FLEX NEWSLETTER NO. 1
February 1979

P.O. Box 2574, West Lafayette, Indiana 47906

Here's the first edition of the FLEX Newsletter. We've got some helpful tips on using FLEX, some fixes for FLEX utilities, news on the latest TSC software, and more.

First of all let us tell you about some of our new offerings for the FLEX operating system. Later in this newsletter there is a reprint of a recent ad showing most of our new software. The first item listed is a new version of FLEX for owners of the SWTPc MF-68 disk system and for Smoke Signal disk system owners. This version, called FLEX 2.0, is EXACTLY like the version of FLEX SWTPc includes with their 8" DMAF-1 system (called FLEX 1.0) except for the disk driver routines. In order to avoid confusion, let us reiterate on some nomenclature.

mini FLEX
: This is the version which SWTPc includes with the 5" MF-68 disk system. It is located at $7000.

FLEX 1.0
: This is the version which SWTPc includes with the 8" DMAF-1 disk system. It is located at $A000.

FLEX 2.0
: This is a new version which can be implemented on the MF-68. It is software equivalent to FLEX 1.0 except for the disk drivers. It is located at $A000 hex.

The only drawback to FLEX 2.0 besides laying out another $75.00 is the fact that the software for mini FLEX will not be compatible with FLEX 2.0. This is, of course, due to the fact that they are located at different memory addresses. Thus calls to routines in mini FLEX would have to be changed to the addresses in FLEX 2.0 in order to function. If you have the source listing of a program which references mini FLEX on disk, you could edit the source and reassemble. This is the case with the mini FLEX utilities, for example. You could edit the source, changing the equates at the beginning of each utility to point to FLEX 2.0 instead of mini FLEX and reassemble. Then load this binary and save it out in FLEX 2.0 with the SAVE.LOW command.

There will be a utility available to convert text files from mini FLEX to FLEX 2.0. This is possible because both operating systems can be in memory at once. Any binary files which do not have jumps or calls to FLEX can be converted with this utility or can be loaded into memory from mini FLEX and saved back out under FLEX 2.0. Note that FLEX 2.0 includes the TSC Text Editing System and the TSC Mnemonic Assembler so there will be no need to upgrade these programs. They do not, however, include the source listing. It can be purchased separately at the

normal price as listed in the latest TSC software catalog.

The Sort/Merge package is an extremely powerful tool for use with any type of data file you wish to have sorted. The short description in the ad does no justice to the completeness of the package. The 70 page manual was written with the non-computerist in mind and smoothly guides the operator through the use of the Sort/Merge package with many examples.

TSC BASIC is finally getting near! The first version will be a cassette version and will be available around March 1st. It is part number AP68-11C and sells for $39.95. There is no source listing included in this package. It is about 9K in length and VERY fast! In fact, it's the fastest floating point BASIC for ANY micro. It will be comparable in command scope to the latest Microsoft 8K BASIC. It includes such features as six digit floating point math, full transcendental functions, unlimited string length, IF/THEN/ELSE construct, logical operators, and two-dimensional arrays (including string arrays). In all there are over 50 commands and functions. There will be FLEX versions available around April 1st. The FLEX 1.0 and FLEX 2.0 versions will have random access file capabilities while the mini FLEX version will have only sequential. Note that there will be no upgrade policy for those who purchase the cassette version and then wish to purchase a disk version. Also in the works is a 6809 version and a business version which will have 12 digit math capability and extended output formatting such as is required by business applications.

1) USE OF THE "P" COMMAND

There seems to be much confusion among beginning user's of FLEX regarding the use of the "P" command or about sending data to the line printer device in general. This section is intended to clear up this confusion and get you on your way to full use of FLEX's output versatility.

When FLEX is booted up, ANYTHING which is output is sent to the command terminal or console. If, for example, you do a "LIST" of some text file, you will see the contents of that file printed on the terminal. If you perform an assembly with the TSC Mnemonic Assembler, the output (if selected) will be sent to the terminal or console. Suppose we wanted the output of the assembly sent to the printer instead. How would we go about doing that? It's really very easy. You simply precede the assemble command with a "P" command. For example, instead of the command:

```
ASMB PROG1 +BG
```

You would enter the command:

```
P ASMB PROG1 +BG
```

The "P" command simply says to FLEX that anything output during the following command should go to the printer instead of the terminal.

This works for ANY command. Thus we could "LIST" a file called "TEXTFILE" and have the result printed on the line printer rather than the CRT or terminal by simply entering the command:

```
P LIST TEXTFILE
```

You see how simple it really is? ANYTHING which normally goes to the CRT screen through FLEX I/O routines (like PUTCHR & GETCHR) can be routed to the line printer by preceding the command with a "P" command.

There is, of course, a catch. The "P" command requires a couple of driver routines in order to know how and where to output characters so that they go to the printer. These driver routines cannot be supplied by the distributor of FLEX as he has no way of knowing what your printer requires in the way of drivers nor where it is located in the address space. For this reason, you (the user) must supply your own driver routines. They should be placed in a file called "PRINT.SYS" as explained in the FLEX User's Manual and Advanced Programmer's Guide. Usually FLEX is supplied with a sample PRINT.SYS file which has driver routines for a parallel port on port #7 which will operate a standard Centronics interface printer (such as the SWTPc PR40 or the Okidata model 22). As we have received many requests from people who were unable to develop their own drivers for a serial port, a sample set of drivers for an ACIA is listed in section 5 of this newsletter.

Now, how does the "P" command perform this magic? It's really quite simple also. The "P" command simply changes the output character routine in FLEX from that of your terminal to that of the printer driver which is supplied in PRINT.SYS. Then it sets the command buffer to point to the next thing on the command line and continues to read the command as if it had not even seen the "P" which preceded it.

Now if that was all there was to it, you can see that we would never get anything printed on the terminal again unless we went back in and changed the output character routine address back to that of the terminal. FLEX does that for us, automatically. FLEX always keeps the address of the terminal output routine at OUTCH2. Everytime a warm start is executed (usually indicated by the 3 plus signs), the output character routine address is always reset to the terminal's output routine. Thus the "P" command will only be in effect during execution of the command which it preceded.

There is one more trick which can be made use of by a FLEX programmer. It allows you to send data to BOTH the printer and the terminal during execution of a particular program or utility. An example of this is in the TSC Assembler where there are prompts to the user such as "DELETE EXISTING BINARY (Y/N)?". These should obviously be sent to the terminal even though the output has been routed to the printer via the "P" command. This is done by use of the OUTPUT SWITCH flag. If this flag is set to some non-zero value, anything output via the PUTCHR routine will be sent to the terminal regardless of whether the "P" command has been typed or not. If the OUTPUT SWITCH is zero, output will be sent through the normal OUTCH routine which will be to the printer if the "P" command was typed. Note again that a warm start will clear this OUTPUT SWITCH to zero.

