```
A---E000+    A  0 100 0000 000,0000    TERMINATION SIGNAL
A---E001     A  1 000 0000 001,6001    1
A---E002     A  2 000 0000 001,6002    2
A---E003     A  3 000 0000 001,5003    3
A---E004     A  4 000 0000 001,6004    4
A---E005     A  5 000 0000 001,5005    5
A---E006     A  6 000 0000 001,5006    6
A---E007     A  7 000 0000 001,4007    7
A---E008     A  8 000 0000 001,6010    8
A---E009     A  9 000 0000 001,5011    9
A---E010     A 10 000 0000 000,4016
A---E011     A 11 000 0000 000,4016
A---E012     A 12 000 0000 000,4016
A---E013     A 13 000 0000 000,4016
A---E014     A 14 000 0000 000,4016    SPACE
A---E015     A 15 000 0000 000,4016
A---E016     A 16 000 0000 000,6020    SEMICOLON
A---E017     A 17 001 V  1 002,5021    LCA
A---E018     A 18 002 V  2 002,5022    LCB
A---E019     A 19 003 V  3 002,4023    LCC
A---E020     A 20 004 V  4 002,5024    LCD
A---E021     A 21 005 V  5 002,4025    LCE
A---E022     A 22 006 V  6 002,4026    LCF
A---E023     A 23 007 V  7 002,3027    LCG
A---E024     A 24 010 V  8 002,5030    LCH
A---E025     A 25 011 V  9 002,4031    LCI
A---E026     A 26 000 0000 000,4016
A---E027     A 27 000 0000 000,3033    .
A---E028     A 28 000 0000 000,4016
A---E029     A 29 000 0000 000,4016
A---E030     A 30 000 0000 000,3036    EJECT+CR+EOM
A---E031     A 31 000 0000 000,2037    EJECT+CR
A---E032     A 32 000 = 33 240,6040    MULTIPLY DOT
A---E033     A 33 012 V 10 002,5041    LCJ
A---E034     A 34 013 V 11 002,5042    LCK
A---E035     A 35 014 V 12 002,4043    LCL
A---E036     A 36 015 V 13 002,5044    LCM
A---E037     A 37 016 V 14 002,4045    LCN
A---E038     A 38 017 V 15 002,4046    LCO
A---E039     A 39 020 V 16 002,3047    LCP
A---E040     A 40 021 V 17 002,5050    LCQ
A---E041     A 41 022 V 18 002,4051    LCR
A---E042     A 42 100 0000 000,4052    CR+EOM+TERMINAL SIGNAL IN SIGN
A---E043     A 43 002 0000 004,3053    =
A---E044     A 44 000 = 32 140,4054    -
A---E045     A 45 000 0000 000,3416    CONDITION FLAG (CHAR 055 CAN'T GET IN)
A---E046     A 46 000 0000 000,4016
A---E047     A 47 000 0000 000,4016
A---E048     A 48 000 0000 001,5060    ZERO
A---E049     A 49 000 = 34 240,4061    /
A---E050     A 50 023 V 19 002,4062    LCS
A---E051     A 51 024 V 20 002,3063    LCT
A---E052     A 52 025 V 21 002,4064    LCU
A---E053     A 53 026 V 22 002,3065    LCV
A---E054     A 54 027 V 23 002,3066    LCW
A---E055     A 55 030 V 24 002,2067    LCX
A---E056     A 56 031 V 25 002,4070    LCY
A---E057     A 57 032 V 26 002,3071    LCZ
A---E058     A 58 000 0000 000,4016
```

```
A---E059    A 59 000 0000 000,2073    ,
A---E060    A 60 000 = 31 140,3074    +
A---E061    A 61 000 0000 000,2075    TAB
A---E062    A 62 000 0000 000,4016
A---E063    A 63 000 0000 000,4016
A---E064    A 64 000 0000 000,4016
A---E065    A 65 000 0000 000,5101    PRIME
A---E066    A 66 000 0000 000,5102    QUOTE MARKS
A---E067    A 67 000 0000 000,4016    STRIKE OUT CHARACTER = SPACE
A---E068    A 68 000 0000 000,5104    $
A---E069    A 69 003 0000 004,4105    LESS THAN OR =
A---E070    A 70 006 0000 004,4106    GREATER THAN OR =
A---E071    A 71 001 0000 004,3107    LESS THAN
A---E072    A 72 004 0000 004,5110    GREATER THAN
A---E073    A 73 000 A112 020,4111    (
A---E074    A 74 000 0000 000,4016
A---E075    A 75 000 0000 000,4016
A---E076    A 76 000 0000 000,4016
A---E077    A 77 000 0000 000,4016
A---E078    A 78 000 0000 000,4016
A---E079    A 79 000 0000 000,4016
A---E080    A 80 000 0000 000,5120    COLON
A---E081    A 81 041 V 27 002,4121    A
A---E082    A 82 042 V 28 002,4122    B
A---E083    A 83 043 V 29 002,3123    C
A---E084    A 84 044 V 30 002,4124    D
A---E085    A 85 045 V 31 002,3125    E
A---E086    A 86 046 V 32 002,3126    F
A---E087    A 87 047 V 33 002,2127    G
A---E088    A 88 050 V 34 002,4130    H
A---E089    A 89 051 V 35 002,3131    I
A---E090    A 90 000 0000 000,4016
A---E091    A 91 000 A123 010,2133    RIGHT BRACKET
A---E092    A 92 000 0000 000,4016
A---E093    A 93 000 0000 000,4016
A---E094    A 94 000 0000 000,4016
A---E095    A 95 000 0000 000,4016
A---E096    A 96 000 A 96 030,5140    ABS VAL BAR
A---E097    A 97 052 V 36 002,4141    J
A---E098    A 98 053 V 37 002,4142    K
A---E099    A 99 054 V 38 002,3143    L
A---E100    A100 055 V 39 002,4144    M
A---E101    A101 056 V 40 002,3145    N
A---E102    A102 057 V 41 002,3146    O
A---E103    A103 060 V 42 002,2147    P
A---E104    A104 061 V 43 002,4150    Q
A---E105    A105 062 V 44 002,3151    R
A---E106    A106 000 0000 000,4016
A---E107    A107 005 0000 004,2153    NOT =
A---E108    A108 000 0000 000,3154    UNDERSCORE
A---E109    A109 000 0000 000,2433    PERIOD WITH FLAG (CHAR 155 CAN'T GET IN)
A---E110    A110 000 0000 000,4016
A---E111    A111 000 0000 000,4016
A---E112    A112 000 A 73 010,4160    )
A---E113    A113 000 0000 000,3161    QUESTION MARK
A---E114    A114 063 V 45 002,3162    S
A---E115    A115 064 V 46 002,2163    T
A---E116    A116 065 V 47 002,3164    U
A---E117    A117 066 V 48 002,2165    V
A---E118    A118 067 V 49 002,2166    W
```

```
A---E119    A119 070 V 50 002,1167      X
A---E120    A120 071 V 51 002,3170      Y
A---E121    A121 072 V 52 002,2171      Z
A---E122    A122 000 0000 000,4016
A---E123    A123 000 A091 020,1173      LEFT BRACKET
A---E124    A124 000 = 35 440,2174      *
A---E125    A125 000 0000 000,1175      TAB
A---E126    A126 000 0000 000,4016
A---E127    A127 000 0000 000,4016
A---E128    A128 000 0000 000,0037      EJECT
A---E129    A129 000 0000 000,0056      CARRIAGE RETURN
A---E130    A130 000 0000 000,0155      PSEUDO-PERIOD (CAN'T BE GENERATED)
E---E000+   E  0 020 *  0 010 $  0      OUT OF SPACE
E---E001    E  1 020 *  1 010 $  0      OVERLFOW
E---E002    E  2 020 *  2 010 $  0      UNDEFINED VALUE
E---E003    E  3 020 *  3 010 $  0      DIVISION BY ZERO
E---E004    E  4 020 *  4 010 $  0      ILLEGAL INDEX
E---E005    E  5 020 *  5 010 $  0      NUMERAL MORE THAN NINE DIGITS
E---E006    E  6 020 *  6 010 $  0      MALFORMED
E---E007    E  7 020 *  7 010 $  0      ILLEGAL LIMIT
E---E008    E  8 020 *  8 010 $  0      OPEN TASK + DO
E---E009    E  9 020 *  9 010 $  0      ILLEGAL INDIRECT
E---E010    E 10 020 * 10 010 $  0      ILLEGAL DIRECT
E---E011    E 11 020 * 11 010 $  0      ILLEGAL DECK NUMBER
E---E012    E 12 020 * 12 010 $  0      ILLEGAL FORM NUMBER
E---E013    E 13 020 * 13 010 $  0      ILLEGAL STEP NUMBER
E---E014    E 14 020 * 14 010 $  0      ILLEGAL ARG FOR SQRT
E---E015    E 15 020 * 15 010 $  0      CONDITIONAL DIRECT
E---E016    E 16 020 * 16 010 $  0      NO SUCH STEP
E---E017    E 17 020 * 17 010 $  0      NO SUCH PART
E---E018    E 18 020 * 18 010 $  0      ILLEGAL PART NUMBER
E---E019    E 19 020 * 19 010 $  0      NO SUCH FORM
E---E020    E 20 020 * 20 010 $  0      ILLEGAL ARG FOR LOG
E---E021    E 21 020 * 21 010 $  0      NEGATIVE * NON INTEGER
E---E022    E 22 020 * 22 010 $  0      ILLEGAL ARG FOR SIN OR COS
E---E023    E 23 020 * 23 010 $  0      ZERO * NEGATIVE
E---E100              050 * 45 010 $  1
E---E110              020 S  0 050 P 12
E---E120              020 Q  9 002 * 41
E---E130              020 $  0 010 = 64
E---E140                 * 40     * 40
E---E150              010 * 44
E---E160    * 40 052,4511,222,3051      'ERROR IN LINE ABOVE '
E---E170              007,0311,120,7043
E---E180              014,4450,520,7021
E---E190              011,0461,521,2520
E---E200              007,0164,000,0000
E---E210    * 41 020 $  0 010 = 64
E---E220                 * 42     * 42
E---E230              010 * 43
E---E240    * 42 052,4511,222,3051      'ERROR IN STEP'
E---E250              007,0311,120,7062
E---E260              031,4251,160,7200
E---E270    * 43 020 P 16 056 P  3
E---E280              020 $  0 010 = 63      COPY STEP NUMBER FROM R TO S.
E---E290              020 A 80 050 P 11
E---E300              020 $  0 010 = 66      ADD COLON
E---E310              020 $  0 010 = 62      SPACE
E---E320              020 $  0 010 = 62      SPACE
E---E330    * 44 020 $  0 010 = 64
```

```
E---E340      * 45
E---E350            020 $  0 010 = 25      SEND ERROR MSG
E---E360      * 46  020 Q  2 001 * 47      CLEAN UP OPERAND STACK
E---E370            020 $  0 010 = 13
E---E380            010 * 46
E---E390      * 47  020 Q  4 001 * 48      CLEAN UP OPERATOR STACK
E---E400            020 $  0 010 = 15
E---E410            010 * 47
E---E420      * 48  020 Q  6 001 * 49      CLEAN UP AUXILIARY STACK
E---E430            020 $  0 010 = 17
E---E440            010 * 48
E---E450      * 49  020 $  0 010 = 55      SWITCH
E---E460            020 $  0 010 = 53      KICK OUT
E---E470            010 X  1
E---E500      * 50  054,4451,443,2026      'INSUFFICIENT STORAGE SPACE.'
E---E501            013,0310,461,4425
E---E502            022,4630,343,1063
E---E503            023,0510,421,3425
E---E504            007,0621,161,0423
E---E505            012,4331,250,0000
E---E510      * 51  063,0650,522,4426      'OVERFLOW.'
E---E511            021,4461,541,5452
E---E512            100,0000,000,0000
E---E520      * 52  072,0450,501,2426      'UNDEFINED VALUE.'
E---E521            014,4450,521,2016
E---E522            032,4211,063,2025
E---E523            015,4524,000,0000
E---E530      * 53  050,4631,461,2444      'ATTEMPTING DIVISION BY ZERO.'
E---E531            023,4630,622,2427
E---E532            007,0240,623,2431
E---E533            031,0311,142,2416
E---E534            011,0700,343,4425
E---E535            024,4460,662,5200
E---E540      * 54  054,4450,501,2467      'INDEX VALUE MUST BE INTEGER AND
E---E541            007,0650,422,1464        0=INDEX=99.'
E---E542            012,4161,103,2062
E---E543            031,4160,441,2416
E---E544            014,4451,461,2427
E---E545            012,4510,341,0445
E---E546            012,0161,404,2431
E---E547            022,4240,523,3505
E---E548            004,4110,662,5200
E---E550      * 55  062,4641,101,2451      'NUMERALS MUST BE LESS THAN TEN
E---E551            010,4431,440,7044        DIGITS LONG.'
E---E552            032,0621,460,7022
E---E553            012,4161,061,2462
E---E554            031,0161,461,4021
E---E555            022,4161,461,2445
E---E556            007,0240,621,3431
E---E557            031,4620,342,1446
E---E558            022,4270,662,5200
E---E560      * 56  062,0211,061,3046      'MALFORMED.'
E---E561            024,4440,521,2033
E---E562            025,2000,000,0000
E---E570      * 57  061,4311,101,4463      'LIMIT MUST BE INTEGER AND
E---E571            007,0441,503,1063        /LIMIT/-10*9.'
E---E572            007,0220,520,7031
E---E573            022,4630,521,3425
E---E574            024,4160,422,2424
E---E575            007,1401,061,4444
```

```
E---E576          014,4633,004,3401
E---E577          030,1740,221,5452
E---E578          100,0000,000,0000
E---E580     * 58 063,4430,521,0462      'PLEASE 'CANCEL' OR 'GO' BEFORE
E---E581          012,4162,041,1421        ANOTHER 'DO'.'
E---E582          022,4230,522,1502
E---E583          007,0461,220,7102
E---E584          013,4462,040,7022
E---E585          012,4261,142,4425
E---E586          007,0211,122,3063
E---E587          031,4300,522,4416
E---E588          041,0241,144,1033
E---E589          025,2000,000,0000
E---E590     * 59 071,4300,623,1016      'THIS COMMAND MUST NOT BE
E---E591          011,4461,102,2021        GIVEN INDIRECTLY.'
E---E592          022,4240,342,2064
E---E593          031,0630,342,2446
E---E594          031,4160,441,2416
E---E595          013,4311,521,2445
E---E596          007,0311,121,2031
E---E597          024,4250,463,1443
E---E598          034,0331,250,0000
E---E600     * 60 071,4300,623,1016      'THIS COMMAND MUST NOT BE
E---E601          011,4461,102,2021        GIVEN DIRECTLY.'
E---E602          022,4240,342,2064
E---E603          031,0630,342,2446
E---E604          031,4160,441,2416
E---E605          013,4311,521,2445
E---E606          007,0240,622,4425
E---E607          011,4631,063,4033
E---E608          025,2000,000,0000
E---E610     * 61 052,0250,462,1016      'DECK NUMBER MUST BE INTEGER AND
E---E611          022,4641,101,1025        0-DECK-10*9.'
E---E612          024,4161,103,2062
E---E613          031,4160,441,2416
E---E614          014,4451,461,2427
E---E615          012,4510,341,0445
E---E616          012,0161,404,3424
E---E617          012,4231,044,3401
E---E618          030,1740,221,5452
E---E619          100,0000,000,0000
E---E620     * 62 053,0461,222,2016      'FORM NUMBER MUST BE INTEGER AND
E---E621          022,4641,101,1025        0-FORM-10*9.'
E---E622          024,4161,103,2062
E---E623          031,4160,441,2416
E---E624          014,4451,461,2427
E---E625          012,4510,341,0445
E---E626          012,0161,404,3426
E---E627          023,0511,104,3401
E---E628          030,1740,221,5452
E---E629          100,0000,000,0000
E---E630     * 63 071,0630,522,3416      'STEP NUMBER MUST SATISFY 1-STEP-10*9.'
E---E631          022,4641,101,1025
E---E632          024,4161,103,2062
E---E633          031,4161,441,0463
E---E634          014,4620,543,4016
E---E635          000,5051,443,1425
E---E636          023,5070,023,0174
E---E637          004,4331,250,0000
E---E640     * 64 050,4631,461,2444      'ATTEMPTING SQRT OF A NEGATIVE NUMBER.'
```

```
   E---E641          023,4630,622,2427
   E---E642          007,0621,202,4463
   E---E643          007,0460,540,7021
   E---E644          007,0450,521,3421
   E---E645          031,4311,521,2416
   E---E646          022,4641,101,1025
   E---E647          024,4331,250,0000
   E---E650     * 65 052,0311,221,2423     'DIRECT COMMANDS MUST BE UNCONDITIONAL.'
   E---E651          031,4160,462,3044
   E---E652          022,0211,121,2062
   E---E653          007,0441,503,1063
   E---E654          007,0220,520,7064
   E---E655          022,4231,142,2424
   E---E656          014,4630,622,3045
   E---E657          010,4430,662,5200
   E---E660     * 66 071,4300,520,7062     'THE STEP CALLED FOR DOESN'T EXIST.'
   E---E661          031,4251,160,7023
   E---E662          010,4431,061,2424
   E---E663          007,0261,142,4416
   E---E664          012,0460,523,1045
   E---E665          040,4630,341,2467
   E---E666          014,4621,461,5452
   E---E667          100,0000,000,0000
   E---E670     * 67 071,4300,520,7047     'THE PART CALLED FOR DOESN'T EXIST.'
   E---E671          010,4511,460,7023
   E---E672          010,4431,061,2424
   E---E673          007,0261,142,4416
   E---E674          012,0460,523,1045
   E---E675          040,4630,341,2467
   E---E676          014,4621,461,5452
   E---E677          100,0000,000,0000
   E---E680     * 68 063,4211,223,1416     'PART NUMBER MUST BE INTEGER AND
   E---E681          022,4641,101,1025       0-PART-10*9.'
   E---E682          024,4161,103,2062
   E---E683          031,4160,441,2416
   E---E684          014,4451,461,2427
   E---E685          012,4510,341,0445
   E---E686          012,0161,404,3447
   E---E687          010,4511,464,3401
   E---E688          030,1740,221,5452
   E---E689          100,0000,000,0000
   E---E690     * 69 071,4300,520,7026     'THE FORM CALLED FOR DOESN'T EXIST.'
   E---E691          023,0511,100,7023
   E---E692          010,4431,061,2424
   E---E693          007,0261,142,4416
   E---E694          012,0460,523,1045
   E---E695          040,4630,341,2467
   E---E696          014,4621,461,5452
   E---E697          100,0000,000,0000
   E---E700     * 70 050,4631,461,2444     'ATTEMPTING LOG OF A NONPOSITIVE
   E---E701          023,4630,622,2427       NUMBER.'
   E---E702          007,0431,141,3416
   E---E703          023,0260,341,0416
   E---E704          022,4461,122,3446
   E---E705          031,0311,461,4465
   E---E706          012,4161,123,2044
   E---E707          011,0251,221,5452
   E---E708          100,0000,000,0000
   E---E710     * 71 050,4631,461,2444     'ATTEMPTING NEGATIVE BASE TO
   E---E711          023,4630,622,2427       NONINTEGER POWER.'
```

```
   E---E712          007,0450,521,3421
   E---E713          031,4311,521,2416
   E---E714          011,0211,441,2416
   E---E715          031,4460,342,2446
   E---E716          022,4311,123,1425
   E---E717          013,4251,220,7047
   E---E718          023,0660,522,4433
   E---E719          025,2000,000,0000
   E---E720     * 72 050,4631,461,2444      'ATTEMPTING SIN OR COS
   E---E721          023,4630,622,2427        OF MAGNITUDE -100.'
   E---E722          007,0620,622,2416
   E---E723          023,0510,341,1446
   E---E724          031,0161,141,3016
   E---E725          022,0210,562,2431
   E---E726          031,4640,501,2416
   E---E727          043,0011,403,0033
   E---E728          025,2000,000,0000
   E---E730     * 73 050,4631,461,2444      'ATTEMPTING ZERO TO A NEGATIVE POWER.'
   E---E731          023,4630,622,2427
   E---E732          007,0710,522,4446
   E---E733          007,0631,140,7021
   E---E734          007,0450,521,3421
   E---E735          031,4311,521,2416
   E---E736          023,4461,541,2451
   E---E737          015,4524,000,0000
   E---E900      *  0      * 50      * 50
   E---E901      *  1      * 51      * 51
   E---E902      *  2      * 52      * 52
   E---E903      *  3      * 53      * 53
   E---E904      *  4      * 54      * 54
   E---E905      *  5      * 55      * 55
   E---E906      *  6      * 56      * 56
   E---E907      *  7      * 57      * 57
   E---E908      *  8      * 58      * 58
   E---E909      *  9      * 59      * 59
   E---E910      * 10      * 60      * 60
   E---E911      * 11      * 61      * 61
   E---E912      * 12      * 62      * 62
   E---E913      * 13      * 63      * 63
   E---E914      * 14      * 64      * 64
   E---E915      * 15      * 65      * 65
   E---E916      * 16      * 66      * 66
   E---E917      * 17      * 67      * 67
   E---E918      * 18      * 68      * 68
   E---E919      * 19      * 69      * 69
   E---E920      * 20      * 70      * 70
   E---E921      * 21      * 71      * 71
   E---E922      * 22      * 72      * 72
   E---E923      * 23      * 73      * 73
   E---E999          100 0000 000 0000,
   F000E000+    F  0 010 $  0                  Y = SQRT(X)
   F000E010          024 K  2 052 *  9         SET EXIT
   F000E020          023 Q  3 002 *  9         TRANSFER IF X=0
   F000E030          020 $  0 010 = 11         UNPACK X INTO P4, P5, P6
   F000E040          023 P  5 001 E 14         TRANSFER IF X NEGATIVE
   F000E050          020 P  4 071    8         COMPUTE 1/2 SFX
   F000E060          050 T  1 060 T  2         STORE IN P4
   F000E070          071   32 072    1
   F000E080          050 P  4 020 T  1         TEST FOR ODD OR EVEN
   F000E090          004 P  6 002 *  1         IF ODD 10.P6 TO T0
```

```
F000E100          032 N  1 060 T  0      AND 10E9 TO P6 AS Y(0)
F000E110          020 N  9 010 *  2
F000E120     *  1 060 T  0 020 * 90      SFX EVEN, X TO TO
F000E130     *  2 050 P  6 004 T  0      AND SQRT(10) + EPSILON TO P6
F000E140          032 N  8 044 P  6      T1=Y(N)-(10E8*X/Y(N))=DELTA
F000E150          020 P  6 065 T  1
F000E160          020 P  6 073    1      P6=Y(N+1)=(2Y-DELTA)/2
F000E170          025 T  1 072    1
F000E180          050 P  6 020 T  1      IF DELTA GREATER THAN OR EQUAL TO
F000E190          025 N  4 006 *  2      10E4, ITERATE
F000E200          020 P  6 004 T  0
F000E210          037 N  8 044 P  6      IF (Y(N+1) + (Y(N+1)-A*10E8)/Y(N+1)))
F000E220          020 P  6 064 T  2      IS LESS THAN ZERO, ADD ONE TO
F000E230          002 *  3 020 P  6      Y(N+1)
F000E240          024 K  1 050 P  6
F000E250     *  3 020 S  0 010 = 30      PACK RESULT
F000E260     *  9 010 ====               EXIT
F000E270     * 90+   316227800   39,     SQRT(10) + EPSILON
F001E000+    F  1 010 S  0               Y=LOG(X)---BASE E
F001E010          024 K  2 052 *  9      SET EXIT
F001E020          023 Q  3 002 E 20      X=0 IS AN ERROR
F001E030          020 S  0 010 = 11      UNPACK
F001E040          023 P  5 001 E 20      X NEGATIVE IS AN ERROR
F001E050          020 P  4 072  31       SFX AS AN INTEGER
F001E060          050 P  4 020 * 96      INITIALIZE ADDRESS
F001E070          052 *  6 120
F001E080     *  1 050 T  1 020 P  6      SET Q=0
F001E090          025 * 91 002 *  2      IF COEF(X) LESS THAN LOWER BOUND,
F001E100          020 P  6 073    1      2 COEF(X) REPLACES IT AND
F001E110          050 P  6 020 T  1      Q+1 REPLACES Q
F001E120          024 I  1 010 *  1      COMPUTE Z FOR SERIES AS EQUAL TO
F001E130     *  2 020 P  6 024 * 92      (COEF(X)-(10*8)(2*3) /
F001E140          050 T  2 020 P  6      (COEF(X)+(10*8)(2*3)
F001E150          025 * 92 040 T  2
F001E160          060 T  2 030 T  2      Z IN T2, Z*2 IN T3
F001E170          050 T  3 120           COMPUTE 1/2.LOG(X)=Z+1/3(Z*3)+
F001E180          050 T  0 020 * 95      1/5(Z*5)+...+1/13(Z*13)
F001E190     *  3 050 T  4 020 I  1      T4=DELTA=13
F001E200          040 T  4 020 T  0      T0=LOG(X)=0, INITIALLY
F001E210          064 T  0 004 T  0      T0=LOG(X)+(1/DELTA)
F001E220          030 T  3 050 T  0      T0=(LOG(X)+(1/DELTA)).Z*2
F001E230          020 T  4 025 I  4      TRANSFER IF DELTA WAS 3
F001E240          005 *  4 024 I  2      DELTA=DELTA-2
F001E250     *  4 010 *  3 004 T  0      (Z.LOG(X))+Z
F001E260          030 T  2 024 T  2      T0=LOG(X)/4
F001E270          072    1 050 T  0
F001E280          020 I  3 025 T  1
F001E290          050 T  1 004 * 93
F001E300          032 T  1 060 T  5      T5=(3-Q).LOG(2)/4
F001E310          024 T  0 004 * 94      LOG(X)=4.(EXP(X).LOG(10)+T5)
F001E320          036 P  4 077    2      INTEGER IN ACCUMULATOR AND T0, T5
F001E330          050 T  0 050 T  5      FRACTION IN MQ AND T1
F001E340          001 * 12 060 T  1      TRANSFER IF LOG IS NEGATIVE
F001E350     *  5 023 T  0 006 * 14      ..... NORMALIZE .....
