                  00889700 T  0027
                SR := DI ;                                                          00889800 T  0028
                DI := D ;                                                           00889900 T  0028
                DI := DI - J ;                                                      00890000 T  0028
                DS := CHR ;                                                         00890100 T  0029
                TALLY := J ;                                                        00890200 T  0029
                JUMP OUT TO AROUND) ;                                               00890300 T  0030
             DS := J CHR ;                                                          00890400 T  0030
             D := DI ;                                                              00890500 T  0031
             SI := SR ;                                                             00890600 T  0031
             8 (SI := SI - I) ;                                                     00890700 T  0031
             SR := SI ;                                                             00890800 T  0032
             TALLY := I ;                                                           00890900 T  0033
AROUND:                                                                             00891000 T  0033
             TALLY := TALLY + 63 ;                                                  00891100 T  0033
             K := TALLY) ;                                                          00891200 T  0034
          TALLY := T1 ;                                                             00891600 T  0035
          T := TALLY) ;                                                             00891700 T  0035
       CI := CI + CHANGED ;                                                         00891800 T  0036
       GO TO EXIT ;                                                                 00891900 T  0036
       M (K (DS := N CHR ;                                                          00892000 T  0037
             TALLY := K ;                                                           00892100 T  0039
             TALLY := TALLY + 63 ;                                                  00892200 T  0040
             K := TALLY ;                                                           00892300 T  0040
             JUMP OUT)) ;                                                           00892400 T  0040
       TALLY := 2 ;                                                                 00892500 T  0042
       K (CHANGED := TALLY ;                                                        00892600 T  0042
          JUMP OUT TO EXIT) ;                                                       00892700 T  0043
       SI := LOC SR ;                                                               00892800 T  0044
       DI := LOC SS ;                                                               00892900 T  0044
       6 (IF SC NEQ "0" THEN JUMP OUT TO UNDERFLOW ; SI := SI + 1) ;                00893000 T  0045
       DI := DI + 7 ;                                                               00893100 T  0046
       DS := CHR ;                                                                  00893200 T  0047
       SI := S ;                                                                    00893300 T  0047
       DI := D ;                                                                    00893400 T  0047
```

```
          SS (DS := 32 CHR ; DS := 32 CHR) ;                                    00893500 T  0047
          DS := SR CHR ;                                                        00893600 T  0049
          S := SI ;                                                             00893700 T  0050
          GO TO UNDERFLOW ;                                                     00893710 T  0050
EXIT:                                                                           00893720 T  0050
          GO TO XIT ;                                                           00893730 T  0050
UNDERFLOW:                                                                      00893800 T  0051
          N (SI := LOC TOTAL ;                                                  00893900 T  0051
             DI := LOC K ;                                                      00894000 T  0053
             SI := SI + 6 ;                                                     00894100 T  0053
             DI := DI + 7 ;                                                     00894200 T  0053
             DS := 1 CHR ;                                                      00894300 T  0054
             SI := S ;                                                          00894400 T  0054
             K (2 (32 (IF SC NEQ " " THEN                                       00894500 T  0054
                       BEGIN                                                    00894600 T  0056
                         CHANGED := TALLY ;                                     00894700 T  0056
                         JUMP OUT 4 TO XIT ;                                    00894800 T  0057
                       END ;                                                    00894900 T  0058
                       SI := SI + 1))) ;                                        00895000 T  0058
             TOTAL (IF SC NEQ " " THEN                                          00895100 T  0059
                    BEGIN                                                       00895200 T  0061
                      CHANGED := TALLY ;                                        00895300 T  0061
                      JUMP OUT 2 TO XIT ;                                       00895400 T  0061
                    END ;                                                       00895500 T  0062
                    SI := SI + 1)) ;                                            00895600 T  0062
XIT:                                                                            00895700 T  0063
       END CHANGED ;                                                            00895800 T  0063
BOOLEAN STREAM PROCEDURE EDITS (F, S, D, N) ;                                   00895900 T  0065
VALUE N ;                                                                       00896000 T  0065
       BEGIN                                                                    00896100 T  0065
LABEL XIT ;                                                                     00896200 T  0065
          DI := D ;                                                             00896300 T  0065
          DS := 8 LIT " " ;                                                     00896400 T  0065
          SI := D ;                                                             00896500 T  0066
          DS := 9 WDS ;                                                         00896600 T  0066
          DI := D ;                                                             00896700 T  0067
          D := TALLY ;                                                          00896800 T  0067
          2 (N (SI := F ;                                                       00896900 T  0067
          IF SC = "@" THEN                                                      00897000 T  0069
          BEGIN                                                                 00897100 T  0069
            SI := SI + 1 ;                                                      00897200 T  0069
            F := SI ;                                                           00897300 T  0070
            SI := S ;                                                           00897400 T  0070
            DS := CHR ;                                                         00897500 T  0070
            S := SI ;                                                           00897600 T  0071
          END                                                                   00897700 T  0071
          ELSE IF SC = "#" THEN                                                 00897800 T  0071
          BEGIN                                                                 00897900 T  0072
            SI := SI + 1 ;                                                      00898000 T  0072
            F := SI ;                                                           00898100 T  0072
            SI := S ;                                                           00898200 T  0073
            SI := SI + 1 ;                                                      00898300 T  0073
            S := SI ;                                                           00898400 T  0073
          END                                                                   00898500 T  0073
          ELSE                                                                  00898600 T  0073
          BEGIN                                                                 00898700 T  0074
            DS := CHR ;                                                         00898800 T  0074
```

```
                F := SI ;                                                    00898900 T  0074
                SI := D ;                                                    00899000 T  0074
                SI := SI + 8 ;                                               00899100 T  0074
                D := SI ;                                                    00899200 T  0075
              END)) ;                                                        00899300 T  0075
              SI := LOC D ;                                                  00899400 T  0075
              DI := LOC N ;                                                  00899500 T  0076
              SI := SI + 6 ;                                                 00899600 T  0076
              DI := DI + 7 ;                                                 00899700 T  0076
              DS := 1 CHR ;                                                  00899800 T  0076
              SI := S ;                                                      00899900 T  0077
              N ( 2 ( 32 (IF SC NEQ " " THEN                                 00900000 T  0077
                           BEGIN                                             00900100 T  0079
                             TALLY := 1 ;                                    00900200 T  0079
                             EDITS := TALLY ;                                00900300 T  0080
                             JUMP OUT 3 TO XIT ;                             00900400 T  0080
                           END ;                                             00900500 T  0081
                           SI := SI + 1))) ;                                 00900600 T  0081
              D (IF SC NEQ " " THEN                                          00900700 T  0082
                 BEGIN                                                       00900800 T  0083
                   TALLY := 1 ;                                              00900900 T  0083
                   EDITS := TALLY ;                                          00901000 T  0084
                   JUMP OUT ;                                                00901100 T  0084
                 END ;                                                       00901200 T  0085
                 SI := SI + 1) ;                                             00901300 T  0085
XIT:                                                                         00901400 T  0086
        END EDITS ;                                                          00901500 T  0086
        REAL L ;                                                             00901600 T  0087
              IF CHANGETOG THEN                                              00901700 T  0087
              BEGIN                                                          00901710 T  0087
                PARAMETER1 := STRINGIREPEAT DIV 2 ;                          00901720 T  0088
                PARAMETER2 := STRINGIREPEAT - PARAMETER1 ;                   00901730 T  0089
                PARAMETER3 := FULLLENGTH - STRINGILEFT ;                     00901740 T  0091
                PARAMETER4 := MIN (PARAMETER3, 63) ;                         00901750 T  0094
              END ;                                                          00901760 T  0097
              WHILE N := (L := LL [AT]).S LEQ M DO                           00901900 T  0097
              BEGIN                                                          00902000 T  0103
               RDISC (AT, ZIPPY) ;                                           00902100 T  0103
               IF SCANTOG THEN                                               00902200 T  0106
               BEGIN                                                         00902300 T  0106
                 IF STRINGFOUND THEN                                         00902400 T  0107
                 BEGIN                                                       00902700 T  0107
                   WRITEME (N, ZIPPY) ;                                      00902900 T  0108
                   N := N + 1 ;                                              00902910 T  0110
                   GO TO NEXT ;                                              00903000 T  0112
                 END ;                                                       00903100 T  0112
               END                                                           00903700 T  0112
               ELSE IF CHANGETOG THEN                                        00903800 T  0112
               BEGIN                                                         00903900 T  0113
                 IF I := CHANGED (ZIPPY, IMAGE, STRINGI, STRINGJ,            00904000 T  0114
                           STRING, STRINGILEFT, PARAMETER1, PARAMETER2,      00904100 T  0117
                           PARAMETER3, PARAMETER4, STRINGI LSS STRINGJ,      00904200 T  0119
                           STRINGID) = 1 THEN                                00904300 T  0121
                 BEGIN                                                       00904500 T  0122
                   RESETN := N ;                                             00904600 T  0123
REWRITE:                                                                     00904700 T  0124
                   IF ECHO THEN                                              00904800 T  0124
```

```
                    WRITEME (N, IMAGE) ;                                                    00904900 T  0124
                  WDISC ;                                                                   00905000 T  0127
                END ELSE                                                                    00905100 T  0129
                IF I = 2 THEN                                                               00905200 T  0129
OVERFLOW:                                                                                   00905250 T  0130
                  ERROR (NEXT, 0, PARAMETER0, " OVRFLW") ;                                  00905300 T  0131
              END                                                                           00905400 T  0134
              ELSE                                                                          00905500 T  0134
              BEGIN                                                                         00905600 T  0134
                IF EDITS (RECORD, ZIPPY, IMAGE, HALFFULLLENGTH) THEN                        00905700 T  0134
                  GO TO OVERFLOW ;                                                          00905800 T  0139
                GO TO REWRITE ;                                                             00905900 T  0140
              END ;                                                                         00906000 T  0140
              INTERRUPT (1) ;                                                               00906100 T  0140
              AT := L.T ;                                                                   00906110 T  0146
            END ;                                                                           00906200 T  0147
            IF SCANTOG THEN                                                                 00906300 T  0148
              ERRORX (0, "EOF NO ", "STRING.") ;                                            00906400 T  0148
NEXT:                                                                                       00906500 T  0150
            IF CHANGETOG THEN                                                               00906600 T  0151
              N := RESETN ;                                                                 00906700 T  0151
          END SPECIAL ;                                                                     00906800 T  0153
                                                                                    35 IS  158 LONG, NEXT SEG   31
BOOLEAN COMPLEX ;                                                                           00906900 T  0000
REAL L ;                                                                                    00906910 T  0000
      IF COMPLEX := SCANTOG OR CHANGETOG OR EDITTOG THEN                                    00907000 T  0000
        GETSTRINGS ;                                                                        00907100 T  0003
      IF NUM1 AND (NUM2 OR EMPTY2 OR CHANGETOG) THEN                                        00907200 T  0005
      BEGIN                                                                                 00907300 T  0009
        N := PARAMETER1 ;                                                                   00907400 T  0009
        IF NUM2 THEN                                                                        00907500 T  0010
          M := PARAMETER2                                                                   00907600 T  0011
        ELSE IF SCANTOG THEN                                                                00907700 T  0011
          M := FINITY                                                                       00907800 T  0013
        ELSE M := N ;                                                                       00907900 T  0014
      END                                                                                   00908000 T  0017
      ELSE IF NOT (EMPTY1 OR CHANGETOG) THEN                                                00908100 T  0017
      BEGIN                                                                                 00908200 T  0020
        IF XFILE (12, 0, 2) LSS 2 THEN                                                      00908300 T  0020
          GO TO NEXT ;                                                                      00908400 T  0022
        IF LOCKED OR NOT POSTING THEN                                                       00908500 T  0023
          WAIT ((IF NUM3 AND NUM4 THEN                                                      00908600 T  0024
            MIN (PARAMETER4, INPUT [5]) ELSE INPUT [5]) =                                   00908700 T  0024
            (IF NUM3 THEN PARAMETER3 ELSE 0), LOCKED) ;                                     00908800 T  0024
        EXTERNALFILE (LISTTYPE) ;                                                           00908900 T  0036
        GO TO NEXT ;                                                                        00909000 T  0037
      END                                                                                   00909100 T  0038
      ELSE                                                                                  00909200 T  0038
      BEGIN                                                                                 00909300 T  0038
        IF NOT COMPLEX THEN                                                                 00909400 T  0038
        BEGIN                                                                               00909500 T  0039
          AT := 0 ;                                                                         00909600 T  0039
          N := 1 ;                                                                          00909700 T  0040
        END ;                                                                               00909800 T  0041
        IF CHANGETOG THEN                                                                   00909900 T  0041
          M := N                                                                            00910000 T  0041
        ELSE                                                                                00910100 T  0042
```