One final point to consider. There is nothing special about the "P" utility itself. It is listed in full in the Advanced Programmer's Guide. There is no reason a user cannot write another command similar to the "P" command which would send the output to a paper tape punch, cassette, or even a second printer. Simply call the new routine "PT", "CAS", "P2", or whatever you like. The new routine will also require a new file similar to PRINT.SYS which contains the appropriate driver routines. You would have to call the new driver routine something different such as "PRINT2.SYS".

2) DRIVE AND EXTENSION DEFAULTS

Another commonly misunderstood feature of FLEX is that of drive number and filename extension defaulting. To FLEX, a file specification consists of three parts, the drive number, the filename, and the extension. Refer to the FLEX User's Manual for further details on these specifications. All specifications of files, whether it be in a command line or within the operation of some program (such as when the TSC Editor asks you for a filename when you perform a READ from disk) can have all three of these parts. It is often possible to omit two of these parts when typing a file specification. These are the drive number and the extension. Note that these values are still supplied to FLEX for operation on that file, it's just that they will take on some default value so that the user doesn't have to type so much. Note that the filename portion of the file spec must ALWAYS be supplied.

Let's start with the drive number. If you type "0.NAME.TXT", FLEX will look on drive #0 for a file named "NAME.TXT". If you just type "NAME.TXT", a default drive number will be used. This default number can be one of two values. If the file is specified as a command file (the first filename given on a command line) the default drive number will be set to the assigned system drive (see the ASN command description in FLEX). ANY OTHER file specification without a drive number will default to the assigned working drive. Thus as an example,

```
EDIT NAME.TXT
```

would go to the assigned system drive to get the editor and would go to the assigned working drive to get the file "NAME.TXT".

Now on to the extension defaults. It is not always necessary to include the extension in a file specification. If it is omitted, some default extension will be assumed. The actual extension used as the default can vary. If the file specification is a command file specification (again, the first on a command line) the default extension will be ".CMD". Thus in the edit example given directly above, FLEX would go to the system drive and look for a file named "EDIT.CMD".

For file specifications other than those which are first on a command line, you will have to consult the description of the particular command or program in order to determine what the default extension will be. For example, the TSC Text Editing System always assumes a default of ".TXT" for files being edited. Thus if you don't specify an extension

it will always use ".TXT". Our most recent example could have been equivalently entered as:

```
EDIT NAME
```

Some other common defaults are:

Command	Default Extension
APPEND	TXT
LIST	TXT
SAVE	BIN
ASMB	TXT for file1 - BIN for file2

Note that in the TSC Assembler there can be two files specified on the command line and they have different defaults for the extensions. Some commands, such as DELETE, don't allow defaults. It is necessary that you specifically enter an extension when using the DELETE command and you will be prompted for such if not included.

3) FULL DIRECTORY PROBLEM IN MINI FLEX

There is a problem in mini FLEX (the version supplied with the MF-68 disk system from SWTPc) which does not occur often, but is not too pleasant when it does. It occurs when you have filled the directory of a disk (exactly 75 files), then delete one or more of those files, and then try to write another file to the disk. This will crash the directory on the disk and thus cause you to lose data. Unfortunately, there is no simple patch to fix this problem. The best solution is to just be aware of it and not get into such a situation. If you get a directory full error, be SURE not to attempt to delete a file. The disk will be OK to use for reading files, but cannot be used for any other purpose since trying to write to it will give you a directory full error and trying to delete a file will give you the fatal opportunity to write to the disk which will cause the directory to be lost. It's best not to perform a delete at all. As stated, if you only wish to read from the disk, there is no problem and the disk could be used for such purposes forever. However, to make things safer (so that you don't unintentionally perform a delete and write) you should copy the files from the disk to two other disks (so that their directories aren't full) and then reformat or "NEWDISK" the one with the full directory. If you don't need all the files on the disk with a full directory, simply copy those you want to a single other disk and then reformat the filled one.

This problem does not exist in FLEX 1.0 or FLEX 2.0. In fact, the number of files in these versions is not limited to 75.

4) FIXES FOR BUGS IN FLEX UTILITIES (UV1-6)

Four bugs have been reported in the additional utilities (volumes 1-6) for the mini FLEX operating system. Two of these are also in the utilities for the 8" FLEX 1.0. The fixes for these bugs follow.

A) The FIND Utility - Volume 1, Number 1

The find utility does not always perform an FMS close function before returning to warm start. This can cause problems if using the FIND utility in an EXEC command. The problem is in both the mini FLEX and 8" FLEX 1.0 versions. The fixes are not the same in both cases. First for the mini FLEX version:

Change the instruction at $01B7 from FIND35 JMP WARMS
to FIND35 JMP ERROR8

For the large disk FLEX 1.0:

Change the instruction at $A17F from JMP WARMS
to JMP FIND35
Then change the instruction at $A195
from FIND35 JMP WARMS
to FIND35 JSR FMSCLS
followed by JMP WARMS

B) The MAP Utility - Volume 3, Number 1

The mini FLEX version of MAP has a bug which can be fixed by replacing the routine called "PRTEND" at $01F5 thru $0201 with the following:

```
PRTEND  LDX   PREV     GET ADDRESS
        DEX            DECREMENT IT
        STX   PREV     SAVE BACK OUT
        LDX   #PREV    GET PREVIOUS ADDRESS
        JSR   OUTHEX   OUTPUT IT
        INX
        JSR   OUTHEX
        RTS
```

This change has already been made in the 8" FLEX 1.0 version of MAP.

C) The TEST Utility - Volume 3, Number 6

The TEST Utility in the mini FLEX version has a call to a routine called OUTADR which is non-existant in mini FLEX. OUTADR is in FLEX 1.0 so the TEST utility for that DOS is OK. The mini version can be fixed as follows:

Replace the instruction at $7635 (JSR OUTADR)
with the following instructions
JSR OUTHEX
INX
JSR OUTHEX

D) The PDEL Utility - Volume 5, Number 2

This problem is in both versions of the utilities and the fix is the same for each:

Change the instruction at $029B in the mini FLEX
version or $A29E in the 8" FLEX 1.0 version
from BEQ DODLR5
to BNE DODLR6

5) "PRINT.SYS" ROUTINES FOR A SERIAL PORT

The following routines may be used as a guide to assembling your own PRINT.SYS file for a serial port. You should read the instructions in the FLEX User's Manual and the FLEX Advanced Programmer's Guide to learn just where to place these routines and in the mini FLEX version where to place the calling addresses for these routines. Note that the "PCHK" routine given is not needed for the mini FLEX version.

