Waterloo Structured BASIC For The PET

Feed Your PET Some APPLESOFT

COMPUTE!

$2.00
September/October, 1980
Vol. 1, Issue 6

The Journal for Progressive Computing™

The Resource Magazine For Apple, Atari, and Commodore

Teaching Basic Academic Skills- Can Micros Make A Difference?

Mixing Atari Graphics Modes

Thesus Versus The Minotaur- PASCAL Visits Ancient Greece

RS-232 Communications

TelePET

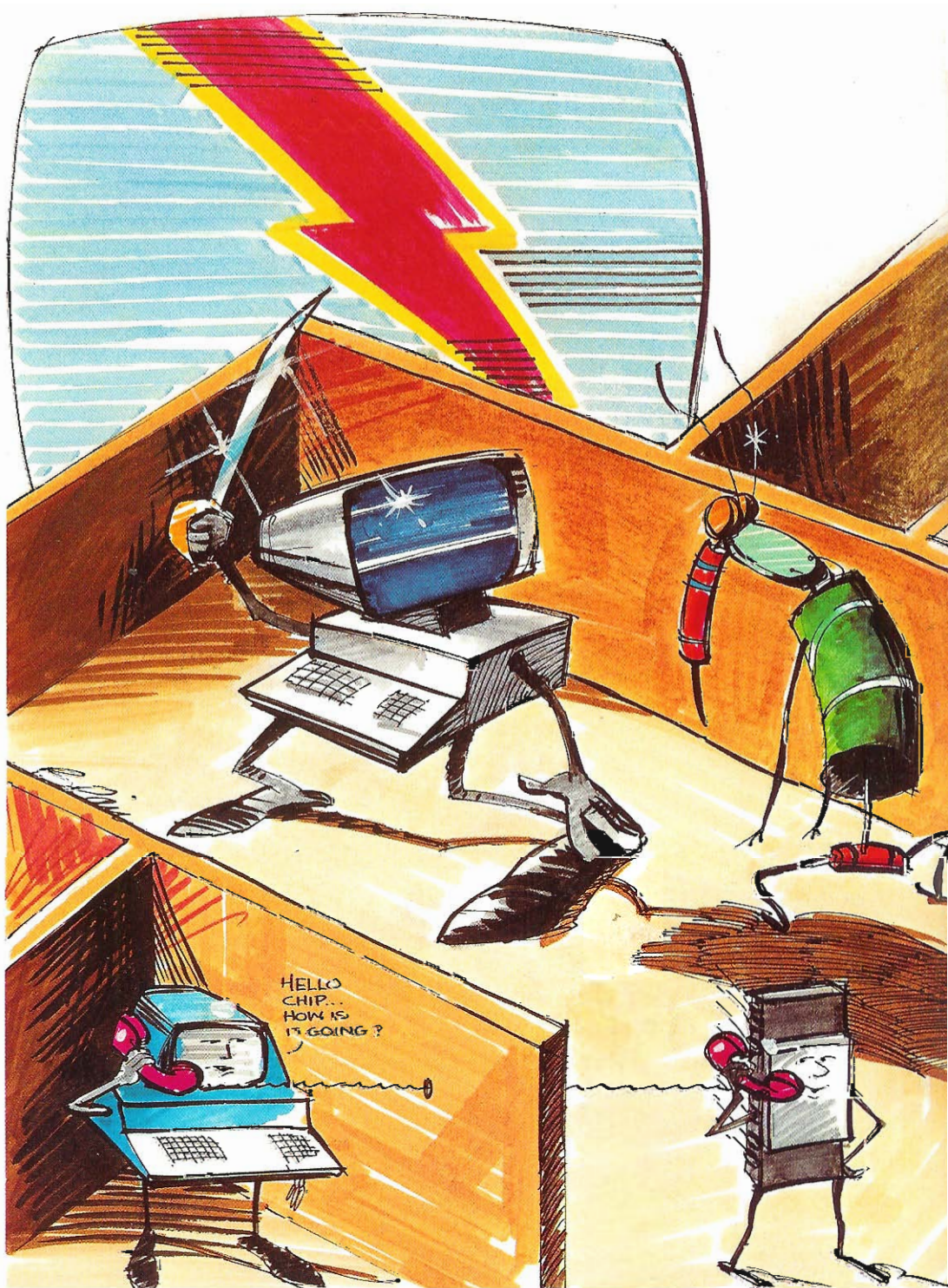
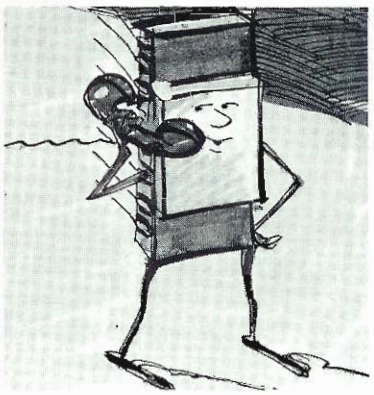
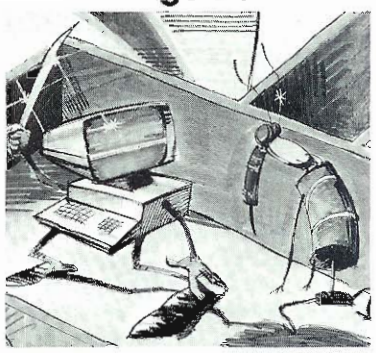


