

::: STRINGY-FLOPPY DATA I/O DEMONSTRATION ROUTINES :::

----- PRIMER PARTS 1 AND 2 -----

0 ' ***** THIS LISTING IS AN EXPANDED UNPACKED VERSION
OF THE ACTUAL LISTING IN YOUR COMPUTER. IT
WAS DONE TO MAKE IT EASIER FOR YOU TO FOLLOW
AND STUDY THE PROGRAM ACTION.

1 ' ***** USE THIS LISTING AS A STUDY REFERENCE ON HOW
THESE DATA I/O DEMOS. EXECUTE. WHEN YOU
UNDERSTAND CLEARLY HOW THESE TWO DEMOS. WORK,
YOU SHOULD HAVE A GOOD FEELING FOR THE POWER

2 ' ***** OF THE STRINGY DATA HANDLING COMMANDS. THROUGH
YOUR OWN INDEPENDENT STUDY OF THE SUBJECT OF
DATA BASED MANAGEMENT AND THE STRINGY DATA I/O
COMMANDS, YOU SHOULD BE WELL PREPARED TO WRITE

3 ' ***** AND MODIFY DATA BASED PROGRAMS IN THE STRINGY-
FLOPPY FORMAT.

10 CLS
20 CLEAR 75
30 PRINT CHR\$(23)
40 PRINT @ 400, "DATA I/O DEMO."
50 FOR T=1 TO 700
60 NEXT
70 CLS
80 GOTO 450 GO GET INITIALIZED THEN COME
 BACK TO LINE 110

90 THESE DEMO. PROGRAMS ARE DONATED BY THE AUTHORS TO ESFOA.
WE HOPE THEY WILL HELP YOU IN LEARNING THE NEW STRINGY
DATA COMMANDS.

100 *****
* DATA PROCESSING PRIMER DEMO. *
* THIS PROGRAM ILLUSTRATES THE BASIC FILE COMMANDS *

110 CLS
120 PRINT CHR\$(23)
130 PRINT @ 400, "PRIMER PART 1
 BY
 M. N. KIDDER"
140 FOR T=1 TO 1000
150 NEXT
160 CLS
170 PRINT "PRESS A KEY TO WRITE AND THEN READ A FILE"
180 PRINT @ 128, ; ' INFORM USER
190 IF PEEK(14591) THEN 200 ELSE 190
200 @ OPEN 1 ' OPEN FILE FOR EITHER READ OR WRITE
210 ' TO WRITE A FILE, WE NEED DATA TO WRITE.

```

                WE WILL CREATE A LOOP AND WRITE AND
                PRINT 64 CONSECUTIVE INTEGERS.
220 FOR I%=1 TO 64      ' 64 STEP LOOP
230 @ PRINT I%          ' WRITE ONE INTEGER
240 PRINT USING"####"; I%; ' PRINT THE SAME INTEGER
250 NEXT I%             ' REPEAT THE LOOP
260                     ' NOW THE FILE MUST BE CLOSED TO WRITE
                        OUT THE LAST BUFFER AND LABEL FOR
                        FILE #2.

270 @ CLOSE            ' FILE NUMBER IS NOT NEEDED
280 PRINT
290 PRINT"READ FILE"
300 @ OPEN 1           ' THE DATA I/O PROGRAM WILL FIND THE
                        BEGINNING OF FILE #1.
310                     ' WE WILL USE A SIMILAR LOOP
                        TO READ THE FILE.
320 FOR I%=1 TO 64      ' 64 STEP LOOP
330 @ INPUT J%          ' READ ONE INTEGER. NOTE J% NOT I%
340 PRINT USING"####"; J%; ' PRINT THE INTEGER READ
350 NEXT I%             ' REPEAT THE LOOP
360 @ CLOSE            ' AGAIN CLOSE THE FILE SO THE DATA I/O
                        PROGRAM CAN COMPLETE IT'S PROCESS.

370 PRINT
380 PRINT"  PRESS ENTER TO RUN AGAIN OR PRESS SPACE BAR FOR PRIMER PART 2.";
390 I$=INKEY$
400 I$=INKEY$
410 IF I$="" THEN 400
420 IF I$=" " THEN 790
430 IF ASC(I$)=13 THEN CLS: PRINT @ 128,;: GOTO 200 ELSE 400

440 SOME LINES FROM 450 THROUGH 770 ARE FROM THE COMPLETE
    PROCESSING PROGRAM BY M. N. KIDDER. THAT PROGRAM IS FILE
    3 ON THIS WAFER. BE SURE TO STUDY THE LISTING. IT WILL
    COVER ARRAYS.

450 PRINT"DATA I/O DEMONSTRATION PROGRAM
    (LIST PROGRAM FOR COMMENTS)"
460 ON ERROR GOTO 660
470 POKE 16396, 23
480 @ CLEAR
490 PRINT"CREATING A FILE" ' WE WILL GENERATE A TEST FILE
                        THIS IS WHERE YOU WOULD CREATE YOUR FILE
500 GOSUB 600
510 IF I=0 PRINT"REMOVE WAFER"
520 GOSUB 600
540 IF I=0 THEN 520
550 PRINT"INSERT A 5 FT. WAFER WITH REFLECTIVE STICKER"
560 GOSUB 600
580 IF I=1 THEN 560
590 GOTO 110          ' GO RUN PRIMER DATA DEMO.
600 I=INP(240)AND 1
610 IF INKEY$ =CHR$(2)THEN I=2
620 RETURN

630 POKE 16396, 201
640 ON ERROR GOTO 0
650 END
660 PRINT