```
          M := FINITY ;                                                                    00910200 T  0043
        END ;                                                                              00910300 T  0044
        OPENCHECK ;                                                                        00910400 T  0044
        IF COMPLEX THEN                                                                    00910500 T  0045
          WAIT (KOUNT (N, M, CLOCK), FALSE) ;                                              00910600 T  0046
        IF ITSOLD (N) THEN ;                                                               00910700 T  0050
        IF COMPLEX THEN                                                                    00910900 T  0052
          SPECIAL (LISTTYPE, TEMPTOG EQV (IF CHANGETOG THEN CHANGEECHO                     00911000 T  0052
                                          ELSE EDITECHO))                                  00911050 T  0055
        ELSE                                                                               00911100 T  0057
        BEGIN                                                                              00911200 T  0057
          WHILE N := (L := LL [AT]).S LEQ M DO                                             00911300 T  0060
          BEGIN                                                                            00911310 T  0065
            WRITEAT ;                                                                      00911400 T  0065
            INTERRUPT (1) ;                                                                00911410 T  0069
            AT := L.T ;                                                                    00911420 T  0074
          END ;                                                                            00911500 T  0076
          N := LL [L.F].S + INC ;                                                          00911600 T  0076
        END ;                                                                              00911700 T  0081
NEXT:                                                                                      00911800 T  0081
    END LISTIT ;                                                                           00911900 T  0082
                                                                                  31 IS   87 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    00912000 T  0051
PROCEDURE EXECUTE ;                                                                        00950000 T  0051
BEGIN                                                                                      00950100 T  0051
LABEL NEXT ;                                                                               00950200 T  0051
                                                                                 START OF SEGMENT **********   36
INTEGER XSUB ;                                                                             00950300 T  0000
REAL *START,                                                                               00950400 T  0000
  YLAST,                                                                                   00950500 T  0000
  YFILETYPE,                                                                               00950600 T  0000
  YREPEAT,                                                                                 00950700 T  0000
  YNCHRS ;                                                                                 00950800 T  0000
BOOLEAN VERBISEXECUTE ;                                                                    00950900 T  0000
   IF VERBISEXECUTE := PARAMETER0 = RSWD [0] THEN                                          00951000 T  0000
     IF PARAMETER1 = "LIBRARY" THEN                                                        00951100 T  0001
     BEGIN                                                                                 00951200 T  0002
       IF EMPTY2 THEN                                                                      00951300 T  0003
         ERROR (NEXT, 7, "MACRO=/", MACROLIBRARY) ;                                        00951400 T  0004
       IF NUM2 THEN                                                                        00951500 T  0009
         MACROLIBRARY := OCTDEC (PARAMETER2)                                               00951600 T  0009
       ELSE                                                                                00951700 T  0010
         MACROLIBRARY := PARAMETER2 ;                                                      00951800 T  0012
       GO TO NEXT ;                                                                        00951900 T  0013
     END ELSE                                                                              00952000 T  0013
     IF PARAMETER1 = "ECHO    " THEN                                                       00952100 T  0013
     BEGIN                                                                                 00952200 T  0015
       EXECUTEECHO := TOGGLE (EXECUTEECHO, 2) ;                                            00952300 T  0015
       GO TO NEXT ;                                                                        00952400 T  0018
     END ;                                                                                 00952500 T  0020
   IF XDEX + 1 GEQ XMAX THEN                                                               00952600 T  0020
     ERROR (NEXT, 0, PARAMETER0, " OVRFLW") ;                                              00952700 T  0021
   XSUB := (XDEX + 1) * 13 ;                                                               00952800 T  0025
   YFILETYPE := DATA ;                                                                     00952900 T  0026
   IF NOT VERBISEXECUTE THEN                                                               00953000 T  0027
   BEGIN                                                                                   00953100 T  0028
     XPARAMETERS [0] := PARAMETER1 ;                                                       00953200 T  0028
```

```
    XPARAMETERS [1] := PARAMETER2 ;                                                  00953300 T  0031
    XPARAMETERS [2] := PARAMETER3 ;                                                  00953400 T  0033
    XPARAMETERS [3] := PARAMETER4 ;                                                  00953500 T  0036
    XPARAMETERS [4] := ="#000000" ;                                                  00953600 T  0038
    YREPEAT := 1 ;                                                                   00953700 T  0041
    PARAMETER2 := INPUT [2] ;                                                        00953800 T  0042
    PARAMETER1 := PARAMETER0 ;                                                       00953900 T  0043
    %LAST := INPUT [5] + 1 ;                                                         00954000 T  0043
  END ELSE                                                                           00954100 T  0045
  IF FILEOPEN AND (NUM1 OR EMPTY1) AND (NUM2 OR EMPTY2) THEN                         00954200 T  0045
  BEGIN                                                                              00954300 T  0051
    IF NUM1 THEN                                                                     00954400 T  0052
    BEGIN                                                                            00954500 T  0053
      PARAMETER3 := PARAMETER1 ;                                                     00954600 T  0053
      IF NUM2 THEN                                                                   00954700 T  0054
        PARAMETER4 := PARAMETER2                                                     00954800 T  0055
      ELSE                                                                           00954900 T  0055
        PARAMETER4 := PARAMETER3 ;                                                   00955000 T  0056
    END ELSE                                                                         00955100 T  0057
    BEGIN                                                                            00955200 T  0057
      PARAMETER3 := 1 ;                                                              00955300 T  0058
      PARAMETER4 := FINITY ;                                                         00955400 T  0058
    END ;                                                                            00955500 T  0059
    PARAMETER1 := OCTDEC(XDEX+1+10×STATION,[14:4]+1000×STATION,[9:4]);               00955600 T  0059
    PARAMETER2 := "#MACRO#" ;                                                        00955700 T  0064
    IF YFILETYPE := XFILE (PARAMETER1, PARAMETER2, =1) = 7 THEN                      00955800 T  0065
    BEGIN                                                                            00955900 T  0068
      READ (LIBRARY) ;                                                               00956000 T  0068
      DETACH ;                                                                       00956100 T  0072
      CLOSE (LIBRARY, PURGE) ;                                                       00956200 T  0073
      REATTACH ;                                                                     00956300 T  0075
    END ELSE                                                                         00956400 T  0076
    IF YFILETYPE GEQ 0 THEN                                                          00956500 T  0076
      ERROR (NEXT, 4, PARAMETER1, PARAMETER2) ;                                      00956600 T  0079
    YLAST := KOUNT (PARAMETER3, PARAMETER4, =1) ;                                    00956700 T  0082
    I := SAVEFACTOR ;                                                                00956710 T  0084
    FREEFILE (STATION) ;                                                             00956800 T  0084
    THERMOFAX (SAVEFACTOR := 0, (YLAST + 14) DIV 15 × 15) ;                          00956900 T  0085
    UNFREEFILE (STATION) ;                                                           00957000 T  0093
    SAVEFACTOR := I ;                                                                00957010 T  0094
    %FILETYPE := FILEINFO ;                                                          00957100 T  0095
  END ELSE                                                                           00957200 T  0096
  BEGIN                                                                              00957300 T  0096
    IF XFILE (12, 0, 2) LSS 2 THEN                                                   00957400 T  0096
      GO TO NEXT ;                                                                   00957500 T  0098
    %LAST := INPUT [5] + 1 ;                                                         00957600 T  0099
    IF NUM3 THEN                                                                     00957700 T  0100
    BEGIN                                                                            00957800 T  0101
      IF YSTART := PARAMETER3 = 1 GTR YLAST THEN                                     00957900 T  0101
        ERROR (NEXT, 0, "USE REC", "ORD #S.") ;                                      00958000 T  0103
      IF NUM4 THEN                                                                   00958100 T  0108
        IF PARAMETER4 LSS YLAST THEN                                                 00958200 T  0108
          YLAST := PARAMETER4 ;                                                      00958300 T  0110
    END ;                                                                            00958400 T  0111
  END ;                                                                              00958500 T  0111
  IF XDEX LSS 0 THEN                                                                 00958600 T  0111
    XECHO := TEMPTOG EQV EXECUTEECHO ;                                               00958700 T  0112
```

```
    IF VERBISEXECUTE THEN                                                                00958800 T  0115
      IF YREPEAT := GETPARAMETERS (63) = 0 THEN                                          00958900 T  0116
        GO TO NEXT ;                                                                     00959000 T  0118
    WAIT ((YLAST - YSTART) * YREPEAT * 3, FALSE) ;                                       00959100 T  0118
    IF MOREINPUT THEN                                                                    00959200 T  0123
    BEGIN                                                                                00959300 T  0123
      READ (IO [USER + MAXUSERS], 30, IMAGE [*]) ;                                       00959400 T  0124
      WRITE (IO [2*MAXUSERS+XMAX*USER+XDEX+1], 30, IMAGE [*]) ;                          00959410 T  0130
      YNCHRS := NCHRS & 1[1:47:1] ;                                                      00959500 T  0137
      MOREINPUT := FALSE ;                                                               00959600 T  0140
    END ;                                                                                00959700 T  0142
    XDEX := XDEX + 1 ;                                                                   00959800 T  0142
    XN := XSTART := YSTART ;                                                             00959900 T  0143
    XLAST := YLAST ;                                                                     00960000 T  0148
    XFILETYPE := YFILETYPE ;                                                             00960100 T  0150
    XREPEAT := YREPEAT ;                                                                 00960200 T  0153
    XNCHRS := YNCHRS ;                                                                   00960300 T  0155
    XPREFIX := PARAMETER1 ;                                                              00960400 T  0158
    XSUFFIX := PARAMETER2 ;                                                              00960500 T  0160
NEXT:                                                                                    00960600 T  0163
END EXECUTE ;                                                                            00960700 T  0164
                                                                                  36 IS  168 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *       00970000 T  0051
PROCEDURE XVERBS (K) ; VALUE K ; INTEGER K ;                                             00970100 T  0051
BEGIN                                                                                    00970200 T  0051
DEFINE                                                                                   00970300 T  0051
                                                                                START OF SEGMENT **********   37
  REPLACE =                                                                              00970400 T  0000
   BEGIN                                                                                 00970500 T  0000
     IF NUM2 OR EMPTY2 THEN                                                              00970600 T  0000
       ERROR (NEXT, 0, PARAMETER2, " IS BAD") ;                                          00970700 T  0000
     M := -1 ;                                                                           00970800 T  0000
     FOR I := 0 STEP 1 UNTIL RSWDM DO                                                    00970900 T  0000
       IF PARAMETER0 := RSWD [I] = PARAMETER1 THEN                                       00971000 T  0000
         M := I                                                                          00971100 T  0000
       ELSE IF PARAMETER0 = PARAMETER2 THEN                                              00971200 T  0000
         ERROR (NEXT, 0, "DUP    ", PARAMETER2) ;                                        00971300 T  0000
     IF M LSS 0 THEN                                                                     00971400 T  0000
       ERROR (NEXT, 0, "NO VERB", PARAMETER1) ;                                          00971500 T  0000
     RSWD [M] := PARAMETER2 ;                                                            00971600 T  0000
   END#,                                                                                 00971700 T  0000
  DELETE =                                                                               00971800 T  0000
   BEGIN                                                                                 00971900 T  0000
     OPENCHECK ;                                                                         00972000 T  0000
     IF NOT NUM1 THEN                                                                    00972100 T  0000
       PARAMETER1 := N ;                                                                 00972200 T  0000
     INORDER := READONLYFILE ;                                                           00972300 T  0000
     IF NOT NUM2 OR PARAMETER2 LSS PARAMETER1 THEN                                       00972400 T  0000
       PARAMETER2 := PARAMETER1 ;                                                        00972500 T  0000
     I := LL [LOC (PARAMETER1)] . F ;                                                    00972600 T  0000
     IF ITSOLD (PARAMETER2) THEN                                                         00972700 T  0000
       AT := LL [AT] . T ;                                                               00972800 T  0000
     LL [I] . T := AT ;                                                                  00972900 T  0000
     MODIFY (I) ;                                                                        00973000 T  0000
     LL [AT] . F := I ;                                                                  00973100 T  0000
     MODIFY (AT) ;                                                                       00973200 T  0000
     N := LL [I] . S + INC ;                                                             00973300 T  0000
```