Table of Contents

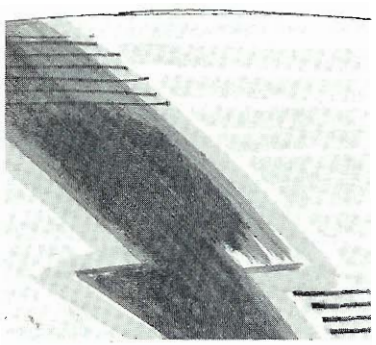
September/October 1980. Volume 1. Issue 6



Page 26



Page 64



Page 71

Page 86

COMPUTE. **The Journal for Progressive Computing** is published six times each year by Small System Services, Inc., P.O. Box 5406, Greensboro, NC 27403 USA. Phone: (919) 275-9809. Editorial Offices are located at 200 East Bessemer Ave., Greensboro, NC 27401.

Domestic Subscriptions: 12 issues, $16.00. Send subscription orders or change of address (P.O. Form 3579) to Circulation Dept., COMPUTE. Magazine, P.O. Box 5406, Greensboro, NC 27403. Controlled circulation postage paid at Greensboro, NC 27403. ISSN 0194-357X.

Word Pro Converter

Robert W. Baker, BAKER ENTERPRISES,
15 Windsor Drive, Atoo, NJ 08004

An ever increasing number of programs make use of Commodore's Word Pro program with its excellent editing facilities to generate files for their own use. However, disk files created by Word Pro 3 are not fully compatible with those created or used by Word pro 4 on the 8016/8032.

If you create any files on a 2001 series PET/CBM using Word Pro 3, you will have to do some editing to be able to use the same file on an 8016/8032 CBM with its 80 column screen. This simple utility program will eliminate the boring task of editing the file, and do all the necessary changes for you automatically. It will run on either a 2001 PET/CBM or an 8016/8032 CBM; using a 2040 disk. Remember, though, that the 2040 disk must have the DOS 2.0 ROMs if you are using an 8016/8032 CBM.

The Word Pro Converter program is very straight forward in operation and no fancy frills or options are included. The file to be converted must be on the diskette in Drive #0. The new file created will be written on the diskette on Drive #1 with the same name. If the file all ready exists on Drive #1, it will be deleted first. The only input to the program is the name of the file to be converted. It should be very simple to add an output file name option along with drive number selections if desired. During program execution, any disk error will be displayed and terminate the program.

In theory, the program simply copies the file byte-by-byte while counting characters and looking for a RETURN within each original 40 character line. Straight text that continues over several 40 character lines is copied as-is, creating new 80 character lines. If a RETURN is detected in any line, an extra 40 spaces are added at the end of the line whenever required to make the line 80 characters long.