F001E360     * 18 020 I  1 050 T  3      IF T0 IS GREATER THAN 10*I, THEN
F001E370     *  6 021 N  1 024 T  0      SET I=I+1, ITERATE
F001E380          005 *  7 020 *  6
F001E390          024 K  2 052 *  6
F001E400          020 T  3 024 I  1
F001E410     *  7 014 * 18 020 T  3      IF T0 IS LESS THAN 10*I, THEN
```

```
F001E420          025 I  1 050 P  4     P4=EXP(Y)=I-1 AND THE MAGNITUDE
F001E430          020 * 97 025 T  3     /FY=(TO+(2*(-39)).T1).10*(9-I)
F001E440          056 *  8 056 * 10
F001E450     *  8 004 T  1 030 ----
F001E460     * 10 004 T  0 036 ----
F001E470          060 P  6 020 P  4     P6=Y, SHIFT EXP(Y)
F001E480     * 11 073   31 050 P  4     P4=SFY
F001E490          020 T  5 124 K  4     SET SIGN OF LOG
F001E500          070    9 050 P  5
F001E510          020 $  0 010 = 30     PACK RESULT
F001E520     *  9 010 ----               EXIT
F001E530     * 12 061 T  1 006 * 13
F001E540          125 K  4 050 T  1     GET ABSOLUTE VALUE OF NEGATIVE LOG
F001E550          022 T  0 025 I  1
F001E560     * 20 050 T  0 010 *  5
F001E570     * 13 050 T  1 022 T  0
F001E580     * 14 010 * 20 023 T  1     LOG IS LESS THAN ONE, TEST FOR
F001E590          002 * 19 004 T  1     LOG=ZERO
F001E600     * 15 020 I  1 050 T  3
F001E610          032 N  1 025 I  1
F001E620          002 * 16 020 T  3
F001E630          024 I  1 014 * 15
F001E640     * 16 020 T  3 025 I  2
F001E650          024 * 96 056 * 17
F001E660     * 17 004 T  1 032 ----
F001E670          032 N  9 050 P  6
F001E680          021 T  3 010 * 11
F001E690     * 19 050 Q  3 010 *  9
F001E700     * 90+  316227800    39     SQRT(10)+EPSILON
F001E710     * 91+  565685425    39     (2E3)(10E8)(SQRT(2)/2)
F001E720     * 92+  800000000    39     (2E3)(10E8)
F001E730     * 93+.17328679514   00     LOG(2)/4
F001E740     * 94+.57564627325   00     LOG(10)/4
F001E750     * 95+          13   39
F001E760     * 96     N  1      N  1
F001E770     * 97     N  9      N  9,
F002E000+    F  2 010 $  0               Y = E*X
F002E010          024 K  2 052 *  9     SET EXIT
F002E020          023 Q  3 001 *  1     IF X=0, SET Y=I AND EXIT
F002E030     *  3 020 N  8 050 Q  3
F002E040          010 *  9
F002E050     *  1 020 $  0 010 = 11     UNPACK NON-ZERO X
F002E060          020 P  6 040 N  9     NEW COEF(X) IS COEF(X) / 10*9
F002E070          060 T  0 020 P  4     SFX SHIFTED TO 2*-39 POSITION
F002E080          072   31 050 T  1     STORE IN T1
F002E090          002 *  2 024 I  9     IF, SFX IS LESS THAN -9, SET Y=1
F002E100          001 *  3 022 T  1
F002E105          025 I  1 010 $  1     ADJUST SFX BY ONE
F002E110          024 * 91 056 *  4
F002E120          120      050 P  4     SFX FOR RESULT
F002E130     *  4 004 T  0 044 ----
F002E140          060 T  0 070    2     COEF(X) /4 FOR SERIES
F002E150          050 T  0 014 *  5     IF SFX IS 3 OR MORE, CHECK SIGN
F002E160     *  2 025 I  3 001 *  6     OF COEF(X)
F002E170     *  8 023 P  5 002 E  1     OVERFLOW FOR POSITIVE NO.
F002E180          120      014 *  3     SET Y=0 FOR NEGATIVE NO.
F002E190     *  6 024 I  1 005 *  7     IF SFX=2, COEF(X) MUST BE LESS
F002E200          020 P  6 025 * 92     THAN LOG(10) TIMES (10*8) + 1
F002E210     *  7 002 *  8 020 T  1     IF GREATER, TEST SIGN OF COEF
F002E220          024 I  1 024 * 91
```

```
F002E230          056 * 10 004 T  0      (COEF(X)) (LOG(E)) (LESS THAN ONE)
F002E240          030 * 93 050 T  0      IN TO
F002E250     * 10 004 T  0 032 ----      TO TIMES 10*(SFX PLUS 1)
F002E260          050 P  4 060 T  0      INTEGER PART TO SFX FOR RESULT
F002E270          072    2 044 * 93      FRACTIONAL PART /4 TO XBAR
F002E280     *  5 060 T  0 023 P  5
F002E290          006 * 11 021 T  0      XBAR FOR SERIES IN TO, USE 12 TERMS
F002E300          050 T  0 021 P  4
F002E310     * 11 050 P  4 020 * 95      SET DIVISOR FOR SERIES
F002E320          050 T  2 020 * 94      SET LOG = 1/2
F002E330     * 12 050 T  1 020 T  0
F002E340          076   39 044 T  2      COMPUTE 1/2 (E*XBAR) = Z
F002E350          030 T  1 024 * 94
F002E360          050 T  1 020 T  2
F002E370          025 I  1 050 T  2
F002E380          025 I  1 006 * 12      (Z*4) (10*10) (16) = E*X
F002E390          004 T  1 030 T  1
F002E400          050 T  1 004 T  1
F002E410          030 T  1 010 * 18
F002E420     * 18 050 T  1 004 T  1
F002E430          032 N 10 077    4
F002E440          050 P  6 020 P  4      ADJUST SFX BY 2
F002E450          025 I  2 050 P  4
F002E460          004 I  3 020 P  6      NORMALIZE E*X
F002E470          025 N 11 002 * 13      RESULT LESS THAN (1.6)(10*11)
F002E480          004 I  2 020 P  6      AND GREATER THAN OR EQUAL TO 10*8
F002E490          025 N 10 002 * 13
F002E500          004 I  1 020 P  6
F002E510          025 N  9 005 * 14
F002E520     * 13 020 * 91 064 T  1
F002E530          056 * 15 020 P  4      ADJUST SFX
F002E540          064 P  4 004 P  6
F002E550     * 15 120      044 ----      SCALE E*X
F002E560     * 14 060 P  6 120           CLEAR SIGN OF E*X
F002E570          050 P  5 020 P  4
F002E580          073   31 050 P  4
F002E620     * 17 020 $  0 010 = 30      PACK RESULT
F002E630     *  9 010 ----               EXIT
F002E640     * 91                 N  0
F002E650     * 92+   230258510    39     LOG(10), BASE E, TIMES 10*8 PLUS 1
F002E660     * 93+.43429448190    00     LOG(E), BASE 10
F002E670     * 94+.50000000000    00     1/2
F002E680     * 95+           12   39
F003E000+    F  3 000 K  0 010 $  0      SIN(X)--SET FLAG IN MQ=-1
F003E010          024 K  2 052 *  9      SET EXIT
F003E020          060 T  0 002 $  1      STORE FLAG IN TO
F003E050          020 $  0 010 = 11      UNPACK
F003E055          020 T  0 002 *  1      FOR COSINE SET P5 EQUAL TO ZERO
F003E060          020 P  5 071    9      SHIFT SIGN OF X TO SIGN BIT
F003E070     *  1 050 P  5 020 P  4      SHIFT SFX TO 2*(-39) POSITION
F003E080          072   31 050 P  4      TRANSFER IF SFX IS NEGATIVE
F003E090          005 *  2 025 I  2      NO. OUT OF RANGE IF SFX IS 2 OR MORE
F003E100          002 E 22 024 I  4      LOCATION (10*(SFX+2))
F003E110          024 J 11 056 $  1
F003E120          004 P  6 032 ----      X=X.(10*(SFX+10))
F003E130          044 * 91 050 T  1      COMPUTE X/2PI
F003E140          025 * 92 001 *  3      X=REMAINDER IF X LESS THAN PI,
F003E150          050 T  1 020 P  5      OTHERWISE X=X-PI
F003E160          024 K  4 050 P  5      CHANGE SIGN
F003E170     *  3 020 T  1 040 * 92      X/PI IN MQ
```

```
F003E180      *  2 014 *  4 024 * 93      IF SFX LESS THAN -11, SET X=0
F003E190           006 *  5 004 K  0
F003E200      *  5 014 *  4 024 J 11      LOCATION (10*(SFX+11))
F003E210           056 * 15 020 P  6
F003E220      * 15 040 N  9 032 ----      X=((X/10*9).10*(11+SFX))/PI.10*10
F003E230      *  4 044 * 92 020 * 94      X=1/2-X FOR COS
F003E240           065 T  1 020 T  0      TRANSFER TO SINE OR COS
F003E250           002 *  6 020 * 94      FOR SINE X=1/2-ABSOLUTE VALUE
F003E260           027 T  1 050 T  1      OF T1 (1/2-X)
F003E270      *  6 004 T  1 033 T  1      -X*2
F003E280           077    2 050 T  2      -4(X*2)
F003E290           020 * 98 052 *  7      INITIALIZE SERIES FOR 1/4 SINE
F003E300      *  8 004 * 95 031 T  2
F003E310      *  7 024 ---- 050 P  6
F003E320           004 P  6 020 *  7
F003E330           024 K  2 052 *  7
F003E340           025 * 97 005 *  8
F003E350           032 T  1 050 P  6
F003E360           124 K  4 024 P  5
F003E370           070    9 050 P  5
F003E380           007 $  0 022 P  6
F003E390           073    2 003 * 10      TEST FOR OVERFLOW (RESULT=1)
F003E400           071   40 030 N  9
F003E410           050 P  6 025 N  9
F003E420           002 * 10 023 P  6      TEST FOR RESULT=0
F003E430           002 *  9 021 I  9
F003E440           050 P  4 010 S  1
F003E450      * 13 020 * 99 050 * 11      PRESET INSTRUCTION FOR NORMALIZATION
F003E460      * 11 020 P  6 025 N  8
F003E470           002 * 12 020 * 11
F003E480           025 K  1 014 * 13      NORMALIZE RESULT AND
F003E490      * 12 020 * 99 025 * 11
F003E500           050 T  1 024 J 11      ADJUST SCALE FACTOR
F003E510           056 * 14 021 T  1
F003E520           024 I  8 024 P  4
F003E530           073   31 050 P  4
F003E540      * 14 004 P  6 032 ----
F003E550           060 P  6 010 $  1
F003E560           020 $  0 010 = 30      PACK RESULT
F003E570      *  9 010 ----               EXIT
F003E580      * 10 020 P  5 024 N  8      RESULT IS ONE
F003E590           050 Q  3 010 *  9      COMBINE WITH SIGN AND EXIT
F003E620      * 91+ 62831853072   39      2 PI . 10*10
F003E630      * 92+ 31415926536   39      PI.10*10
F003E640      * 93+           11  39      ELEVEN
F003E650      * 94 040,0000,000,0000      1/2
F003E660      * 95 000,0000,003,5173      COEFFICIENTS A13,A11,...,A1
F003E670      * 96 177,7777,416,5664      TAKEN FROM ILLIAC SINE ROUTINE T5
F003E680           000,0025,016,6553
F003E690           177,6632,264,6777
F003E700           002,4315,361,4634
F003E710           153,2504,143,1673
F003E720           062,2077,325,0420      A1
F003E730      * 97 024 * 97 050 P  6      TEST WORD FOR END OF SERIES
F003E740      * 98      * 96      * 96
F003E750      * 99 020 P  6 025 N  8,     INITIALIZER FOR *11
F004E000      F  4 004 K  0 014 F  3,     COS(X)--SET FLAG IN MQ=0
F005E000+     F  5 010 $  0               IP(X)
F005E010           024 K  2 052 *  9
F005E020           020 $  0 010 = 11      UNPACK (Q3)
```

```
F005E030          023 Q  3 002 *  9      ZERO RESULT FOR ZERO ARGUMENT
F005E040          020 P  4 006 *  1
F005E050          120    0 050 Q  3      ZERO RESULT IF NEGATIVE SFX
F005E060    *  1 010 *  9 072    31
F005E070          050 T  0 025 I  8
F005E080          002 *  9 020 I  8      RESULT = ARGUMENT IF SFX BIG AS 8.
F005E090          025 T  0 024 J 11
F005E095          056 *  2 010 $  1
F005E100          056 *  3 120    0
F005E110    *  2 004 P  6 044 ----       CHOP FRACTIONAL PART
F005E120    *  3 000    0 032 ----
F005E130          060 P  6 010 $  1
F005E140          020 $  0 010 = 30      PACK RESULT INTO Q3
F005E150    *  9 010 ----          ,
F006E000+   F  6 010 $  0                FP(X)
F006E010          024 K  2 052 *  9
F006E020          020 $  0 010 = 12      PUSH OPERANDS
F006E030          020 $  0 010 F  5      IP(X)
F006E040          020 $  0 010 = 32      X-IP(X)=FP(X)
F006E050    *  9 010 ----          ,
F007E000+   F  7 010 $  0                SFX(X)
F007E010          024 K  2 052 *  9
F007E020          020 $  0 010 = 11      UNPACK (Q3)
F007E040          020 P  4 001 *  1
F007E050          120    0 014 *  2
F007E060    *  1 021 P  4 050 P  4
F007E070    *  2 020 K 12 050 P  5
F007E080          020 P  4 072    31
F007E090          050 P  6 025 N  1
F007E100          002 *  3 120    0
F007E110          004 N  8 010 *  4
F007E120    *  3 020 K 14 004 N  7
F007E130    *  4 050 P  4 032 P  6
F007E140          060 P  6 010 $  1
F007E150          020 $  0 010 = 30      PACK RESULT INTO Q3
F007E160    *  9 010 ----          ,
F008E000+   F  8 024 K  2 010 $  0       CF(X)
F008E010          052 *  9 020 Q  3
F008E020          125 K 11 050 Q  3
F008E030    *  9 010 ----          ,
F009E000+   F  9 010 $  0                SGN(X)
F009E010          024 K  2 052 *  9
F009E020          023 Q  3 002 *  9
F009E030          020 Q  3 124 K 12
F009E040          024 N  8 050 Q  3
F009E050    *  9 010 ----          ,
F011E000+   F 11 004 * 91 010 $  0       MAX
F012E000    F 12 004 * 92 010 $  0       MIN
F012E010          024 K  2 052 *  9
F012E020          060 *  2 010 $  1
F012E030          020 Q  6 001 E  6      MALFORMED IF NOT MULTIPLE ARGUMENT
F012E040    *  1 020 Q  3 050 H 21
F012E050          020 Q  7 050 H 22
F012E060          020 $  0 010 = 17
F012E070          020 $  0 010 = 70      COMPARE
F012E080    *  2 --- ---- --- ----
F012E090    *  3 020 H 22 050 Q  3
F012E100    *  4 020 Q  7 002 *  1
F012E110    *  9 010 ----
F012E120    * 91 071    4 001 *  4       FOR MAX
```

```
F012E130      * 92 071     6 001 *  4,    FOR MIN
G000E000+     G  0 010 $   0              SET (TEMPORARY) (NO ARRAYS YET)
G000E010           020 $   0 010 = 39     VERIFY SPACE AND ADVANCE
G000E020           052 *   2 124 K 34
G000E030           025 K  34 001 E  6     ERROR IF NOT LETTER
G000E040           020 $   0 010 =  2     ADVANCE TO NONSPACE
G000E050           124 A  43 025 A 43
G000E060           002 *   1 010 E  6     ERROR IF NOT '='
G000E070      *  1 020 $   0 010 =  2     ADVANCE TO NONSPACE
G000E080           020 $   0 010 = 21     EVALUATE EXPRESSION
G000E090           020 P   1 124 K 19
G000E100           025 K  19 001 E  6     ERROR IF NOT AT TERMINAL
G000E110           020 Q   3 010 $  1
G000E120      *  2 050 ---- 010 $  1      SET VALUE
G000E130           020 $   0 010 = 13     POP Q3
G000E140           010 X   5          ,
G001E000+     G  1 010 $   0              DO (TEMPORARY)
G001E005           020 Q   9 002 *  3
G001E006           020 Q  13 002 E 10     ERROR IF GIVEN DIRECTLY AND TASK OPEN
G001E010      *  3 020 $   0 010 = 39     VERIFY SPACE AND ADVANCE TO NONSPACE
G001E020           020 $   0 010 = 27     ACCUMULATE WORD
G001E030           025 W   7 050 T  0
G001E040           023 T   0 001 E  6     MALFORMED IF NOT 'PART'
G001E050           020 $   0 010 = 39     VERIFY SAPCE AND ADVANCE
G001E060           020 $   0 010 = 21     EVALUATE PART NUMBER
G001E070           020 $   0 010 = 44     VALIDATE PART NUMBER
G001E080           020 P   1 124 K 19     VERIFY TERMINAL
G001E090           025 K  19 001 E  6
G001E100           020 Q   3 050 P 20     SET CONTEXT
G001E110           020 $   0 010 = 13     POP Q3
G001E120           020 $   0 010 = 50     FIND PART
G001E130           001 E  17 052 *  1     ERROR IF CAN'T
G001E140           024 K   2 052 *  2
G001E150      *  1 020 ---- 050 P 16      SET P16 FOR FIRST STEP
G001E160           020 $   0 010 = 18     DESCEND
G001E170      *  2 020 ---- 050 Q  9      SET Q9 TO FIRST STEP
G001E180           010 X   6          ,   JUMP TO FETCH
G002E000+     G  2 010 $   0              TO
G002E010           020 Q   9 001 E 10     ERROR IF DIRECT
G002E020           020 $   0 010 = 39     VERIFY SPACE AND ADVANCE
G002E030           020 $   0 010 = 27     PICK UP WORD
G002E040           025 W   9 050 T  0
G002E050           023 T   0 001 *  1     JUMP IF NOT 'STEP'
G002E060           020 $   0 010 = 39     VERIFY SPACE AND ADVANCE
G002E070           020 $   0 010 = 21     EVALUATE EXPRESSION
G002E080           020 $   0 010 = 45     VALIDATE
G002E085           020 P   1 124 K 19     VERIFY TERMINAL
G002E086           025 K  19 001 E  6
G002E090           020 Q   3 050 P 21     SET CONTEXT
G002E100           020 $   0 010 = 13     POP Q3
G002E110           020 $   0 010 = 51     FIND STEP
G002E120           001 E  16 050 P 16     ERROR IF CAN'T ELSE SET P16
G002E125           020 P  21 050 Q  9     SET Q9
G002E130           010 X   6              JUMP TO FETCH
G002E140      *  1 020 H  19 025 W  7
G002E150           050 T   0 023 T  0
G002E160           002 *   2 010 E  6     ERROR IF NOT 'PART'
G002E170      *  2 020 $   0 010 = 39     VERIFY SPACE AND ADVANCE
G002E180           020 $   0 010 = 21     EVALUATE EXPRESSION
G002E190           020 $   0 010 = 44     VALIDATE
```

```
G002E195          020 P  1 124 K 19      VERIFY TERMINAL
G002E196          025 K 19 001 E  6
G002E200          020 Q  3 050 P 20      SET CONTEXT
G002E210          020 $  0 010 = 13      POP Q3
G002E220          020 $  0 010 = 50      FIND PART
G002E230          001 E 17 052 *  3      ERROR IF CAN'T
G002E235          024 K  2 052 *  4
G002E240     *  3 020 ---- 050 P 16      ELSE SET P16 FOR FIRST STEP
G002E245     *  4 020 ---- 050 Q  9      SET Q9
G002E250          010 X  6          ,    JUMP TO FETCH
G003E000+    G  3 010 $  0               DONE
G003E010          020 $  0 010 = 38      SIMPLE-INDIRECT TEST
G003E020          010 X  7          ,    JUMP TO ASCEND
G004E000+    G  4 010 $  0               CANCEL
G004E010          020 $  0 010 = 37      SIMPLE-DIRECT TESTS
G004E020          020 Q 12 050 Q  8
G004E030          020 Q 13 050 Q  9
G004E040          020 K  4 050 Q 12
G004E050          050 Q 13 010 $  1
G004E060          020 K  4 050 P 17      RESET STOP FLAG
G004E070          020 Q  8 001 *  5
G004E080     *  1 052 *  2 052 *  3
G004E090     *  2 020 ---- 001 *  4
G004E100          050 T  0 004 T  0
G004E110          020 $  0 010 = 29
G004E120     *  3 020 ---- 010 *  1
G004E130     *  4 004 Q  8 010 $  1
G004E140          020 $  0 010 = 28
G004E150          020 $  0 010 = 19
G004E160          020 Q  8 002 *  1
G004E170     *  5 020 $  0 010 = 55      SWITCH TO USER
G004E180          020 $  0 010 = 53      KICK OUT PROGRAM
G004E190          010 X  1          ,
G005E000+    G  5 010 $  0               ERASE TEMPORARY
G005E010          020 $  0 010 = 39      VERIFY SPACE AND ADVANCE
G005E020          124 K 34 025 K 34
G005E030          001 E  6 010 $  1
G005E040          020 P  1 052 *  1
G005E050          020 $  0 010 =  2      ADVANCE TO NONSPACE
G005E060          124 K 19 025 K 19
G005E070          001 E  6 020 K  4
G005E080     *  1 050 ---- 010 X  5,
G006E000+    G  6 010 $  0               TYPE
G006E010          020 $  0 010 = 39      VERIFY SPACE AND ADVANCE TO NONSPACE
G006E020          020 P  2 050 * 90
G006E030          020 $  0 010 = 27      ACCUMULATE WORD
G006E040          025 W  0 050 T  0
G006E050          023 T  0 002 *  1      JUMP IF 'ALL'
G006E060          004 H 19 010 $  1
G006E070          020 W  7 065 T  0
G006E080          023 T  0 002 *  5      JUMP IF 'PART'
G006E090          020 W  9 065 T  0
G006E100          023 T  0 002 * 10      JUMP IF 'STEP'
G006E110          020 W  3 065 T  0
G006E120          023 T  0 002 * 15      JUMP IF 'FORM'
G006E130          020 W 13 065 T  0