```
      END#,                                                              00973400 T  0000
    PRINTORPUNCH =                                                       00973500 T  0000
      BEGIN                                                              00973600 T  0000
        OPENCHECK ;                                                      00973700 T  0000
        IF NOT NUM3 THEN                                                 00973800 T  0000
          PARAMETER3 := 1 ;                                              00973900 T  0000
        IF NOT NUM4 THEN                                                 00974000 T  0000
          PARAMETER4 := FINITY ;                                         00974100 T  0000
        THERMOFAX (K, 0) ;                                               00974200 T  0000
      END# ;                                                             00974300 T  0000
   LABEL NEXT ;                                                          00974400 T  0000
      IF BOOLEAN (K) THEN                                                00974500 T  0000
        IF K = 1 THEN                                                    00974600 T  0000
          REPLACE                                                        00974700 T  0001
        ELSE                                                             00974800 T  0025
          DELETE                                                         00974900 T  0025
      ELSE IF K = 0 THEN                                                 00975000 T  0067
        CLOSEMYFILE                                                      00975100 T  0068
      ELSE                                                               00975200 T  0082
        PRINTORPUNCH ;                                                   00975300 T  0082
   NEXT:                                                                 00975400 T  0095
   END XVERBS ;                                                          00975500 T  0095
                                                            37 IS    97 LONG, NEXT SEG     2
   DEFINE CLOSEFILE = XVERBS (0)#,                                       00975600 T  0051
     REPLACE = XVERBS (1)#,                                              00975700 T  0051
     PRINT = XVERBS (2)#,                                                00975800 T  0051
     DELETE = XVERBS (3)#,                                               00975900 T  0051
     PUNCH = XVERBS (4)# ;                                               00976000 T  0051
   % * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *      01023000 T  0051
   PROCEDURE MAIL ;                                                      01023500 T  0051
      BEGIN                                                              01024000 T  0051
   LABEL NEXT ;                                                          01024100 T  0051
                                                           START OF SEGMENT **********    38
   BOOLEAN STREAM PROCEDURE POSTFROM (SENDER, MESSAGE, Z) ;              01024500 T  0000
        BEGIN                                                            01025000 T  0000
   LABEL OK,                                                             01025500 T  0000
     EXIT ;                                                              01026000 T  0000
          SI := Z ;                                                      01026500 T  0000
          DI := Z ;                                                      01027000 T  0000
          DS := 8 LIT "        " ;                                       01027500 T  0000
          DS := 8 WDS ;                                                  01028000 T  0001
          SI := MESSAGE ;                                                01028500 T  0002
          20 (IF SC = ":" THEN                                           01029000 T  0002
            JUMP OUT TO OK ;                                             01029500 T  0003
          SI := SI + 1) ;                                                01030000 T  0003
          TALLY := 1 ;                                                   01030500 T  0004
          POSTFROM := TALLY ;                                            01031000 T  0004
          GO TO EXIT ;                                                   01031500 T  0004
   OK:                                                                   01032000 T  0004
          SI := SI + 1 ;                                                 01032500 T  0004
          DI := Z ;                                                      01033000 T  0005
          63 (IF SC = ";" THEN                                           01033500 T  0005
            JUMP OUT ;                                                   01034000 T  0006
          DS := 1 CHR) ;                                                 01034500 T  0006
          DS := 1 LIT "-" ;                                              01035000 T  0008
          SI := SENDER ;                                                 01035500 T  0008
          SI := SI + 1 ;                                                 01036000 T  0008
```

```
        DS := 7 CHR ;                                                                    01036500 T  0009
EXIT:                                                                                    01037000 T  0009
     END POSTFROM ;                                                                      01037500 T  0009
     IF NUM2 THEN                                                                        01038000 T  0011
       PARAMETER2 := OCTDEC (PARAMETER2) ;                                               01038500 T  0011
     I := XFILE ("MAIL % ", IF EMPTY1 THEN USERCODE ELSE PARAMETER2,                     01039000 T  0014
          -1) ;                                                                          01039500 T  0016
     IF EMPTY1 THEN                                                                      01040500 T  0018
     BEGIN                                                                               01041000 T  0018
       IF I LSS 7 THEN                                                                   01041500 T  0019
         ERROR (NEXT, 0, "SORRY, ", "NO MAIL") ;                                         01042000 T  0020
       PARAMETER1 := "MAIL % " ;                                                         01042500 T  0026
       NUM1 := FALSE ;                                                                   01043500 T  0026
       PARAMETER2 := USERCODE ;                                                          01044000 T  0028
       NUM2 := FALSE ;                                                                   01045000 T  0029
       NUM3 := FALSE ;                                                                   01045500 T  0031
       LISTIT (17) ;% POSTING AND QUICK                                                  01046000 T  0032
     END                                                                                 01047000 T  0033
     ELSE                                                                                01047500 T  0033
     BEGIN                                                                               01048000 T  0033
       IF PARAMETER1 NEQ "TO      " THEN                                                 01048500 T  0035
         ERROR (NEXT, 0, "MISSING", " TO,    ") ;                                       01049000 T  0035
       IF POSTFROM (USERCODE, IMAGE, RECORD) THEN                                        01049500 T  0041
         ERROR (NEXT, 0, "MISSING", " COLON.") ;                                         01050000 T  0043
       IF I LSS 0 THEN                                                                   01050500 T  0048
         BEGIN                                                                           01051000 T  0048
           FREEFILE (STATION) ;                                                          01051500 T  0049
           PARAMETER1 := "MAIL % " ;                                                     01052000 T  0050
           CREATEFILE (15) ;                                                             01053000 T  0051
           UNFREEFILE (STATION) ;                                                        01053500 T  0070
         END                                                                             01054000 T  0071
         ELSE IF I GTR 2 THEN                                                            01054500 T  0071
         BEGIN                                                                           01055000 T  0077
           WRITE (LIBRARY [INPUT [5] + 1], 10, RECORD [*]) ;                             01055500 T  0078
           CLOSE (LIBRARY) ;                                                             01056000 T  0084
         END ELSE                                                                        01056500 T  0085
           ERRORX (1, "MAIL % ", PARAMETER2) ;                                           01056600 T  0085
     END ;                                                                               01057500 T  0087
NEXT:                                                                                    01058000 T  0087
   END POSTMAN ;                                                                         01058500 T  0088
                                                                          38 IS   90 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * 01059000 T  0051
PROCEDURE COPY ;                                                                         01059500 T  0051
   BEGIN                                                                                 01060000 T  0051
BOOLEAN B,                                                                               01060100 T  0051
                                                                    START OF SEGMENT **********   39
  MERGE ;                                                                                01060200 T  0000
LABEL NEXT ;                                                                             01060500 T  0000
     IF PARAMETER1 = "OVERITE" THEN                                                      01060550 T  0000
     BEGIN                                                                               01060600 T  0000
       COPYCLOBBER := TOGGLE (COPYCLOBBER, 2) ;                                          01060700 T  0001
       GO TO NEXT ;                                                                      01060850 T  0004
     END ;                                                                               01060900 T  0006
     READONLYCHECK ;                                                                     01060950 T  0006
     IF XFILE (12, 0, 2) LSS 2 THEN                                                      01061500 T  0007
       GO TO NEXT ;                                                                      01062000 T  0009
     IF INPUT [3] NEQ 10 OR INPUT [4] MOD 30 NEQ 0 THEN                                  01063500 T  0009
```

```
    ERROR (NEXT, 3, PARAMETER1, PARAMETER2) ;                                  01064000 T  0012
  IF NUM3 THEN                                                                 01064500 T  0014
  BEGIN                                                                        01065000 T  0015
    I := PARAMETER3 - 1 ;                                                      01065500 T  0016
    IF I GTR INPUT [5] THEN                                                    01066000 T  0017
      ERROR (NEXT, 0, "USE REC", "ORD #S,") ;                                  01066500 T  0018
    IF NUM4 THEN                                                               01067000 T  0023
      M := MIN (PARAMETER4 - 1, INPUT [5])                                     01067500 T  0023
    ELSE M := I ;                                                              01070000 T  0025
  END                                                                          01070500 T  0029
  ELSE                                                                         01071000 T  0029
  BEGIN                                                                        01071500 T  0029
    I := 0 ;                                                                   01072000 T  0029
    M := INPUT [5] ;                                                           01072500 T  0030
    IF DATAFILE AND MERGE := PARAMETER3 = "MERGE  " THEN                       01072600 T  0031
      ERROR (NEXT, 5, " TYPE: ", PARAMETER3) ;                                 01072700 T  0033
  END ;                                                                        01073000 T  0038
  WAIT (M = I, FALSE) ;                                                        01073500 T  0038
  READ SEEK (LIBRARY [I]) ;                                                    01075000 T  0041
  B := NOT (COPYCLOBBER EQV TEMPTOG) ;                                         01075100 T  0046
  FOR I := I STEP 1 UNTIL M DO                                                 01075500 T  0049
  BEGIN                                                                        01076000 T  0050
    READ (LIBRARY, 10, IMAGE [*]) ;                                            01076100 T  0050
    IF MERGE THEN                                                              01076200 T  0054
      N := IF COBOLFILE THEN DEC (IMAGE [0], 6)                                01076300 T  0054
                        ELSE DEC (IMAGE [9], 8) ;                              01076400 T  0055
      IF ITSOLD (N) AND B THEN                                                 01076500 T  0059
        ERROR (NEXT, 0, "OVERITE", " IS OFF") ;                                01076600 T  0061
      WDISC ;                                                                  01077000 T  0066
      INTERUPT (1, 1, I + 1) ;                                                 01078000 T  0068
    END ;                                                                      01078500 T  0074
NEXT:                                                                          01079500 T  0076
    CLOSE (LIBRARY) ;                                                          01079600 T  0077
  END ;                                                                        01080000 T  0078
                                                                         39 IS   81 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     01086500 T  0051
PROCEDURE ZIPIT ;                                                              01087000 T  0051
  BEGIN                                                                        01087500 T  0051
ALPHA STREAM PROCEDURE ENDCHECK (S) ;                                          01087550 T  0051
                                                                       START OF SEGMENT **********   40
BEGIN                                                                          01087600 T  0000
  SI := S ;                                                                    01087650 T  0000
  IF SC = "?" THEN                                                             01087700 T  0000
  BEGIN                                                                        01087710 T  0000
    DI := LOC ENDCHECK ;                                                       01087750 T  0000
    DI := DI + 3 ;                                                             01087800 T  0001
    DS := CHR ;                                                                01087810 T  0001
    63 (IF SC NEQ " " THEN JUMP OUT ;                                          01087850 T  0002
        SI := SI + 1) ;                                                        01087900 T  0003
    4 (IF SC = ALPHA THEN DS := 1 CHR ELSE JUMP OUT) ;                         01087950 T  0004
  END ;                                                                        01087960 T  0006
END ENDCHECK ;                                                                 01088000 T  0006
LABEL NEXT ;                                                                   01088050 T  0007
    READONLYCHECK ;                                                            01088100 T  0007
    RDISC (FIRST . T, RECORD) ;                                                01089500 T  0008
    IF ENDCHECK (RECORD) = 0 THEN                                              01090000 T  0012
      ERROR (NEXT, 0, "INV FIR", "ST CARD") ;                                  01090500 T  0014
```