Before using the routines below, you will have to set up the proper value for the ACIA output port you wish to use. This is done with an equate statement. For example to output via an ACIA on port 7 you should use:

```
ACIA EQU $801C
```

The output character routine (OUTCHR or POUT) should look like this:

```
OUTCHR  PSH B          SAVE B ACC.
OUTCH2  LDA B  ACIA    GET STATUS
        ASR B          GET TDR BIT
        ASR B          INTO CARRY
        BCC    OUTCH2  LOOP IF NOT READY
        PUL B          RESTORE B ACC.
        STA A  ACIA+1  WRITE OUT THE CHAR.
        RTS            RETURN
```

The initialization routine (INIT or PINIT) should look like:

```
INIT    LDA A  #$13    RESET ACIA
        STA A  ACIA
        LDA A  #$11    SET 8 BITS & 2 STOP
        STA A  ACIA
        RTS            RETURN
```

If using FLEX 1.0 or FLEX 2.0 you'll need a printer check routine (PCHK)

like this:

```
PCHK    PSH B          SAVE B ACC.
        LDA B  ACIA    GET STATUS
        ROR B          GET TDR BIT INTO
        ROR B          SIGN POSITION
        ROR B
        PUL B          RESTORE B ACC.
        RTS            RETURN
```

Recall that if using mini FLEX you'll have to setup the proper addresses with statements like:

```
        ORG    $0010
        FDB    INIT    SETUP INITIALIZATION ADDR.

        ORG    $710D
        FDB    OUTCHR SETUP OUTPUT CHAR. ADDR.
```

6) BASIC RENUMBER UTILITY by Ron Anderson

It is not our intention to distribute free software through the FLEX Newsletter. Under certain circumstances we might do so, however, with user-submitted utilities. That is the case in this issue. Ron Anderson of Ann Arbor, Michigan, submitted this BASIC Renumber Utility for which we have seen much interest. We are reprinting it here for FLEX Newsletter subscribers exactly as Ron sent it to us. Although we have run the program several times with no problems whatsoever, we make no guarantees on its operation and will not support technical calls regarding the utility. Instructions for use are included in the source listing and as you can see, Ron has done extensive commenting of the source code.

Ron Anderson, an avid computerist who has written several articles for Kilobaud magazine, is very much into the use of FLEX. In fact, he is starting a FLEX User's Group which will be dedicated to distributing user-submitted programs for essentially the cost of copying and distribution. According to Ron, there will be about 12 issues per year. If you are interested in joining this group, write to the following address for more information:

The FLEX User's Group
Attn: Ron Anderson
3540 Sturbridge Court
Ann Arbor, MI 48105

The assembled source listing for the BASIC Renumbering Utility is included at the end of this newsletter.

Technical Systems Consultants, Inc.

The Software Company

TSC, Technical Systems Consultants, is **the software company** for all the newest, most innovative ideas in computer software. TSC builds a variety of programs, packages and games so you can get down to business or just some fun.

FLEX for SWTPc and SSB

Now owners of Smoke Signal Broadcasting's BFD-68 or LFD disk systems can enjoy all the power and convenience of the FLEX disk operating system. SWTPc MF-68 owners can step up to an enhanced version of FLEX. The SWTPc MF-68 comes with a version called "mini FLEX", while the SWTPc DMAF-1 comes with "FLEX 1.0". There are three new versions, all identical to FLEX 1.0 except for the disk drivers. They are FLEX 1.0 for the SSB LFD, FLEX 2.0 for the SSB BFD-68 and FLEX 2.0 for the SWTPc MF-68. The new FLEX's require 8K of RAM at location $A000. This means all FLEX based software can now be run on a Smoke Signal system.

Some of FLEX's features are:

- Simple command structure
- Dynamic file allocation
- Automatic space compression
- Extensive software support

Enhancements to FLEX 1.0 & 2.0:

- *Full 32K available to user*
- Printer Spooling (requires timer)
- Random Access files
- Input/Output file capability
- File protection and dating
- More data per disk

Included are an object code disk with FLEX, the utility command set, the TSC Text Editing System, and the TSC Mnemonic Assembler, the FLEX User's Manual, FLEX Advanced Programmer's Guide, Text Editor manual, and Assembler manual (no source listings included).

FLEX 1.0 for SSB LFD	**$100.00**
FLEX 2.0 for SSB BFD-68	**$ 90.00**
FLEX 2.0 for SWTPc MF-68	**$ 75.00**

SORT/MERGE Package

This FLEX compatible package allows any size and type file to be sorted according to parameters you specify. Written in 6800 assembly language, it is extremely fast. Sort parameters may be supplied in three ways: as part of the command line, through use of a "parameter editor", or through an existing parameter file. The package is a full-disk sort/merge meaning that files too large to fit in memory will be broken into multiple, temporary work files which are individually sorted and then merged into one. The final output file may be routed to disk, CRT, or printer. Features include:

- Fixed or variable length records
- Fixed or variable length fields
- Definable record & field terminators
- Accepts multiple input files
- *Up to 20 input or output keys*
- Sort key up to 250 characters
- Ascending or descending keys
- Right or left justified keys
- Select/exclude capability
- Non-ASCII collating sequences
- Sorts upper case equal to lower
- Merge-only capability

Includes extensive user's manual and object code diskette. No source listing is included. Specify FLEX 1.0, FLEX 2.0, or mini FLEX.

AP68-10 6800 Sort/Merge Package $75.00
Manual Only $15.00

FLEX Software

TSC plans full support of the FLEX disk operating system. Several software packages are already available as listed below. Watch our ads for FLEX compiler BASIC available soon!

TSC Text Processing System

SL68-29D	with Mini FLEX disk	$ 40.00
SL68-29F1	with FLEX 1.0 disk	$ 75.00
SL68-29F2	with FLEX 2.0 disk	$ 65.00

TSC 6800 Debug Package

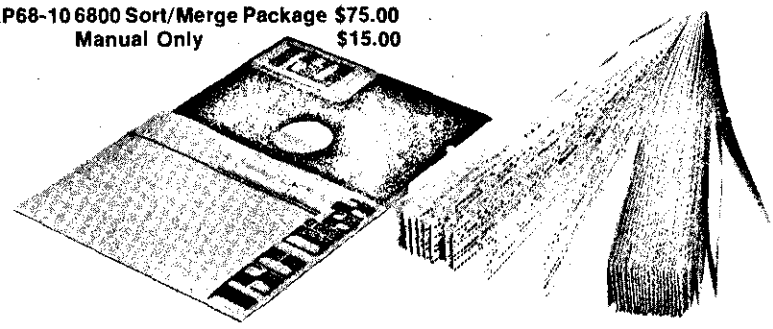
SL68-30D	with Mini FLEX disk	$ 43.00
SL68-30F1	with FLEX 1.0 disk	$ 55.00
SL68-30F2	with FLEX 2.0 disk	$ 50.00

36 Additional Disk Utility Commands

UV1-6D	with Mini FLEX disk	$ 99.95
UV1-6F1	with FLEX 1.0 disk	$124.95
UV1-6F2	with FLEX 2.0 disk	$119.00

FLEX Newsletter
Published irregularly (about 4 per year) with latest news, hints, and user feedback. $4.00 for four issues.