Files stored by Word Pro 3 contain 40 characters per display line regardless of content. Thus, if you have a single FP command on a line, there is a 37 byte overhead with Word Pro 3. Word pro 4, on the other hand, stores 80 characters per display line regardless of content. The same FP command in Word Pro 4 will then have a 77 byte overhead! While Word Pro 4 has its advantages with the 80 column screen, the disk files created will be generally bigger than those created by Word Pro 3 for the same text. This is especially true when there are a large number of formatting commands or blank lines.

Program Variables

E	input file status, 64 = end-of-file
N	#characters in input file line, 40 max
P	#characters in output file line, 80 max.
R	RETURN character flag: 0 = no 1 = yes
B$	character (byte) being copied

```
 100 REM ******************************
 110 REM
 120 REM    SIMPLE UTILITY PROGRAM
 130 REM     TO CONVERT DISK FILES
 140 REM    CREATED BY WORD PRO III
 150 REM  FOR LOADING BY WORD PRO IV
 160 REM
 170 REM ----------------------------
 180 REM
 190 REM       BY: ROBERT W. BAKER
 200 REM
 210 REM        BAKER ENTERPRISES
 220 REM 15 WINDSOR DR., ATCO, NJ 08004
 230 REM
 240 REM ******************************
 250 :
 260 :
 270 PRINT"ĥ  W O R D   P R O   C O N V E ¬
     ¬R T E R
 280 PRINT"ϕϕTHE FILE TO BE CONVERTED MUST ¬
     ¬BE ON
 290 PRINT"ON THE DISKETTE IN DRIVE #0ϕ
 300 PRINT"THE NEW FILE GENERATED WILL BE ¬
     ¬WRITTEN
 310 PRINT"ON THE DISKETTE IN DRIVE #1,
 320 PRINT"WITH THE SAME FILE NAME.ϕϕ
 330 INPUT"FILE NAME   .<<<";FI$
 340 IF FI$="." THEN 330
 350 PRINT"ϕrCONVERTING FILE, PLEASE ¬
     ¬WAIT...
 360 OPEN 15,8,15
 370 OPEN 5,8,8,"0:"+FI$+",P,R"
 380 GOSUB 560
 390 PRINT#15,"S1:"+FI$
 400 OPEN 6,8,9,"1:"+FI$+",P,W"
 410 GOSUB 560
 420 GET#5,A$,B$:GOSUB 560
 430 PRINT#6,A$;B$;:GOSUB 560
 440 P=0
 450 N=0:R=0
 460 GET#5,B$:E=ST:GOSUB 560
 470 PRINT#6,B$;:GOSUB 560
 480 P=P+1:IF P=80 THEN P=0
 490 IF E=64 THEN PRINT"ϕrDONE !!!ϕ":
     ¬GOTO 610
 500 N=N+1
 510 IF ASC(B$)=31 THEN R=1
 520 IF N<40 THEN 460
 530 IF R=0 OR P=0 THEN 450
 540 FOR N=1 TO 40: PRINT#6," ";GOSUB 560:
     ¬NEXT
 550 GOTO 440
 560 INPUT#15,EN,EM$,ET$,ES$
 570 IF EN=0 THEN RETURN
 580 PRINT"ϕrDISK ERROR_!!!ϕ
 590 PRINT EN;EM$,ET$,ES$
 600 PRINT"ϕOPERATION ABORTED!
 610 CLOSE 5: CLOSE 6: CLOSE 15
READY.
```

Compactor

Robert W. Baker,
Baker Enterprises,
15 Windsor Drive,
Atco, NJ 08004

This program is the result of several days of experimenting with BASIC program structures and the 2040 disk. In short, the program will read a BASIC program that was saved on disk and create a new, compacted copy. The program will delete all REMarks, unnecessary spaces, and leading colons. Much of this is similar to other utility programs currently available. However, this program goes one step further. It combines program lines whenever possible to eliminate the link, line number, and line-end-flag overheads normally associated with each line. It will make a program as small as possible, and most likely faster running.