```
RDISC (LAST , F, IMAGE) ;                                                    01092000 T  0019
IF ENDCHECK (IMAGE) NEQ "?ENDO" THEN                                         01093100 T  0022
  ERROR (NEXT, 0, "NO END ", "CARD.  ") ;                                    01093200 T  0024
WAIT (KOUNT (1, FINITY, CLOCK) × 2, XLOCKED) ;                               01095500 T  0030
FILL LIBRARY WITH PREFIX, SUFFIX ;                                           01096000 T  0034
READ SEEK (LIBRARY [M := (AT := FIRST.T) - 2]) ;                             01096500 T  0038
I := 0 ;                                                                     01096600 T  0047
WHILE AT := LL [AT] . T NEQ 1 DO                                             01097000 T  0047
BEGIN                                                                        01097500 T  0053
  RDISC (AT, IMAGE) ;                                                        01098000 T  0053
  I := I + 1 ;                                                               01098100 T  0055
  IF ENDCHECK (IMAGE) NEQ 0 THEN                                             01098500 T  0056
  BEGIN                                                                      01099000 T  0058
    RECORD [9] := I ;                                                        01100000 T  0059
    WRITE (LIBRARY, 10, RECORD [*]) ;                                        01100500 T  0060
    IF M + 1 NEQ M := AT - 2 THEN                                            01100510 T  0064
      READ SEEK (LIBRARY [M]) ;                                              01100600 T  0067
    READ (IMAGE [*], 10, RECORD [*]) ;                                       01101600 T  0073
  END ;                                                                      01102000 T  0075
  INTERUPT (1, 2, M) ;                                                       01102500 T  0075
END ;                                                                        01103000 T  0080
IMAGE [9] := I ;                                                             01104000 T  0083
WRITE (LIBRARY, 10, IMAGE [*]) ;                                             01104500 T  0084
CLOSE (LIBRARY) ;                                                            01104600 T  0088
IF NOT EMPTY1 THEN                                                           01105000 T  0090
BEGIN                                                                        01105500 T  0091
  PARAMETER3 := 1 ;                                                          01106100 T  0091
  PARAMETER4 := FINITY ;                                                     01106200 T  0092
  THERMOFAX (8, (D + 14) DIV 15 × 15) ;                                      01107500 T  0093
END                                                                          01108500 T  0100
ELSE                                                                         01109000 T  0100
BEGIN                                                                        01109500 T  0100
  FILEINFO := DATA ;                                                         01109600 T  0103
  CLOSEFILE ;                                                                01110000 T  0103
  ZIP WITH DISC ;                                                            01111000 T  0104
END ;                                                                        01111500 T  0105
NEXT:                                                                        01111600 T  0105
  CLOSE (LIBRARY) ;                                                          01111700 T  0106
END ;                                                                        01112000 T  0107
                                                                      40 IS  108 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     01113000 T  0051
DEFINE CLOSEIT =                                                             01113500 T  0051
  BEGIN                                                                      01113600 T  0051
    OPENCHECK ;                                                              01113800 T  0051
    CLOSEFILE ;                                                              01114000 T  0051
  END# ;                                                                     01116500 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     01117000 T  0051
PROCEDURE OPEN ;                                                             01117500 T  0051
  BEGIN                                                                      01118000 T  0051
LABEL NEXT ;                                                                 01118500 T  0051
                                                                  START OF SEGMENT **********   41
    IF FILEOPEN THEN CLOSEFILE ;                                             01119000 T  0000
    TABAMOUNT := 0 ;                                                         01119500 T  0002
    PREWHERE := - 1 ;                                                        01121000 T  0002
    IF FILEINFO := FILETYPE (PARAMETER3) - 0 THEN                            01122000 T  0003
      ERROR (NEXT, 5, " TYPE: ", PARAMETER3) ;                               01126000 T  0005
    I := XFILE (12, 0, -1) ;                                                 01126100 T  0009
```

```
FILL DISC WITH PREFIX := PARAMETER1, SUFFIX := PARAMETER2 ;                    01126200 T  0011
IF PARAMETER4 = "NEW     " THEN                                                01126500 T  0015
BEGIN                                                                          01127000 T  0016
  IF I GEQ 0 THEN                                                              01128500 T  0017
    ERROR (NEXT, 4, PARAMETER1, PARAMETER2) ;                                  01129000 T  0017
  CREATEFILE (450) ;                                                           01130000 T  0021
  FILEACCESS := 7 ;                                                            01130500 T  0040
  FIRST := D := 1 ;                                                            01130650 T  0041
  LAST := 1 & INFINITY [SF] ;                                                  01130700 T  0044
  MODIFIED := TRUE ;                                                           01130750 T  0047
  N := 0 ;                                                                     01130800 T  0047
  INORDER := FALSE ;                                                           01130850 T  0048
  GO TO NEXT ;                                                                 01130900 T  0050
END ;                                                                          01131000 T  0055
IF I LEQ 0 THEN                                                                01133500 T  0055
  ERROR (NEXT, 1 - I, PARAMETER1, PARAMETER2) ;                                01134000 T  0055
IF INPUT [3] NEQ 10 OR INPUT [4] MOD 30 NEQ 0 THEN                             01134500 T  0058
  ERROR (NEXT, 3, PARAMETER1, PARAMETER2) ;                                    01135000 T  0061
IF INPUT [6] NEQ 0 THEN                                                        01135500 T  0063
  ERROR (NEXT, 0, "FILE IN", " USE.  ") ;                                      01136000 T  0064
IF D := INPUT [5] + 2 GTR MAXFILELENGTH THEN                                   01146000 T  0069
  ERROR (NEXT, 0, "FILE TO", " LONG. ") ;                                      01147500 T  0071
IF PARAMETER4 = "OLD     " OR DATAFILE THEN                                    01155000 T  0077
BEGIN                                                                          01155500 T  0078
  INORDER := DATAFILE OR READONLYFILE ;                                        01156000 T  0079
  N := 0 ;                                                                     01157500 T  0082
  FOR AT := 2 STEP 1 UNTIL D DO                                                01157600 T  0083
    LL [AT] := (AT+1) & (N:=N+INC)[SF] & (AT=1)[FF] ;                          01157700 T  0085
END ELSE                                                                       01158000 T  0095
BEGIN                                                                          01158500 T  0095
  WAIT (D, FALSE) ;                                                            01158600 T  0095
  M := 0 ;                                                                     01159000 T  0098
  FOR AT := 2 STEP 1 UNTIL D DO                                                01160000 T  0099
  BEGIN                                                                        01160500 T  0100
    READ (LIBRARY, 10, IMAGE [*]) ;                                            01161000 T  0100
    N := IF COBOLFILE THEN DEC (IMAGE [0], 6)                                  01161500 T  0104
                      ELSE DEC (IMAGE [9], 8) ;                                01162000 T  0104
    IF M GTR N THEN                                                            01164500 T  0109
      ERROR (NEXT, 0, "SEQERR ", OCTDEX (M)) ;                                 01166000 T  0109
    LL [AT] := (AT+1) & (M:=N)[SF] & (AT=1)[FF] ;                              01167500 T  0114
    INTERUPT (1, 2, AT = 1) ;                                                  01167600 T  0121
  END ;                                                                        01168000 T  0127
  FILEACCESS := I ;                                                            01168100 T  0130
  MODIFIED := NOT FALSE ;                                                      01168200 T  0130
  LL [0] . T := 1 ;                                                            01168210 T  0130
  FIRST := 2 ;                                                                 01168220 T  0131
  LAST := 1 & INFINITY [SF] & D [FF] ;                                         01168230 T  0137
  LL [2] . F := 0 ;                                                            01168240 T  0139
NEXT:                                                                          01168250 T  0143
  CLOSE (LIBRARY) ;                                                            01168500 T  0148
  N := N + INC ;                                                               01168600 T  0149
  AT := 0 ;                                                                    01169000 T  0150
  IF READONLYFILE THEN                                                         01169100 T  0152
    ERRORX (7, "READ ON", "LY FILE") ;                                         01169500 T  0152
END ;                                                                          01170000 T  0153
                                                                               01171000 T  0155
```

41 IS 159 LONG, NEXT SEG 2

```
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *          01171500 T  0051
DEFINE INCREMENT =                                                           01172000 T  0051
   BEGIN                                                                     01172100 T  0051
     IF NOT NUM1 THEN                                                        01172500 T  0051
       ERRORX (7, PARAMETER0, OCTDEX (INC))                                  01173000 T  0051
     ELSE INC := PARAMETER1 ;                                                01173500 T  0051
   END# ;                                                                    01174000 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *          01174500 T  0051
PROCEDURE RESEQ ;                                                            01175000 T  0051
   BEGIN                                                                     01175500 T  0051
REAL L ;                                                                     01175600 T  0051
                                                          START OF SEGMENT **********    42
LABEL NEXT ;                                                                 01176000 T  0000
     OPENCHECK ;                                                             01176100 T  0000
     IF NUM2 THEN                                                            01176500 T  0001
     BEGIN                                                                   01177000 T  0002
       IF NOT NUM1 THEN                                                      01177500 T  0002
         ERROR (NEXT, 0, PARAMETER1, "INVALID") ;                            01178000 T  0003
       IF NUM4 THEN                                                          01178500 T  0007
         INC := PARAMETER4 ;                                                 01179000 T  0007
       IF NUM3 THEN                                                          01179500 T  0009
         M := PARAMETER3 - INC                                               01180000 T  0009
       ELSE M := PARAMETER1 - INC ;                                          01180500 T  0010
       IF M + INC × KOUNT (PARAMETER1,PARAMETER2,-1) GEQ LL [AT].S THEN      01180600 T  0013
         ERROR (NEXT, 0, PARAMETER0, " ERROR.") ;                            01180700 T  0018
       AT := LOC (PARAMETER1) ;                                              01181000 T  0023
       IF M + INC LEQ LL [LL [AT].F].S THEN                                  01181010 T  0024
         ERROR (NEXT, 0, PARAMETER0, " ERROR.") ;                            01181020 T  0033
       N := M ;                                                              01181500 T  0037
       WHILE (L := LL [AT]).S LEQ PARAMETER2 DO                              01182000 T  0037
       BEGIN                                                                 01182500 T  0043
         LL [AT] . S := N := N + INC ;                                       01183000 T  0043
         MODIFY (AT) ;                                                       01183500 T  0049
         AT := L.T ;                                                         01183600 T  0052
       END ;                                                                 01184000 T  0053
     END                                                                     01185500 T  0054
     ELSE                                                                    01186000 T  0054
     BEGIN                                                                   01186500 T  0054
       IF NUM1 THEN                                                          01187000 T  0054
         INC := PARAMETER1 ; ;                                               01187500 T  0055
       IF INC × KOUNT (1, FINITY, -1) GEQ INFINITY THEN                      01187600 T  0056
         ERROR (NEXT, 0, PARAMETER0, " ERROR.") ;                            01187700 T  0059
       N := 0 ;                                                              01188000 T  0065
       AT := 0 ;                                                             01188500 T  0065
       WHILE AT := LL [AT] . T NEQ 1 DO                                      01189000 T  0066
         LL [AT] . S := N := N + INC ;                                       01189500 T  0072
       MODIFIED := NOT FALSE ;                                               01189600 T  0079
     END ;                                                                   01190000 T  0080
     N := N + INC ;                                                          01190500 T  0080
     IF NOT DATAFILE THEN                                                    01191000 T  0081
       INORDER := READONLYFILE ;                                             01191500 T  0082
NEXT:                                                                        01192000 T  0084
   END RESEQ ;                                                               01192500 T  0085
                                                            42 IS   88 LONG, NEXT SEG     2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *          01193000 T  0051
DEFINE TAB =                                                                 01193500 T  0051
   BEGIN                                                                     01193600 T  0051
```