All orders should include check or money order. Add 3% for postage (8% foreign) and for orders under $10, please add $1 handling. Send 25¢ for a complete software catalog.



```
                    * THIS IS A UTILITY PROGRAM FOR THE SWTPC TSC FLEX
                    * OPERATING SYSTEM.  IT WILL RENUMBER A BASIC PROGRAM FILE
                    * WITH THE STARTING LINE NO.  AND THE INCREMENT SPECIFIED.
                    *
                    * THE COMMAND FORMAT IS:  RENUM,FILENAME,START,INCREMENT
                    * START AND INCREMENT DEFAULT TO 10,10
                    *
                    * START IS IN LOC $01B5 AND $01B6, NOW 0010.  MAY BE
                    * CHANGED AS DESIRED, IE.  SET TO 0100 TO DEFAULT TO
                    * STARTING LINE 100
                    *
                    * INCREMENT IS IN LOC $01BF AND $01C0 AND MAY BE
                    * ALTERED SIMILARLY AS DESIRED.
                    *
                    * THE DEFAULT EXTENSION IS .BAS.  ONLY BASIC FILES WILL
                    * BE RENUM'D.  START AND INC. NEED NOT HAVE LEADING
                    * ZEROES INCLUDED.  THE FILE TO BE RENUMBERED NEED NOT
                    * HAVE 4 DIGIT LINE NO'S OR 4 DIGIT REF'S FOLLOWING
                    * THEN, GOSUB, AND GOTO.  RENUM WILL IGNORE A THEN
                    * FOLLOWED BY A STATEMENT, BUT WILL CHANGE A LINE NUMBER
                    * FOLLOWING 'THEN'.  RENUM DOES CHECK TO SEE IF
                    * LINE NUMBERS ARE INCREMENTED BEYOND 9999.
                    * RENUM VER3 DELETES ALL LEADING ZEROES FROM
                    * LINE NO'S IN THE RENUM'D FILE.  THE ORIGINAL FILE IS
                    * SAVED AS (FILENAME).BAK.  THE NEW FILE RETAINS THE
                    * ORIGINAL (FILENAME).BAS

0020                        ORG     $20

0020                BUFPTR  RMB     2       BUFFER POINTER
0022                TABPTR  RMB     2       LINE NUMBER TABLE POINTER
0024                FSTLIN  RMB     2       FIRST LINE NUMBER STORAGE
0026                INCR    RMB     2       INCREMENT
0028                OLDLIN  RMB     2       OLD LINE NUMBER
002A                ASCII   RMB     4       USED BY HEX ASCII CONVERSION
002E 00             NFLG    FCB     0       FLAG TO CONTROL PADDING OF NO'S
002F 00             PCN     FCB     0       PREV. CHAR. WAS NO.  IF NOT 0
0030 00             PCD     FCB     0       PREV CHAR.  WAS DEC.  PT.

                    * SYSTEM EQUATES (FLEX)

7740                FCB     EQU     $7740   FILE CONTROL BLOCK
710F                GETCHR  EQU     $710F   GET CHARACTER
7112                PUTCHR  EQU     $7112   PUT CHARACTER
7118                PSTRNG  EQU     $7118   PRINT STRING
712D                SETEXT  EQU     $712D   SET EXTENSION
7806                FMS     EQU     $7806   FILE MANAGEMENT SYSTEM
7803                FMSCLS  EQU     $7803   CLOSE FILES ON ERROR
7103                WARMS   EQU     $7103   FLEX WARMSTART
7139                OUTHEX  EQU     $7139   OUTPUT HEX NUMBER
7127                GETFIL  EQU     $7127   GET FILE SPEC FROM LINE BUFFER
```

```
713F             GETHEX  EQU     $713F       GET HEX NUMBER FROM LINE BUFFER
713C             RPTERR  EQU     $713C       REPORT ERROR

0100                     ORG     $100
0100 20 01       START   BRA     BEGIN
0102 03          VER     FCB     3           VERSION # OF UTILITY PROGRAM

                 * ROUTINE TO GET FILE TO BE RENUMBERED FROM
                 * DISK INTO MEMORY STARTING AT LOCATION
                 * 'BUFFER'.

0103 CE 77 40    BEGIN   LDX     #FCB        POINT AT FCB
0106 C6 08               LDA B   #8          SET UP CLEAR
0108 6F 04       CLEAR   CLR     4,X
010A 08                  INX
010B 5A                  DEC B
010C 26 FA               BNE     CLEAR       CLEARED FILENAME SPACE IN FCB
010E CE 77 40            LDX     #FCB
0111 C6 01               LDA B   #1
0113 E7 00               STA B   0,X         OPEN FOR READ CODE
0115 E7 03               STA B   3,X         DRIVE NUMBER
0117 BD 71 27            JSR     GETFIL      FILENAME FROM CMD LINE TO FCB
011A 86 03       DONFIL  LDA A   #3
011C CE 77 40            LDX     #FCB
011F BD 71 2D            JSR     SETEXT      SET EXT. TO .BAS IF NOT GIVEN
0122 CE 77 40            LDX     #FCB
0125 BD 78 06            JSR     FMS         GO OPEN FILE
0128 26 69               BNE     ERROR       HANDLE ERROR FLAG
012A CE 0A B0            LDX     #BUFFER     POINT AT BUFFER STARTING LOC.
012D DF 20               STX     BUFPTR      SAVE POINTER
012F 7F 77 40            CLR     FCB
0132 CE 77 40    READ    LDX     #FCB
0135 BD 78 06            JSR     FMS
0138 27 08               BEQ     CONTIN      NO ERROR DETECTED
013A E6 01               LDA B   1,X         GET ERROR TYPE
013C C1 08               CMP B   #8          IS IT END OF FILE?
013E 27 36               BEQ     CLOSE       END OF FILE
0140 20 51               BRA     ERROR       ANOTHER ERROR TYPE
0142 DE 20       CONTIN  LDX     BUFPTR      HAVE FIRST BYTE OF FILE IN ACCA
0144 16                  TAB                 PUT IN ACCB FOR TESTING
0145 C1 2E               CMP B   #$2E        DECIMAL POINT?
0147 27 02               BEQ     CONT4       Y. DONT PAD NOS. FOLL. WITH 0'S
0149 20 06               BRA     CONT3       NO, CONTINUE TESTS
014B C6 FF       CONT4   LDA B   #$FF        SET NFLG AS THO PAD ZEROS
014D D7 2E               STA B   NFLG        ---WERE ALREADY ADDED
014F 20 19               BRA     CONT2       NO MORE TESTS REQ'D
0151 BD 01 99    CONT3   JSR     NUMBER      TEST FOR NUMBER
0154 25 1B               BCS     CONT1
0156 7D 00 2E            TST     NFLG        ALREADY PADDED?
0159 26 0F               BNE     CONT2       YES, DONE TESTING
015B C6 FF               LDA B   #$FF        SET NFLG
015D D7 2E               STA B   NFLG
015F C6 30               LDA B   #$30        PAD NO. W/3 LEADING 0'S
0161 E7 00               STA B   0,X
```

```
0163 08                  INX
0164 E7 00               STA B  0,X
0166 08                  INX
0167 E7 00               STA B  0,X
0169 08                  INX
016A A7 00      CONT2    STA A  0,X        STORE NO. AFTER PAD
016C 08                  INX
016D DF 20               STX    BUFPTR     SAVE BUFFER PTR
016F 20 C1               BRA    READ       GET ANOTHER CHARACTER
0171 7F 00 2E   CONT1    CLR    NFLG       CHAR. WAS NOT NO.
0174 20 F4               BRA    CONT2

                * FILE NOW IN BUFFER

0176 DE 20      CLOSE    LDX    BUFPTR     GET END OF FILE LOC.
0178 86 1A               LDA A  #$1A