While creating this program, I came across a few undocumented "quirks" of Commodore BASIC. Since many people are currently experimenting with the capabilities of having programs "write" programs on disk, this information may be of interest:

Zero Length Lines:

Normally, it is impossible to create a zero length line using the screen editor on the PET. By zero length line, I mean a line with a link, line number, and end-of-line flag; but no BASIC commands or text. If you were to type just a line number using the screen editor, you would actually delete a line instead of entering a zero length line. However, when writing a BASIC program on disk as a data file there is nothing stopping you from entering a zero length line. But if you want the program to run, you cannot have any zero length lines in the program. BASIC cannot link the program lines correctly whenever there is a zero length line in the program.

Long Lines:

At the other extreme, you cannot create a BASIC line that is longer than 255 bytes. Again, using the PET screen editor you could not create such a line. You are normally limited to a maximum of 78 bytes due to the line wrapping characteristics and at least a one digit line number. When writing a BASIC program on disk as a data file, be careful not to create a line greater than 255 bytes. Otherwise the program will usually not load from the disk. If it does load, the program will be totally destroyed and unuscable.

Printing Long Lines:

Here's a quick comment on the Commodore printers. If you list a program that contains lines longer than 80 characters, the printed listing may be incorrect. It appears that the printer occasionally switches out of listing mode and into print mode when a line exceeds 80 characters. At the start of the next line everything is ok again.

Program Description

When running the COMPACTOR program, the BASIC program to be compacted must be on the diskette inserted in drive #0. The new compacted version will be written on the diskette in drive #1 with the same filename, but with a "/C" suffix. The program will read the program to be compacted as a sequential disk data file, and the file will be read twice.

The first pass checks for line numbers within the subject program that are the targets of: GOTO, GOSUB, or IF...THEN..(line#) statements. When a target line number is found, it is saved in matrix TL if not all ready saved. A check is also made for multiple target lines in ON..GOTO and ON.. GOSUB statements. Each target line will be displayed on the PET screen in the order found. This helps give some indication of the scanning progress since it can be rather slow.

During the second pass, each line is copied, deleted, or compacted as appropriate. The line number will be displayed as each line is processed to let you know how the program is progressing. The rules followed by the COMPACTOR are as follows:

Any leading colons and/or spaces on a line are deleted.

A line that has only REMarks is deleted if it is not a target line. The remark will be replaced with a single colon if the line is a target line and must be retained. This may produce a leading colon if the next line is not a target line and is combined with this line. The line cannot be reduced to a zero length line since BASIC cannot link a program correctly with a zero length line, as mentioned earlier.

If any line contains an IF...THEN or GOTO statement, another line cannot be combined with this line. Any line combined after these BASIC commands would never be executed, thus the program would not function properly.

Any spaces within a line, not enclosed in quotes, are deleted.

Any REMarks at the end of a BASIC line are deleted to the end of the line.

Anything within quotes is copied, untouched. If an ending quote is missing from the line, one is

added if another line could be combined with this line. Therefore, if a line does not contain an IF.. THEN or GOTO statement, an ending quote is added.

When a colon is found within a BASIC line and not within quotes, the next non-space character is checked before copying the colon. If a REMark follows the colon, the colon and the rest of the line is deleted. Otherwise, the colon is copied and processing continues as normal.

At the end of each BASIC line, a check is made to see if the next line can be combined with this line. If there were no IF...THEN or GOTO commands, and the next line is not a target line, the lines are combined. When combining lines, the line and line number are discarded, a colon is written, and the next line is processed as part of the previous line.

If the next line cannot be combined with the current line, the end of line flag is copied along with a dummy link and the next line number. A dummy link is used to avoid excessive processing and working buffers necessary with calculating program links. Besides, the links are automatically corrected by PET BASIC with the RUN or CLR commands. As a standard operating procedure, the newly created program outputted by COMPACTOR should be loaded and re-linked, then re-saved onto disk. The program can be re-linked by issuing a CLR command after being loaded.