```
      IF NOT NUM1 THEN                                                                  01194000 T  0051
      BEGIN                                                                             01194010 T  0051
        IF NOT EMPTY1 THEN                                                              01194100 T  0051
          TABON := TOGGLE (TABON, 1)                                                    01194300 T  0051
        ELSE                                                                            01194450 T  0051
          ERRORX (7, PARAMETER0, ONOFF (TABON) &                                        01194500 T  0051
            OCTDEX (IF COBOLFILE THEN TABAMOUNT + 7 ELSE TABAMOUNT + 1)                 01194600 T  0051
            [36:36:12]) ;                                                               01194700 T  0051
      END ELSE                                                                          01194800 T  0051
      BEGIN                                                                             01194900 T  0051
        IF RELATIVENUMBER.[2:2] NEQ 0 THEN                                              01194910 T  0051
          PARAMETER1 := TABAMOUNT + 1 +                                                 01194920 T  0051
            (RELATIVENUMBER & RELATIVENUMBER[1:3:3])                                    01194930 T  0051
        ELSE IF COBOLFILE THEN                                                          01194940 T  0051
          PARAMETER1 := PARAMETER1 - 6 ;                                                01194950 T  0051
        IF TABAMOUNT := PARAMETER1 GTR 55 THEN                                          01195000 T  0051
          TABAMOUNT := 55 ;                                                             01195500 T  0051
        IF TABAMOUNT := TABAMOUNT - 1 LSS 0 THEN                                        01197000 T  0051
          TABAMOUNT := 0 ;                                                              01197500 T  0051
      END ;                                                                             01197600 T  0051
    END#,                                                                               01198000 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     01198500 T  0051
 SAVEIT =                                                                               01199000 T  0051
   BEGIN                                                                                01199100 T  0051
     IF NOT NUM1 THEN                                                                   01199500 T  0051
       ERRORX (7, PARAMETER0, OCTDEX (SAVEFACTOR))                                      01200000 T  0051
     ELSE SAVEFACTOR := PARAMETER1 ;                                                    01200500 T  0051
   END# ;                                                                               01201000 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     01211500 T  0051
PROCEDURE COMPILE ;                                                                     01212000 T  0051
  BEGIN                                                                                 01212500 T  0051
LABEL NEXT ;                                                                            01213000 T  0051
                                                                  START OF SEGMENT **********    43
     OPENCHECK ;                                                                        01213100 T  0000
     IF EMPTY2 THEN                                                                     01213500 T  0001
       ERROR (NEXT, 3, PARAMETER1, PARAMETER2) ;                                        01214000 T  0002
     IF DATAFILE AND EMPTY3 THEN                                                        01216000 T  0004
       ERROR (NEXT, 3, PREFIX, SUFFIX) ;                                                01216500 T  0006
     IF NOT EMPTY3 THEN                                                                 01217000 T  0008
       IF XFILE (PARAMETER3, "DISK   ", 2) LSS 2 THEN                                   01218000 T  0009
         GO TO NEXT ;                                                                   01218500 T  0011
     IF PARAMETER0 := XFILE ("LINE   ", USERCODE, -1) = 7 THEN                          01221500 T  0012
     BEGIN                                                                              01223000 T  0014
       READ (LIBRARY) ;                                                                 01223500 T  0015
       DETACH ;                                                                         01224000 T  0019
       CLOSE (LIBRARY, PURGE) ;                                                         01224500 T  0020
       REATTACH ;                                                                       01225000 T  0022
     END ELSE                                                                           01225500 T  0023
     IF PARAMETER0 GEQ 0 THEN                                                           01226000 T  0023
       ERROR (NEXT, 4, "LINE   ", USERCODE) ;                                           01226500 T  0026
     IF XFILE (12, 0, -1) GEQ 0 THEN                                                    01227000 T  0030
       ERROR (NEXT, 4, PARAMETER1, PARAMETER2) ;                                        01228500 T  0032
     CLOSEFILE ;                                                                        01229000 T  0034
     IF EMPTY3 THEN                                                                     01230000 T  0035
       IF COMPILER = ALGOL THEN                                                         01230500 T  0035
         PARAMETER3 := "ALGOL  "                                                        01231000 T  0037
       ELSE IF COMPILER = FORTRAN THEN                                                  01231500 T  0037
```

```
        PARAMETER3 := "FORTRAN"                                                  01232000 T  0040
      ELSE IF COMPILER = XALGOL THEN                                             01232500 T  0041
        PARAMETER3 := "XALGOL "                                                  01233000 T  0044
      ELSE IF COMPILER = BASIC THEN                                              01233500 T  0045
        PARAMETER3 := "BASIC  "                                                  01234000 T  0048
      ELSE                                                                       01234500 T  0049
        PARAMETER3 := "COBOL  " ;                                                01235000 T  0050
    WRITE (ZIPPY [*], ZIPPER, PARAMETER1,[6:6],                                  01238000 T  0052
      PARAMETER1, PARAMETER2,[6:6], PARAMETER2, PARAMETER3,[6:6],                01238500 T  0061
      PARAMETER3, PREFIX,[6:6], PREFIX, SUFFIX,[6:6], SUFFIX,                    01239000 T  0068
      USERCODE , [6 : 6], USERCODE) ;                                            01239500 T  0077
    ZIP WITH ZIPPY [*] ;                                                         01240000 T  0084
NEXT:                                                                            01242000 T  0085
  END COMPILE ;                                                                  01242500 T  0086
                                                                          43 IS   87 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *     01243000 T  0051
PROCEDURE DITTO ;                                                                01243500 T  0051
  BEGIN                                                                          01244000 T  0051
BOOLEAN B ;                                                                      01244100 T  0051
                                                                        START OF SEGMENT **********   44
REAL L ;                                                                         01244200 T  0000
PROCEDURE LINK (X, Y) ; VALUE X, Y ; INTEGER X, Y ;                              01244300 T  0000
BEGIN                                                                            01244310 T  0000
  LL [X].T := Y ;                                                                01244320 T  0000
  MODIFY (X) ;                                                                   01244330 T  0005
  LL [Y].F := X ;                                                                01244340 T  0008
  MODIFY (Y) ;                                                                   01244350 T  0013
END LINK ;                                                                       01244360 T  0016
LABEL NEXT ;                                                                     01244500 T  0016
    IF PARAMETER1 = "OVERITE" THEN                                               01244550 T  0016
    BEGIN                                                                        01244600 T  0017
      DITTOCLOBBER := TOGGLE (DITTOCLOBBER, 2) ;                                 01244700 T  0018
      GO TO NEXT ;                                                               01244850 T  0021
    END ;                                                                        01244900 T  0023
    READONLYCHECK ;                                                              01245000 T  0023
    IF NOT NUM1 THEN                                                             01246000 T  0024
      ERROR (NEXT, 0, PARAMETER0, " ERROR,") ;                                   01246500 T  0025
    IF PARAMETER2 = "MOVE   " OR PARAMETER3 = "MOVE   " THEN                     01246510 T  0029
    BEGIN                                                                        01246520 T  0030
      IF NOT NUM2 THEN                                                           01246530 T  0031
        PARAMETER2 := PARAMETER1 ;                                               01246540 T  0032
      B := ITSOLD (N) ;                                                          01246550 T  0033
      PARAMETER4 := LL [PARAMETER3 := AT] ;                                      01246560 T  0035
      M := LL [I := LOC (PARAMETER1)].F ;                                        01246570 T  0039
      IF PARAMETER0 := KOUNT (PARAMETER1,PARAMETER2,"1) - 1 LSS 0 THEN           01246580 T  0045
        GO TO NEXT ;                                                             01246590 T  0047
      IF ITSOLD (PARAMETER2) THEN                                                01246600 T  0048
        L := LL [AT].T                                                           01246610 T  0050
      ELSE                                                                       01246620 T  0053
        AT := LL [L := AT].F ;                                                   01246630 T  0054
      IF (B AND B := LL [M].S GEQ N OR N GEQ LL [L].S) OR                        01246640 T  0061
        N+INC*PARAMETER0 GEQ (IF B THEN PARAMETER4 ELSE LL [L]).S THEN           01246650 T  0071
        ERROR (NEXT, 0, "NO ROOM", "! MOVE ") ;                                  01246660 T  0078
      IF B THEN                                                                  01246670 T  0083
      BEGIN                                                                      01246680 T  0083
        LINK (M, L) ;                                                            01246690 T  0083
        LINK (AT, PARAMETER3) ;                                                  01246700 T  0084
```

```
    LINK (PARAMETER4.F, I) ;                                                    01246710 T  0085
  END ELSE                                                                      01246720 T  0087
    PARAMETER3 := L ;                                                           01246730 T  0087
  DO BEGIN                                                                      01246740 T  0088
    LL [I].S := N ;                                                             01246750 T  0089
    N := N + INC ;                                                              01246760 T  0094
    MODIFY (I) ;                                                                01246770 T  0095
  END UNTIL I := LL [I].T = PARAMETER3 ;                                        01246780 T  0098
  INORDER := FALSE ;                                                            01246790 T  0103
  GO TO NEXT ;                                                                  01246800 T  0105
END ;                                                                           01246810 T  0105
CLOSE (DISC) ;                                                                  01247000 T  0105
PREWHERE := PARAMETER3 := "1 ;                                                  01247100 T  0107
IF NUM2 THEN                                                                    01247500 T  0109
  WAIT (KOUNT (PARAMETER1, PARAMETER2, CLOCK), FALSE)                           01248500 T  0109
ELSE PARAMETER2 := PARAMETER1 ;                                                 01250000 T  0109
FILL LIBRARY WITH PREFIX, SUFFIX ;                                              01250500 T  0115
I := LOC (PARAMETER1) ;                                                         01251600 T  0119
M := D ;                                                                        01252000 T  0120
B := NOT (DITTOCLOBBER EQV TEMPTOG) ;                                           01252100 T  0121
WHILE (L := LL [I].S LEQ PARAMETER2 AND I LEQ M DO                              01252500 T  0124
BEGIN                                                                           01253500 T  0130
  IF PARAMETER3 + 1 NEQ PARAMETER3 := I THEN                                    01254000 T  0130
    READ SEEK (LIBRARY [I * 2]) ;                                               01254500 T  0132
  IF ITSOLD (N) AND B THEN                                                      01255000 T  0138
    ERROR (NEXT, 0, "OVERITE", " IS OFF") ;                                     01255500 T  0140
  I := L.T ;                                                                    01256000 T  0145
  READ (LIBRARY, 10, IMAGE [*]) ;                                               01256500 T  0146
    WDISC ;                                                                     01257000 T  0150
    INTERUPT (1, 2, I * 2) ;                                                    01257500 T  0152
  END ;                                                                         01258000 T  0158
NEXT:                                                                           01259500 T  0159
    CLOSE (LIBRARY) ;                                                           01260000 T  0160
  END DITTO ;                                                                   01261000 T  0161
                                                                          44 IS  164 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    01261500 T  0051
PROCEDURE REMOVE ;                                                              01262000 T  0051
  BEGIN                                                                         01262500 T  0051
LABEL NEXT ;                                                                    01263000 T  0051
                                                                      START OF SEGMENT **********   45
    IF EMPTY2 AND PARAMETER1 = "LISTING" THEN                                   01263100 T  0000
    BEGIN                                                                       01263200 T  0001
      PARAMETER1 := "LINE    " ;                                                01263300 T  0002
      PARAMETER2 := USERCODE ;                                                  01263400 T  0003
    END ;                                                                       01263500 T  0003
    IF XFILE (12, 0, 4) LSS 4 THEN                                              01263600 T  0003
      GO TO NEXT ;                                                              01263700 T  0005
    IF PARAMETER1 = PREFIX THEN                                                 01264000 T  0006
      IF PARAMETER2 = SUFFIX AND READWRITEFILE THEN                             01264500 T  0006
      BEGIN                                                                     01265000 T  0009
        READ (DISC [0]) ;                                                       01265500 T  0009
        DETACH ;                                                                01266000 T  0014
        CLOSE (DISC, PURGE) ;                                                   01266500 T  0015
        REATTACH ;                                                              01267500 T  0017
        FILEACCESS := 0 ;                                                       01268000 T  0018
        INORDER := TRUE ;                                                       01268500 T  0019
        GO TO NEXT ;                                                            01269000 T  0021
```