G006E140          023 T  0 002 * 20      JUMP IF 'SIZE'
G006E150          020 P  1 124 A 66
G006E160          025 A 66 002 * 30      JUMP IF QUOTES
G006E170          010 * 50               JUMP TO EVALUATION SECTION
```

```
G006E180      *  1 020 $  0 010 = 39      VERIFY SPACE AND ADVANCE TO NONSPACE
G006E190          020 $  0 010 = 27      ACCUMULATE WORD
G006E200          004 H 19 010 $  1
G006E210          020 W  8 065 T  0
G006E220          023 T  0 002 * 35      JUMP IF 'PARTS'
G006E230          020 W 10 065 T  0
G006E240          023 T  0 002 * 35      JUMP IF 'STEPS'
G006E250          020 W  4 065 T  0
G006E260          023 T  0 002 * 40      JUMP IF 'FORMS'
G006E270          010 E  6               ELSE MALFORMED
G006E280      *  5 020 $  0 010 = 39      VERIFY SPACE AND ADVANCE
G006E290          020 $  0 010 = 21      EVALUATE PART NUMBER
G006E300          020 $  0 010 = 44      VALIDATE PART NUMBER
G006E310          020 $  0 010 * 25      VERIFY TERMINAL
G006E320          020 Q  3 050 P 20
G006E330          020 $  0 010 = 13      POP Q3
G006E340          020 $  0 010 = 50      FIND PART
G006E350          002 *  8 010 E 17      ERROR IF CAN'T
G006E360      *  8 020 H 23 050 P 19
G006E370      *  6 020 P 19 052 *  7
G006E380      *  7 020 ---- 001 X  5      TO ADVANCE WHEN DONE
G006E390          050 P 19 056 P  3
G006E400          020 $  0 010 =  7      UNPACK TO R
G006E410          020 $  0 010 = 67      R TO S
G006E415          020 P 15 002 X 13      JUMP IF INTERRUPT
G006E420          020 $  0 010 = 25      TYPE
G006E430          010 *  6
G006E440      * 10 020 $  0 010 = 39      VERIFY SPACE AND ADVANCE
G006E450          020 $  0 010 = 21      EVALUATE STEP NUMBER
G006E460          020 $  0 010 = 45      VALIDATE STEP NUMBER
G006E470          020 $  0 010 * 25      VERIFY TERMINAL
G006E480          020 Q  3 050 P 21
G006E490          020 $  0 010 = 13      POP Q3
G006E500          020 $  0 010 = 51      FIND STEP
G006E510          002 * 11 010 E 16      ERROR IF CAN'T
G006E520      * 11 020 H 26 056 P  3
G006E530          020 $  0 010 =  7      UNPACK TO R
G006E540          020 $  0 010 = 67      R TO S
G006E550          020 $  0 010 = 25      TYPE
G006E560          010 X  5               TO ADVANCE
G006E570      * 15 020 $  0 010 = 39      VERIFY SPACE AND ADVANCE
G006E580          020 $  0 010 = 21      EVALUATE FORM NUMBER
G006E590          020 $  0 010 = 46      VALIDATE FORM NUMBER
G006E600          020 $  0 010 * 25      VERIFY TERMINAL
G006E610          020 Q  3 050 P 22
G006E620          020 $  0 010 = 13      POP Q3
G006E630          020 $  0 010 = 52      FIND FORM
G006E640          002 * 11 010 E 19      ERROR IF CAN'T
G006E650      * 25 024 K  2 052 * 29      LOCAL SUBROUTINE TO VERIFY TERMINAL
G006E660          020 P  1 124 K 19
G006E670          025 K 19 001 E  6
G006E680      * 29 010 ----
G006E700      * 20 020 $  0 010 =  3      ELIMINATE SPACES
G006E702          020 $  0 010 * 25      VERIFY TERMINAL
G006E704          120 ---- 0 050 H 20      COMPUTE SIZE
G006E706          020 * 22 056 * 21
G006E708      * 21          020 ----
G006E710          001 * 23 056 * 21
G006E712          020 H 20 024 I  1
G006E714          050 H 20 014 * 21
```

```
G006E716      * 22                  Q  0
G006E718      * 23 020 S  0 050 P 12
G006E720           020 $  0 010 = 36
G006E722                1          4
G006E724           020 A 42 050 P 11
G006E726           020 $  0 010 = 66
G006E728           020 $  0 010 = 25
G006E730           010 X  5
G006E750      * 30 020 S  0 050 P 12      SET OUTPUT FOR QUOTED MESSAGE
G006E752      * 31 020 $  0 010 =  1      ADVANCE
G006E754           001 E  6 124 A 66
G006E756           025 A 66 002 * 32      JUMP IF QUOTES
G006E758      * 33 020 $  0 010 =  4
G006E760           010 * 31
G006E762      * 32 020 P  2 050 * 90      SAVE P2
G006E764           020 $  0 010 =  2      ADVANCE TO NON-SPACE
G006E766           020 * 90 050 P  2      RESTORE P2
G006E768           020 P  1 124 K 19
G006E770           025 K 19 002 * 34      JUMP IF END
G006E772           020 A 66 050 P  1
G006E774           010 * 33
G006E776      * 34 020 A 42 050 P 11      FINISH LINE
G006E778           020 $  0 010 = 66
G006E780           020 $  0 010 = 25      TRANSMIT
G006E782           010 X  5
G006E800      * 35 020 $  0 010 =  3      ELIMINATE SPACES
G006E802           124 K 19 025 K 19
G006E804           002 $  1 010 E  6      ERROR IF NOT END
G006E806           020 Q 10 050 P 18
G006E808      * 36 020 A 42 050 S  1      TYPE PARTS
G006E810           020 $  0 010 = 25
G006E812           020 P 18 056 * 37
G006E814      * 37                  020 ----
G006E816           001 X  5 050 P 18
G006E818           050 P 19 010 $  1
G006E820      * 38 020 P 19 052 * 39      TYPE STEPS WITHIN PART
G006E822      * 39 020 ---- 001 * 36
G006E824           050 P 19 056 P  3
G006E826           020 $  0 010 =  7
G006E828           020 $  0 010 = 67
G006E829           020 P 15 002 X 13      JUMP IF INTERRUPT
G006E830           020 $  0 010 = 25
G006E832           010 * 38
G006E850      * 40 023 Q 11 002 * 45      TYPE ALL FORMS
G006E852           020 Q 11 050 P 19
G006E854           020 A 42 050 S  1
G006E856           020 $  0 010 = 25      LINE FEED
G006E858      * 41 020 P 19 052 * 42
G006E860           024 K  2 052 * 43
G006E862      * 42 020 ---- 001 * 45
G006E864           050 P 19 056 P  3
G006E866           020 S  0 050 P 12
G006E868           020 $  0 010 = 64
G006E870                * 46      * 46
G006E872      * 43 020 ---- 050 T  0
G006E874           124 K 13 050 T  1
G006E876           020 T  0 124 K 11
G006E878           070    10 050 T  2
G006E880           020 J 14 025 T  2
G006E882           052 * 44 010 $  1
```

```
G006E884          120 ---- 0 004 T  1
G006E886    * 44 044 ---- 060 H 20
G006E888          020 $  0 010 = 36
G006E890              1          9
G006E892          020 A 80 050 P 11
G006E894          020 $  0 010 = 66
G006E896          020 A 42 050 P 11
G006E898          020 $  0 010 = 66
G006E899          020 P 15 002 X 13      JUMP IF INTERRUPT
G006E900          020 $  0 010 = 25
G006E902          020 $  0 010 =  7
G006E904          020 $  0 010 = 67
G006E906          020 $  0 010 = 25
G006E908          010 * 41
G006E910    * 45  020 A 42 050 S  1
G006E912          020 $  0 010 = 25      LINE FEED
G006E914          010 X  5
G006E916    * 46  053,0461,222,2016      'FORM'
G006E918          100,0000,000,0000
G006G000    * 50  020 * 90 050 P  2      TYPE VALUES
G006G010          052 * 51 010 $  1
G006G020    * 51  020 ---- 050 P  1      RESTORE P1,P2
G006G030    * 52  020 $  0 010 = 21      EVALAUTE EXP
G006G040          020 P  1 124 A 59
G006G050          025 A 59 001 * 53      JUMP IF NOT COMMA
G006G060          020 $  0 010 = 14
G006G070          020 P  1 050 Q  5      PUSH COMMA INTO OPERATOR STACK
G006G080          020 $  0 010 =  1      ADVANCE
G006G090          010 * 52               LOOP
G006G100    * 53  020 P  1 124 K 19
G006G110          025 K 19 001 E  6      JUMP IF NOT TERMINAL
G006G120    * 54  020 $  0 010 = 13      POP Q3 TILL CLEAN
G006G130          020 Q  2 002 * 54
G006G140          020 Q  4 001 * 56
G006G150    * 55  020 $  0 010 = 15      POP Q5 TILL CLEAN
G006G160          020 Q  4 002 * 55
G006G170    * 56  020 * 90 050 P  2
G006G180          052 * 57 010 $  1
G006G190    * 57  020 ---- 050 P  1      RESTORE P1,P2
G006G200    * 58  020 P 15 002 X 13      JUMP IF INTERRUPT
G006G210          020 $  0 050 P 12
G006G220          020 $  0 010 = 21      EVALUATE EXPRESSION
G006G230          020 $  0 010 = 11      UNPACK
G006G240          020 $  0 010 = 13      POP Q3
G006G250          020 P  2 050 P 18
G006G260          020 * 90 050 P  2
G006G270          052 * 59 010 $  1
G006G280    * 59  020 ---- 050 P  1      RESTORE P1,P2
G006G290    * 60  020 $  0 010 =  4      COPY CHARACTER TO OUTPUT
G006G300          020 $  0 010 =  1
G006G310          020 P  2 025 P 18
G006G320          001 * 60 010 $  1      REPEAT UNTIL EXPRESSION IS COPIED
G006G330          020 P 12 025 * 91
G006G340          001 * 61 010 $  1      JUMP IF ROOM FOR VALUE ON SAME LINE
G006G350          020 A 42 050 P 11
G006G360          020 $  0 010 = 66
G006G370          020 $  0 010 = 25
G006G380          020 S  0 050 P 12
G006G390    * 61  020 $  0 010 = 62      INSERT SPACE
G006G400          020 A 43 050 P 11
```

```
G006G410          020 $  0 010 = 66      INSERT '='
G006G420          020 $  0 010 = 62      INSERT SPACE
G006G430          023 P  6 001 * 62      JUMP IF NOT ZERO
G006G440          020 $  0 010 = 62      INSERT SPACE
G006G450          020 A 48 050 P 11
G006G460          020 $  0 010 = 66      INSERT ZERO
G006G470          010 * 66
G006G480     * 62 021 P  5 001 * 63
G006G490          020 A 60 014 * 63
G006G500     * 63 020 A 44 050 P 11
G006G510          020 $  0 010 = 66      INSERT SIGN
G006G520          120    0 004 P  6
G006G530          044 N  8 050 H 20
G006G540          060 P 11 010 $  1
G006G550          020 $  0 010 = 66      INSERT FIRST DIGIT
G006G560          020 A 27 050 P 11
G006G570          020 $  0 010 = 66      INSERT DECIMAL POINT
G006G580          020 $  0 010 = 36
G006G590                  3        8
G006G600          023 P  4 002 * 66      JUMP IF SFX = 0
G006G610          020 $  0 010 = 64
G006G620              * 92     * 92
G006G630          020 P  4 001 * 64
G006G640          020 A 60 014 * 64
G006G650     * 64 020 A 44 050 P 11
G006G660          020 $  0 010 = 66      INSERT SIGN OF SFX
G006G670          020 P  4 006 * 65
G006G680     * 65 021 P  4 072   31
G006G690          050 H 20 010 $  1
G006G700          020 $  0 010 = 36      INSERT SFX DIGITS
G006G710                  3        2
G006G720          020 A112 050 P 11
G006G730          020 $  0 010 = 66      INSERT RIGHT PAREN
G006G740     * 66 020 A 42 050 P 11
G006G750          020 $  0 010 = 66      INSERT CR+EOM
G006G760          020 $  0 010 = 25      TRANSMIT
G006G770          020 P  1 124 K 19
G006G780          025 K 19 002 X  5      JUMP IF AT TERMINAL
G006G790          020 $  0 010 =  2      ADVANCE TO NONSPACE
G006G800          020 P  2 050 * 90
G006G810          010 * 58
G006G820     * 90
G006G830     * 91      S 50      S 50
G006G840     * 92 020,0011,407,6111      '.10*('
G006G850          100,0000,000,0000,
G007E000+    G  7 010 $  0               RETURN
G007E010          020 $  0 010 = 38      SIMPLE-INDIRECT TESTS
G007E020          020 A 42 050 S  1      SET UP CR+EOM
G007E030          020 $  0 010 = 25      TRANSMIT
G007E040          010 X  5         ,
G008E000+    G  8 010 $  0               PAGE
G008E010          020 $  0 010 = 38      SIMPLE-INDIRECT TESTS
G008E020          020 $  0 010 = 22
G008E030          010 X  5         ,
G011E000+    G 11 010 $  0               FORM
G011E010          020 $  0 010 = 39      VERIFY SPACE AND ADVANCE
G011E015          020 Q  9 002 E  9      ERROR IF INDIRECT
G011E020          020 $  0 010 = 20      CONVERT FORM NUMBER
G011E030          020 $  0 010 = 46      VALIDATE
G011E040          020 Q  3 050 P 22      SET CONTEXT
```

```
G011E050          020 $  0 010 = 13      POP Q3
G011E060          020 $  0 010 = 3       ELIMINATE SPACES
G011E070          124 A 80 025 A 80
G011E080          002 *  1 010 E  6      MALFORMED IF NOT COLON
G011E090      *   1 020 $  0 010 = 1       STEP ONE CHARACTER
G011E100          001 *  2 010 E  6      MALFORMED IF NOT TERMINAL MINUS
G011E110      *   2 020 I  1 050 P 13      SET SUBSTATE FOR FORM
G011E120          020 $  0 010 = 55      SWITCH TO USER
G011E130          020 $  0 010 = 53      KICK OUT PROGRAM
G011E140          010 X  1         ,     JUMP TO Q SERVICE
G012E000+     G 12 010 $  0               STOP
G012E010          020 $  0 010 = 38      SIMPLE-INDIRECT TESTS
G012E020          120    0 050 P 17      SET 'STOPPED' FLAG
G012E030          020 S  0 050 P 12
G012E040          020 $  0 010 = 64      PREPARE MESSAGE
G012E050             * 90     * 90
G012E060          020 $  0 010 = 63      COPY STEP NUMBER FROM R TO S
G012E070          020 A 27 050 P 11
G012E080          020 $  0 010 = 66      ADD PERIOD
G012E090          020 A 42 050 P 11
G012E100          020 $  0 010 = 66      ADD CR+EOM
G012E110          020 $  0 010 = 25      SEND 'STOP' MESSAGE
G012E120          020 $  0 010 = 55      SWITCH TO USER
G012E130          020 $  0 010 = 53      KICK OUT PROGRAM
G012E140          010 X  1
G012E150      * 90 071,0631,142,3447      'STOPPED BY STEP'
G012E160          012,4240,341,1070
G012E170          007,0621,461,2447
G012E180          007,2000,000,0000,
G013E000+     G 13 010 $  0               GO
G013E010          020 $  0 010 = 37      SIMPLE-DIRECT TESTS
G013E020          020 Q 12 050 Q  8
G013E030          020 Q 13 050 Q  9
G013E040          020 K  4 050 Q 12
G013E050          050 Q 13 010 $  1
G013E060          020 P 17 001 *  1      JUMP IF NOT 'STOP'
G013E070          020 K  4 050 P 17      RESET 'STOP' FLAG
G013E080          010 X  5               JUMP TO ADVANCE
G013E090      *   1 020 Q  9 001 *  3      JUMP IF NO OPEN TASK
G013E100          050 P 21 010 $  1
G013E110          020 $  0 010 = 51      FIND CURRENT STEP
G013E120          002 *  2 010 $  1      JUMP IF FOUND
G013E130          020 H 23 001 X  7      JUMP TO ASCEND IF DONE
G013E140          020 H 26 001 X  7      JUMP TO ASCEND IF DONE
G013E150      *   2 050 P 16 010 X  6      JUMP TO FETCH
G013E160      *   3 020 $  0 010 = 55      SWITCH TO USER
G013E170          020 $  0 010 = 53      KICK OUT PROGRAM
G013E180          010 X  1         ,     BACK TO Q SERVICE
G014E000+     G 14 010 $  0               DELETE
G014E010          020 Q  9 002 E  9      ERROR IF INTERNAL
G014E020          020 $  0 010 = 39      VERIFY SPACE AND ADVANCE
G014E030          020 $  0 010 = 27      ACCUMULATE WORD
G014E040          025 W  0 050 T  0
G014E050          023 T  0 001 * 60      JUMP IF NOT 'ALL'
G014E060          020 $  0 010 = 39      VERIFY SPACE AND ADVANCE
G014E160          020 $  0 010 = 27      ACCUMULATE WORD
G014E170          004 H 19 010 $  1
G014E180          020 W  8 065 T  0
G014E190          023 T  0 002 * 10      JUMP IF 'PARTS'
G014E200          020 W 10 065 T  0
```

```
G014E210          023 T  0 002 * 10      JUMP IF 'STEPS'
G014E220          020 W  4 065 T  0
G014E230          023 T  0 002 * 20      JUMP IF 'FORMS'
G014E240          010 E  6               ELSE MALFORMED
G014E250      * 10 020 $  0 010 * 50      VERIFY TERMINAL
G014E260          020 $  0 010 * 30      DELETE ALL STEPS
G014E270          010 X  5
G014E280      * 20 020 $  0 010 * 50      VERIFY TERMINAL
G014E290          020 $  0 010 * 40      DELETE ALL FORMS
G014E300          010 X  5
G014E310      * 30 024 K  2 052 * 39      LOCAL SUBR TO DEL ALL STEPS (HENCE PARTS)
G014E320          020 Q 10 050 * 90
G014E330      * 31 020 * 90 056 * 32
G014E340          124 K  5 025 K  1
G014E350      * 32 005 * 36 020 ----
G014E360          050 * 90 050 * 91
G014E370      * 33 020 * 91 052 * 34
G014E380          124 K  6 025 K  1
G014E390          005 * 35 010 $  1
G014E400      * 34 004 ---- 060 * 91
G014E410          020 $  0 010 = 29
G014E420      * 35 010 * 33 004 * 90
G014E430          020 $  0 010 = 28
G014E440      * 36 010 * 31 004 Q 10
G014E450          020 $  0 010 = 29
G014E460          120    0 050 Q 10
G014E470      * 39 010 ----
G014E480      * 40 024 K  2 052 * 49      LOCAL SUBR TO DELETE ALL FORMS
G014E490          020 Q 11 050 * 91
G014E500      * 41 020 * 91 052 * 42
G014E505          124 K  6 025 K  1
G014E510          005 * 43 010 $  1
G014E515      * 42 004 ---- 060 * 91
G014E520          020 $  0 010 = 29
G014E525      * 43 010 * 41 004 Q 11
G014E530          020 $  0 010 = 28
G014E535          120    0 050 Q 11
G014E540      * 49 010 ----
G014E545      * 50 024 K  2 052 * 59      LOCAL SUBR TO VERIFY TERMINAL
G014E550          020 $  0 010 =  3       ELIMINATE SPACES
G014E555          124 K 19 025 K 19
G014E560      * 59 002 ---- 010 E  6      MALFORMED IF NOT TERMINAL
G014E565      * 60 020 $  0 010 = 39      VERIFY SPACE AND ADVANCE
G014E570          004 H 19 010 $  1
G014E575          020 W  7 065 T  0
G014E580          023 T  0 002 * 65      JUMP IF 'PART'
G014E585          020 W  9 065 T  0
G014E590          023 T  0 002 * 70      JUMP IF 'STEP'
G014E595          020 W  3 065 T  0
G014E600          023 T  0 002 * 75      JUMP IF 'FORM'
G014E605          010 E  6               ELSE MALFORMED
G014E610      * 65 020 $  0 010 = 21      EVALUATE PART NUMBER
G014E615          020 $  0 010 = 44      VALIDATE PART NUMBER
G014E620          020 $  0 010 * 50      VERIFY TERMINAL
G014E625          020 Q  3 050 P 20
G014E630          020 $  0 010 = 13      POP Q3
G014E635          020 $  0 010 = 50      FIND PART
G014E640          002 $  1 010 E 17      ERROR IF CAN'T
G014E645          020 H 23 050 * 91
G014E650      * 66 020 * 91 052 * 67
```

```
G014E655          124 K  6 025 K  1
G014E660          001 *  68 010 $  1
G014E665      * 67 004 ---- 060 *  91
G014E670          020 $  0 010 =  29
G014E675          010 *  66
G014E680      * 68 004 H 22 075    21
G014E685          020 $  0 010 =  28
G014E690          020 H 21 056 *  69
G014E695      * 69 020 H 23 056 ----
G014E700          010 X  5                    EXIT
G014E705      * 70 020 $  0 010 =  21         EVALUATE STEP NUMBER
G014E710          020 $  0 010 =  45          VALIDATE STEP NUMBER
G014E715          020 $  0 010 *  50          VERIFY TERMINAL
G014E720          020 Q  3 050 P  21
G014E725          020 $  0 010 =  13          POP Q3
G014E730          020 $  0 010 =  51          FIND STEP
G014E735          002 *  71 010 E  16         ERROR IF CAN'T
G014E740      * 71 020 $  0 010 =  56          ERASE STEP A/C H21-H26.