As mentioned previously, a BASIC program line cannot exceed 255 bytes in length. If it does, the program may not load from disk or it may be totally unuseable. To protect against this, the COMPACTOR program stops combining lines if more than 170 bytes have been written in a single BASIC line. Since any normal line cannot exceed 78 bytes in length, this should insure than no program generated lines are longer than the maximum length.

As an example of what this program will do, I included a listing of a compacted version of the COMPACTOR program itself. Since this program has many REMarks, compacting saves over 3000 bytes for about a 50% saving in memory space. On most programs the savings will be much smaller, depending on the programming style. A side benefit, however, is the increase in the operating speed of compacted programs. I should warn, though, that the compacting process can be rather slow. Compacting of the COMPACTOR program (a 6K program with all the REMarks) takes about 16 minutes. But all you have to do is start it off and then go get a cup of coffee while the PET does the work! And you only have to run it once for any given program!

For those too lazy to type in the program, I'll be happy to provide copies on tape at $2 each.

```
10 REM ***************************
20 REM *    C O M P A C T O R    *
30 REM *  --------------------   *
40 REM *    BY:  ROBERT BAKER    *
50 REM *                         *
60 REM *    BAKER ENTERPRISES    *
70 REM *     15 WINDSOR DR.      *
80 REM *    ATCO, N.J.  08004    *
90 REM ***************************
100 :
110 CLR : DIM TL(1000)
120 :
130 REM ****************
140 REM READY DISK FILES
150 REM ****************
160 :
170 PRINT"î"SPC(15)"rCOMPACTOR↓↓
180 PRINT" rINPUTî FILE IN rDRIVE #0↓
190 PRINT"rOUTPUTî FILE IN rDRIVE #1↓↓
200 INPUT"rINPUT FILE NAMEî";FL$
210 PRINT"îSCANNING FILE
220 PRINT"   FOR TARGET LINES.....↓↓
230 OPEN 15,8,15 : GOSUB 2370
240 OPEN 5,8,5,"0:"+FL$+",P,R"
250 :
260 REM ***************************
270 REM READ LOAD ADR, LINK & LINE#
280 REM ***************************
290 :
300 GOSUB 2370 : GOSUB 2310
310 GOSUB 2310 : IF V+V1=0 THEN 790
320 GOSUB 2310 : LN=V1+(256*V)
330 :
340 REM ****************************
350 REM         SCAN BASIC LINES
360 REM FOR GOTO, GOSUB & THEN TOKENS
370 REM ****************************
380 :
390 GOSUB 2330
400 IF V=0 THEN 310
410 IF V=137 OR V=141 THEN 480
420 IF V<>167 THEN 390
430 :
440 REM ****************
450 REM GET TARGET LINE#
460 REM ****************
470 :