```
      END ;                                                                                    01269500 T  0024
    IF INPUT [6] NEQ 0 THEN                                                                    01273000 T  0024
      ERROR (NEXT, 0, "FILE IN", " USE.  ") ;                                                  01273500 T  0025
    READ (LIBRARY) ;                                                                           01274000 T  0030
    DETACH ;                                                                                   01274500 T  0034
    CLOSE (LIBRARY, PURGE) ;                                                                   01275000 T  0035
    REATTACH ;                                                                                 01275500 T  0036
NEXT:                                                                                          01276000 T  0037
  END REMOVE ;                                                                                 01276500 T  0038
                                                                                        45 IS   39 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *   01277000 T  0051
PROCEDURE LISTING ;                                                                            01277200 T  0051
  BEGIN                                                                                        01277400 T  0051
BOOLEAN LOCKED ;                                                                               01277500 T  0051
                                                                                      START OF SEGMENT **********   46
INTEGER P5 ;                                                                                   01277510 T  0000
LABEL NEXT ;                                                                                   01277600 T  0000
FILE LINE 15 (2, 15) ;                                                                         01277800 T  0000
FILE FEEDBACK DISK SERIAL (2, 15, 30) ;                                                        01278000 T  0003
REAL STREAM PROCEDURE READZ (Z, SKP, A, N) ;                                                   01278200 T  0007
VALUE SKP, A, N ;                                                                              01278210 T  0007
BEGIN                                                                                          01278220 T  0007
LABEL EXIT ;                                                                                   01278230 T  0008
  SI := Z ;                                                                                    01278240 T  0008
  SI := SI + SKP ;                                                                             01278250 T  0008
  DI := LOC READZ ;                                                                            01278260 T  0008
  A (DI := DI + 8 ; DI := DI - N ; DS := N CHR ; JUMP OUT TO EXIT) ;                           01278270 T  0009
  DS := N OCT ;                                                                                01278280 T  0012
EXIT:                                                                                          01278290 T  0012
END READZ ;                                                                                    01278300 T  0012
    IF XFILE ("LINE   ", USERCODE, 1) LSS 1 THEN                                               01278400 T  0014
      GO TO NEXT ;                                                                             01278600 T  0015
    IF NOT EMPTY1 THEN                                                                         01279200 T  0016
      IF I := FILETYPE (PARAMETER1) = 0 OR I = DATA THEN                                       01279400 T  0017
        ERROR (NEXT, 5, "TYPE:  ", PARAMETER1) ;                                               01281600 T  0020
    WAIT (INPUT [5], YLOCKED) ;                                                                01281800 T  0025
    YLOCKED := LOCKED := TRUE ;                                                                01282000 T  0028
    FILL FEEDBACK WITH "LINE   ", USERCODE ;                                                   01282200 T  0030
    IF NUM2 AND NUM3 AND NUM4 THEN                                                             01282400 T  0034
    BEGIN                                                                                      01282600 T  0037
      PARAMETER1 := 1 ;                                                                        01282800 T  0037
      WRITESEGMENT ;                                                                           01283200 T  0038
      PARAMETER0 := IF I=FORTRAN THEN 10 ELSE REAL(I GEQ ALGOL)+12 ;                           01283300 T  0039
      P5 := IF I GEQ ALGOL THEN PARAMETER0 - 1 ELSE 0 ;                                        01283310 T  0043
    END                                                                                        01283600 T  0046
    ELSE IF PARAMETER2.[6:30] = "ERROR" OR PARAMETER2                                          01283800 T  0046
      = "SYNTAX " THEN                                                                         01284000 T  0049
    BEGIN                                                                                      01284100 T  0050
      PARAMETER1 := 2 ;                                                                        01284200 T  0050
      P5 := IF I=FORTRAN THEN 9 ELSE 12 ;                                                      01284300 T  0051
    END                                                                                        01284310 T  0054
    ELSE IF EMPTY2 THEN                                                                        01284400 T  0054
    BEGIN                                                                                      01284600 T  0057
      FILL LINE WITH "LINE   ", USERCODE ;                                                     01284800 T  0058
      DETACH ;                                                                                 01285000 T  0062
      WRITE (LINE) ;                                                                           01285200 T  0063
      REATTACH ;                                                                               01285400 T  0067
```

```
    PARAMETER1 := 3 ;                                                                    01285600 T  0068
  END                                                                                    01285800 T  0068
  ELSE                                                                                   01286000 T  0068
    ERROR (NEXT, 0, PARAMETER0, " ERROR.") ;                                             01286200 T  0068
  DO BEGIN                                                                               01286400 T  0074
    READ (FEEDBACK, 15, ZIPPY [*]) [NEXT] ;                                              01287000 T  0074
    IF PARAMETER1 = 1 THEN                                                               01287200 T  0079
    BEGIN                                                                                01287400 T  0080
      IF I = FORTRAN THEN                                                                01287600 T  0080
      BEGIN                                                                              01287800 T  0081
        IF N := READZ (ZIPPY [11], 4, 1, 4) NEQ "LONG" THEN                              01288000 T  0081
          IF N := READZ (ZIPPY [11], 3, 0, 4) NEQ 0 THEN                                 01288200 T  0085
            M := N ;                                                                     01288400 T  0088
      END                                                                                01288800 T  0090
      ELSE IF I GEQ ALGOL THEN                                                           01289000 T  0090
      BEGIN                                                                              01289200 T  0092
        IF N := READZ (ZIPPY [14], 4, 0, 4) NEQ 0 THEN                                   01289400 T  0093
          M := N ;                                                                       01289600 T  0096
      END                                                                                01289800 T  0097
      ELSE M := READZ (ZIPPY [12], N := 0, 0, 4) ;                                       01293000 T  0097
      IF M = PARAMETER2 AND N = 0 THEN                                                   01293200 T  0101
      BEGIN                                                                              01293400 T  0103
        N := READZ (ZIPPY [PARAMETER0], REAL (I=COBOL) + 4, 0, 4) ;                      01293600 T  0103
        IF N GTR PARAMETER4 THEN                                                         01294600 T  0107
          GO TO NEXT ;                                                                   01294800 T  0108
        IF PARAMETER3 LEQ N THEN                                                         01295000 T  0108
        BEGIN                                                                            01295200 T  0109
          PARAMETER3 := N ;                                                              01295400 T  0110
          N := READZ (ZIPPY [P5], IF I=COBOL THEN 3 ELSE 0, 0, 8) ;                      01295600 T  0110
          WRITESEQ ;                                                                     01296600 T  0115
          WRITERELADDR ;                                                                 01297000 T  0116
        END ;                                                                            01297200 T  0117
      END ;                                                                              01297400 T  0117
    END                                                                                  01297600 T  0117
    ELSE IF PARAMETER1 = 2 THEN                                                          01297800 T  0117
    BEGIN                                                                                01298000 T  0118
      IF I = COBOL THEN                                                                  01298200 T  0118
      BEGIN                                                                              01298400 T  0119
        M := READZ (ZIPPY [0], 0, 1, 1) ;                                                01298600 T  0120
        IF M = " " OR M = "[" THEN                                                       01298800 T  0122
          M := READZ (ZIPPY [0], 5, 0, 6)                                                01299000 T  0124
        ELSE M := 0 ;                                                                    01299200 T  0126
      END                                                                                01299400 T  0129
      ELSE M := READZ (ZIPPY [P5], 0, 0, 8) ;                                            01300000 T  0129
      IF M = 0 AND N GTR 0 THEN                                                          01300200 T  0132
      BEGIN                                                                              01301200 T  0134
        WRITESEQ ;                                                                       01301600 T  0134
        WRITEROW (ZIPPY, TRUE, DATA) ;                                                   01301800 T  0135
      END ELSE N := M ;                                                                  01302000 T  0137
    END                                                                                  01302200 T  0138
    ELSE                                                                                 01302400 T  0138
      WRITE (LINE [DBL], 15, ZIPPY [*]) ;                                                01302600 T  0138
    INTERRUPT (1) ;                                                                      01302800 T  0143
  END UNTIL BOOLEAN (BREAKI) ;                                                           01303000 T  0148
NEXT:                                                                                    01303200 T  0150
  N := RESETN ;                                                                          01303400 T  0151
  IF LOCKED THEN                                                                         01303500 T  0151
```

```
      YLOCKED := FALSE ;                                                                    01303600 T  0152
   END LISTING ;                                                                            01303800 T  0154
                                                                                          46 IS  158 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    01345000 T  0051
PROCEDURE INLINE ;                                                                        01345500 T  0051
   BEGIN                                                                                  01345550 T  0051
LABEL NEXT ;                                                                              01345560 T  0051
                                                                                        START OF SEGMENT **********   47
DEFINE QUICK = FALSE# ;                                                                   01345570 T  0000
     IF PARAMETER1 = "ECHO    " THEN                                                      01345600 T  0000
     BEGIN                                                                              01345610 T  0000
       INLINEECHO := TOGGLE (INLINEECHO, 2) ;                                           01345620 T  0001
       GO TO NEXT ;                                                                     01345660 T  0004
     END ;                                                                              01345670 T  0006
     READONLYCHECK ;                                                                    01345700 T  0006
     IF NUM1 THEN                                                                       01346000 T  0007
     BEGIN                                                                              01346500 T  0008
       N := PARAMETER1 ;                                                                01347000 T  0008
       IF NOT ITSOLD (N) THEN                                                           01347500 T  0009
         ERROR (NEXT, 0, "MISSING", OCTDEX (N)) ;                                       01348000 T  0011
       IF NOT MOREINPUT THEN                                                            01348500 T  0016
         WRITEAT ;                                                                      01349000 T  0017
       I := PARAMETER2,[6:6] ;                                                          01349500 T  0020
     END                                                                                01350000 T  0021
     ELSE                                                                               01350500 T  0021
     BEGIN                                                                              01351000 T  0021
       AT := LL [LOC (N)].F ;                                                           01351500 T  0022
       N := LL [AT].S ;                                                                 01352500 T  0027
       I := PARAMETER1,[6:6] ;                                                          01353000 T  0031
     END ;                                                                              01353500 T  0032
     IF NOT NUM1 OR MOREINPUT THEN                                                      01353600 T  0032
       RDISC (AT, RECORD) ;                                                             01354000 T  0034
     INLINETOG := TRUE ;                                                                01354500 T  0037
     IF I = "I" THEN                                                                    01356000 T  0039
       M := 1                                                                           01356500 T  0040
     ELSE IF I = "D" THEN                                                               01357000 T  0040
       M := 2                                                                           01357500 T  0042
     ELSE IF I = "R" THEN                                                               01358000 T  0043
       M := 3                                                                           01358500 T  0045
     ELSE M := 0 ;                                                                      01359000 T  0045
NEXT:                                                                                   01359500 T  0047
   END INLINE ;                                                                         01359600 T  0048
                                                                                        47 IS   49 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *    01360000 T  0051
PROCEDURE COLUMN ;                                                                      01360100 T  0051
     BEGIN                                                                              01360150 T  0051
       INTEGER STREAM PROCEDURE GETCHAR (S) ;                                           01360200 T  0051
                                                                                        START OF SEGMENT **********   48
       BEGIN                                                                            01360250 T  0000
        LABEL NOPE, YES, XIT ;                                                          01360300 T  0000
            DI := LOC GETCHAR ;                                                         01360350 T  0000
            SI := S ;                                                                   01360400 T  0000
            2(40(IF SC = ALPHA THEN ELSE IF SC = " " THEN ELSE                          01360450 T  0000
                 IF SC = """ THEN                                                       01360500 T  0002
                   JUMP OUT 2 TO YES                                                    01360550 T  0003
                 ELSE IF SC = "," THEN                                                  01360600 T  0003
```