017A A7 00               STA A  0,X        MARK END OF FILE
017C CE 77 40            LDX    #FCB
017F 86 04               LDA A  #4
0181 A7 00               STA A  0,X        SET UP CLOSE FILE COMMAND
0183 BD 78 06            JSR    FMS        DO IT
0186 26 0B               BNE    ERROR      HANDLE ERROR
0188 20 1B               BRA    GETINF     GO GET LINE AND INCREMENT INFO

                * ROUTINE FOR ERROR HANDLING

018A BD 71 18   EREXIT   JSR    PSTRNG     PRINT ERROR MESSAGE
018D BD 78 03   EREX1    JSR    FMSCLS     CLOSE FILES
0190 7E 71 03            JMP    WARMS      EXIT TO DOS


0193 BD 71 3C   ERROR    JSR    RPTERR
0196 7E 01 8D            JMP    EREX1

0199 C1 39      NUMBER   CMP B  #$39
019B 2E 06               BGT    NOT
019D C1 2E               CMP B  #$2E
019F 2F 02               BLE    NOT
01A1 0C                  CLC
01A2 39                  RTS
01A3 0D         NOT      SEC
01A4 39                  RTS

01A5 CE 0A B0   GETINF   LDX    #BUFFER
01A8 DF 20               STX    BUFPTR     SAVE POINTER
01AA CE 04 B0            LDX    #TABLE     POINT AT NUMBER TABLE START
01AD DF 22               STX    TABPTR     SAVE POINTER
01AF BD 71 3F            JSR    GETHEX     GET FIRST LINE NO.
01B2 26 03               BNE    GET1       VALID IF NOT ZERO
01B4 CE 00 10            LDX    #$10       DEFAULT FIRST LINE NUMBER
01B7 DF 24      GET1     STX    FSTLIN     HEX LINE NUMBER
01B9 BD 71 3F            JSR    GETHEX
01BC 26 03               BNE    GET2
01BE CE 00 10            LDX    #$10
```

```
01C1 DF 26      GET2    STX     INCR        INCRENEMT

                * NOW SEARCH FILE FOR OLD LINE NUMBERS.  THE FIRST IS
                * THE FIRST NUM. ASCII CHARACTERS IN BUFFER, THE REST
                * FOLLOW CARRIAGE RETURN (#0D).

01C3 DE 20      LINNO   LDX     BUFPTR      POINT AT FIRST LINE NUMBER
01C5 BD 03 C0   LINNO1  JSR     SETPTR      ADJ PTR TO 4TH NO. FROM RT.
01C8 BD 02 23           JSR     ASHEX       GO CONVERT ASCII TO HEX
01CB DF 20              STX     BUFPTR      SAVE POINTER FOR FURTHER SEARCH
01CD DE 22              LDX     TABPTR      POINT AT FIRST LOC. IN TABLE
01CF D6 28              LDA B   OLDLIN      HAS HEX LINE NO. FROM ASHEX
01D1 E7 00              STA B   0,X         PUT IN TABLE
01D3 D6 29              LDA B   OLDLIN+1    LOW ORDER DIGITS
01D5 E7 01              STA B   1,X
01D7 08                 INX
01D8 08                 INX                 SKIP 2 LOCS.  FOR NEW NUMBER
01D9 08                 INX
01DA 08                 INX
01DB DF 22              STX     TABPTR      SAVE POINTER
01DD 8C 0A B0           CPX     #BUFFER
01E0 26 28              BNE     LINNO2
01E2 CE 01 E8           LDX     #MSG2       [TABLE TOO SMALL FOR PROGRAM
01E5 7E 01 8A           JMP     EREXIT
01E8 50         MSG2    FCC     /PROGRAM TOO LONG - 384 LINES MAX./
0209 04                 FCB     4
020A DE 20      LINNO2  LDX     BUFPTR      CONT.  SEARCH FOR NEXT LINE NO.
020C A6 00      SEARCH  LDA A   0,X
020E 08                 INX
020F 81 0D              CMP A   #$0D        IS IT CARRIAGE RETURN?
0211 26 F9              BNE     SEARCH      NO, GET NEXT CHAR.
0213 A6 00              LDA A   0,X         YES, TEST NEXT CHAR.
0215 81 1A              CMP A   #$1A        SEE IF END OF FILE
0217 27 02              BEQ     ENDFIL
0219 20 AA              BRA     LINNO1      NOT EOF, FOUND NEXT LINE NO.
021B DE 22      ENDFIL  LDX     TABPTR      EOF, MARK END OF NUMBER TABLE
021D 86 FF              LDA A   #$FF
021F A7 00              STA A   0,X         EOF MARK
0221 20 25              BRA     NEWNUM

                * ASCII TO HEX SUBROUTINES

0223 C6 01      ASHEX   LDA B   #1          BYTE COUNTER
0225 37                 PSH B               SAVE IT
0226 BD 02 43   ASHEX1  JSR     CONVT
0229 48                 ASL A               GOT FIRST DIGIT
022A 48                 ASL A               SHIFT IT LEFT TO PACK
022B 48                 ASL A
022C 48                 ASL A
022D 16                 TAB                 SAVE IT IN ACCB
022E 08                 INX                 POINT AT SECOND ASCII DIGIT
022F BD 02 43           JSR     CONVT
0232 1B                 ABA                 PUT TWO DIGITS TOGETHER
0233 33                 PUL B               GET BYTE COUNTER
```

```
0234 5D                 TST B               HAVE WE GONE AROUND BEFORE?
0235 27 07              BEQ     LOW         IF SO, LOW ORDER DIGITS
0237 97 28              STA A   OLDLIN      HIGH ORDER DIGITS CONVERTED SAVE
0239 5A                 DEC B               COUNT
023A 37                 PSH B               SAVE AGAIN
023B 08                 INX
023C 20 E8              BRA     ASHEX1      GET LOW ORDER DIGITS
023E 97 29      LOW     STA A   OLDLIN+1    THIS TIME, LOW ORDER
0240 08                 INX
0241 0C                 CLC
0242 39                 RTS

0243 A6 00      CONVT   LDA A   0,X         ASCII CHARACTER
0245 84 0F              AND A   #$0F        THROW AWAY '3'
0247 39                 RTS

                * NOW HAVE TABLE OF ALL OLD LINE NUMBERS.
                * NEXT FILL THE SPACES IN THE TABLE WITH THE NEW
                * ONES CALCULATED FROM FSTLIN AND INCR.

0248 CE 04 B0   NEWNUM  LDX     #TABLE
024B 08         NEWNU1  INX
024C 08                 INX                 POINT AT FIRST NEW LINE NO. LOC.
024D D6 24              LDA B   FSTLIN      HIGH ORD. TWO DIG. IN PACKED BCD
024F E7 00              STA B   0,X
0251 08                 INX
0252 D6 25              LDA B   FSTLIN+1
0254 E7 00              STA B   0,X
0256 08                 INX                 NOW POINTS AT SEC. LIN. OLD NO.
0257 E6 00              LDA B   0,X
0259 C1 FF              CMP B   #$FF        END OF TABLE?
025B 27 37              BEQ     CHNUM       IF SO, EXIT TO CHANGE ROUTINE
025D 96 25              LDA A   FSTLIN+1    NOT DONE, CALC. NEXT NO.
025F 9B 27              ADD A   INCR+1      ADD LOW ORDER DIGITS
0261 19                 DAA                 DECIMAL ADJUST
0262 97 25              STA A   FSTLIN+1    SAVE LOW ORDER
0264 24 E5              BCC     NEWNU1      DONE IF NO CARRY
0266 96 24              LDA A   FSTLIN
0268 8B 01              ADD A   #1          CARRY 1
026A 19                 DAA                 HIGH ORD. DIG. DEC. ADJUST
026B 25 06              BCS     OFLO
026D 97 24              STA A   FSTLIN      SAVE HIGH ORDER
026F E6 00              LDA B   0,X
0271 20 D8              BRA     NEWNU1      CONT. GEN. NEW NO.'S IN TABLE
0273 CE 02 79   OFLO    LDX     #MSG1
0276 7E 01 8A           JMP     EREXIT