G014E745          010 X  5                    EXIT
G014E750      * 75 020 $  0 010 =  21         EVALUATE FORM NUMBER
G014E755          020 $  0 010 =  46          VALIDATE FORM NUMBER
G014E760          020 $  0 010 *  50          VERIFY TERMINAL
G014E765          020 Q  3 050 P  22
G014E770          020 $  0 010 =  13          POP Q3
G014E775          020 $  0 010 =  52          FIND FORM
G014E780          002 *  76 010 E  19         ERROR IF CAN'T
G014E781      * 76 020 $  0 010 =  57          ERASE FORM A/C H24-H26.
G014E782          010 X  5
G014E785      * 90                           LOCAL STORAGE
G014E790      * 91                           LOCAL STORAGE
H---E000+     H  0                           LAST 16 BIT CLOCK READING
H---E001      H  1+      1080000    39       TEMPORARY INITIAL TIME
H---E002      H  2 000,4021,300,1401         TEMPORARY DATE
H---E003      H  3 026,0100,070,0000         TEMPORARY DATE
H---E004      H  4 100 0000 000 0000         STN DUE FOR Q MSG
H---E005      H  5                           NEXT DECK NUMBER
H---E006      H  6                           NEXT STATISTICAL CARD NUMBER
H---E007      H  7      ----      ----       STN IN CORE
H---E008      H  8                           START TIME OF CURRENT SHOT
H---E009      H  9                           SCR CONTENTS
H---E010      H 10         0         0       NEXT AVAILABLE BUFFER
H---E011      H 11         0         0       NEXT AVAILABLE DRUM
H---E012      H 12      ----      ----       STN
H---E013      H 13      ----      ----       BUFFER
H---E014      H 14      ----      ----       DRUM
H---E015      H 15 100 0000 000 0000         += 'OF' DELAYED A/C READING
H---E016      H 16 100 0000 000 0000         += 'OF' DELAYED A/C PUNCHING
H---E017      H 17
H---E018      H 18
H---E019      H 19                           ACCUMULATED CHARACTERS RESULT OF '27
H---E020      H 20                           INTEGER FOR OUTPUT
H---E021      H 21                           SEARCH POINTER
H---E022      H 22                           SEARCH POINTER
H---E023      H 23                           SEARCH POINTER
H---E024      H 24                           SEARCH POINTER
H---E025      H 25                           SEARCH POINTER
H---E026      H 26                           SEARCH POINTER
I---E000+     I  0              0            SMALL INTEGERS
I---E001      I  1              1
I---E002      I  2              2
```

```
   I---E003     I  3                          3
   I---E004     I  4                          4
   I---E005     I  5                          5
   I---E006     I  6                          6
   I---E007     I  7                          7
   I---E008     I  8                          8
   I---E009     I  9                          9
   J---E000+    J  0                         60      MINUTES PER HOUR
   J---E001     J  1                       1800      COUNTS PER MINUTE
   J---E002     J  2                         55      LINE NUMBER FOR EJECT
   J---E003     J  3
   J---E004     J  4                         60      SHOT TIME FOR PROCESSING
   J---E005     J  5                        120      SHOT TIME FOR READING
   J---E006     J  6                        120      SHOT TIME FOR PUNCHING
   J---E007     J  7                        450      OVERDUE TIME FOR PROCESSING
   J---E008     J  8                        300      OVERDUE TIME FOR READING
   J---E009     J  9                        300      OVERDUE TIME FOR PUNCHING
   J---E010     J 10
   J---E011     J 11      N  0      N  0
   J---E012     J 12      R  0      R  0
   J---E013     J 13      A  0      A  0
   J---E014     J 14      N  8      N  8
   J---E015     J 15
   J---E016     J 16                          4      CHOKE NUMBER
   J---E017     J 17                          1      UNCHOKE NUMBER
   K---E000+    K  0 000,0000,000,0000             ZERO   (GENERAL CONSTANTS)
   K---E001     K  1 000,0000,000,0001             ADDRESS MODIFIER
   K---E002     K  2 000,0001,000,0000             ADDRESS MODIFIER
   K---E003     K  3 000,0001,000,0001             ADDRESS MODIFIER
   K---E004     K  4 100,0000,000,0000             SIGN BIT
   K---E005     K  5 000,0000,000,7777             EXTRACTOR RIGHT ADDRESS
   K---E006     K  6 000,7777,000,0000             EXTRACTOR LEFT ADDRESS
   K---E007     K  7 000,7777,000,7777             EXTRACTOR BOTH ADDRESSES
   K---E008     K  8 000,0000,177,0000             MASK INPUT MESSAGES
   K---E009     K  9 000,0000,020,0000             COUNTER FOR EXTENDED CLOCK
   K---E010     K 10 000,0000,033,0000             MASK FOR DRUM SECTION CODE
   K---E011     K 11 077,6000,000,0000             EXTRACTOR SFX
   K---E012     K 12 000,1000,000,0000             EXTRACTOR SIGN(CF)
   K---E013     K 13 000,0777,777,7777             EXTRACTOR MAG(CF)
   K---E014     K 14 000,2000,000,0000             UNIT FOR SFX (AND TEST IN =27)
   K---E015     K 15 177,7777,777,7777             ALL ONES
   K---E016     K 16 000,0017,000,0000             BUFFER MASK
   K---E017     K 17 000,0000,000,0177             CHARACTER CODE MASK AND STN MASK.
   K---E018     K 18 000,0017,000,0177             BUFFER + STN MASK
   K---E019     K 19 000,0000,000,0400             BIT OF DISTINCTION
   K---E020     K 20 000,0000,000,0017             EXTRACTOR NUMERIC CODE
   K---E021     K 21 040,0000,000,0000             EN BIT
   K---E022     K 22 020,0000,000,0000             DS BIT
   K---E023     K 23 010,0000,000,0000             RO BIT
   K---E024     K 24 004,0000,000,0000             TL BIT OR GREATER
   K---E025     K 25 002,0000,000,0000             CL BIT OR EQUAL
   K---E026     K 26 001,0000,000,0000             SU BIT OR LESS
   K---E027     K 27 000,0000,400,0000             *      OR FUNCTION OR SUBSCRIPT
   K---E028     K 28 000,0000,200,0000             MPY, /
   K---E029     K 29 000,0000,100,0000             ON BIT OR +, -
   K---E030     K 30 000,0000,040,0000             OF BIT OR ARITH OPERATOR
   K---E031     K 31 000,0000,020,0000             TC BIT OR LEFT GROUPER
   K---E032     K 32 000,0000,010,0000             RI BIT OR RIGHT GROUPER
   K---E033     K 33 000,0000,004,0000             RC BIT OR RELATION
   K---E034     K 34 000,0000,002,0000             EJ BI) OR LETTER
```

```
K---E035    K 35 000,0000,001,0000      TO BIT OR DIGIT
K---E036    K 36 003,0000,000,0000      CL+SU
K---E037    K 37 000,0000,700,0000      PRECEDENCE BITS
K---E038    K 38
K---E039    K 39
K---E040    K 40 000,0000,177,0000      MAJOR CYCLE MSGS
K---E041    K 41 000,0000,177,0000      MINOR CYCLE MSGS
L---E000+   L  0          0         0   Q FOR ENABLE
L---E001    L  1 100 0000 000 0000      L ENABLED AND IDLE
L---E002    L  2 100 0000 000 0000      Q FOR JOSS ID MESSAGE
L---E003    L  3 100 0000 000 0000      Q FOR JOSS ID MESSAGE, RI ALREADY NOTED
L---E004    L  4 100 0000 000 0000      L ON AND WAITING FOR RI
L---E005    L  5 100 0000 000 0000      Q FOR DRUM ASSIGNMENTS
L---E006    L  6 100 0000 000 0000      L DRAINING BUFFERS BEFORE CL+SU
L---E007    L  7 100 0000 000 0000      Q FOR INPUT BUFFER AND CL+SU
L---E008    L  8 100 0000 000 0000
L---E009    L  9 100 0000 000 0000
L---E010    L 10 100 0000 000 0000      L GREEN (P13 HOLDS SUBSTATE)
L---E011    L 11 100 0000 000 0000      Q FOR BUFFER AND PREFERRED PROCESSING
L---E012    L 12 100 0000 000 0000      Q FOR PROCESSING
L---E013    L 13 100 0000 000 0000      S PROCESSING
L---E014    L 14 100 0000 000 0000      L CHOKED
L---E015    L 15 100 0000 000 0000
L---E016    L 16 100 0000 000 0000      Q FOR CARD READER
L---E017    L 17 100 0000 000 0000      S WAITING FOR READER-READY SIGNAL
L---E018    L 18 100 0000 000 0000      S READING CARDS
L---E019    L 19 100 0000 000 0000      Q FOR CARD PUNCH
L---E020    L 20 100 0000 000 0000      S PUNCHING CARDS
N---E000+   N  0+                1  39  POWERS OF TEN.
N---E001    N  1+               10  39
N---E002    N  2+              100  39
N---E003    N  3+             1000  39
N---E004    N  4+            10000  39
N---E005    N  5+           100000  39
N---E006    N  6+          1000000  39
N---E007    N  7+         10000000  39
N---E008    N  8+        100000000  39
N---E009    N  9+       1000000000  39
N---E010    N 10+      10000000000  39
N---E011    N 11+     100000000000  39
P---E000+   P  0                    --  LINE NUMBER ON TYPEWRITER PAGE
P---E001    P  1 --- ---- --- ----      PRESENT CHARACTER WORD IN R
P---E002    P  2 ----------- ----      ADDRESS OF PRESENT CHARACTER WORD IN R
P---E003    P  3                ----    LOCN FOR PACKING OR UNPACKING
P---E004    P  4                        SFX   (EXTENDED TO SIGN POSITION)
P---E005    P  5                        SIGN   (EXTRACTED)
P---E006    P  6                        CF   (MAGNITUDE)
P---E007    P  7                        SFX   (EXTENDED TO SIGN POSITION)
P---E008    P  8                        SIGN   (EXTRACTED)
P---E009    P  9                        CF   (MAGNITUDE)
P---E010    P 10 ----------- ----      LOCN OF SPACE AFER 'IF', ELSE -1.
P---E011    P 11 --- ---- --- ----      PRESENT CHARACTER WORD IN S
P---E012    P 12 ----------- ----      ADDRESS OF PRESENT CHARACTER WORD INS
P---E013    P 13                   2    SUBSTATE OF L10 0=INST 1=FORM 2=INITIALS
P---E014    P 14 010 ----              LINK FROM =25 (OUTPUT LINE)
P---E015    P 15 100 0000 000 0000      + = RI WHILE IN L11 OR L14.
P---E016    P 16 100 0000 000 0000      + = LOCN OF CURRENT STEP IN RT ADDR
P---E017    P 17 100 0000 000 0000      + = STOPPED (USED BY G13 'GO')
P---E018    P 18                        WORK SPACE FOR G6 TYPE
P---E019    P 19                        WORK SPACE FOR G6 TYPE
```

```
P---E020    P 20                                PART (CONTEXT)
P---E021    P 21                                STEP (CONTEXT)
P---E022    P 22                                FORM (CONTEXT)
Q---E000+   Q  0 100 0000 000 0000              AVAILABLE SPACE PDL LINK
Q---E001    Q  1 --- ---- --- ----              USER'S INITIALS
Q---E002    Q  2 100 0000 000 0000              OPERAND PDL LINK
Q---E003    Q  3 100 0000 000 0000              OPERAND
Q---E004    Q  4 100 0000 000 0000              OPERATOR PDL LINK
Q---E005    Q  5 100 0000 000 0000              OPERATOR
Q---E006    Q  6 100 0000 000 0000              AUXILIARY PDL LINK
Q---E007    Q  7 100 0000 000 0000              AUXILIARY ITEM
Q---E008    Q  8 100 0000 000 0000              CONTROL PDL LINK
Q---E009    Q  9 100 0000 000 0000              CONTROL (STEP NUMBER)
Q---E010    Q 10 000 0000 000 0000              LIST OF STEPS (AND PARTS)
Q---E011    Q 11 000 0000 000 0000              LIST OF FORMS
Q---E012    Q 12 100 0000 000 0000              HIDEOUT FOR (Q8)
Q---E013    Q 13 100 0000 000 0000              HIDEOUT FOR (Q9)
R---E000+   R  0 100 0000 000,0000              R BLOCK LOWER LIMIT
S---E000    S  0     S  0     S  0              ORIGIN-OUTPUT-BLOCK-S0-S73
U000E000+   U  0     $  0     $  0              STN TABLE -- STATE
U000E001    *  0     L  0     L  0
U000E002    *  1     L  0     L  0
U000E003    *  2     L  0     L  0
U000E004    *  3     L  0     L  0
U000E005    *  4     L  0     L  0
U000E006    *  5     L  0     L  0
U000E007    *  6     L  0     L  0
U000E008    *  7     L  0     L  0
U000E009    *  8     L  0     L  0
U000E010    *  9     L  0     L  0
U000E011    * 10     L  0     L  0
U000E012    * 11     L  0     L  0
U000E013    * 12     L  0     L  0
U000E014    * 13     L  0     L  0
U000E015    * 14     L  0     L  0
U000E016    * 15     L  0     L  0,
U001E000+   U  1     $  0     $  0              STN TABLE -- CURRENT BUFFER
U001E001    *  0 100 0000 000 0000
U001E002    *  1 100 0000 000 0000
U001E003    *  2 100 0000 000 0000
U001E004    *  3 100 0000 000 0000
U001E005    *  4 100 0000 000 0000
U001E006    *  5 100 0000 000 0000
U001E007    *  6 100 0000 000 0000
U001E008    *  7 100 0000 000 0000
U001E009    *  8 100 0000 000 0000
U001E010    *  9 100 0000 000 0000
U001E011    * 10 100 0000 000 0000
U001E012    * 11 100 0000 000 0000
U001E013    * 12 100 0000 000 0000
U001E014    * 13 100 0000 000 0000
U001E015    * 14 100 0000 000 0000
U001E016    * 15 100 0000 000 0000,
U002E000+   U  2     $  0     $  0              STN TABLE -- DRUM ASSIGNMENT
U002E001    *  0 100 0000 000 0000
U002E002    *  1 100 0000 000 0000
U002E003    *  2 100 0000 000 0000
U002E004    *  3 100 0000 000 0000
U002E005    *  4 100 0000 000 0000
U002E006    *  5 100 0000 000 0000
```

```
U002E007     *  6 100 0000 000 0000
U002E008     *  7 100 0000 000 0000
U002E009     *  8 100 0000 000 0000
U002E010     *  9 100 0000 000 0000
U002E011     * 10 100 0000 000 0000
U002E012     * 11 100 0000 000 0000
U002E013     * 12 100 0000 000 0000
U002E014     * 13 100 0000 000 0000
U002E015     * 14 100 0000 000 0000
U002E016     * 15 100 0000 000 0000,
U003E000+    U  3     $  0     $  0     STN TABLE -- NEXT STN IN SAME STATE
U003E001     *  0        1        1
U003E002     *  1        2        2
U003E003     *  2        3        3
U003E004     *  3        4        4
U003E005     *  4        5        5
U003E006     *  5        6        6
U003E007     *  6        7        7
U003E008     *  7        8        8
U003E009     *  8        9        9
U003E010     *  9       10       10
U003E011     * 10       11       11
U003E012     * 11       12       12
U003E013     * 12       13       13
U003E014     * 13       14       14
U003E015     * 14       15       15
U003E016     * 15 100 0000 000 0000,
U004E000+    U  4     $  0     $  0     STN TABLE -- TIME OF LAST ACTIVITY
U004E001     *  0 100 0000 000 0000
U004E002     *  1 100 0000 000 0000
U004E003     *  2 100 0000 000 0000
U004E004     *  3 100 0000 000 0000
U004E005     *  4 100 0000 000 0000
U004E006     *  5 100 0000 000 0000
U004E007     *  6 100 0000 000 0000
U004E008     *  7 100 0000 000 0000
U004E009     *  8 100 0000 000 0000
U004E010     *  9 100 0000 000 0000
U004E011     * 10 100 0000 000 0000
U004E012     * 11 100 0000 000 0000
U004E013     * 12 100 0000 000 0000
U004E014     * 13 100 0000 000 0000
U004E015     * 14 100 0000 000 0000
U004E016     * 15 100 0000 000 0000,
U005E000+    U  5     $  0     $  0     BUFFER TABLE -- NEXT BUFFER IN LIST
U005E001     *  0        1        1
U005E002     *  1        2        2
U005E003     *  2        3        3
U005E004     *  3        4        4
U005E005     *  4        5        5
U005E006     *  5        6        6
U005E007     *  6        7        7
U005E008     *  7        8        8
U005E009     *  8        9        9
U005E010     *  9       10       10
U005E011     * 10       11       11
U005E012     * 11       12       12
U005E013     * 12       13       13
U005E014     * 13       14       14
U005E015     * 14       15       15
```

```
U005E016      * 15 100 0000 000 0000,
U006E000+     U  6      $  0      $  0      DRUM TABLE -- NEXT DRUM IN LIST
U006E001      *  0           1         1
U006E002      *  1           2         2
U006E003      *  2           3         3
U006E004      *  3 100 0000 000 0000,
U007E000+     U  7      $  0      $  0      DRUM TABLE -- FIRST CONTROL WORD
U007E001      *  0           0,000,0777
U007E002      *  1           0,002,0777
U007E003      *  2           0,020,0777
U007E004      *  3           0,022,0777,
V---E000+     V  0      V  0      V  0      ORIGIN  VARIABLES
V---E001      V  1 100 0000 000 0000      LCA
V---E002      V  2 100 0000 000 0000      LCB
V---E003      V  3 100 0000 000 0000      LCC
V---E004      V  4 100 0000 000 0000      LCD
V---E005      V  5 100 0000 000 0000      LCE
V---E006      V  6 100 0000 000 0000      LCF
V---E007      V  7 100 0000 000 0000      LCG
V---E008      V  8 100 0000 000 0000      LCH
V---E009      V  9 100 0000 000 0000      LCI
V---E010      V 10 100 0000 000 0000      LCJ
V---E011      V 11 100 0000 000 0000      LCK
V---E012      V 12 100 0000 000 0000      LCL
V---E013      V 13 100 0000 000 0000      LCM
V---E014      V 14 100 0000 000 0000      LCN
V---E015      V 15 100 0000 000 0000      LCO
V---E016      V 16 100 0000 000 0000      LCP
V---E017      V 17 100 0000 000 0000      LCQ
V---E018      V 18 100 0000 000 0000      LCR
V---E019      V 19 100 0000 000 0000      LCS
V---E020      V 20 100 0000 000 0000      LCT
V---E021      V 21 100 0000 000 0000      LCU
V---E022      V 22 100 0000 000 0000      LCV
V---E023      V 23 100 0000 000 0000      LCW
V---E024      V 24 100 0000 000 0000      LCX
V---E025      V 25 100 0000 000 0000      LCY
V---E026      V 26 100 0000 000 0000      LCZ
V---E027      V 27 100 0000 000 0000      A
V---E028      V 28 100 0000 000 0000      B
V---E029      V 29 100 0000 000 0000      C
V---E030      V 30 100 0000 000 0000      D
V---E031      V 31 100 0000 000 0000      E
V---E032      V 32 100 0000 000 0000      F
V---E033      V 33 100 0000 000 0000      G
V---E034      V 34 100 0000 000 0000      H
V---E035      V 35 100 0000 000 0000      I
V---E036      V 36 100 0000 000 0000      J
V---E037      V 37 100 0000 000 0000      K
V---E038      V 38 100 0000 000 0000      L
V---E039      V 39 100 0000 000 0000      M
V---E040      V 40 100 0000 000 0000      N
V---E041      V 41 100 0000 000 0000      O
V---E042      V 42 100 0000 000 0000      P
V---E043      V 43 100 0000 000 0000      Q
V---E044      V 44 100 0000 000 0000      R
V---E045      V 45 100 0000 000 0000      S
V---E046      V 46 100 0000 000 0000      T
V---E047      V 47 100 0000 000 0000      U
V---E048      V 48 100 0000 000 0000      V
```

```
V---E049      V 49 100 0000 000 0000      W
V---E050      V 50 100 0000 000 0000      X
V---E051      V 51 100 0000 000 0000      Y
V---E052      V 52 100 0000 000 0000      Z
W---E000+     W  0 000,0000,001,1414      ALL
W---E001      W  1 000,0000,405,0313      DECK
W---E002      W  2 000,0000,006,1722      FOR
W---E003      W  3 000,0000,617,2215      FORM
W---E004      W  4 000,0061,722,1523      FORMS
W---E005      W  5 000,0000,000,1106      IF
W---E006      W  6 000,0000,000,1116      IN
W---E007      W  7 000,0002,001,2224      PART
W---E008      W  8 000,0200,122,2423      PARTS
W---E009      W  9 000,0002,324,0520      STEP
W---E010      W 10 000,0232,405,2023      STEPS
W---E011      W 11 000,0000,000,2417      TO (LOWER CASE T)
W---E012      W 12 002,6011,425,0523      VALUES
W---E013      W 13 000,0002,311,3205      SIZE
X001E000+     X  1 010 $  0               QUEUE SERVICE
X001E010      *  1 020 L  0 001 *  2      TO *2 IF NO L0 QUEUE FOR ENABLE
X001E020            050 H 12 024 K 21     SERVICE L0 BY ENABLING
X001E030            142 K 21 010 $  1
X001E040            020 $  0 010 = 47     CHANGE STATE TO L1.
X001E050               L  1     L  1
X001E060            010 *  1
X001E070      *  2 020 H 10 001 *  4      TO *4 IF NO BUFFERS AVAILABLE
X001E080            020 L  3 001 *  3     TO *3 IF NO L3 QUEUE FOR JOSS ID
X001E090            050 H 12 010 $  1
X001E095            120    0 050 P  0
X001E100            020 $  0 010 = 65     SEND JOSS ID MESSAGE
X001E110               *  90    *  90
X001E120            020 $  0 010 = 47     CHANGE STATE TO L5.(Q FOR DRUM)
X001E130               L  5     L  5
X001E140            020 H  4 002 *  2
X001E150            020 H 12 050 H  4     ARRANGE FOR Q MSG
X001E160            010 *  2
X001E170      *  3 020 L  2 001 *  4      TO *4 IF NO L2 QUEUE FOR JOSS ID.
X001E180            050 H 12 010 $  1
X001E185            120    0 050 P  0
X001E190            020 $  0 010 = 65     SEND JOSS ID MESSAGE
X001E200               *  90    *  90
X001E210            020 $  0 010 = 47     CHANGE STATE TO L4.
X001E220               L  4     L  4
X001E230            010 *  2
X001E240      *  4 020 H 11 001 *  7      TO *7 IF NO DRUMS AVAILABLE
X001E250            020 L  5 001 *  7     TO *7 IF NO L5 Q FOR DRUM
X001E260            050 H 12 025 H  4
X001E270            001 *  6 025 K  1
X001E280            002 *  6 020 H  4     DELETE STN FROM H4 IF NOW THERE.
X001E290            024 U  3 052 *  5
X001E300      *  5 020 ---- 050 H  4
X001E310      *  6 020 $  0 010 = 60     INITIALIZE DRUM FOR STN(H12).(P13)=2.