480 LT=0
490 GOSUB 2330 : IF V=32 THEN 490
500 IF V<48 OR V>57 THEN 580
510 LT=(10*LT)+VAL(C$)
520 GOSUB 2330 : GOTO 500
530 :
540 REM ************************
550 REM CHECK IF ALL READY FOUND
560 REM ************************
570 :
580 FOR X=0 TO N
590 IF TL(X)=LT THEN 710
600 NEXT X
610 TL(N)=LT : N=N+1
620 PRINT LT,
630 IF N<1000 THEN 710
640 PRINT"↓↓TOO MANY TARGET LINES!
650 GOTO 2430
660 :
670 REM ***************************
680 REM CHECK FOR 'ON...GOTO/GOSUB'
690 REM ***************************
700 :
710 IF V=44 THEN 480
720 IF V<>32 THEN 400
730 GOSUB 2330 : GOTO 710
740 :
```

```
750 REM *****************
760 REM SORT TARGET LINES
770 REM *****************
780 :
790 IF N<2 THEN 900
800 FOR X=0 TO N-1
810 FOR Y=0 TO N-2
820 IF TL(Y) < TL(X) THEN 840
830 V=TL(Y) : TL(Y)=TL(X) : TL(X)=V
840 NEXT Y,X
850 :
860 REM *********************
870 REM GET READY FOR COMPACT
880 REM *********************
890 :
900 PRINT "ñCOMPACTING LINES....↓↓
910 CLOSE 5
920 OPEN 5,8,5,"0:"+FL$+",P,R"
930 GOSUB 2370
940 FO$=LEFT$(FL$,14)+"/C"
950 PRINT#15,"S1:"+FO$
960 OPEN 6,8,6,"1:"+FO$+",P,W"
970 GOSUB 2370
980 :
990 REM  **************
1000 REM COPY LOAD ADR
1010 REM **************
1020 :
1030 GOSUB 2310
1040 PRINT#6,CHR$(V1);
1050 PRINT#6,CHR$(V); : R=0
1060 :
1070 REM ************************
1080 REM COPY LINK & LINE NUMBER
1090 REM ************************
1100 :
1110 GOSUB 2310 : K1=V1 : K2=V
1120 F=0 : IF V+V1=0 THEN 2230
1130 GOSUB 2310 : L1=V1 : L2=V
1140 LN=L1+(256*L2) : PRINT LN,
1150 GOSUB 2330
1160 IF V=32 OR V=58 THEN 1150
1170 IF V=0 THEN 1200
1180 IF V<> 143 THEN 1240
1190 GOSUB 2330 : IF V>0 THEN 1190
1200 F=1 : FOR X=0 TO N
1210 IF TL(X)<LN THEN NEXT X
1220 IF TL(X)=LN THEN 1240
1230 GOTO 1110
1240 PRINT#6,CHR$(K1);CHR$(K2);
1250 PRINT#6,CHR$(L1);CHR$(L2); : R=4
1260 IF F THEN PRINT#6,":"; : R=5
1270 F=0 : GOTO 1360
1280 :
1290 REM ***************************
1300 REM ****  SCAN BASIC LINE  ***
1310 REM **** & COMPACT PROGRAM ***
1320 REM ***************************
1330 :
1340 PRINT#6,C$; : R=R+1
1350 GOSUB 2330
1360 IF V=137 THEN F=1
1370 IF V=139 OR V=167 THEN F=1
1380 IF V=0 THEN 1820
1390 IF V=32 THEN 1350
1400 :
1410 REM ********************
1420 REM 'REM' TOKEN -
1430 REM DISCARD REST OF LINE
1440 REM ********************
1450 :
1460 IF V<>143 THEN 1550
1470 GOSUB 2330 : IF V>0 THEN 1470
1480 GOTO 1820
1490 :
1500 REM ************************
1510 REM QUOTE -
1520 REM COPY TILL NEXT OR LINE END
1530 REM ************************
1540 :
1550 IF V<>34 THEN 1690