0279 4E         MSG1    FCC     /NEW LINE NO'S EXCEED 9999./
0293 04                 FCB     4

                * NOW REPLACE OLD LINE NO'S WITH NEW

0294 CE 04 B0   CHNUM   LDX     #TABLE
0297 DF 22              STX     TABPTR      SET UP POINTER FOR TABLE
0299 CE 0A B0           LDX     #BUFFER
```

```
029C DF 20      CHNUM1  STX     BUFPTR      SET UP POINTER FOR BUFFER
029E DE 22              LDX     TABPTR
02A0 BD 02 CD           JSR     HEAS        CONVERT FIRST NEW NO. TO ASCII
02A3 DF 22              STX     TABPTR      SAVE POINTER FOR NEXT NUMBER
02A5 DE 20              LDX     BUFPTR
02A7 BD 03 C0           JSR     SETPTR      PT AT 4TH DIG FRM RT.
02AA D6 2A              LDA B   ASCII       PUT NEW NUMBER IN BUFFER
02AC E7 00              STA B   0,X
02AE 08                 INX
02AF D6 2B              LDA B   ASCII+1
02B1 E7 00              STA B   0,X
02B3 08                 INX
02B4 D6 2C              LDA B   ASCII+2
02B6 E7 00              STA B   0,X
02B8 08                 INX
02B9 D6 2D              LDA B   ASCII+3
02BB E7 00              STA B   0,X
02BD 08                 INX
02BE E6 00      CHNUM2  LDA B   0,X
02C0 08                 INX
02C1 C1 0D              CMP B   #$0D        LOOK FOR END OF LINE
02C3 26 F9              BNE     CHNUM2      KEEP LOOKING
02C5 E6 00              LDA B   0,X
02C7 C1 1A              CMP B   #$1A        SEE IF END OF FILE
02C9 27 2C              BEQ     CHREF       IF SO GO CHANGE REFERENCES
02CB 20 CF              BRA     CHNUM1      IF NOT GO CHANGE MORE NUMBERS

                * HEX TO ASCII CONVERSION SUBROUTINE

02CD C6 01      HEAS    LDA B   #1          COUNTER
02CF 08                 INX
02D0 08         HEAS1   INX                 NOW POINTING AT NEW NUMBER
02D1 A6 00              LDA A   0,X         GET FIRST DIGITS
02D3 84 F0              AND A   #$F0        MASK OFF LOW ORDER
02D5 44                 LSR A               SHIFT HI ORDER TO RIGHT
02D6 44                 LSR A
02D7 44                 LSR A
02D8 44                 LSR A
02D9 8A 30              ORA A   #$30        MAKE IT ASCII
02DB 5D                 TST B               COUNTER
02DC 27 0D              BEQ     HEAS2
02DE 97 2A              STA A   ASCII       FIRST DIGIT
02E0 A6 00              LDA A   0,X         GET PAIR AGAIN
02E2 84 0F              AND A   #$0F        THIS TIME MASK OFF HI ORD.
02E4 8A 30              ORA A   #$30        MAKE IT ASCII
02E6 97 2B              STA A   ASCII+1     SAVE SECOND DIGIT
02E8 5A                 DEC B               COUNT
02E9 20 E5              BRA     HEAS1       DO SEC. DIGIT PAIR
02EB 97 2C      HEAS2   STA A   ASCII+2     THIRD DIGIT
02ED A6 00              LDA A   0,X
02EF 84 0F              AND A   #$0F
02F1 8A 30              ORA A   #$30        MASK
02F3 97 2D              STA A   ASCII+3     LAST DIGIT
02F5 08                 INX
```

```
02F6 39                 RTS

                * NOW CHANGE REFERENCES
                * FIND GOTO, GOSUB, AND THEN FOLLOWED BY LINE NUMBER
                * THEN MATCH NUMBER TO OLD NUMBER IN TABLE AND REPLACE
                * WITH NEW NUMBER NEXT IN TABLE.

02F7 CE 0A B0  CHREF   LDX     #BUFFER   SET UP TO SCAN BUFFER
02FA DF 20             STX     BUFPTR
02FC A6 00     CHREF1  LDA A   0,X       GET CHARACTER
02FE 08                INX
02FF 81 1A             CMP A   #$1A      EOF
0301 26 03             BNE     CHRE15
0303 7E 03 F7          JMP     WRITE     DONE FIXING REFERENCES
0306 81 54     CHRE15  CMP A   #'T
0308 27 2F             BEQ     THENS     IT'S A T, SEE IF 'THEN'
030A 81 47             CMP A   #'G       NOT 'T' MAYBE 'G'
030C 26 EE             BNE     CHREF1    IF NO MATCH START AGAIN
030E A6 00             LDA A   0,X
0310 08                INX
0311 81 4F             CMP A   #'O       G MATCHED, SEE IF 'O' NEXT
0313 26 E7             BNE     CHREF1
0315 A6 00             LDA A   0,X
0317 08                INX
0318 81 54             CMP A   #'T       ETC.
031A 26 09             BNE     SUBSER    IF NOT 'GOTO' MAYBE 'GOSUB'
031C A6 00             LDA A   0,X
031E 08                INX
031F 81 4F             CMP A   #'O
0321 26 D9             BNE     CHREF1
0323 20 29             BRA     CHREF2    IT'S A 'GOTO'
0325 81 53     SUBSER  CMP A   #'S
0327 26 D3             BNE     CHREF1
0329 A6 00             LDA A   0,X
032B 08                INX
032C 81 55             CMP A   #'U
032E 26 CC             BNE     CHREF1
0330 A6 00             LDA A   0,X
0332 08                INX
0333 81 42             CMP A   #'B
0335 26 C5             BNE     CHREF1
0337 20 15             BRA     CHREF2    IT'S A GOSUB
0339 A6 00     THENS   LDA A   0,X
033B 08                INX
033C 81 48             CMP A   #'H
033E 26 BC             BNE     CHREF1
0340 A6 00             LDA A   0,X
0342 08                INX
0343 81 45             CMP A   #'E
0345 26 B5             BNE     CHREF1
0347 A6 00             LDA A   0,X       INX
0349 08                INX
034A 81 4E             CMP A   #'N
034C 26 AE             BNE     CHREF1
```

```
034E A6 00      CHREF2  LDA A   0,X        WHAT'S NEXT CHAR.
0350 08                 INX
0351 81 20              CMP A   #$20        IF SPACE IGNORE IT
0353 27 F9              BEQ     CHREF2      IGNORE SPACES
0355 09         CHREF3  DEX                 BACK UP
0356 84 F0              AND A   #$F0        SEE IF IT'S ASCII NUMBER
0358 81 30              CMP A   #$30
035A 26 A0              BNE     CHREF1      NO, CONTINUE SEARCH
035C BD 03 C0           JSR     SETPTR
035F BD 02 23           JSR     ASHEX       YES, GET HEX EQUIV. FOR MATCH
0362 DF 20              STX     BUFPTR      SAVE POINTER
0364 CE 04 B0           LDX     #TABLE
0367 DF 22              STX     TABPTR      SET UP TABLE SEARCH
0369 A6 00      FIND    LDA A   0,X         FIND MATCH
036B 81 FF              CMP A   #$FF        END OF TABLE ?
036D 26 23              BNE     FIND1       NO, TRY AGAIN
036F CE 03 A2           LDX     #MSG        TABLE END ERROR
0372 BD 71 18           JSR     PSTRNG      PRINT MESSAGE
0375 DE 20              LDX     BUFPTR      GET REF. LINE NO. NOT FOUND
0377 09                 DEX                 BACK UP POINTER
0378 09                 DEX
0379 09                 DEX
037A 09                 DEX
037B A6 00              LDA A   0,X         OUT. LINE NO. AFTER MESSAGE
037D BD 71 12           JSR     PUTCHR
0380 A6 01              LDA A   1,X
0382 BD 71 12           JSR     PUTCHR
0385 A6 02              LDA A   2,X
0387 BD 71 12           JSR     PUTCHR
038A A6 03              LDA A   3,X