```
                    JUMP OUT 2 TO YES                                                    01360650 T  0004
                  ELSE IF SC = "(" THEN                                                  01360700 T  0005
                    JUMP OUT 2 TO YES                                                    01360750 T  0006
                  ELSE IF SC = "[" THEN                                                  01360800 T  0006
                    JUMP OUT 2 TO YES                                                    01360850 T  0007
                  ELSE IF SC = ";" THEN                                                  01360900 T  0008
                    JUMP OUT 2 TO NOPE ;                                                 01360950 T  0009
                  SI := SI + 1)) ;                                                       01361000 T  0009
NOPE:                                                                                    01361050 T  0010
            DS := 8 LIT "+0000001" ;                                                     01361100 T  0010
            GO TO XIT ;                                                                  01361150 T  0012
YES:                                                                                     01361200 T  0012
            SI := SI + 1 ;                                                               01361250 T  0012
            DI := DI + 7 ;                                                               01361300 T  0013
            DS := CHR ;                                                                  01361350 T  0013
XIT:                                                                                     01361400 T  0013
      END GETCHAR ;                                                                      01361450 T  0013
      IF I := GETCHAR (IMAGE) GEQ 0 THEN                                                 01361500 T  0015
        CHARACTER := I ;                                                                 01361550 T  0017
      IF NUM1 THEN                                                                       01361600 T  0019
      BEGIN                                                                              01361650 T  0020
        COLSTOP1 := MIN (PARAMETER1, 80) ;                                               01361700 T  0020
        IF NUM2 THEN                                                                     01361900 T  0024
        BEGIN                                                                            01361950 T  0024
          COLSTOP2 := MIN (MAX (PARAMETER2, COLSTOP1), 80) ;                             01362000 T  0025
          IF NUM3 THEN                                                                   01362050 T  0031
          BEGIN                                                                          01362100 T  0032
            COLSTOP3 := MIN (MAX (PARAMETER3, COLSTOP2), 80) ;                           01362150 T  0032
            IF NUM4 THEN                                                                 01362200 T  0038
            BEGIN                                                                        01362250 T  0039
              COLSTOP4 := MIN (MAX (PARAMETER4, COLSTOP3), 80) ;                         01362300 T  0039
              COLSTOPS := 4 ;                                                            01362350 T  0045
            END ELSE                                                                     01362400 T  0047
              COLSTOPS := 3 ;                                                            01362450 T  0047
          END ELSE                                                                       01362500 T  0048
            COLSTOPS := 2 ;                                                              01362550 T  0048
        END ELSE                                                                         01362600 T  0050
          COLSTOPS := 1 ;                                                                01362650 T  0050
        MAXCOLSTOP := COLSTOP [COLSTOPS] ;                                               01362675 T  0052
      END ELSE                                                                           01362700 T  0054
      IF EMPTY1 THEN                                                                     01362750 T  0054
      BEGIN                                                                              01363100 T  0055
        SHOW (PARAMETER0, ONOFF (COLUMNS) & (CHARACTER)[42:42:6]) ;                      01363200 T  0056
        IF COLSTOPS LSS 1 THEN                                                           01363210 T  0060
          PARAMETER1 := 0 & "#"[6:42:6]                                                  01363220 T  0061
        ELSE PARAMETER1 := OCTDEX (COLSTOP1) ;                                           01363230 T  0063
        IF COLSTOPS LSS 2 THEN                                                           01363240 T  0069
          PARAMETER2 := 0 & "#"[6:42:6]                                                  01363250 T  0070
        ELSE PARAMETER2 := OCTDEX (COLSTOP2) ;                                           01363260 T  0071
        SHOW (PARAMETER1, PARAMETER2) ;                                                  01363270 T  0074
        IF COLSTOPS LSS 3 THEN                                                           01363280 T  0076
          PARAMETER3 := 0 & "#"[6:42:6]                                                  01363290 T  0077
        ELSE PARAMETER3 := OCTDEX (COLSTOP3) ;                                           01363300 T  0078
        IF COLSTOPS LSS 4 THEN                                                           01363310 T  0081
          PARAMETER4 := 0 & "#"[6:42:6]                                                  01363320 T  0082
        ELSE PARAMETER4 := OCTDEX (COLSTOP4) ;                                           01363330 T  0084
        ERRORX (7, PARAMETER3, PARAMETER4) ;                                             01363410 T  0087
```

```
      END ELSE                                                                    01363420 T  0088
        COLUMNS := TOGGLE (COLUMNS, 1) ;                                          01363430 T  0088
    END COLUMN ;                                                                  01364500 T  0092
                                                                          48 IS   93 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *       01365000 T  0051
PROCEDURE TEACH ;                                                                 01365500 T  0051
  BEGIN                                                                           01366000 T  0051
LABEL NEXT ;                                                                      01366500 T  0051
                                                                        START OF SEGMENT **********   49
    IF NOT EMPTY1 THEN                                                            01367000 T  0000
    BEGIN                                                                         01367500 T  0001
      M := -1 ;                                                                   01368000 T  0001
      FOR I := 0 STEP 1 UNTIL RSWDM DO                                            01368500 T  0002
        IF PARAMETER1 = RSWD [I] THEN                                             01369000 T  0004
        BEGIN                                                                     01369500 T  0005
          M := I ;                                                                01370000 T  0005
          I := RSWDM ;                                                            01370500 T  0006
        END ;                                                                     01371000 T  0007
      IF M LSS 0 THEN                                                             01371500 T  0009
      BEGIN                                                                       01371600 T  0010
        IF I := XFILE (PARAMETER1, PARAMETER2:=MACROLIBRARY, -1) LSS 2            01371700 T  0010
          AND MACROLIBRARY NEQ "MACRO  " THEN                                     01371800 T  0013
          I := XFILE (PARAMETER1, PARAMETER2:="MACRO  ", -1) ;                    01371830 T  0014
        IF I LSS 2 THEN                                                           01371900 T  0017
        BEGIN                                                                     01372000 T  0018
          SHOW (PARAMETER1, " INVALI") ;                                          01372020 T  0018
          ERROR (NEXT, 0, "D:*    ", RWTEACH) ;                                   01372040 T  0020
        END ;                                                                     01372050 T  0025
        NUM2 := FALSE ;                                                           01372100 T  0025
        NUM3 := BOOLEAN (2) ;                                                     01372200 T  0026
        LISTIT (0) ;                                                              01372300 T  0028
        GO TO NEXT ;                                                              01372400 T  0029
      END ;                                                                       01372450 T  0029
      PARAMETER1 := "TEACHER" ;                                                   01372500 T  0029
      PARAMETER2 := OCTDEC (VERSION) ;                                            01373000 T  0030
      IF XFILE (PARAMETER1, PARAMETER2, 2) LSS 2 THEN                             01373500 T  0032
        GO TO NEXT ;                                                              01374000 T  0034
      READ (LIBRARY [M], 1, IMAGE [*]) ;                                          01375000 T  0034
      N := DEC (IMAGE [0], 8) ;                                                   01376000 T  0040
      CLOSE (LIBRARY) ;                                                           01376500 T  0042
      PARAMETER3 := N DIV 10000 ;                                                 01377000 T  0043
      NUM3 := TRUE ;                                                              01377500 T  0045
      PARAMETER4 := N MOD 10000 ;                                                 01378000 T  0046
      NUM4 := TRUE ;                                                              01378500 T  0048
      N := RESETN ;                                                               01379000 T  0049
      LISTIT (17) ;% POSTING AND QUICK                                            01379500 T  0050
    END ELSE                                                                      01380500 T  0051
    BEGIN                                                                         01380600 T  0051
      WRITE (PRETANK [*], TEACH1) ;                                               01381500 T  0054
      WRITETWX ;                                                                  01382000 T  0058
      FOR I := 0 STEP 7 UNTIL RSWDM DO                                            01382500 T  0058
      BEGIN                                                                       01383000 T  0060
        WRITE (IMAGE [*], TEACH2, FOR M := 0 STEP 1 UNTIL 6 DO                    01383500 T  0060
          [(PARAMETER0 := RSWD [I + M]).[6:6], PARAMETER0]) ;                     01384000 T  0066
        WRITEROW (IMAGE, FALSE, COBOL) ;                                          01384500 T  0076
      END ;                                                                       01386500 T  0078
      WRITE (IMAGE [*], TEACH3) ;                                                 01387000 T  0080
```

```
      WRITEROW (IMAGE, FALSE, COBOL) ;                                                                    01387500 T  0084
    END ;                                                                                                 01390500 T  0086
NEXT:                                                                                                     01390600 T  0086
  END TEACH ;                                                                                             01390700 T  0086
                                                                                                  49 IS   87 LONG, NEXT SEG    2
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *       01391500 T  0051
DEFINE PERCENT =                                                                                          01392000 T  0051
  BEGIN                                                                                                   01392500 T  0051
    TRANSLATING := BOOLEAN (I := REAL (TOGGLE (TRANSLATING, 1))) ;                                        01393000 T  0051
    TRANSLATEI := I ;                                                                                     01393500 T  0051
  END# ;                                                                                                  01394500 T  0051
% * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *       01402000 T  0051
PROCEDURE STOP ;                                                                                          01402500 T  0051
  BEGIN                                                                                                   01403000 T  0051
DEFINE DIRCTRY = CONTROLS# ;                                                                              01403100 T  0051
                                                                                                START OF SEGMENT **********   50
LABEL NEXT ;                                                                                              01403500 T  0000
    IF BOOLEAN (ABNORMALEND) THEN                                                                         01403600 T  0000
    BEGIN                                                                                                 01403610 T  0001
      EMPTY1 := ABNORMALEND = 3 ;                                                                         01403620 T  0001
      ABNORMALEND := BREAKI := 0 ;                                                                        01403630 T  0005
      IF BOOLEAN (INREADYQ) THEN                                                                          01403650 T  0008
      BEGIN                                                                                               01403690 T  0010
        FOR I := 1 STEP 1 WHILE READYQ [I] NEQ USER DO ;                                                  01403700 T  0010
        FOR I := I + 1 STEP 1 UNTIL READYQTOP DO                                                          01403750 T  0015
          READYQ [I - 1] := READYQ [I] ;                                                                  01403775 T  0019
        READYQTOP := READYQTOP - 1 ;                                                                      01403800 T  0022
        INREADYQ := 0 ;                                                                                   01403810 T  0023
      END ;                                                                                               01403825 T  0025
    END ELSE                                                                                              01403850 T  0025
    BEGIN                                                                                                 01403875 T  0025
      IF FILEOPEN AND PARAMETER1 NEQ "DS      " THEN                                                      01403900 T  0025
        CLOSEFILE ;                                                                                       01404500 T  0027
      WRITE (PRETANK [*], EOJ) ;                                                                          01407000 T  0028
      WRITETWX ;                                                                                          01407500 T  0032
      IF NOT EMPTY1 THEN                                                                                  01407600 T  0033
        SAVESTATE ;                                                                                       01407700 T  0034
      IF COUNTI GEQ 0 THEN                                                                                01407710 T  0035
      BEGIN                                                                                               01407720 T  0037
        ABNORMALEND := IF EMPTY1 THEN 2 ELSE 4 ;                                                          01407730 T  0037
        STATION := 0 ;                                                                                    01407740 T  0041
        GO TO NEXT ;                                                                                      01407750 T  0042
      END ;                                                                                               01407760 T  0044
    END ;                                                                                                 01407800 T  0044
    FORGET (STATIONI) ;                                                                                   01407900 T  0044
    I := 2 * SLOTI ;                                                                                      01408000 T  0046
    READ (R1 [45], 90, DIRCTRY [*]) ;                                                                     01408100 T  0048
    IF EMPTY1 THEN                                                                                        01408500 T  0053
      DIRCTRY [I] := 0                                                                                    01408600 T  0054
    ELSE                                                                                                  01408700 T  0055
      DIRCTRY [I].[1:1] := 0 ;                                                                            01408800 T  0056
    DIRCTRY [I + 1] := 0 ;                                                                                01409000 T  0059
    WRITE (R1 [45], 90, DIRCTRY [*]) ;                                                                    01409500 T  0061
    STATION := 0 ;                                                                                        01410500 T  0066
    IF USER NEQ BIGBIRD THEN                                                                              01415100 T  0066
    BEGIN                                                                                                 01415110 T  0067
      WRITE (BUFFERS [USER, *], 45, BUFFERS [BIGBIRD, *]) ;                                               01415130 T  0068
```