1560 PRINT#6,C$; : R=R+1
1570 GOSUB 2330
1580 IF V=34 THEN 1340
1590 IF V>0 THEN 1560
1600 IF F THEN V=0 : GOTO 1050
1610 PRINT#6,CHR$(34); : R=R+1
1620 GOTO 1820
1630 :
1640 REM ***********************
1650 REM IF COLON - CHK NEXT CHAR
1660 REM       ELSE COPY CHAR
1670 REM ***********************
1680 :
1690 IF V<>58 THEN 1340
1700 GOSUB 2330
1710 IF V=32 OR V=58 THEN 1700
1720 IF V=143 THEN 1470
1730 IF V=0 THEN 1820
1740 PRINT#6,":"; : R=R+1
1750 GOTO 1360
1760 :
1770 REM ****************************
1780 REM END OF LINE -
1790 REM CAN WE COMPACT THESE LINES ?
1800 REM ****************************
1810 :
1820 IF F OR (R>170) THEN V=0:GOTO 1050
1830 GOSUB 2310
1840 IF V+V1=0 THEN 2230
1850 GOSUB 2310 : LN=V1+(256*V)
1860 L1=V1 : L2=V : PRINT LN,
1870 :
1880 REM ***********************
1890 REM CHK IF LINE# IS A TARGET
1900 REM ***********************
1910 :
1920 FOR X=0 TO N
1930 IF TL(X)<LN THEN NEXT X
1940 IF TL(X)=LN THEN 2110
1950 :
1960 REM ********************
1970 REM NOT USED -
1980 REM DISCARD LINK & LINE#
1990 REM ********************
2000 :
2010 GOSUB 2330 : IF V=143 THEN 1470
2020 IF V=32 OR V=58 THEN 2010
2030 IF V=0 THEN 1830
2040 PRINT#6,":"; : R=R+1 : GOTO 1360
2050 :
2060 REM ****************************
2070 REM LINE# NEEDED -
2080 REM WRITE LINE END, LINK & LINE#
2090 REM ****************************
2100 :
2110 PRINT#6,CHR$(0);CHR$(1);CHR$(1);
2120 PRINT#6,CHR$(L1);CHR$(L2); : R=4
2130 GOSUB 2330
2140 IF V=32 OR V=58 THEN 2130
2150 IF V=0 OR V=143 THEN PRINT#6,":";
2160 F=0 : GOTO 1360
2170 :
2180 REM ********************
2190 REM END OF COMPACT -
2200 REM WRITE END OF PROGRAM
2210 REM ********************
2220 :
```

```
 2230 PRINT#6,CHR$(0);CHR$(0);CHR$(0);
 2240 PRINT"ĥ↓DONE↓↓"
 2250 GOTO 2430
 2260 :
 2270 REM ************************
 2280 REM *****  SUBROUTINES  *****
 2290 REM ************************
 2300 :
 2310 GOSUB 2330 : V1=V
 2320 :
 2330 GET#5,C$ : GOSUB 2370
 2340 IF C$="" THEN V=0 : RETURN
 2350 V=ASC(C$) : RETURN
 2360 :
 2370 INPUT#15,EN,EM$,ET,ES
 2380 IF EN=0 THEN RETURN
 2390 :
 2400 PRINT : PRINT"↓↓↓↓DISK ERROR↓
 2410 PRINT EN;EM$;ET;ES
 2420 :
 2430 CLOSE 5 : CLOSE 6 : CLOSE 15
READY.
```

```
 110 CLR:DIMTL(1000):PRINT"ĥ"SPC(15)"↓COMPA
       ¬CTOR↓↓":PRINT" ↓INPUTî FILE IN ¬
       ¬↓DRIVE #0↓":PRINT"↓OUTPUTî FILE IN ¬
       ¬↓DRIVE #1↓↓":INPUT"
↓INPUT FILE NAMEî";FL$:PRINT"ĥSCANNING ¬
       ¬FILE":PRINT"   FOR TARGET LINES.....
       ¬↓↓"
 230 OPEN15,8,15:GOSUB2370:OPEN5,8,5,"0:
       ¬"+FL$+",P,R":GOSUB2370:GOSUB2310