038C BD 71 12           JSR     PUTCHR
038F 7E 01 8D           JMP     EREX1
0392 91 28      FIND1   CMP A   OLDLIN      REF PUT HERE BY ASHEX
0394 26 06              BNE     NEXT        DOESNT MATCH
0396 A6 01              LDA A   1,X
0398 91 29              CMP A   OLDLIN+1
039A 27 31              BEQ     REPLAC      MATCH, REPLACE OLD WITH NEW
039C 08         NEXT    INX
039D 08                 INX
039E 08                 INX
039F 08                 INX
03A0 20 C7              BRA     FIND        LOOK AT NEXT OLD NO.
03A2 52         MSG     FCC     /REFERENCED LINE NOT FOUND, # /
03BF 04                 FCB     4
03C0 08         SETPTR  INX
03C1 E6 00              LDA B   0,X
03C3 BD 01 99           JSR     NUMBER
03C6 24 F8              BCC     SETPTR
03C8 09         SETPT1  DEX     NOW         BACK UP 4 DIGITS
03C9 09                 DEX
03CA 09                 DEX
03CB 09                 DEX
03CC 39                 RTS
03CD BD 02 CD   REPLAC  JSR     HEAS        CONV. NEW TO ASCII
```

```
03D0 DE 20              LDX     BUFPTR
03D2 09                 DEX
03D3 09                 DEX
03D4 09                 DEX
03D5 09                 DEX                     BACK UP TO START OF OLD REF.
03D6 D6 2A              LDA B   ASCII           REPLACE WITH NEW NUMBER
03D8 E7 00              STA B   0,X
03DA 08                 INX
03DB D6 2B              LDA B   ASCII+1
03DD E7 00              STA B   0,X
03DF 08                 INX
03E0 D6 2C              LDA B   ASCII+2
03E2 E7 00              STA B   0,X
03E4 08                 INX
03E5 D6 2D              LDA B   ASCII+3
03E7 E7 00              STA B   0,X
03E9 08                 INX
03EA A6 00              LDA A   0,X
03EC 08                 INX
03ED 81 2C              CMP A   #',             LOOK FOR MORE REFS ON N GOTO
03EF 26 03              BNE     CH1             IF SO, CHANGE NEXT REF. TOO
03F1 7E 03 4E           JMP     CHREF2
03F4 7E 02 FC  CH1      JMP     CHREF1          IF NOT, FIND NEXT REF.

                * NOW DONE CHANGING REFERENCES, SAVE RENUMBERED FILE

03F7 CE 77 40  WRITE    LDX     #FCB
03FA C6 0D              LDA B   #$0D            RENAME CODE
03FC E7 00              STA B   0,X
03FE C6 0A              LDA B   #10             SET UP MOVE TO RENAME BUFFER
0400 CE 77 44           LDX     #FCB+4          START OF NAME IN FCB
0403 DF 28              STX     OLDLIN          USE AS XTEMP FROM
0405 CE 77 75           LDX     #FCB+53
0408 DF 26              STX     INCR            USE AS XTEMP TO
040A DE 28     MOFILE   LDX     OLDLIN
040C A6 00              LDA A   0,X
040E 08                 INX
040F DF 28              STX     OLDLIN
0411 DE 26              LDX     INCR
0413 A7 00              STA A   0,X
0415 08                 INX
0416 DF 26              STX     INCR
0418 5A                 DEC B
0419 26 EF              BNE     MOFILE
041B 86 4B              LDA A   #'K             CHANGE EXT FROM BAS TO BAK
041D A7 00              STA A   0,X
041F CE 77 40           LDX     #FCB
0422 BD 78 06           JSR     FMS             RENAME ORIGINAL FILE
0425 26 4F              BNE     ERROR1
0427 CE 77 40           LDX     #FCB
042A 86 53              LDA A   #'S             SET UP NAME FOR NEW FILE
042C A7 0E              STA A   14,X
042E C6 02              LDA B   #2              OPEN FOR WRITE CODE
0430 E7 00              STA B   0,X
```

```
0432 BD 78 06          JSR    FMS         DO IT
0435 26 3F             BNE    ERROR1
0437 7F 77 40          CLR    FCB
043A CE 0A AF          LDX    #BUFFER-1 POINT AT START OF FILE
043D DF 20             STX    BUFPTR
043F 7F 00 2F          CLR    PCN         FLGS USED TO STRIP LEADING 0'S
0442 7F 00 2E          CLR    NFLG        ---ZEROS BUT LEAVE N=0,
0445 7F 00 30          CLR    PCD         ---RND (0), ETC
0448 DE 20     WRLOP   LDX    BUFPTR
044A 08                INX
044B DF 20             STX    BUFPTR
044D A6 00             LDA A  0,X         GET CHARACTER TO WRITE
044F E6 01             LDA B  1,X         PREVIEW NEXT CHAR.
0451 BD 01 99          JSR    NUMBER      IS IT A NUMBER
0454 25 3F             BCS    WRLOP2      NO, WRITE PRESENT ONE
0456 81 30             CMP A  #$30        PRESENT CHAR. ZERO?
0458 26 31             BNE    CNF         NO, CLR NFLG
045A 7D 00 2F          TST    PCN         PREV CHAR. A NUMBER?
045D 26 31             BNE    TSTNFL      YES
045F 7D 00 30          TST    PCD         WAS PREV. CHAR. A DEC. PT. ?
0462 26 31             BNE    WRLOP2      NO,WRITE IT
0464 C6 FF             LDA B  #$FF        Y SET NFLG SO DON'T STRIP
0466 D7 2E             STA B  NFLG        ---ZERO'S AFTER D. PT.
0468 20 06             BRA    SETPCN      IF WE GOT HERE CHAR. IS NO.
046A 16        TSTNUM  TAB                SET UP TEST FOR NUMBER
046B BD 01 99          JSR    NUMBER
046E 25 09             BCS    CLRPCN
0470 C6 FF     SETPCN  LDA B  #$FF        IF CUR. CHAR. IS NO.
0472 D7 2F             STA B  PCN         --- SET PCN
0474 20 D2             BRA    WRLOP       GET ANOTHER CHARACTER
0476 7E 01 93  ERROR1  JMP    ERROR       JMP FOR TOO LONG BRA
0479 7F 00 2F  CLRPCN  CLR    PCN
047C 81 2E             CMP A  #'.         IS IT DEC PT.
047E 26 06             BNE    CLRDPT      NO, CLEAR FLAG
0480 C6 FF             LDA B  #$FF        Y SET FLG FOR NEXT CHAR.
0482 D7 30             STA B  PCD
0484 20 C2             BRA    WRLOP       GET NEXT CHAR.
0486 7F 00 30  CLRDPT  CLR    PCD         CHAR. NOT A D. PT.
0489 20 BD             BRA    WRLOP       GET NEXT CHAR.
048B 7F 00 2E  CNF     CLR    NFLG
048E 20 05             BRA    WRLOP2      GO WRITE CHAR.
0490 7D 00 2E  TSTNFL  TST    NFLG
0493 26 D5             BNE    TSTNUM      SEE IF NUMBER
0495 CE 77 40  WRLOP2  LDX    #FCB        POINT AT FCB
0498 BD 78 06          JSR    FMS         WRITE CHARACTER
049B 26 D9             BNE    ERROR1
049D 81 1A             CMP A  #$1A        END OF FILE?
049F 26 C9             BNE    TSTNUM      IF NOT CONTINUE WRITING
04A1 CE 77 40          LDX    #FCB
04A4 86 04             LDA A  #4          SET UP CLOSE FILE CODE
04A6 A7 00             STA A  0,X
04A8 BD 78 06          JSR    FMS
04AB 27 00             BEQ    DONWR       NO ERROR
04AD 7E 71 03  DONWR   JMP    WARMS       EXIT PROGRAM
```

```
                    * DEFINE TABLE AND BUFFER START ADDRESSES

 04B0               TABLE   RMB     1

                    * TABLE BIG ENOUGH FOR 384 LINE PROGRAM

 0AB0                       ORG     TABLE+$600

 0AB0               BUFFER  RMB     1
                            END     START

NO ERROR(S) DETECTED
```