X001E320            020 $  0 010 = 47     CHANGE STATE TO L10.
X001E330               L 10     L 10
X001E335            120    0 050 P  0
X001E340            020 $  0 010 = 65     SEND INITIALS REQUEST MESSAGE
X001E350               *  91    *  91
X001E360            020 $  0 010 = 55
X001E370            020 $  0 010 = 53
X001E380            010 *  4
```

```
X001E390      *  7 020 H  4 001 *  20      TO *20 IF NO Q MSGS DUE
X001E400           020 H 10 001 *  20      TO *20 IF NO BUFFER AVAILABLE
X001E410           020 H  4 050 H  12
X001E420           024 U  3 052 *   8
X001E430      *  8 020 ---- 050 H   4      ADVANCE H4
X001E440           020 S  0 050 P  12      COMPOSE Q MESSAGE
X001E450           020 $  0 010 = 61       TIME
X001E460           020 $  0 010 = 64
X001E470                *  92    *  92
X001E480           120    0 050 H  20
X001E490           020 L  5 050 T   0
X001E500      *  9 020 H 20 024 I   1
X001E510           050 H 20 020 T   0
X001E520           025 H 12 001 *  10
X001E530           025 K  1 001 *  12
X001E540      * 10 020 T  0 024 U   3
X001E550           052 *  11 010 $   1
X001E560      * 11 020 ---- 050 T   0
X001E570           002 *  9 134 $   0
X001E580      * 12 020 $  0 010 = 36
X001E590                   1         2
X001E600           020 $  0 010 = 64
X001E610                *  93    *  93
X001E615           120    0 050 P   0
X001E620           020 $  0 010 = 25       SEND Q MESSAGE
X001E630           010 *  7
X001E640      *  90 063,4510,523,1062      MESSAGE
X001E650           007,1012,626,2501       'PRESS 'IN' FOR JOSS SERVICE.'
X001E660           007,0261,142,4416
X001E670           060,5463,447,1016
X001E680           031,0251,223,2431
X001E690           011,4250,662,5200
X001E700      *  91 063,4430,521,0462      MESSAGE
X001E710           012,4161,463,4047       'PLEASE TYPE YOUR INITIALS.'
X001E720           012,4161,602,3064
X001E730           024,4160,622,2431
X001E740           031,4310,422,1462
X001E750           015,4524,000,0000
X001E760      *  92 007,1701,143,2016      MESSAGE COMPONENTS
X001E770           010,4510,520,7045       ' YOU ARE NOW NUMBER'
X001E780           023,0660,342,2464
X001E790           022,0220,522,4416
X001E800           100,0000,000,0000
X001E810      * 93 007,0311,120,7063       MESSAGE COMPONENT
X001E820           014,0250,342,4064       ' IN THE QUEUE.'
X001E830           012,4640,521,5452
X001E840           100,0000,000,0000
X001E850      * 20 010 X  2          ,      SERVE OPERATOR MSG (TEMP INCOMPLETE)
X002E000+     X  2 010 $  0                 SIGNAL SERVICE
X002E010           120    0 145 K  40
X002E020           001 X  3 050 H   9       TO X3 IF NO MESSAGES. SET H9.
X002E030           124 K  5 050 H  12
X002E040           071    21 052 H  12      SET H12
X002E050           020 $  0 010 =   0       READ CLOCK
X002E060           020 H 12 024 U   4       UPDATE ACTIVITY
X002E070           052 *  1 020 H   1
X002E080      *  1 050 ---- 010 $   1
X002E090           020 H  9 124 K  35       TO *10 IF 'TO'
X002E100           025 K 35 002 *  10
X002E110           020 H  9 124 K  31       TO *20 IF 'TC'
```

```
X002E120          025 K 31 002 * 20
X002E130          020 H  9 124 K 29      TO *30 IF 'ON'
X002E140          025 K 29 002 * 30
X002E150          020 H  9 124 K 30      TO *40 IF 'OF'
X002E160          025 K 30 002 * 40
X002E170          020 $  0 010 = 54      BRING IN PROGRAM FOR STN
X002E180          020 H  9 124 K 33      TO *50 IF 'RC'
X002E190          025 K 33 002 * 50
X002E200          020 H  9 124 K 34      TO *60 IF 'EJ'
X002E210          025 K 34 002 * 60
X002E220          020 H  9 124 K 32      TO *70 IF 'RI'
X002E230          025 K 32 002 * 70
X002E240          130 $  0
X002E250     * 10 020 $  0 010 = 48      'TO' TREAT BY SUBROUTINE
X002E260          010 X  2
X002E270     * 20 020 H 12 024 K 26      'TC' SWITCH BACK TO USER
X002E280          142 K 26 010 X  2      (SU ERASES TC)
X002E290     * 30 020 H 12 142 K 29      'ON' ERASE SIGNAL
X002E300          020 $  0 010 = 47      CHANGE STATE TO L2
X002E310              L  2     L  2
X002E320          010 X  1               GO BACK TO Q SERVICE
X002E330     * 40 020 H 12 142 K 30      'OF' ERASE SIGNAL
X002E340          024 U  0 052 * 41
X002E350     * 41 020 ---- 124 K  5
X002E360          025 * 91 001 * 42
X002E370          025 I  2 001 * 45      TO *45 IF L17 OR L18
X002E380          025 I  1 001 * 42
X002E390          025 I  1 001 * 46      TO *46 IF L20
X002E400     * 42 020 H  4 025 H 12      ADJUST H4 IF NECESSARY
X002E410          001 * 44 025 K  1
X002E420          002 * 44 020 H  4
X002E430          024 U  3 052 * 43
X002E440     * 43 020 ---- 050 H  4
X002E450     * 44 020 $  0 010 = 58      RELEASE BUFFERS (IF ANY)
X002E460          002 * 44 010 $  1
X002E470          020 $  0 010 = 59      RELEASE DRUM (IF ANY)
X002E475          020 $  0 010 = 47      CHANGE STATE TO L0
X002E476              L  0     L  0
X002E480          010 X  2
X002E490     * 45 120    0 050 H 15      SET DELAYED 'OF' FLAG FOR READING
X002E500          010 X  2
X002E510     * 46 120    0 050 H 16      SET DELAYED 'OF' FLAG FOR PUNCHING
X002E520          010 X  2
X002E530     * 50 020 H 12 142 K 33      'RC' ERASE SIGNAL
X002E540          020 $  0 010 = 47      CHANGE STATE TO L13
X002E550              L 13     L 13
X002E552          020 H  9 141 R  1      READ BUFFER
X002E554          020 $  0 010 = 58      RELEASE BUFFER
X002E556          020 $  0 010 =  5      CONVERT
X002E560          020 P  0 024 I  1
X002E570          050 P  0 025 J  2
X002E580          001 * 51 010 $  1      TO *51 IF MORE ROOM ON THIS PAGE
X002E590          020 $  0 010 = 22      TYPE TIME LINE AT TOP OF NEXT PAGE
X002E600     * 51 010 X 10               TO X10 FOR INPUT PROCESSING
X002E610     * 60 020 H 12 142 K 34      'EJ' ERASE SIGNAL
X002E612          020 $  0 010 = 47      CHANGE STATE TO L13
X002E613              L 13     L 13
X002E615          020 H  9 141 R  1      READ BUFFER
X002E616          020 $  0 010 = 58      RELEASE BUFFER
X002E617          020 $  0 010 =  5      CONVERT
```

```
X002E620          020 $  0 014 = 22     TYPE TIME LINE
X002E630          010 X 10              TO X10 FOR INPUT PROCESSING
X002E640    * 70  020 H 12 142 K 32     'RI' RESET
X002E642          024 U  0 052 * 71
X002E644    * 71  020 ---- 124 K  5
X002E646          025 * 90 001 * 72
X002E648          025 I  1 001 * 74     TO *74 IF L2
X002E650          025 I  1 001 * 72
X002E652          025 I  1 001 * 75     TO *75 IF L4
X002E654          025 I  1 001 * 72
X002E656          025 I  2 001 * 73     TO *73 IF L6 OR L7
X002E658          025 I  3 001 * 72
X002E660          025 I  1 001 * 78     TO *78 IF L11
X002E662          025 I  1 001 X 13     TO X13 IF L12
X002E664          025 I  1 001 * 72
X002E666          025 I  1 001 * 78     TO *78 IF L14
X002E668          025 I  1 001 * 72
X002E670          025 I  1 001 X 13     TO X13 IF L16
X002E672          025 I  2 001 * 73     TO *73 IF L17 OR L18
X002E674          025 I  1 001 X 13     TO X13 IF L19
X002E676          025 I  1 001 * 73     TO *73 IF L20
X002E678    * 72 130 $  0               HALT INCONSISTENT STATE
X002E680    * 73  020 $  0 010 = 53     KICK OUT PROGRAM
X002E682          010 X  2              BACK TO SIGNAL SERVICE
X002E684    * 74  020 $  0 010 = 47     CHANGE STATE TO L3
X002E686                L  3      L  3
X002E688          010 X  2              BACK TO SIGNAL SERVICE
X002E690    * 75  020 H  4 002 * 76     SET H4 IF NECESSARY
X002E692          020 H 12 050 H  4
X002E694    * 76  020 $  0 010 = 47     CHANGE STATE TO L5
X002E696                L  5      L  5
X002E698          010 X  1              BACK TO Q SERVICE
X002E730    * 78  120 ----  0 050 P 15  SET RI FLAG FOR L11, L14
X002E732          020 $  0 010 = 53     KICK OUT PROGRAM
X002E734          010 X  2              BACK TO SIGNAL SERVICE
X002E900    * 90                L  2
X002E902    * 91                L 17,
X003E000+   X  3 010 $  0               TASK SELECTION
X003E010          011 *  1 010 *  2     TO *1 IF 'READER READY', ELSE TO *2
X003E020    *  1 020 L 17 001 *  2     TO *2 IF NO STN REALLY WAITING FOR READE
X003E030          020 L 20 001 X  8     TO X8 IF NO STN PUNCHING (INITIALIZE REA
X003E040    *  2 020 $  0 010 =  0     READ CLOCK
X003E050          020 L 18 001 *  3     TO *3 IF NO STN READING
X003E060          024 U  4 052 $  1
X003E070          020 ---- 024 J  8
X003E080          025 H  1 005 X  8     TO X8 IF READING IS OVERDUE.
X003E090          010 *  5
X003E100    *  3 020 L 16 002 * 20     TO *20 IF STN WTG TO READ
X003E110          020 L 20 001 *  4     TO *4 IF NO STN PUNCHING
X003E120          024 U  4 052 $  1
X003E130          020 ---- 024 J  9
X003E140          025 H  1 005 X  9     TO X9 IF PUNCHING IS OVERDUE
X003E150          010 *  5
X003E160    *  4 020 L 19 001 *  5     TO *5 IF NO STN WTG TO PUNCH
X003E170          020 L 18 001 X  9     TO X9 IF NO STN READING (INITIALIZE PUNCH
X003E180    *  5 020 L 12 001 *  6     TO *6 IF NO STN WTG FOR NORMAL PROCESSIN
X003E190          024 U  4 052 $  1
X003E200          020 ---- 024 J  7
X003E210          025 H  1 001 * 40     TO *40 IF STN OVERDUE FOR PROCESSING
X003E220    *  6 020 L 11 001 *  7     TO *7 IF NO STN WTG FOR BUFFER
```

```
X003E230           020 H 10 002 * 50      TO *50 IF BUFFER AVAILABLE
X003E240      *  7 020 L 12 002 * 40      TO *40 IF STN WTG FOR PROCESSING
X003E250           020 L 18 006 X  8      TO X8 IF STN READING
X003E260           020 L 20 006 X  9      TO X9 IF STN PUNCHING
X003E270           010 X  1               NOTHING TO DO,GO BACK TO Q SERVICE
X003E280      * 20 105    0 011 $  0      LOOP TILL T1 TURNED OFF
X003E500      * 40 020 L 12 050 H 12      SET STN CONTEXT
X003E510           020 $  0 010 = 47      CHANGE STATE TO L13
X003E520               L 13     L 13
X003E530           020 $  0 010 = 54      BRING IN PROGRAM
X003E540           020 $  0 010 = 49      UPDATE STN ACTIVITY
X003E550           050 H  8 010 X  4      SET START TIME AND FIRE
X003E560      * 50 020 L 11 050 H 12      SET STN CONTEXT
X003E570           020 $  0 010 = 47      CHANGE STATE TO L13
X003E580               L 13     L 13
X003E590           020 $  0 010 = 54      BRING IN PROGRAM
X003E600           020 $  0 010 = 49      UPDATE STN ACTIVITY
X003E610           050 H  8 014 = 25,     SET START TIME AND REENTER =25
X004E000+     X  4 010 $  0               FIRE
X004E010           020 P 10 001 *  1      JUMP IF UNCONDITIONAL
X004E020           050 P  2 010 $  1
X004E030           020 A 14 050 P  1
X004E040           020 $  0 010 = 23      EVALUATE CONDITION
X004E050           002 *  1 010 X  5      JUMP TO ADVANCE IF DOESN'T HOLD
X004E060      *  1 020 J 12 050 P  2      FIND VERB
X004E070      *  2 020 P  2 024 K  3
X004E080           050 P  2 052 *  3
X004E090      *  3 020 ---- 001 E  6
X004E100           050 P  1 124 K 34
X004E110           025 K 34 001 *  2
X004E120           020 $  0 010 = 27      INTERPRET VERB
X004E130           020 * 50 052 * 11
X004E140      * 11 020 ---- 001 E  6
X004E150           025 H 19 001 * 12
X004E160           025 K  1 001 * 13
X004E170      * 12 020 * 11 024 K  2
X004E180           052 * 11 010 * 11
X004E190      * 13 020 * 11 025 * 50
X004E200           024 * 51 052 *  9
X004E210      *  9 010 ----               GO TO GXXX
X004E220      * 50      * 60
X004E230      * 51      G  0
X004E240      * 60 000,0000,063,0524      SET
X004E250           000,0000,000,4417      DO
X004E260           000,0000,000,6417      TO
X004E270           000,0004,417,1605      DONE
X004E280           004,3011,603,0514      CANCEL
X004E290           000,0452,201,2305      ERASE
X004E300           000,0006,431,2005      TYPE
X004E310           006,2052,425,2216      RETURN
X004E320           000,0006,001,0705      PAGE
X004E330           000,0006,205,0104      READ
X004E340           000,0602,516,0310      PUNCH
X004E350           000,0004,617,2215      FORM
X004E360           000,0006,324,1720      STOP
X004E370           000,0000,000,4717      GO
X004E380           004,4051,405,2405      DELETE
X004E390           100,0000,000,0000,
X005E000+     X  5 010 $  0               ADVANCE
X005E010           020 P 16 002 *  1      TO *1 IF P16 STILL GOOD
```

```
X005E020          020 Q  9 001 *  5
X005E030          050 P 21 010 $  1
X005E040          020 $  0 010 = 51      FIND CURRENT STEP
X005E050          002 *  1 010 $  1      TO *1 IF FOUND
X005E060          020 H 23 001 X  7      TO X7 (ASCEND) IF DONE.
X005E070          020 H 26 001 X  7      TO X7 (ASCEND) IF DONE.
X005E080          020 H 25 010 *  1      GIMMICK TO GET NEXT
X005E090     *  1 052 *  2 010 $  1
X005E095          024 K  2 052 *  6
X005E096     *  6 020 ---- 050 Q  9      SET Q9
X005E100     *  2 020 ---- 001 *  4
X005E110     *  3 050 P 16 010 X  6      SET P16 FOR NEXT STEP AND GO TO X6(FETCH
X005E120     *  4 050 P 16 010 X  7      SET P16 AND GO TO X7 (ASCEND)
X005E130     *  5 020 S  0 010 = 55      SWITCH TO USER--TASK COMPLETE
X005E140          020 $  0 010 = 53      KICK OUT PROGRAM
X005E150          010 X  1         ,     GO BACK TO Q SERVICE.
X006E000+    X  6 010 $  0               FETCH
X006E005          020 P 15 002 X 13      JUMP IF INTERRUPT
X006E010          020 P 16 056 P  3
X006E020          020 S  0 010 =  7      UNPACK
X006E030          020 $  0 010 =  0      READ CLOCK
X006E040          025 H  8 025 J  4
X006E050          002 *  2 120    0      JUMP IF SHOT TIME IS UP
X006E060     *  1 145 K 41 001 X  4      JUMP UNLESS MINOR CYCLE MSG
X006E070          050 H  9 124 K 35
X006E080          025 K 35 001 *  2      JUMP IF NOT 'TO'
X006E090          020 $  0 010 = 48      TREAT 'TO' BY SUBROUTINE
X006E095          020 L 13 050 H 12      RESTORE H12 CONTEXT
X006E100          120    0 010 *  1
X006E110     *  2 020 L 13 050 H 12      CUT OFF THIS SHOT
X006E120          020 $  0 010 = 47      CHANGE STATE TO L12
X006E130             L 12    L 12
X006E140          020 $  0 010 = 53      KICK OUT
X006E150          010 X  1         ,     JUMP TO Q SERVICE
X007E000+    X  7 010 $  0               ASCEND          (TEMPORARY)
X007E010          020 K  4 050 P 16      RESET P16 FLAG
X007E020          020 $  0 010 = 19      POP Q8-9 CONTROL STACK
X007E030          020 Q  8 002 *  1      TO *1 IF TASK NOT FINISHED
X007E040          020 $  0 010 = 55      SWITCH TO USER
X007E050          020 $  0 010 = 53      KICK OUT PROGRAM
X007E060          010 X  1               TO X1 FOR QUEUE SERVICE
X007E070     *  1 010 X  5         ,     TO X5 ADVANCE (TEMPORARY TO GET GOING)
X010E000+    X 10 010 $  0               PROCESS INPUT
X010E040          023 P 13 002 * 10      JUMP IF WTG FOR INSTRUCTION
X010E050          024 I  1 002 * 30      JUMP IF WTG FOR FORM
X010E060          010 * 40               ELSE JUMP WTG FOR INITIALS
X010E070     * 10 020 R  1 001 * 50      JUMP IF LINE WAS EMPTY
X010E080          020 P  1 124 A124
X010E090          025 A124 002 * 50      JUMP IF LINE ENDS WITH '*'
X010E100          020 R  1 124 A124
X010E110          025 A124 002 * 50      JUMP IF LINE BEGINS WITH '*'
X010E120          020 R  1 124 K 34
X010E130          025 K 34 002 * 11      JUMP IF LINE BEGINS WITH LETTER
X010E140          020 R  1 124 K 35
X010E150          025 K 35 002 * 13      JUMP IF LINE BEGINS WITH DIGIT
X010E160          010 E  6               ELSE MALFORMED STEP
X010E170     * 11 020 P 10 002 E 15      ERROR IF CONDITIONAL
X010E180          020 P  1 124 A 27
X010E190          025 A 27 001 * 19
X010E200          020 P  2 056 * 12      REPLACE FINAL PERIOD
```

```
X010E205      * 12 020 A 109 050 ----
X010E220      * 19 010 X   4                FIRE
X010E230      * 13 020 P   2 052 * 14       SET LAST CHARACTER ADDRESSES
X010E240          056 * 15 010 $   1
X010E250          020 J 12 050 P   2
X010E260          020 $   0 010 =   1
X010E270          020 $   0 010 = 20        EVALUATE STEP NUMBER
X010E280          020 $   0 010 = 45        VALIDATE
X010E285          020 P   1 001 $   2       SKIP IF NOW AT TERMINAL MINUS
X010E290          020 $   0 010 = 39        VERIFY SPACE
X010E300          020 Q   3 050 P 21        SET STEP NUMBER IN CONTEXT
X010E305          020 $   0 010 = 13        POP Q3
X010E310      * 14 020 ---- 124 A 27
X010E315          025 A 27 001 * 16         JUMP IF LAST CHARACTER NOT PERIOD
X010E320      * 15 020 A 130 050 ----        REPLACE FINAL PERIOD BY PSEUDO-PERIOD
X010E325      * 16 020 P 10 001 * 17         JUMP IF NO CONDITION
X010E330          025 I   3 056 $   1
X010E340          120     0 050 ----        REPLACE SP BEFORE 'IF' BY ZERO
X010E350      * 17 020 $   0 010 = 51        FIND STEP IF CAN
X010E360          001 * 18 010 $   1        JUMP IF CAN'T
X010E370          020 $   0 010 = 56        ERASE DUPLICATE
X010E380          020 $   0 010 = 51        FIND PLACE AGAIN
X010E390      * 18 020 $   0 010 =   6       PACK STRING
X010E400          020 H 23 002 * 20         JUMP IF PART EXISTS
X010E410          020 $   0 010 = 68        COMPUTE PART (P20) FOR STEP (P21)
X010E420          020 H 21 056 * 21         SET UP PART ID IN Q10 STRUCTURE
X010E430          020 $   0 010 = 10
X010E440      * 21 056 * 22 056 ----
X010E450          024 K   1 056 * 23
X010E460      * 22 020 H 22 056 ----
X010E470      * 23 020 P 20 050 ----
X010E480          020 * 22 124 K   5
X010E490          071     21 050 H 24
X010E500          120     0 050 H 25
X010E510      * 20 020 H 24 052 * 24        SET UP STEP IN Q10 STRUCTURE
X010E520          020 $   0 010 = 10
X010E530          071     21 052 * 26
X010E540      * 24 052 ---- 024 K   2
X010E550          052 * 25 020 P 21
X010E560      * 25 050 ---- 020 H 25
X010E570          124 K   6 024 P   3
X010E580      * 26 050 ---- 010 * 50
X010E590      * 30 120     0 050 P 13       CLEAR 'WTG FOR FORM' SIGNAL
X010E600          020 $   0 010 = 52        FIND FORM IF CAN A/C P22
X010E610          001 * 31 010 $   1        JUMP IF CAN'T
X010E620          020 $   0 010 = 57        ERASE DUPLICATE
X010E630          020 $   0 010 = 52        FIND PLACE AGAIN
X010E640      * 31 020 $   0 010 =   6       PACK STRING
X010E650          020 H 24 052 * 32
X010E660          020 $   0 010 = 10
X010E670          071     21 052 * 34
X010E680      * 32 052 ---- 024 K   2
X010E690          052 * 33 020 P 22
X010E700      * 33 050 ---- 020 H 25
X010E710          124 K   6 024 P   3
X010E720      * 34 050 ---- 010 * 50
X010E740      * 40 020 R   1 001 * 45       JUMP IF NO CHARACTERS
X010E750          020 R   5 002 * 45        JUMP IF 5 OR MORE CHARACTERS
X010E760          020 K 15 050 Q   1        PRIME Q1
X010E770      * 41 020 P   1 124 K 34
```

```
X010E780          025 K 34 001 * 45      JUMP IF NOT LETTER
X010E790          020 P  1 004 Q  1
X010E800          075    32 050 Q  1
X010E810          020 P  2 025 K  3
X010E820          054 P  2 052 * 42
X010E830     * 42 020 ---- 050 P  1
X010E840          006 * 41 010 $  1      GO BACK IF MORE
X010E850          120    0 050 P 13      CLEAR 'WTG FOR INITIALS' SIGNAL
X010E860          020 $  0 010 = 22      TYPE TIME LINE AT TOP OF PAGE
X010E870          010 * 50
X010E880     * 45 020 $  0 010 = 65      ERROR, TYPE 'AGAIN'
X010E890              * 46      * 46
X010E900          010 * 50
X010E910     * 46 050,4270,421,4445      'AGAIN.'