 310 GOSUB2310:IFV+V1=0THEN790
 320 GOSUB2310:LN=V1+(256*V)
 390 GOSUB2330
 400 IFV=0THEN310
 410 IFV=137ORV=141THEN480
 420 IFV<>167THEN390
 480 LT=0
 490 GOSUB2330:IFV=32THEN490
 500 IFV<48ORV>57THEN580
 510 LT=(10*LT)+VAL(C$):GOSUB2330:GOTO500
 580 FORX=0TON:IFTL(X)=LTTHEN710
 600 NEXTX:TL(N)=LT:N=N+1:PRINTLT,:
       ¬IFN<1000THEN710
 640 PRINT"↓↓TOO MANY TARGET LINES!":
       ¬GOTO2430
 710 IFV=44THEN480
 720 IFV<>32THEN400
 730 GOSUB2330:GOTO710
 790 IFN<2THEN900
 800 FORX=0TON-1:FORY=0TON-2:IFTL(Y)<TL(X)T
       ¬HEN840
 830 V=TL(Y):TL(Y)=TL(X):TL(X)=V
 840 NEXTY,X
 900 PRINT"ĥCOMPACTING LINES....↓↓":CLOSE5:
       ¬OPEN5,8,5,"0:"+FL$+",P,R":GOSUB2370:
       ¬FO$=LEFT$(FL$,14)+"/C":PRINT#15,"S1:
       ¬"+FO$:OPEN6,8
,6,"1:"+FO$+",P,W":GOSUB2370:GOSUB2310:
       ¬PRINT#6,CHR$(V1);
1050 PRINT#6,CHR$(V);:R=0
1110 GOSUB2310:K1=V1:K2=V:F=0:IFV+V1=0THEN
       ¬2230
1130 GOSUB2310:L1=V1:L2=V:LN=L1+(256*L2):
       ¬PRINTLN,
1150 GOSUB2330:IFV=32ORV=58THEN1150
1170 IFV=0THEN1200
1180 IFV<>143THEN1240
1190 GOSUB2330:IFV>0THEN1190
1200 F=1:FORX=0TON:IFTL(X)<LNTHENNEXTX
1220 IFTL(X)=LNTHEN1240
1230 GOTO1110
1240 PRINT#6,CHR$(K1);CHR$(K2);:PRINT#6,
       ¬CHR$(L1);CHR$(L2);:R=4:IFFTHENPRINT#
       ¬6,":";:R=5
1270 F=0:GOTO1360
1340 PRINT#6,C$;:R=R+1
1350 GOSUB2330
1360 IFV=137THENF=1
1370 IFV=139ORV=167THENF=1
1380 IFV=0THEN1820
1390 IFV=32THEN1350
1460 IFV<>143THEN1550
1470 GOSUB2330:IFV>0THEN1470
1480 GOTO1820
1550 IFV<>34THEN1690
1560 PRINT#6,C$;:R=R+1:GOSUB2330:IFV=34THE
       ¬N1340
1590 IFV>0THEN1560
1600 IFFTHENV=0:GOTO1050
1610 PRINT#6,CHR$(34);:R=R+1:GOTO1820
1690 IFV<>58THEN1340
1700 GOSUB2330:IFV=32ORV=58THEN1700
1720 IFV=143THEN1470
1730 IFV=0THEN1820
1740 PRINT#6,":";:R=R+1:GOTO1360
1820 IFFOR(R>170)THENV=0:GOTO1050
1830 GOSUB2310:IFV+V1=0THEN2230
1850 GOSUB2310:LN=V1+(256*V):L1=V1:L2=V:
       ¬PRINTLN,:FORX=0TON:IFTL(X)<LNTHENNEX
       ¬TX
1940 IFTL(X)=LNTHEN2110
2010 GOSUB2330:IFV=143THEN1470
2020 IFV=32ORV=58THEN2010
2030 IFV=0THEN1830
2040 PRINT#6,":";:R=R+1:GOTO1360
2110 PRINT#6,CHR$(0);CHR$(1);CHR$(1);:
       ¬PRINT#6,CHR$(L1);CHR$(L2);:R=4
2130 GOSUB2330:IFV=32ORV=58THEN2130
2150 IFV=0ORV=143THENPRINT#6,":";
2160 F=0:GOTO1360
2230 PRINT#6,CHR$(0);CHR$(0);CHR$(0);:
       ¬PRINT"ĥ↓DONE↓↓":GOTO2430
2310 GOSUB2330:V1=V
2330 GET#5,C$:GOSUB2370:IFC$=""THENV=0:
       ¬RETURN
2350 V=ASC(C$):RETURN
2370 INPUT#15,EN,EM$,ET,ES:IFEN=0THENRETUR
       ¬N
2400 PRINT:PRINT"↓↓↓↓DISK ERROR↓":
       ¬PRINTEN;EM$;ET;ES
2430 CLOSE5:CLOSE6:CLOSE15
READY.
```