```
          IF BOOLEAN (INREADYQ) THEN                                                    01415140 T  0071
            FOR I := 0 STEP 1 UNTIL READYQTOP DO                                        01415150 T  0072
              IF READYQ [I] = BIGBIRD THEN                                              01415160 T  0074
                READYQ [I] := USER ;                                                    01415170 T  0075
          READ (R1 [IF INREADYQ=3 THEN 46 ELSE SLOTI], 90, CONTROLS [*]) ;              01415180 T  0079
          FILEACCESS := CONTROLS [51] ;                                                 01415190 T  0087
          IF FILEOPEN THEN                                                              01415200 T  0089
          BEGIN                                                                         01415210 T  0090
            N := BIGBIRD * 32 ;                                                         01415215 T  0090
            M := CONTROLS [57].LEFTSIDE ;                                               01415220 T  0092
            FOR I := 0 STEP 1 UNTIL M DO                                                01415230 T  0093
              WRITE (LINKLISTS [USER32 + I, *], 256,                                    01415240 T  0095
                     LINKLISTS [N + I, *]) ;                                            01415250 T  0096
          END ;                                                                         01415260 T  0101
          IF XDEX := CONTROLS [62] GEQ 0 THEN                                           01415265 T  0101
          BEGIN                                                                         01415270 T  0103
            WRITE (XARRAY [USER, *], XMAX * 13, XARRAY [BIGBIRD, *]) ;                  01415275 T  0103
            FOR XDEX := XDEX STEP -1 UNTIL 0 DO                                         01415280 T  0107
              IF BOOLEAN (XNCHRS).[1:1] THEN                                            01415285 T  0109
              BEGIN                                                                     01415290 T  0111
                READ (IO [2*MAXUSERS+XMAX*BIGBIRD+XDEX], 30, IMAGE [*]) ;               01415295 T  0112
                WRITE (IO [2*MAXUSERS+XMAX*USER+XDEX], 30, IMAGE [*]) ;                 01415300 T  0119
              END ;                                                                     01415305 T  0126
          END ;                                                                         01415310 T  0129
        END ;                                                                           01415315 T  0129
        BIGBIRD := BIGBIRD - 1 ;                                                        01415500 T  0129
NEXT:                                                                                   01416500 T  0130
      END ;                                                                             01417000 T  0131
                                                                                  50 IS  132 LONG, NEXT SEG     2
PROCEDURE PROGRAM ;                                                                     01417100 T  0051
BEGIN                                                                                   01417110 T  0051
LABEL NEXT, EXIT ;                                                                      01417120 T  0051
                                                                                START OF SEGMENT **********    51
NEXT:                                                                                   01417150 T  0000
     CASE VERB OF                                                                       01417160 T  0000
     BEGIN                                                                              01417170 T  0000
       EXECUTE ;                                                                        01417175 T  0001
       DITTO ;                                                                          01417180 T  0002
       COPY ;                                                                           01417190 T  0003
       INLINE ;                                                                         01417200 T  0004
       ZIPIT ;                                                                          01417210 T  0005
       CHANGE ;                                                                         01417220 T  0006
       EDIT ;                                                                           01417230 T  0007
       SAVEIT ;                                                                         01417240 T  0008
       RESEQ ;                                                                          01417250 T  0014
       PUNCH ;                                                                          01417260 T  0015
       PRINT ;                                                                          01417270 T  0016
       DELETE ;                                                                         01417280 T  0017
       CLOSEIT ;                                                                        01417300 T  0018
       COMPILE ;                                                                        01417310 T  0021
       COLUMN ;                                                                         01417320 T  0022
       SCAN ;                                                                           01417330 T  0023
       LISTING ;                                                                        01417340 T  0024
       INCREMENT ;                                                                      01417350 T  0025
       TAB ;                                                                            01417360 T  0031
       PERCENT ;                                                                        01417370 T  0061
       QUICKLIST ;                                                                      01417380 T  0067
```

```
      LISTIT (0) ;
      OPEN ;
      MAIL ;
      TEACH ;
      REMOVE ;
      REPLACE ;
      STOP ;
      GO TO EXIT ;
    END ;


  IF BIGBIRD GEQ 0 THEN
    GO TO NEXT ;
EXIT:
END PROGRAM ;

BOOLEAN PROCEDURE RC (START) ;
VALUE START ;
BOOLEAN START ;
BEGIN
SAVE FILE OUT RONE DISK SERIAL [1:47] "R/C" "#1" (1, 90, SAVE 99) ;

SAVE FILE OUT RTWO DISK SERIAL [15:96] "R/C" "#2" (1, 256, SAVE 99) ;
ARRAY DIRCTRY, NEWDIRCTRY [0:90],
  LINKLIST [0:255] ;
LABEL ENDOFPROGRAM ;
   CHARGE (0) ;
   FREEFILE (0) ;
   IF START THEN
   BEGIN
     SEARCH (RONE, IMAGE [*]) ;
     IF IMAGE [6] GTR 0 THEN
     BEGIN
       I := STATUS (IMAGE [*]) ;
       WRITE (TWXOUTPUT (IMAGE [0]), USERUN) ;
       GO TO ENDOFPROGRAM ;
     END ;
     IF IMAGE [0] GEQ 0 THEN
     BEGIN
       READ (R1 [45], 90, DIRCTRY [*]) ;
       DIRCTRY [90] := 12 ;
       FOR I := 0 STEP 2 WHILE USERCODE := DIRCTRY [I] NEQ 12 DO
         DIRCTRY [I] := ABS (USERCODE) ;
       WRITE (R1 [45], 90, DIRCTRY [*]) ;
     END ELSE
     BEGIN
       DIRCTRY [0] := 12 ;
       WRITE (RONE [45], 90, DIRCTRY [*]) ;
     END ;
     SEARCH (RTWO, IMAGE [*]) ;
     IF IMAGE [0] LSS 0 THEN
       WRITE (RTWO[0], 1, IMAGE [*]) ;
   END ELSE
   BEGIN
     READ (R1 [45], 90, DIRCTRY [*]) ;
     USER := -2 ;
     FOR N := 0 STEP 1 UNTIL 1 DO
```

```
                                                    01417390 T  0069
                                                    01417400 T  0070
                                                    01417410 T  0071
                                                    01417420 T  0072
                                                    01417430 T  0073
                                                    01417440 T  0074
                                                    01417450 T  0075
                                                    01417460 T  0076
                                                    01417470 T  0077
                                                START OF SEGMENT **********    52
                                                 52 IS   30 LONG, NEXT SEG    51
                                                    01417480 T  0077
                                                    01417490 T  0078
                                                    01417500 T  0078
                                                    01418000 T  0079
                                                 51 IS   80 LONG, NEXT SEG     2
                                                    01418100 T  0051
                                                    01418200 T  0051
                                                    01418300 T  0051
                                                    01418500 T  0051
                                                    01418600 T  0051
                                                START OF SEGMENT **********    53
                                                    01418700 T  0006
                                                    01418800 T  0012
                                                    01418900 T  0014
                                                    01419000 T  0016
                                                    01419100 T  0016
                                                    01419150 T  0017
                                                    01419200 T  0018
                                                    01419300 T  0018
                                                    01419500 T  0019
                                                    01420000 T  0021
                                                    01420100 T  0022
                                                    01420200 T  0022
                                                    01420300 T  0025
                                                    01420500 T  0030
                                                    01420600 T  0032
                                                    01421000 T  0032
                                                    01421500 T  0033
                                                    01422000 T  0033
                                                    01422500 T  0039
                                                    01423000 T  0040
                                                    01423500 T  0045
                                                    01424000 T  0047
                                                    01424500 T  0052
                                                    01425000 T  0052
                                                    01425500 T  0052
                                                    01426000 T  0053
                                                    01426500 T  0058
                                                    01432500 T  0058
                                                    01433000 T  0060
                                                    01433500 T  0061
                                                    01436500 T  0067
                                                    01436600 T  0067
                                                    01437300 T  0067
                                                    01437500 T  0073
                                                    01437600 T  0074
```

```
    FOR I := 0 STEP 2 WHILE USERCODE := DIRCTRY [I] NEQ 12 DO                    01438000 T  0075
      IF USERCODE NEQ 0 THEN                                                     01439000 T  0079
      BEGIN                                                                      01439500 T  0080
        READ (R1 [I/2], 90, CONTROLS [*]) ;                                      01440000 T  0081
        FILEACCESS := CONTROLS [51] ;                                            01440100 T  0087
        IF FILEOPEN OR BOOLEAN (N) THEN                                          01440500 T  0088
        BEGIN                                                                    01441000 T  0089
          NEWDIRCTRY [USER := USER + 2] := USERCODE ;                            01441500 T  0089
          WRITE (RONE, 90, CONTROLS [*]) ;                                       01442500 T  0092
          IF FILEOPEN THEN                                                       01442600 T  0096
          BEGIN                                                                  01442700 T  0097
            READ SEEK (R2 [16 * I]) ;                                            01443000 T  0097
            NEWDIRCTRY [USER + 1] := DIRCTRY [I + 1] ;                           01443100 T  0103
            M := CONTROLS [57].LEFTSIDE ;                                        01444100 T  0106
            FOR D := 0 STEP 1 UNTIL M DO                                         01444500 T  0107
            BEGIN                                                                01445000 T  0109
              READ (R2, 256, LINKLIST [*]) ;                                     01445500 T  0109
              WRITE (RTWO, 256, LINKLIST [*]) ;                                  01446000 T  0113
            END ;                                                                01446500 T  0117
            IF M NEQ 31 THEN                                                     01446600 T  0119
              WRITE (RTWO [16 * USER + 31], 1, CONTROLS [*]) ;                   01446700 T  0120
            DIRCTRY [I] := 0 ;                                                   01447000 T  0127
          END ;                                                                  01447100 T  0128
        END ;                                                                    01447500 T  0128
      END ;                                                                      01448000 T  0128
    NEWDIRCTRY [USER + 2] := 12 ;                                                01455000 T  0131
      IF USER GEQ 0 THEN                                                         01455500 T  0132
        WRITE (RONE [45], 90, NEWDIRCTRY [*]) ;                                  01456000 T  0133
      CLOSE (R1, PURGE) ;                                                        01456500 T  0139
      READ (R2 [0]) ;                                                            01456600 T  0140
      CLOSE (R2, PURGE) ;                                                        01457000 T  0146
  ENDOFPROGRAM:                                                                  01458100 T  0147
      RC := TRUE ;                                                               01458200 T  0148
    END ;                                                                        01458300 T  0148
   END RC ;                                                                      01458500 T  0148
                                                                          53 IS   154 LONG, NEXT SEG    2
    CONTROLS [90] := 12 ;                                                        01458900 T  0051
    IF NOT RC (TRUE) THEN                                                        01459000 T  0052
    BEGIN                                                                        01459100 T  0053
      BIGBIRD := -1 ;                                                            01459110 T  0054
      TO := 150 ;                                                                01459120 T  0055
      FREEHEAD := MAXFREEHEAD := (XMAX + 2) * MAXUSERS ;                         01459130 T  0056
      PROGRAM ;                                                                  01459200 T  0059
      BOOL := RC (FALSE) ;                                                       01459300 T  0060
      IF XFILE ("TEACHER", OCTDEC (VERSION), -1) GTR 0 THEN                      01459310 T  0061
        READ (LIBRARY) ;                                                         01459320 T  0064
    END ;                                                                        01459400 T  0069
END.                                                                             01459500 T  0069
                                                                           2 IS    73 LONG, NEXT SEG    1
                                                                           1 IS     2 LONG, NEXT SEG    0
                                                                          62 IS    69 LONG, NEXT SEG    0

NUMBER OF ERRORS DETECTED =   0,  COMPILATION TIME =  325 SECONDS.

PRT SIZE = 171; TOTAL SEGMENT SIZE =  4970 WORDS; DISK SIZE = 360 SEGS; NO. PGM. SEGS =  62

ESTIMATED CORE STORAGE REQUIRED = 10779 WORDS.

ESTIMATED AUXILIARY MEMORY REQUIRED =     0 WORDS.
```

NUMBER OF CARD-IMAGES PROCESSED = 4095.

LABEL 000000000LINE 00176194?COMPILE R/C ALGOL LIBRARY ALGOL /R