X010E920          015,4524,000,0000
X010E930     * 50 020 $  0 010 = 55      SWITCH TO USER
X010E935          020 $  0 010 = 53      KICK OUT PROGRAM
X010E940          010 X  2         ,     BACK TO SIGNAL SERVICE
X013E000+    X 13 010 $  0               INTERRUPT TREATMENT
X013E010          020 Q  9 001 *  1
X013E020          020 $  0 050 P 12
X013E030          020 $  0 010 = 64
X013E040              * 90      * 90
X013E050          020 P 16 056 P  3      P16 OK HERE
X013E060          020 $  0 010 =  7      UNPACK
X013E070          020 $  0 010 = 63      COPY STEP NUMBER FROM R TO S.
X013E080          020 $  0 010 = 69      INSERT PERIOD AND END OF MSG
X013E090          020 $  0 010 = 25      SEND
X013E100          010 *  2
X013E110     *  1 020 $  0 010 = 65
X013E120              * 91      * 91
X013E130     *  2 020 $  0 010 = 55      SWITCH TO USER
X013E140          020 $  0 010 = 53      KICK OUT PROGRAM
X013E150          010 X  2               BACK TO SIGNAL SERVICE
X013E160     * 90 054,4451,461,2451      'INTERRUPTED AT STEP'
X013E170          024,4641,163,1425
X013E180          012,0160,423,1416
X013E190          031,0630,522,3416
X013E200          100,0000,000,0000
X013E210     * 91 064,4251,522,3042      'REVOKED BY IN-REQUEST.'
X013E220          012,4240,341,1070
X013E230          007,1313,122,6051
X013E240          012,4501,501,2462
X013E250          031,4331,250,0000,
Z---E000+    Z  0 100 0000 000 0000      USER'S INITIALS
Z---E001     Z  1 100 0000 000 0000      LAST DECK NUMBER CALLED FOR
=000E000+    =  0 010 $  0               READ CLOCK
=000E010          024 K  2 052 *  9
=000E020          107    0 004 H  0
=000E030          050 H  0 065 T  0
=000E040          002 *  1 024 *  90
=000E050     *  1 024 H  1 050 H  1
=000E060     *  9 010 ----               EXITS WITH TIME IN ACC ALSO
=000E070     * 90+            1   23,
=001E000     =  1 010 $  0               STEP ONE CHARACTER
=001E010          024 K  2 052 *  9
=001E020          020 P  2 024 K  3
=001E030          052 *  1 054 P  2
=001E040     *  1 020 ---- 050 P  1      LEAVES NEW CHARACTER IN ACC
=001E050     *  9 010 ----         ,
```

```
=002E000      =  2 024 K  2 010 $  0      ADVANCE TO NONSPACE
=002E010         052 *  9 010 *  1
=003E000      =  3 010 $  0               ELIMINATE SPACES
=003E010         024 K  2 052 *  9
=003E020         020 P  1 010 *  3
=003E030      *  1 020 P  2 024 K  3
=003E040         052 *  2 054 P  2
=003E050      *  2 020 ---- 050 P  1
=003E060      *  3 124 A 14 025 A 14
=003E070         002 *  1 020 P  1
=003E080      *  9 010 ----                ,  LEAVES NONSPACE IN ACC
=004E000+     =  4 010 $  0                  PUT (P1) IN NEXT S CELL (P12)+1 AND P11
=004E010         024 K  2 052 *  9
=004E020         020 P 12 024 K  3
=004E030         050 P 12 056 *  1
=004E040         020 P  1 010 $  1
=004E050      *  1 050 P 11 050 ----
=004E060      *  9 010 ----                ,
=005E000+     =  5 010 $  0               CONVERTER
=005E010         024 K  2 052 *  9
=005E020      *  1 020 A 42 050 R 73      FILL WITH CR+EOM+TERMINAL FROM RIGHT
=005E030         020 *  1 010 *  3
=005E040      *  2 020 *  4 010 $  1
=005E050      *  3 025 K  1 056 *  4
=005E060      *  4 056 *  6 020 ----
=005E070         024 J 13 056 *  5
=005E080      *  5 020 A 14 124 ----
=005E090         025 A 14 001 * 10
=005E100      *  6 020 A 42 050 ----
=005E110         010 *  2
=005E120      * 10 020 J 12 050 P  2
=005E130      * 11 020 K  4 050 P 10      CLEAR 'IF'
=005E140      * 12 020 $  0 010 * 20
=005E150      * 13 020 P  1 124 A 14      SPACE
=005E160         025 A 14 001 * 12
=005E170         020 $  0 010 * 20
=005E180         020 P  1 124 A 25      I
=005E190         025 A 25 001 * 13
=005E200         020 $  0 010 * 20
=005E210         020 P  1 124 A 22      F
=005E220         025 A 22 001 * 13
=005E230         020 $  0 010 * 20
=005E240         020 P  1 124 A 14      SPACE
=005E250         025 A 14 001 * 12
=005E260         020 P  2 050 P 10      RECORD 'IF' BY LOCN OF FOLLOWING SPACE
=005E270      * 14 020 $  0 010 * 20
=005E280         020 P  1 124 A 66
=005E290         025 A 66 001 * 14
=005E300         010 * 11
=005E310      * 20 024 K  2 052 * 29
=005E320         020 P  2 024 K  3
=005E330         050 P  2 056 * 21
=005E340      *  21 056 * 23 020 ----
=005E350         024 J 13 056 * 22
=005E360      *  22 001 * 24 020 ----
=005E370      * 23 050 P  1 050 ----
=005E380      *  29 010 ----               EXIT FROM *20
=005E390      * 24 020 P  2 025 K  3
=005E400         050 P  2 052 * 25
=005E410      * 25 020 ---- 050 P  1
```

```
=005E420    *  9 010 ----                    EXIT WITH LAST CHAR IN CONTEXT AND ACC
=006E000+   =  6 010 $  0                    PACKER
=006E010          024 K  2 052 *  9          LEAVES LINK TO STRING IN P3 AND ACC.
=006E020          120    0 050 P  3          SET P3.
=006E030          020 R  1 001 *  8          TO *8 IF VACUOUS
=006E040          020 Q  0 056 P  3          SET P3.
=006E050          020 J 12 050 P  2
=006E060          020 $  0 010 =  1
=006E070    *  1 020 $  0 010 = 10
=006E080          056 *  3 024 K  1
=006E090          056 *  2 070    0
=006E100          020 $  0 010 * 20
=006E110          020 $  0 010 * 20
=006E120          020 $  0 010 * 20
=006E130          020 $  0 010 * 20
=006E140          060 T  0 071   12
=006E150          050 T  1 070    0
=006E160          020 $  0 010 * 20
=006E170          020 $  0 010 * 20
=006E180          020 $  0 010 * 20
=006E190          020 $  0 010 * 20
=006E200          020 $  0 010 * 20
=006E210          060 T  0 010 $  1
=006E220    *  2 071    5 050 ----
=006E230          020 P  1 001 *  3
=006E240          020 Q  0 056 T  1
=006E250    *  3 020 T  1 050 ----
=006E260          020 P  1 002 *  1
=006E270    *  8 020 P  3 010 $  1
=006E280    *  9 010 ----
=006E290    * 20 024 K  2 052 * 29
=006E300          120    0 075    7
=006E310          020 P  1 124 K 17
=006E320          064 T  0 004 T  0
=006E330          020 P  2 024 K  3
=006E340          052 * 21 054 P  2
=006E350    * 21 020 ---- 050 P  1
=006E360    * 29 010 ----
=007E000+   =  7 010 $  0                    UNPACKER
=007E010          024 K  2 052 *  9
=007E020          020 K  4 050 P 10
=007E030          020 J 12 052 * 23
=007E040          020 P  3 050 T  0          PICK UP SOURCE FROM P3.
=007E050    *  1 023 T  0 001 *  2          TO *2 IF MORE
=007E060          020 A 42 010 * 21          SUPPLY CR+EOM AND EXIT VIA *21.
=007E070    *  2 020 T  0 010 $  1
=007E080          056 *  3 024 K  1
=007E090    *  3 056 *  4 004 ----
=007E100          020 $  1 010 * 20
=007E110          020 $  1 010 * 20
=007E120          020 $  1 010 * 20
=007E130          020 $  1 010 * 20
=007E140          075   12 010 $  1
=007E150    *  4 056 T  0 004 ----
=007E160          020 $  1 010 * 20
=007E170          020 $  1 010 * 20
=007E180          020 $  1 010 * 20
=007E190          020 $  1 010 * 20
=007E200          020 $  1 010 * 20
=007E210          023 T  0 014 *  1
```

```
=007E220      *  9 010 ----                          EXIT
=007E230      * 20 050 * 29 075    7
=007E240      * 21 124 K 17 024 J 13
=007E245           056 * 22 010 $  1
=007E250           020 * 23 024 K  2
=007E260      * 22 052 * 23 020 ----
=007E270      * 23 050 ---- 002 * 29
=007E280           050 T  3 021 T  3
=007E290           002 *  9 020 * 23
=007E300           024 * 90 124 K  6      SET P10 LOCN OF CONDITION
=007E310           050 P 10 074   21
=007E320           056 P 10 020 A 45      SET CONDITION FLAG
=007E330           010 * 23
=007E340      * 29 --- ---- --- ----
=007E350      * 90         3          ,
=008E000+     =  8 010 $  0                ASSIGN BUFFER TO STN
=008E010           024 K  2 052 *  9
=008E020           020 H 10 050 H 13      SET H13
=008E030           001 *  9 024 U  5      EXIT MINUS IF NO BUFFER AVAILABLE
=008E040           052 *  1 056 *  5
=008E050      *  1 020 ---- 050 H 10      UPDATE H10
=008E060           021 J 16 050 T  0
=008E070           020 H 12 024 U  1
=008E080           052 *  2 056 *  6
=008E090      *  2 020 ---- 002 *  4
=008E100           020 H 12 050 T  1      SET BUFFER BITS IN SCR FOR NEW BUFFER
=008E110           020 H 13 052 T  1
=008E120           020 T  1 142 K 16
=008E130           010 *  5
=008E140      *  3 020 ---- 001 *  5
=008E150      *  4 024 U  5 052 *  3
=008E160           056 *  6 020 T  0
=008E170           024 I  1 050 T  0
=008E180           001 *  3 020 H 13
=008E190           050 H 10 020 K  4      BACKTRACK - STN ALREADY HAS LIMIT
=008E200           050 H 13 010 *  9
=008E210      *  5 020 K  4 050 ----      SET SUCCESSOR TO -1.
=008E220      *  6 020 H 13 050 ----      PUT NEW BUFFER ON END OF LIST FOR STN
=008E230      *  9 010 ----          ,     EXIT WITH BUFFER (OR-1) IN H13 AND ACC
=009E000+     =  9 010 $  0                ASSIGN DRUM TO STN (IF CAN)
=009E010           024 K  2 052 *  9
=009E020           020 H 11 050 H 14      SET H14
=009E030           001 *  9 024 U  6
=009E040           052 *  1 056 *  2
=009E050      *  1 020 ---- 050 H 11      UPDATE H11
=009E060      *  2 020 K  4 050 ----      MAKE SUCCESSOR NULL
=009E070           020 H 12 024 U  2
=009E080           056 *  3 010 $  1
=009E090      *  3 020 H 14 050 ----      ASSIGN DRUM
=009E100      *  9 010 ----          ,     EXIT WITH DRUM (OR -1) IN H14 AND ACC
=010E000      = 10 024 K  2 010 $  0      DEAL OUT A STORAGE SPACE
=010E010           052 *  9 020 Q  0
=010E020           001 E  0 056 *  1
=010E030      *  1 050 T  0 020 ----
=010E040           050 Q  0 020 T  0      LEAVES ADDR IN R(ACC)
=010E050      *  9 010 ----          ,
=011E000+     = 11 024 K  2 010 $  0      UNPACK (Q3) TO P4,P5,P6.
=011E010           052 *  9 020 P  4
=011E020           050 P  7 020 P  5
=011E030           050 P  8 020 P  6
```

```
=011E040          050 P  9 020 Q  3
=011E050          124 K 11 071    1
=011E060          072    1 050 P  4
=011E070          020 Q  3 124 K 12
=011E080          050 P  5 020 Q  3
=011E090          124 K 13 050 P  6
=011E100     *  9 010 ----                                    ,
=012E000+    = 12 004 * 92 010 *  0      PUSH Q2-Q3 OPERANDS
=013E000     = 13 004 * 92 010 * 10      POP Q2-Q3 OPERANDS
=014E000     = 14 004 * 94 010 *  0      PUSH Q4-Q5 OPERANDS
=015E000     = 15 004 * 94 010 * 10      POP Q4-Q5 OPERANDS
=016E000     = 16 004 * 96 010 *  0      PUSH Q6-Q7 AUXILIARY
=017E000     = 17 004 * 96 010 * 10      POP Q6-Q7 AUXILIARY
=018E000     = 18 004 * 98 010 *  0      PUSH Q8-Q9 CONTROLS
=019E000     = 19 004 * 98 010 * 10      POP Q8-Q9 CONTROLS
=019E010     *  0 024 K  2 052 *  9
=019E020          060 T  0 056 *  1
=019E030          052 *  2 024 K  3
=019E040          052 *  5 020 Q  0
=019E050     *  1 001 E  0 004 ----
=019E060     *  2 050 ---- 056 *  3
=019E070          056 *  4 024 K  1
=019E080     *  3 056 *  5 020 ----
=019E090     *  4 050 Q  0 060 ----
=019E100     *  5 020 ---- 050 ----
=019E110     *  9 010 ----
=019E120     * 10 024 K  2 052 * 19
=019E130          060 T  0 052 * 11
=019E140          052 * 13 024 K  3
=019E150          052 * 14 004 Q  0
=019E160     * 11 020 ---- 124 K  5
=019E170          050 Q  0 056 * 12
=019E180          056 * 14 024 K  1
=019E190     * 12 056 * 13 020 ----
=019E200     * 13 050 ---- 020 ----
=019E210     * 14 050 ---- 060 ----
=019E220     * 19 010 ----
=019E230     * 92      Q  2      Q  2
=019E240     * 94      Q  4      Q  4
=019E250     * 96      Q  6      Q  6
=019E260     * 98      Q  8      Q  8,
=020E000+    = 20 010 $  0                EVALUATE NUMERICAL EXPRESSION (UNSIGNED)
=020E010          024 K  2 052 *  9
=020E020          020 $  0 010 = 12
=020E030          120    0 050 P  4
=020E040          050 P  6 050 T  1
=020E050          020 K  4 050 T  0
=020E060          020 * 91 056 *  2
=020E070     *  1 020 P  1 124 K 35
=020E080          025 K 35 001 *  7
=020E090          020 P  1 124 K 20
=020E100     *  2 050 T  2 010 ----      SWITCH *3 THEN *4 AFTER FIRST SIGNIF DIG
=020E110     *  3 023 T  2 002 * 10
=020E120          020 P  4 050 T  1
=020E130          020 * 92 056 *  2
=020E140     *  4 020 P  6 073    2      ACCUMULATE
=020E150          024 P  6 073    1
=020E160          024 T  2 050 P  6
=020E170     * 10 020 P  4 024 K  2
=020E180     *  5 050 P  4 020 P  2
```

```
=020E190          024 K  3 056 *  6
=020E200     *  6 054 P  2 020 ----
=020E210          050 P  1 014 *  1
=020E220     *  7 020 P  1 124 A 27
=020E230          025 A 27 001 *  8
=020E240          020 T  0 001 * 20
=020E250     *  8 021 P  4 002 E  6     ERROR IF NO DIGITS
=020E260          024 * 90 001 E  5     ERROR IF MORE THAN 9 DIGITS
=020E270          024 T  1 024 J 11
=020E275          052 * 11 010 $  1
=020E280          023 P  6 006 *  13    JUMP IF ZERO.
=020E290     * 11 004 ---- 032 P  6     SCALE COEFFICIENT TO NORMALIZE
=020E300          060 P  6 020 T  0
=020E310          002 * 12 020 P  4
=020E320     *  12 025 T  1 025 K  2
=020E330          071   10 125 K  4
=020E340     *  13 024 P  6 050 Q  3
=020E350     *  9 010 ----              LEAVES VALUE IN ACC AS WELL AS Q3
=020E360     *  20 020 P  4 050 T  0    RECORD POSITION OF DEC PT
=020E370          014 *  5
=020E380     *  90         9
=020E390     * 91                 *  3
=020E400     *  92               *  4,
=021E000+    = 21 124 K  6 010 $  0     EVALUATE EXPRESSION
=021E010          024 K  2 050 *  91
=021E020          020 $  0 010 = 14
=021E030          020 *  91 050 Q  5    STACK EXIT IN OPERATORS
=021E040     *  1 020 $  0 010 =  3     ELIMINATE SPACES
=021E050     *  9 124 K 29 025 K 29
=021E060          001 *  2 010 $  1
=021E070          020 $  0 010 = 12
=021E080          120    0 050 Q  3
=021E090          020 $  0 010 = 14
=021E100          020 P  1 050 Q  5
=021E110     *  22 020 $  0 010 =  2    ADVANCE
=021E120     *  2 020 P  1 071    23
=021E130          001 *  8 071    3     TO *8 IF LEFT GROUPER
=021E140          001 *  3 071    1     TO *3 IF LETTER
=021E150          001 *  7 020 P  1     TO *7 IF DIGIT
=021E160          124 A 27 025 A 27
=021E170          002 *  7 020 P  1     TO *7 IF DEC PT
=021E180          124 A 68 025 A 68
=021E190          002 *  10 010 E  6    TO *10 IF CC = '$' ELSE MALFORMED
=021E200     *  3 020 P  2 024 K  2
=021E210          052 *  4 010 $  1
=021E220     *  4 020 ---- 050 T  0
=021E230          071   23 001 *  28    TO *28 IF SUBSCRIPT
=021E240          071    3 001 * 13
=021E250          020 P  1 052 *  5     SIMPLE VARIABLE
=021E255          020 $  0 010 = 12
=021E260     *  5 020 ---- 002 *  6
=021E270          050 T  0 023 T  0
=021E280          001 E  2 010 E  8
=021E290     *  6 050 Q  3 010 * 29
=021E300     *  7 020 $  0 010 = 20     CONVERT NUMBER
=021E310          020 $  0 010 =  3
=021E320          010 *  20
=021E330     *  8 020 $  0 010 = 14     STACK LEFT GROUPER
=021E340          020 P  1 050 Q  5
=021E350          020 $  0 010 =  2
```

```
=021E360          010 * 9
=021E370    * 10 020 $  0 010 = 12      CONVERT $
=021E380         020 P  0 025 N  1
=021E390         002 * 11 004 P  0
=021E400         032 N  8 014 * 12
=021E410    * 11 004 P  0 032 N  7
=021E420    * 12 020 K 14 064 Q  3
=021E430         010 * 29
=021E440    * 13 020 $  0 010 = 24      FUNCTION
=021E450    * 14 020 P  1 124 K 31
=021E460         025 K 31 001 E  6      MUST BE LEFT GROUPER NEXT
=021E470    * 15 020 $  0 010 = 14
=021E480         020 P  1 050 Q  5      STACK CHARACTER IN OPERATORS
=021E490         020 $  0 010 =  1
=021E500         020 $  0 010 = 21      EVALUATE EXPRESSION
=021E505         020 P  1 124 K 32
=021E510         025 K 32 002 * 16
=021E515         020 P  1 124 A 59
=021E520         025 A 59 002 * 15
=021E525         010 E  6               ERROR IF NOT COMMA OR RT GROUPER
=021E530    * 16 020 P  1 052 * 17      CC = RT GROUPER
=021E535    * 17 020 ---- 124 Q  5
=021E540         025 Q  5 002 * 18
=021E545         020 Q  5 124 A 59
=021E550         025 A 59 001 E  6
=021E555         020 $  0 010 = 15
=021E560         020 $  0 010 = 16
=021E565         020 Q  3 050 Q  7
=021E570         020 $  0 010 = 13
=021E575         010 * 17
=021E580    * 18 020 Q  5 124 A 96      MATCHED GROUPERS
=021E585         025 A 96 001 * 19
=021E590         020 Q  6 002 E  6      AUX MUST BE EMPTY IF ABS VAL.