SYMBOL TABLE:

ASCII	002A	ASHEX	0223	ASHEX1	0226	BEGIN	0103	BUFFER	0AB0
BUFPTR	0020	CH1	03F4	CHNUM	0294	CHNUM1	029C	CHNUM2	02BE
CHRE15	0306	CHREF	02F7	CHREF1	02FC	CHREF2	034E	CHREF3	0355
CLEAR	0108	CLOSE	0176	CLRDPT	0486	CLRPCN	0479	CNF	048B
CONT1	0171	CONT2	016A	CONT3	0151	CONT4	014B	CONTIN	0142
CONVT	0243	DONFIL	011A	DONWR	04AD	ENDFIL	021B	EREX1	018D
EREXIT	018A	ERROR	0193	ERROR1	0476	FCB	7740	FIND	0369
FIND1	0392	FMS	7806	FMSCLS	7803	FSTLIN	0024	GET1	01B7
GET2	01C1	GETCHR	710F	GETFIL	7127	GETHEX	713F	GETINF	01A5
HEAS	02CD	HEAS1	02D0	HEAS2	02EB	INCR	0026	LINNO	01C3
LINNO1	01C5	LINNO2	020A	LOW	023E	MOFILE	040A	MSG	03A2
MSG1	0279	MSG2	01E8	NEWNU1	024B	NEWNUM	0248	NEXT	039C
NFLG	002E	NOT	01A3	NUMBER	0199	OFLO	0273	OLDLIN	0028
OUTHEX	7139	PCD	0030	PCN	002F	PSTRNG	7118	PUTCHR	7112
READ	0132	REPLAC	03CD	RPTERR	713C	SEARCH	020C	SETEXT	712D
SETPCN	0470	SETPT1	03C8	SETPTR	03C0	START	0100	SUBSER	0325
TABLE	04B0	TABPTR	0022	THENS	0339	TSTNFL	0490	TSTNUM	046A
VER	0102	WARMS	7103	WRITE	03F7	WRLOP	0448	WRLOP2	0495

TECHNICAL SYSTEMS
CONSULTANTS, INC.
P.O. BOX 2574 WEST LAFAYETTE, INDIANA 47906

FEB 20 '79

FIRST CLASS MAIL

TO:

Michael Holley
Lynes Inc.
4520 148th Avenue, N.E.
Redmond, WA 98052