=021E595         020 Q  3 125 K 12
=021E600         050 Q  3 010 $  1
=021E605    * 19 020 $  0 010 = 15
=021E610    * 29 020 $  0 010 =  2
=021E615    * 20 020 P  1 124 K 30
=021E620         025 K 30 001 * 23
=021E625         020 Q  5 124 K 30
=021E630         025 K 30 001 * 21
=021E635         020 P  1 124 K 37
=021E640         050 T  0 020 Q  5
=021E645         124 K 37 025 T  0
=021E650         002 * 24 010 $  1
=021E655    * 21 020 $  0 010 = 14      STACK OPERATOR
=021E660         020 P  1 050 Q  5
=021E665         010 * 22
=021E670    * 23 020 Q  5 124 K 30
=021E675         025 K 30 001 * 26
=021E680    * 24 020 Q  5 070   21
=021E685         056 * 25 010 $  1
=021E690    * 25 020 $  0 010 ----      FIRE OPERATOR
=021E695         020 $  0 010 = 15
=021E700         010 * 20
=021E705    * 26 020 Q  5 001 E  6
=021E710         125 K  6 025 K  1
=021E715         001 * 30 020 P  1
=021E720         124 K 32 025 K 32
=021E722         002 * 16 010 E  6
```

```
=021E724     * 30 020 Q  5 052 * 27
=021E726          020 $  0 010 = 15      POP Q5
=021E730     * 27 010 ----               EXIT
=021E735     * 28 020 $  0 010 = 14      SUBSCRIPT
=021E740          020 P  1 124 K 17
=021E745          024 * 90 050 Q  5
=021E750          020 $  0 010 =  1
=021E755          010 * 15
=021E760     * 90 000 = 26 440 ----
=021E770     * 91                   ,    PRIVATE WORKING CELL
=022E000+    = 22 010 $  0 010 *  1      TYPE TIME LINE (ENTER RT SUPPRESSES EJ)
=022E010          050 * 90 010 $  1
=022E020          020 A 31 050 S  1
=022E030          020 S  0 024 K  3
=022E040          050 P 12 010 *  2
=022E050     *  1 050 * 90 010 $  1
=022E060          020 S  0 050 P 12
=022E070     *  2 020 $  0 010 = 61      TIME
=022E080          020 $  0 010 = 62      SPACE
=022E090          020 $  0 010 = 64      DATE
=022E100               H  2      H  2
=022E110          020 $  0 010 = 62      SPACE
=022E120          020 $  0 010 = 64      INITIALS
=022E130               Q  1      Q  1
=022E140          020 $  0 010 = 64      SIX CR'S AND EOM
=022E150               * 91      * 91
=022E160          120    0 050 P  0      RESET LINE COUNTER
=022E170          020 * 90 010 = 25      EXIT VIA=25 TO OUTPUT LINE FROM S
=022E180     * 90 ----------------       LINK
=022E190     * 91 027,0561,342,7056
=022E200          025,2000,000,0000,
=023E000+    = 23 010 $  0               EVALUATE CONDITION (RESULT + OR -)
=023E010          024 K  2 052 *  9
=023E020          120    0 050 * 50      SET SIGNAL +
=023E030          020 $  0 010 = 21      EVALUATE EXPRESSION
=023E040     *  1 020 Q  3 050 * 51      SAVE IT
=023E050          020 $  0 010 = 13      POP OPERAND
=023E060          020 P  1 124 K 33
=023E070          025 K 33 001 E  6      CC MUST BE RELATION
=023E080          020 P  1 050 * 52      SAVE IT
=023E090          020 $  0 010 =  1      STEP ONE
=023E100          020 $  0 010 = 21      EVALUATE EXPRESSION
=023E110          020 * 51 050 H 21
=023E120          020 Q  3 050 H 22
=023E130          020 $  0 010 = 70
=023E140          124 * 52 025 K  1
=023E150          001 *  2 010 *  3
=023E160     *  2 020 K  4 050 * 50
=023E170     *  3 020 P  1 124 A109
=023E180          025 A109 001 *  1      JUMP IF NOT AT PERIOD
=023E190          020 $  0 010 = 13      POP OPERAND
=023E200          020 * 50 010 $  1
=023E210     *  9 010 ----               EXIT
=023E220     * 50 --- ---- --- ----      SIGNAL
=023E230     * 51 --- ---- --- ----      LEADING TERM
=023E240     * 52 --- ---- --- ----,     RELATION
=024E000+    = 24 010 $  0               CONVERT FUNCTION TO OPERATOR
=024E010          024 K  2 052 *  9
=024E020          020 $  0 010 = 14
=024E030          020 $  0 010 = 27
```

```
=024E040          050 T  0 004 K  2
=024E050          020 * 50 052 *  1
=024E060     * 1  020 ---- 001 E  6
=024E070          025 T  0 001 *  2
=024E080          025 K  1 005 *  3
=024E090     *  2 020 *  1 064 *  1
=024E100     *  3 010 *  1 020 *  1
=024E110          124 K  6 025 * 50
=024E120          024 * 51 050 Q  5     STACK FUNCTION AS OPERATION IN QK AND AC
=024E130     *  9 010 ----
=024E140     * 50      * 60
=024E150     * 51      F  0 440
=024E160     * 60 000,0002,321,2224     SQRT (ORDERED SAME AS F'S)
=024E170          000,0000,014,1707     LOG
=024E180          000,0000,005,3020     EXP
=024E190          000,0000,023,1116     SIN
=024E200          000,0000,003,1723     COS
=024E210          000,0000,000,1120     IP
=024E220          000,0000,000,0620     FP
=024E230          000,0000,023,0630     SFX
=024E240          000,0000,000,0306     CE
=024E250          000,0000,023,0716     SGN
=024E260          000,0000,001,1520     AMP
=024E270          000,0000,015,0130     MAX
=024E280          000,0000,015,1116     MIN
=024E290          100,0000,000,0000,
=025E000+    = 25 010 $  0 010 *  1     OUTPUT LINE FROM S BLOCK TO STN
=025E010          024 K  2 052 P 14     SET EXIT IN P14 (INITIAL ENTRY)
=025E020     *  1 020 $  0 010 =  8     ASSIGN BUFFER (REENTRY POINT)
=025E030          001 *  3 140 S  1     LOAD BUFFER IF ASSIGNED
=025E040          020 H 12 024 U  1
=025E050          052 *  2 020 H 13
=025E060     *  2 025 ---- 001 *  5     EXIT VIA P14 IF CAN'T TL YET
=025E070          025 K  1 002 *  5     EXIT VIA P14 IF CAN'T TL YET
=025E080          020 H 12 024 K 24     OK TO TRANSMIT
=025E090          142 K 24 010 *  5     EXIT VIA P14
=025E100     *  3 020 H 10 001 *  4
=025E110          020 $  0 010 = 47     CHOKED-CHANGE STATE TO L14
=025E120              L 14     L 14
=025E130          010 *  6
=025E140     *  4 020 $  0 010 = 47     Q FOR BUFFER AND PREFERRED PROCESSING
=025E150              L 11     L 11
=025E155     *  6 020 $  0 010 = 53     KICK OUT PROGRAM
=025E160          010 X  1              EXIT TO EXECUTIVE
=025E170     *  5 020 P  0 024 I  1
=025E180          050 P  0 025 J  2
=025E190          001 P 14 020 P 14
=025E200          025 K  2 010 = 22,    EXIT VIA=22 IF NEW PAGE NEEDED
=027E000+    = 27 010 $  0              ACCUMULATE LETTER CODES (6 MAX)
=027E010          024 K  2 052 *  9     RESULT IN ACC WITH SCAN ADVANCED
=027E020          120 ----  0 050 H 19
=027E030     *  1 020 P  1 124 K 34
=027E040          025 K 34 005 *  3
=027E050          020 H 19 004 P  1
=027E060          077 ----  6 050 H 19
=027E070          020 P  2 024 K  3
=027E080          054 P  2 052 *  2
=027E090     *  2 020 ---- 050 P  1
=027E100          020 H 19 025 K 12
=027E110     *  3 001 *  1 020 H 19     LEAVE RESULT IN ACC AND H19
```

```
=027E120      *  9 010 ----                ,
=028E000+     = 28 010 $  0                      ERASE LEFT LINKED LIST A/C MQ
=028E010            024 K  2 052 *  9
=028E020            060 T  0 124 K  6
=028E030            050 T  0 010 $  1
=028E040      *  1 023 T  0 002 *  9
=028E050            020 T  0 052 *  2
=028E060            052 *  3 070    21
=028E070            004 Q  0 050 Q  0
=028E080      *  2 020 ---- 052 T  0
=028E090      *  3 060 ---- 010 *  1
=028E100      *  9 010 ----                ,
=029E000+     = 29 010 $  0                      ERASE RIGHT LINKED LIST A/C MQ
=029E010            024 K  2 052 *  9
=029E020            060 T  0 124 K  5
=029E030            050 T  0 010 $  1
=029E040      *  1 023 T  0 002 *  9
=029E050            020 T  0 056 *  2
=029E060            056 *  3 004 Q  0
=029E070      *  2 050 Q  0 020 ----
=029E080      *  3 056 T  0 060 ----
=029E090            010 *  1
=029E100      *  9 010 ----                ,
=030E000+     = 30 010 $  0                      CHECK RANGE AND PACK RESULT
=030E010            024 K  2 052 *  9
=030E015            023 P  6 006 *  3          CHK FOR ZERO
=030E020            020 P  4 005 *  1
=030E030            025 * 90 005 *  2
=030E040      *  1 010 E  1 024 * 91
=030E050            006 *  2 120    0
=030E060      *  2 014 *  3 020 P  4
=030E070            024 P  5 024 P  6
=030E080      *  3 125 K  4 050 Q  3
=030E090      *  9 010 ----
=030E100      * 90+          100    8
=030E110      * 91+           99    8,
=031E000+     = 31 010 $  0                      ADD
=031E010            024 K  2 052 *  9
=031E020            020 $  0 010 = 11          UNPACK 2ND OPERAND
=031E030            020 *  1 010 = 13          POP Q3 AND MERGE WITH =32
=032E000      = 32 010 $  0                      SUBTRACT
=032E010            024 K  2 052 *  9
=032E020            020 $  0 010 = 11          UNPACK 2ND OPERAND
=032E030            020 K 12 025 P  5          INVERT SIGN OF 2ND OPERAND
=032E040            050 P  5 010 $  1
=032E050      *  1 020 $  0 010 = 13          POP Q3
=032E060            023 P  6 002 *  9          DONE IF 2ND OPERAND IS ZERO
=032E070            023 Q  3 002 *  8          PACK UP IF 1ST OPERAND IS ZERO
=032E080            020 $  0 010 = 11          UNPACK 1ST OPERAND
=032E090            020 P  4 025 P  7
=032E100            001 *  2 025 K  1          TO *2 IF 2ND SFX IS GREATER
=032E110            002 *  3 020 P  6          TO    IF 1ST SFX IS GREATER
=032E120            025 P  9 002 *  3          TO    IF 1ST CF IS GREATER OR EQUAL
=032E130      *  2 020 P  4 004 P  7          INTERCHANGE OPERANDS
=032E140            050 P  7 060 P  4
=032E150            020 P  5 004 P  8
=032E160            050 P  8 060 P  5
=032E170            020 P  6 004 P  9
=032E180            050 P  9 060 P  6
=032E190      *  3 020 P  5 024 P  8
```

```
=032E200          071    9 002 *  4     TO *4 IF SIGNS ARE ALIKE
=032E210          021 P  9 050 P  9     ELSE COMPLEMENT CF OF SMALLER NUMBER
=032E220     *  4 020 P  4 025 P  7
=032E230          025 * 90 002 *  8     TO *8 IF SFX DIFF AS BIG AS 11
=032E240          024 * 90 072   10
=032E250          024 J 11 052 *  5
=032E260     *  5 004 ---- 060 T  0     SET DECIMAL SHIFTER
=032E270          020 P  9 077    1
=032E280          036 P  6 044 T  0     DEVELOP TWICE SUM IN MQ
=032E290          050 T  2 060 T  1     SAVE PIECES
=032E340     * 10 023 T  1 001 * 11
=032E350          023 T  2 001 * 11
=032E360          050 Q  3 010 *  9     SET ZERO RESULT AND EXIT
=032E370     * 11 020 T  1 025 * 91
=032E380          002 *  13 004 N  1
=032E390          032 T  2 044 T  0     SCALE UP ONE DIGIT
=032E400          060 T  3 004 N  1
=032E410          036 T  1 020 P  4
=032E420     *  12 025 K 14 050 P  4     ADJUST SFX
=032E430          060 T  3 025 * 91
=032E440          002 *  13 032 N  1
=032E450          020 P  4 010 * 12     SCALE UP AND GO TO *12
=032E452     *  13 060 T  3 025 *  92
=032E454          001 * 14 020 P  4
=032E456          024 K 14 050 P  4
=032E458          120    0 044 N  1
=032E460     *  14 060 T  3 024 K  1     ROUND
=032E465          072    1 050 P  6       AND SET CF OF RESULT
=032E470     *  8 020 $  0 010 =  30     PACK UP RESULT
=032E480     *  9 010 ----               EXIT
=032E490     *  90+           11    8
=032E500     * 91+   200000000    39
=032E510     *  92+  1999999999   39,
=033E000+    = 33 010 $  0             MULTIPLY
=033E010          024 K  2 052 *  9
=033E020          020 $  0 010 = 11
=033E030          020 $  0 010 =  13
=033E040          023 P  6 001 *  1
=033E050          050 Q  3 010 *  9
=033E060     *  1 023 Q  3 002 *  9
=033E070          020 $  0 010 =  11
=033E080          020 N  9 070    1
=033E090          004 P  9 036 P  6
=033E100          050 T  0 025 * 90
=033E110          001 *  3 025 N  0
=033E120          002 *  2 021 * 91
=033E130          064 T  1 001 *  3
=033E140     *  2 020 T  0 044 N  9
=033E150          060 P  6 020 P  4
=033E160          024 P  7 024 K 14
=033E170          050 P  4 010 *  6
=033E180     *  3 021 * 92 064 T  1
=033E190          004 T  1 001 *  4
=033E200          020 T  0 010 *  5
=033E210     *  4 020 T  0 025 N  0
=033E220     *  5 044 N  8 060 P  6
=033E230          020 P  4 024 P  7
=033E240          050 P  4 010 $  1
=033E250     *  6 020 P  5 024 P  8
=033E260          124 K 12 050 P  5
```

```
=033E270          020 $  0 010 = 30
=033E280     *  9 010 ----
=033E290     * 90+       181898   39
=033E300     * 91+517415400576   39
=033E310     * 92+  450000000    39,
=034E000+    = 34 010 $  0                    DIVIDE
=034E010          024 K  2 052 *  9
=034E020          023 Q  3 002 E  3
=034E030          020 $  0 010 = 11
=034E040          020 $  0 010 = 13
=034E050          023 Q  3 002 *  9
=034E060          020 $  0 010 = 11
=034E070          020 N  8 050 T  0
=034E080          020 P  4 025 P  7
=034E090          050 P  4 020 P  6
=034E100          025 P  9 002 *  1
=034E110          020 N  9 050 T  0
=034E120          020 P  4 025 K 14
=034E130          050 P  4 010 $  1
=034E140     *  1 020 P  9 072    1
=034E150          004 P  6 036 T  0
=034E160          044 P  9 060 P  6
=034E170          020 P  5 024 P  8
=034E180          124 K 12 050 P  5
=034E190          020 $  0 010 = 30
=034E200     *  9 010 ----
=035E000+    = 35 010 $  0                    Y = A*B
=035E010          024 K  2 052 *  9
=035E020          120       050 * 81         SET FLAGS TO ZERO
=035E030          050 * 82 050 * 83
=035E040          023 Q  3 001 *  1          DOES B=0, IF YES
=035E070     *  6 020 N  8 024 * 81          IF NO, RESULT IS ONE, PLUS SIGN
=035E080     * 16 050 Q  3 010 *  9          IN FLAG 1
=035E090     *  1 020 Q  3 050 * 85          STORE B
=035E100          020 $  0 010 = 11          UNPACK B
=035E110          020 P  4 050 * 86          STORE PARTS IN *86,*87,*88
=035E120          020 P  5 050 * 87
=035E130          020 P  6 050 * 88
=035E140          020 $  0 010 = 13          POP B
=035E150          023 Q  3 001 *  2          IF A=0 AND B IS POSITIVE, RESULT IS
=035E160          023 * 87 002 *  9          ZERO.  IF B IS NEGATIVE, ERROR.
=035E172          010 E 23
=035E180     *  2 020 Q  3 050 * 84          STORE A
=035E190          020 $  0 010 = 11          UNPACK A
=035E200          020 P  4 050 * 89          STORE PARTS IN *89,*90,*91
=035E210          020 P  5 050 * 90
=035E220          020 P  6 050 * 91
=035E230          020 * 85 050 Q  3
=035E240          020 $  0 010 F  6          FIND FP OF B
=035E250          023 Q  3 006 *  3          TRANSFER IF B IS AN INTEGER
=035E260          023 * 90 001 E 21          ERROR IF A IS NEGATIVE
=035E270          020 K 12 050 * 82          FLAG 2 IS 2*(-9) FOR B FRACTIONAL
=035E280     *  3 014 *  4 020 * 86
=035E290          072   31 050 T  0          IF SFB IS LESS THAN 9 COMPUTE AND
=035E300          020 I  8 025 T  0          STORE TRUE VALUE OF INTEGRAL B
=035E310          005 *  4 024 J 11          IN *80
=035E320          056 *  5 120
=035E330     *  5 004 * 88 044 ----          TEST UNITS POSITION FOR B ODD OR
=035E340          060 * 80 071   39          EVEN.  (SHIFT TO SIGN BIT)
=035E350          006 *  4 020 * 90          IF B IS ODD STORE SIGN OF A
```

```
=035E360      *  4 050 *  81 020 *  84      IN FLAG 1
=035E370         125 K  12 050 *  84      STORE ABSOLUTE VALUE OF A
=035E380         025 N   8 050 T   1      IF IT EQUALS ONE, THEN RESULT
=035E390         023 T   1 002 *   6      IS PLUS OR MINUS ONE
=035E400         020 *  87 050 *  83      FLAG 3 EQUALS SIGN OF B
=035E410         020 *  85 125 K  12      STORE ABSOLUTE VALUE OF B
=035E420         050 *  85 023 *  82      IF FLAG 2 IS ZERO, B IS AN INTEGER
=035E430         002 *   7 020 *  85      TRANSFER
=035E440         025 *  93 050 T   0
=035E450         020 *  84 050 Q   3      STORE A IN Q3, THEN TEST
=035E460         023 T   0 001 *   8      IF B EQUALS 1/2
=035E470         020 $   0 010 F   0      COMPUTE SQUARE ROOT OF A
=035E480         023 *  83 002 *   9      IF FLAG 3 IS ZERO, EXIT
=035E490         020 I   1 050 *  80      TRANSFER TO INTEGER ARITHMETIC TO
=035E502         050 *  87 020 Q   3      COMPUTE RECIPROCAL OF SQUARE ROOT
=035E504         050 *  84 010 *  29
=035E530      *  8 020 $   0 010 F   1      COMPUTE E*(B.LOG(A)) FOR B
=035E540         020 $   0 010 =  11      FRACTIONAL OR INTEGER GREATER THAN 9
=035E550         020 P   5 024 *  87
=035E560         124 K  12 025 I   1      TEST SIGN FOR B.LOG(A), IF NEGATIVE
=035E570         001 *  26 020 P   4      TEST SF FOR RESULT
=035E580         024 *  86 072     31      IF SF GREATER THAN 2, E*(-B.LOG(A))=0
=035E590         025 I   3 006 *  23
=035E600      * 26 020 $   0 010 =  12
=035E610         020 *  85 024 *  87
=035E620         050 Q   3 010 $   1
=035E630         020 $   0 010 =  33      MULTIPLY B.LOG(A)
=035E680         020 $   0 010 F   2      COMPUTE  E* (B(LOG(A))
=035E690      *  20 020 Q   3 014 *   6      TRANSFER TO ADD FLAG 1
=035E700      *   7 020 *  91 025 N   8      B IS AN INTEGER, TEST A
=035E710         025 I   1 002 *  14      IF A IS NOT A POWER OF 10, TRANSFER
=035E720         022 *  86 025 K  14      IF ABSOLUTE VALUE OF SFB IS
=035E730         025 K  14 005 *  15      GREATER THAN ONE, TEST SIGN FOR
=035E740         020 *  87 024 I   1      SF FOR RESULT, + IS OVERFLOW,
=035E745         071     49 032 *  89      -IS UNDERFLOW
=035E750      *  23 002 E   1 120
=035E755      *  15 010 *  16 004 *  80
=035E760         023 *  87 002 *  18      ATTACH SIGN TO B
=035E765         061 *  80 004 *  80      B. SFA
=035E770      *  18 032 *  89 076      0      SHIFT SIGN BIT
=035E771         001 *  25 025 I   1      TEST FOR NON-ZERO BIT IN ACCUMULATOR
=035E772         002 E   1 010 *  24      (IS SF MAGNITUDE GREATER THAN 511)
=035E773      *  25 024 I   1 005 *  23
=035E775      *  24 060 P   4 020 *  81
=035E780         050 P   5 020 *  91
=035E785         050 P   6 010 *  17
=035E790      *  17 020 $   0 010 =  30      PACK
=035E795      *   9 010 ----      EXIT
=035E800      *  14 020 *  84 050 Q   3      SET Q3 EQUAL TO ABSOLUTE VALUE OF A
=035E810         020 K  14 025 *  86      IF B IS GREATER THAN 29, TRANSFER TO
=035E812         001 *   8 020 *  80      COMPUTE E*(B.LOG(A))
=035E814         025 *  94 002 *   8
=035E816      *  29 020 N   8 050 Q   3      INITIALIZE PRODUCT AND SET Q3=1
=035E818         050 *  88 010 $   1
=035E820         020 $   0 010 =  12      PUSH
=035E822         020 *  84 050 Q   3      Q3 = A
=035E824         023 *  87 002 *  27
=035E826         020 $   0 010 =  34      IF B NEGATIVE, COMPUTE 1/A
=035E828      *  27 020 Q   3 050 *  91      STORE MULTIPLIER
=035E830         020 *  80 072     1      TEST SUCCESSIVE DIGITS OF B, IF ONE
```

```
=035E832          050 * 80 061 T  0     MULTIPLY CURRENT PRODUCT BY CURRENT
=035E834          002 * 28 010 $  1     POWER OF A*N OR (1/A)*N
=035E836          020 $  0 010 = 12     PUSH A*N
=035E838          020 * 88 050 Q  3
=035E840          020 $  0 010 = 33     MULTIPLY
=035E842          020 Q  3 050 * 88
=035E844    * 28 023 * 80 002 * 20      IF B=0, DONE, OTHERWISE
=035E846          020 * 91 050 Q  3     COMPUTE NEXT POWER
=035E848          020 $  0 010 = 12     PUSH A*N
=035E850          020 $  0 010 = 33     COMPUTE A*(2N)
=035E852          010 * 27
=035E870    * 80                        B IF IT IS AN INTEGER
=035E875    * 81                        FLAG 1
=035E880    * 82                        FLAG 2
=035E885    * 83                        FLAG 3
=035E890    * 84                        A PACKED -- THEN ABSOLUTE A PACKED
=035E895    * 85                        B PACKED -- THEN ABSOLUTE B PACKED
=035E900    * 86                        SFX B
=035E905    * 87                        SIGN B
=035E910    * 88                        COEF B, REPLACED BY 10*8 IN * 29
=035E915    * 89                        SFX A
=035E920    * 90                        SIGN A
=035E925    * 91                        COEF A, REPLACED BY A*N IN * 27
=035E935    * 93 077,6356,326,2400       1/2(10*8)
=035E940    * 94                  30,
=036E000+   = 36 010 $  0               CONVERT H20 INTEGER FOR OUTPUT FROM S
=036E010          024 K  2 052 *  1       020 $  0 010 = 36
=036E020          024 K  2 052 *  9             K        N
=036E030    *  1 020 ---- 024 * 90       WHERE K=OPTION AND N=NBR OF DIGITS
=036E040          052 * 10 124 K  5       K=1 SKIP LEADING ZEROS
=036E050          025 I  1 050 T  0       K=2 SPACE FOR LEADING ZEROS
=036E060          120    0 050 *  3       K=3 FORCE LEADING ZEROS
=036E070    * 10 010 ----                 K=4 FORCE LEADING, SKIP TRAILING ZEROS
=036E080    * 11 020 * 91 014 *  2
=036E090    * 12 020 * 92 014 *  2
=036E100    * 13 020 *  6 014 *  2
=036E110    * 14 020 * 94 050 *  3
=036E120    *  2 020 *  6 050 *  5
=036E130    *  3 --- ---- --- ----
=036E140          020 T  0 024 J 11
=036E150          056 *  4 120    0
=036E160    *  4 004 H 20 044 ----
=036E170          050 H 20 063 T  1
=036E180          002 *  5 020 *  6
=036E190          050 *  5 010 * 20
=036E200    *  5 --- ---- --- ----
=036E210    *  6 004 A 48 010 * 20
=036E220    *  7 004 A 14 010 * 20
=036E230    * 20 020 P 12 024 K  3
=036E240          050 P 12 056 * 21
=036E250    * 21 060 P 11 050 ----
=036E260    * 22 020 T  0 025 I  1
=036E270          050 T  0 002 *  3
=036E280    *  9 010 ----               EXIT
=036E290    * 90      * 10
=036E300    * 91 023 T  0 001 * 22
=036E310    * 92 023 T  0 001 *  7
=036E320    * 94 023 H 20 002 *  9,
=037E000+   = 37 010 $  0               SIMPLE-DIRECT TESTS
=037E010          024 K  2 052 *  9
```

```
=037E020          020 Q  9 002 E  9      ERROR IF Q9 SHOWS INDIRECT
=037E030          020 $  0 010 =  3      ELIMINATE SPACES
=037E040          020 P  1 001 E  6      MALFORMED IF NOT NOW AT PERIOD
=037E050          124 A109 025 A109
=037E060     *  9 002 ---- 010 E  6,
=038E000+    = 38 010 $  0               SIMPLE-INDIRECT TESTS
=038E010          024 K  2 052 *  9
=038E020          020 Q  9 001 E 10      ERROR IF Q9 SHOWS DIRECT
=038E030          020 $  0 010 =  3      ELIMINATE SPACES
=038E040          020 P  1 124 K 19      MALFORMED IF NOT NOW AT GOOD TERMINAL
=038E050          025 K 19 001 E  6
=038E060     *  9 010 ----              ,
=039E000+    = 39 010 $  0               VERIFY SPACE AND ADVANCE
=039E010          024 K  2 052 *  9
=039E020          020 P  1 124 A 14
=039E030          025 A 14 001 E  6      MALFORMED IF NOT SPACE
=039E040     *  1 020 P  2 024 K  3
=039E050          052 *  2 054 P  2
=039E060     *  2 020 ---- 050 P  1
=039E070          124 A 14 025 A 14
=039E080          002 *  1 020 P  1
=039E090     *  9 002 ---- 010 E  6,   EXIT WITH NONSPACE NONTERMINAL IN ACC
=040E000+    = 40 010 $  0               POSITIVE INTEGER TEST ON (Q3)
=040E010          024 K  2 052 *  9
=040E020          020 $  0 010 = 11
=040E030          021 P  6 001 *  1
=040E040     *  2 020 K  4 010 *  9
=040E050     *  1 021 P  5 001 *  9
=040E060          020 P  4 001 *  9
=040E070          025 * 90 002 *  2
=040E080          021 P  4 072   10
=040E090          024 * 91 052 *  3
=040E100          120    0 004 P  6
=040E110     *  3 044 ---- 025 K  1
=040E120          002 *  2 120    0
=040E130     *  9 010 ----              EXIT + OR - A/C TEST
=040E140     * 90+          9    8
=040E150     * 91      N  8          ,
=041E000+    = 41 010 $  0               SUBSCRIPT RANGE TEST AND REPL (Q3)
=041E010          024 K  2 052 *  9
=041E020          023 Q  3 006 *  3
=041E030          020 $  0 010 = 11
=041E040          023 P  5 001 E  4
=041E050          020 P  4 001 E  4
=041E060          025 K 14 002 *  1
=041E070          020 N  8 014 *  2
=041E080     *  1 025 K 14 002 E  4
=041E090     *  2 020 N  7 050 T  0
=041E100          120    0 004 P  6
=041E110          044 T  0 025 K  1
=041E120          002 E  4 060 T  0
=041E130     *  3 071   12 050 Q  3
=041E140     *  9 010 ----              ,
=042E000+    = 42 010 $  0               LIMIT RANGE TEST ON (Q3)
=042E010          024 K  2 052 *  9
=042E020          020 $  0 010 = 11
=042E030          020 P  4 001 E  7
=042E040          025 * 90 002 E  7
=042E050          021 P  4 072   10
=042E060          024 * 91 052 *  1
```

```
=042E070          120    0 004 P  6
=042E080    *  1 044 ---- 025 K  1
=042E090    *  9 001 ---- 010 E  7
=042E100    * 90+          9    8
=042E110    * 91    N  8          ,
=043E000+   = 43 010 S  0          DECK NUMBER TEST ON (Q3)
=043E010          024 K  2 052 *  9
=043E020          020 S  0 010 = 40
=043E030    *  9 002 ---- 010 E 11,
=044E000+   = 44 010 S  0          PART NUMBER TEST ON (Q3)
=044E010          024 K  2 052 *  9
=044E020          020 S  0 010 = 40
=044E030    *  9 002 ---- 010 E 18,
=045E000+   = 45 010 S  0          STEP NUMBER TEST ON (Q3)
=045E010          024 K  2 052 *  9
=045E020          023 Q  3 002 E 13
=045E030          020 Q  3 071    1
=045E040          001 E 13 071    8
=045E050          001 E 13 020 Q  3
=045E060          025 * 90 002 E 13
=045E070    *  9 010 ----
=045E080    * 90+          9    8,
=046E000+   = 46 010 S  0          FORM NUMBER TEST ON (Q3)
=046E010          024 K  2 052 *  9
=046E020          020 S  0 010 = 40
=046E030    *  9 002 ---- 010 E 12,
=047E000+   = 47 010 S  0          PUT STN AT END OF STATE IN $1.
=047E010          024 K  2 052 *  6
=047E020          024 K  2 052 *  9
=047E030          020 H 12 024 U  0
=047E040          052 *  1 056 *  6
=047E050          020 H 12 024 U  3
=047E060          052 *  4 056 *  5
=047E070    *  1 020 ---- 056 *  2
=047E080    *  2 056 *  4 020 ----
=047E090          050 T  0 025 H 12
=047E100          005 *  3 025 K  1
=047E110    *  3 001 *  4 020 T  0
=047E120          024 U  3 014 *  1
=047E130    *  4 020 ---- 050 ----    CLOSE GAP IN OLD STATE LIST
=047E140    *  5 020 K  4 050 ----    SET NULL SUCCESSOR FOR STN
=047E150    *  6 020 ---- 050 ----    SET NEW STATE FOR STN
=047E160    *  7 052 *  8 056 * 10
=047E170    *  8 020 ---- 001 * 10
=047E180          024 U  3 010 *  7
=047E190    * 10 020 H 12 050 ----    ADD STN TO END OF NEW STATE LIST
=047E200    *  9 010 ----          ,
=048E000+   = 48 010 S  0          'TO' ROUTINE
=048E010          024 K  2 052 *  9
=048E020          120    0 145 K 35
=048E030          001 *  9 124 K 17    EXIT IF NO SIGNAL
=048E040          142 K 35 050 H 12    ERASE SIGNAL
=048E050          071   21 052 H 12    SET STN CONTEXT
=048E060          020 S  0 010 = 58    RELEASE CURRENT BUFFER
=048E070          001 *  1 020 H 12
=048E080          024 K 24 142 K 24    TL NEXT BUFFER TO STN
=048E090    *  1 020 H 12 024 U  0
=048E100          052 *  2 020 K  5
=048E110    *  2 124 ---- 025 * 90
=048E120          001 *  9 025 I  1
```

```
=048E130          001 *  3 025 I  7     TO *3 IF STATE=L6
=048E140          001 *  9 025 I  1
=048E150          001 *  5 010 *  9     TO * IF STATE=L14
=048E160    *  3 020 H 13 024 U  5     STATE=L6
=048E170          052 *  4 010 $  1
=048E180    *  4 020 ---- 002 *  9
=048E190          020 $  0 010 = 47     ON LAST BUFFER - SWITCH TO USER
=048E200             L 10     L 10
=048E210          020 H 12 024 K 36
=048E220          142 K 36 010 *  9     CL+SU THEN EXIT
=048E230    *  5 020 J 17 050 T  0     STATE = L14
=048E240          020 H 12 024 U  1
=048E250          052 *  6 010 $  1
=048E260    *  6 020 ---- 001 *  7     TO *7 IF BUFFERS=UNCHOKE NUMBER
=048E270          024 U  5 052 *  6
=048E280          020 T  0 025 I  1
=048E290          050 T  0 002 *  6
=048E300          010 *  9              EXIT IF STILL CHOKING
=048E310    *  7 020 $  0 010 = 47     UNCHOKE
=048E320             L 11     L 11
=048E330    *  9 010 ----              EXIT
=048E340    * 90                 L  6,
=049E000+   =  49 024 K  2 010 $  0     UPDATE ACTIVITY OF STN
=049E010          052 *  9 020 H 12
=049E020          024 U  4 052 *  1
=049E030          020 $  0 010 =  0
=049E040    *  1 050 ---- 010 $  1
=049E050    *  9 010 ----           ,
=050E000+   =  50 010 $  0              FIND PART (P20)
=050E010          024 K  2 052 *  9
=050E020          020 *  90 050 H 22
=050E030          020 Q 10 050 H 23
=050E040    *  1 020 H 22 050 H 21
=050E050          020 H 23 050 H 22
=050E060          056 *  2 024 K  1
=050E070    *  2 056 *  3 020 ----
=050E080          050 H 23 001 *  9     EXIT MINUS IF CAN'T FIND
=050E090    *  3 020 P 20 025 ----
=050E100          001 *  1 025 K  1
=050E110          002 *  1 020 H 23     OUTPUT LOCATOR IN ACC
=050E120    *  9 010 ----              H21, H22, H23 HOLD USEFUL POINTERS
=050E130    * 90                 Q 10,
=051E000+   =  51 010 $  0              FIND STEP (P21)
=051E010          024 K  2 052 *  9
=051E020          020 P 21 070    31
=051E030          124 K  5 050 H 21
=051E040          020 *  90 025 H 21
=051E050          056 *  4 010 $  1
=051E060          020 *  91 050 H 22
=051E070          020 Q 10 050 H 23
=051E080    *  1 020 H 22 050 H 21
=051E090          020 H 23 050 H 22
=051E100          056 *  2 024 K  1
=051E110    *  2 056 *  3 020 ----
=051E120    *  8 050 H 23 001 *  9     EXIT MINUS IF CAN'T FIND
=051E130    *  3 020 P 21 025 ----
=051E140    *  4 001 *  8 025 ----     EXIT MINUS IF CAN'T FIND AND LEAVE H23 M
=051E150          002 *  1 020 H 22
=051E160          071    21 124 K  6
=051E170          050 H 25 020 H 23
```

```
=051E180          050 H 26 010 $  1
=051E190    *  5 020 H 25 050 H 24
=051E200          020 H 26 050 H 25
=051E210          052 *  6 024 K  2
=051E220          052 *  7 010 $  1
=051E230    *  6 020 ---- 050 H 26
=051E240          001 *  9 020 P 21      EXIT MINUS IF CAN'T FIND
=051E250    *  7 025 ---- 001 *  9      EXIT MINUS IF CAN'T FIND
=051E260          025 K  1 002 *  5
=051E270          020 H 26 010 $  1      OUTPUT LOCATOR IN ACC
=051E280    *  9 010 ----                H21-H26 HOLD USEFUL INFO ON EXIT
=051E290    * 90                 N  8
=051E300    * 91                 Q 10,
=052E000+   = 52 010 $  0                FIND FORM (P22)
=052E010          024 K  2 052 *  9
=052E020          020 * 90 050 H 25
=052E030          020 Q 11 050 H 26
=052E040    *  1 020 H 25 050 H 24
=052E050          020 H 26 050 H 25
=052E060          052 *  2 024 K  2
=052E070          052 *  3 010 $  1
=052E080    *  2 020 ---- 050 H 26
=052E090          001 *  9 020 P 22      EXIT MINUS IF CAN'T FIND
=052E100    *  3 025 ---- 001 *  9
=052E110          025 K  1 002 *  1
=052E120          020 H 26 010 $  1      OUTPUT LOCATOR IN ACC AND H26
=052E130    *  9 010 ----                H24 LOCATES PREDECESSOR ON EXIT.
=052E140    * 90      Q 11          ,
=053E000+   = 53 010 $  0                KICK OUT CURRENT PROGRAM (IF ANY)   TEMP
=053E010          024 K  2 052 *  9
=053E020          020 H  7 001 *  9
=053E030          024 U  2 052 *  1
=053E040    *  1 020 ---- 024 U  7
=053E050          052 *  2 010 $  1
=053E060    *  2 004 ---- 111,7000      TEMPORARILY 512 WORDS
=053E070          020 K  4 050 H  7
=053E080    *  9 010 ----          ,
=054E000+   = 54 010 $  0                BRING IN PROGRAM (IF ANY) A/C STN   TEMP
=054E010          024 K  2 052 *  9
=054E020          020 $  0 010 = 53      KICK OUT CURRENT PROGRAM IF ANY
=054E030          020 H 12 024 U  2
=054E040          052 *  1 010 $  1
=054E050    *  1 020 ---- 001 *  9
=054E060          024 U  7 052 *  2
=054E070    *  2 004 ---- 110,7000      TEMPORARILY 512 WORDS
=054E080          020 H 12 050 H  7
=054E090          020 $  0 010 =  0
=054E100          050 H  8 010 $  1      SET START TIME FOR SHOT
=054E110    *  9 010 ----          ,
=055E000+   = 55 010 $  0                SWITCH TO USER (CL+SU)
=055E010          024 K  2 052 *  9      (ASSUMES SUBSTATE OF L10 ALREADY IN P13)
=055E015          020 K  4 050 P 16      CLEAR P16 BEFORE SWITCHING
=055E020          050 P 15 020 H 12      ALSO CLEAR P15
=055E030          024 U  1 052 *  1
=055E032          020 Q  8 001 *  1      JUMP IF Q8 MINUS
=055E034          050 Q 12 020 Q  9      HIDE Q8-Q9
=055E036          050 Q 13 020 K  4
=055E038          050 Q  8 050 Q  9
=055E040    *  1 020 ---- 001 *  3
=055E050          024 U  5 052 *  2
```

```
=055E060      *  2 020 ---- 001 *  4
=055E070           020 $  0 010 = 47      CHANGE STATE TO L6 WHILE DRAINING
=055E080             L  6     L  6
=055E090           010 *  9
=055E100      *  3 020 $  0 010 =  8      ASSIGN INPUT BUFFER
=055E110           002 *  4 010 $  1      TO *4 IF ASSIGNED
=055E120           020 $  0 010 = 47      CHANGE STATE TO L7 IF NO BUFFER AVAILABL
=055E130             L  7     L  7
=055E140           010 *  9
=055E150      *  4 020 H 12 024 K 36      OK TO SWITCH
=055E160           142 K 36 010 $  1      CL+SU
=055E170           020 $  0 010 = 47      CHANGE STATE TO L10 GREEN
=055E180             L 10     L 10
=055E190      *  9 010 ----              ,
=056E000+     = 56 010 $  0              ERASE STEP A/C H21-H26
=056E010           024 K  2 052 *  9
=056E020           020 $  0 010 = 57      STEP LOOKS JUST LIKE FORM HERE
=056E030           020 H 22 056 *  1
=056E040      *  1 020 K  6 124 ----
=056E050           025 K  1 002 *  9      EXIT UNLESS PART NOW EMPTY
=056E060           020 H 21 056 *  2
=056E070      *  2 020 H 23 056 ----
=056E080           020 H 22 071   61
=056E090           020 $  0 010 = 28      ERASE LEFT LINKED LIST (ONE ITEM HERE)
=056E100      *  9 010 ----              ,
=057E000+     = 57 010 $  0              ERASE PART A/C H24-H26
=057E010           024 K  2 052 *  9
=057E020           020 H 24 052 *  1
=057E030           020 H 26 010 $  1
=057E040      *  1 052 ---- 010 $  1
=057E050           020 H 25 071   19
=057E060           020 $  0 010 = 29      ERASE RIGHT LINKED LIST A/C MQ
=057E070      *  9 010 ----              ,
=058E000+     = 58 010 $  0              RELEASE CURRENT BUFFER FOR STN
=058E010           024 K  2 052 *  9        SERVICES L7 WITH FREE BUFFER AND
=058E020           020 H 12 050 * 92        LEAVES NEXT BUFFER OR -1 IN ACC AND H1
=058E030           024 U  1 052 *  1
=058E040           056 *  8 010 $  1
=058E050      *  1 020 ---- 050 H 13
=058E055           002 $  1 010 *  9
=058E060           024 U  5 056 *  2
=058E070      *  2 056 *  3 004 ----      PICK UP NEXT BUFFER (IF ANY)
=058E080      *  3 020 K  4 050 ----      SET SUCCESSOR OF OLD BUFFER TO -1
=058E090           060 * 93 050 T  0
=058E100           020 H 12 056 T  0
=058E110           020 T  0 142 K 16      SET BUFFER BITS IN SCR FOR NEW BUFFER
=058E120           020 * 91 010 $  1      FIND TAIL OF H10 LIST
=058E130      *  4 052 *  5 056 *  6
=058E140      *  5 020 ---- 001 *  6
=058E150           024 U  5 010 *  4
=058E160      *  6 020 H 13 050 ----      PUT OLD BUFFER ON END OF H10 LIST
=058E170           020 L  7 005 *  7      SERVICE L7
=058E180           050 H 12 010 $  1
=058E190           020 $  0 010 =  8      ASSIGN BUFFER
=058E200           020 $  0 010 = 47      CHANGE STATE TO L10
=058E210             L 10     L 10
=058E220           020 H 12 024 K 36      CL+SU
=058E230           142 K 36 020 * 92
=058E240      *  7 050 H 12 020 * 93      RESTORE STN CONTEXT
=058E250      *  8 050 H 13 050 ----      LEAVE NEXT BUFFER OR -1 IN ACC AND H13.
```

```
=058E260      *  9 010 ----                          EXIT
=058E270      * 91      H 10      H 10
=058E280      * 92                                  PRIVATE STORAGE FOR STN
=058E290      * 93                          ,       PRIVATE STORAGE FOR BUFFER
=059E000+     = 59 010 $  0                         RELEASE DRUM FOR STN
=059E010           024 K  2 052 *  9
=059E020           020 H 12 024 U  2
=059E030           052 *  3 056 *  5
=059E035      *  5 056 *  4 020 ----
=059E040           001 *  9 020 * 90                EXIT IF NO DRUM ASSIGNED
=059E050      *  1 052 *  2 056 *  3
=059E060      *  2 020 ---- 001 *  3
=059E070           024 U  6 010 *  1
=059E080      *  3 020 ---- 050 ----                PUT DRUM ON END OF H11 LIST
=059E090      *  4 020 K  4 050 ----                UNASSIGN STN
=059E100      *  9 010 ----                         EXIT
=059E110      * 90      H 11      H 11,
=060E000+     = 60 010 $  0                         INITIALIZE DRUM FOR STN(H12)  (TEMP)
=060E010           024 K  2 052 *  9
=060E020           004 * 90 110,7000
=060E030           020 U  2 024 H 12                ADJUST DRUM TABLES
=060E040           052 *  1 020 H 11
=060E050      *  1 050 ---- 024 U  6
=060E060           052 *  2 056 *  3
=060E070      *  2 020 ---- 050 H 11
=060E080      *  3 020 K  4 050 ----
=060E090           020 H 12 050 H  7                ESTABLISH STN AS IN CORE
=060E100      *  9 010 ----
=060E110      * 90 000,0000,010,0777,
=061E000+     = 61 010 $  0                         CONVERT TIME FOR OUTPUT
=061E010           024 K  2 052 *  9
=061E020           020 $  0 010 =  0                READ THE CLOCK
=061E030           120    0 004 H  1
=061E040           044 J  1 120    0
=061E050           044 J  0 036 N  2
=061E060           060 H 20 010 $  1
=061E070           020 $  0 010 = 36
=061E080                3           4
=061E090      *  9 010 ----                  ,
=062E000+     = 62 050 T  0 010 $  0                INSERT SPACE IN OUTPUT REGION S
=062E010           020 A 14 050 P 11
=062E020           020 T  0 010 = 66,
=063E000+     = 63 010 $  0                         COPY STEP NUMBER FROM R TO S
=063E010           024 K  2 052 *  9
=063E020           020 J 12 050 P  2
=063E030      *  1 020 $  0 010 =  1
=063E040           124 A 14 025 A 14
=063E050      *  9 002 ---- 010 $  1
=063E060           020 $  0 010 =  4
=063E070           010 *  1                  ,
=064E000+     = 64 010 $  0                         UNPACK MSG TO S FOR OUTPUT COMPOSITION
=064E010           024 K  2 052 *  1                   020 $  0 010 = 64
=064E020           024 K  2 052 *  9                          M         M
=064E030      *  1 020 ---- 052 *  2                WHERE 'M' IS LOCN OF MSG
=064E040      *  2 020 ---- 050 T  0
=064E050      *  3 020 T  0 001 *  9
=064E060           071    8 050 T  0
=064E070           020 P 12 024 K  3
=064E080           050 P 12 052 *  4
=064E090      *  4 060 ---- 023 T  0
```

```
=064E100          001 * 3 020 * 2
=064E110          024 K 2 014 * 1
=064E120      * 9 010 ----                ,
=065E000+     = 65 010 $ 0                    OUTPUT CANNED MSG TO STN
=065E010          024 K 2 052 * 1               020 $ 0 010 = 65
=065E020          020 S 0 050 P 12                    M       M
=065E030      * 1 020 ---- 050 * 2
=065E050          020 $ 0 010 = 64
=065E060      * 2 ----    ----
=065E070          020 * 1 010 = 25,
=066E000+     = 66 010 $ 0                    PUT (P11) IN NEXT S CELL (P12)+1
=066E010          024 K 2 052 * 9
=066E020          020 P 12 024 K 3
=066E030          050 P 12 056 * 1
=066E040      * 1 020 P 11 050 ----
=066E050      * 9 010 ----                ,
=067E000+     = 67 004 K 3 010 $ 0            BLOCK TRANSFER R TO S.
=067E010          024 K 2 052 * 9
=067E020          020 * 90 050 * 1
=067E030      * 1 020 ---- 050 ----
=067E040          020 * 1 064 * 1
=067E050          025 * 91 001 * 1
=067E060      * 9 010 ----
=067E070      * 90 020 R 1 050 S 1
=067E080      * 91 020 R 74 050 S 74,
=068E000+     = 68 010 $ 0                    COMPUTE PART (P20) FOR STEP (P21)
=068E010          024 K 2 052 * 9
=068E020          020 P 21 124 K 11
=068E030          050 T 0 070 10
=068E040          050 T 1 020 * 90
=068E045          025 T 1 052 * 1           ASSUMES (P21) IS LEGITIMATE
=068E050          052 * 2 020 P 21
=068E060          124 K 13 072 39
=068E070      * 1 044 ---- 020 T 0
=068E080      * 2 036 ---- 060 P 20
=068E090      * 9 010 ----
=068E100      * 90 N 8                    ,
=069E000+     = 69 010 $ 0                    INSERT PERIOD AND CR+EOM IN OUTPUT
=069E010          024 K 2 052 * 9
=069E020          020 A 27 050 P 11
=069E030          020 $ 0 010 = 66
=069E040          020 A 42 050 P 11
=069E050          020 $ 0 010 = 66
=069E060      * 9 010 ----                ,
=070E000+     = 70 010 $ 0                    COMPARE (H21) W (H22)
=070E010      * 1 024 K 2 052 * 9
=070E020          020 H 21 025 H 22
=070E030          005 * 2 025 K 1
=070E040      * 2 001 * 12 023 H 22
=070E050          005 * 3 020 H 21
=070E060          071 9 001 * 11
=070E070      * 3 010 * 13 023 H 21
=070E080          005 * 4 020 H 22
=070E090          071 9 001 * 13
=070E100      * 4 010 * 11 020 H 21
=070E110          124 K 12 024 H 22
=070E120          124 K 12 025 K 12
=070E130          005 * 5 020 H 21
=070E140          071 9 001 * 11
=070E150      * 5 010 * 13 020 H 22
```

```
=070E160          071     1 072     1
=070E170          050 T   0 020 H 21
=070E180          071     1 072     1
=070E190          025 T   0 001 *   6
=070E200          020 H 21 071      9
=070E210          002 * 13 010 *  11
=070E220      *   6 020 H 21 071      9
=070E230          002 * 11 010 *  13
=070E240      *  11 020 K 26 010 *   9     LESS
=070E250      *  12 020 K 25 010 *   9     EQUAL
=070E260      *  13 020 K 24 010 *   9     GREATER
=070E270      *   9 010 ----           ,    EXIT WITH RELATION IN ACC
```