```

```

670 IF ERR=2 AND ERL=480 PRINT"DATA I/O PGM NOT LOADED": RESUME 650
680 IF ERR=8 AND(ERL=1650 OR ERL=1320)THEN PRINT"FC ERROR (ILLEGAL FUNCTION CALL
). CANNOT CALL FOR A FILE 0": A=1: X=Z: RESUME 1840
690 IF ERR=6 AND(ERL=1690 OR ERL=1750 OR ERL=1800)THEN PRINT"OD ERROR (OUT OF DA
TA). YOU CALLED FOR A NON EXISTENT DATA FILE": RESUME 1840
700 IF ERR<>42 THEN 650
710 IF ERR=42 AND ERL=1650 THEN CLS: RESUME 2730
720 IF ERR=42 AND(X>4 OR ERL=1320 OR ERL=2320 OR ERL=2730 OR ERL=1690 OR ERL=175
0 OR ERL=1800)THENCLS:PRINT@403,"FD ERROR (BAD FILE DATA)":FORT=1TO800:NEXT:CLS:
IF ERL=2320 OR ERL=2730 OR ERL=1690 OR ERL=1750 OR ERL=1800 THEN RESUME 3230 ELS
E RESUME 2610
730 IF ERL<>530 THEN 750
750 PRINT"DATA I/O ERROR - ";
760 @ CLEAR
770 IF ERL<560 OR ERL>590 THEN PRINT: PRINT"TRY AN @NEW ON THIS WAFER.": GOTO 65
0

```

```

780 *****
* PRIMER DEMO. PART2 *
*****

```

```

790 CLS
800 PRINT CHR$(23)
810 PRINT @ 400,"PRIMER PART 2
BY
BILL BURNHAM"
820 FOR T=1 TO 1000
830 NEXT

```

```

840 CLS
850 PRINT"WHEREAS PART 1 OF THIS PRIMER AND THE ADVANCED DATA I/O DEMO. (COMPL
ETE FILE PROCESSING) WERE MEANT TO BE RUN IN CONJUNCTION WITH A THOROUGH STU
DY OF THE LISTING, PRIMER PART 2 IS DESIGNED TO LET YOU INTERACT";
860 PRINT" WITH ME IN MANIPULATING SOME DATA. A STUDY OF THE LISTING FOR PART 2
IS NOT NECESSARY. IF YOU DO WANT TO EXPERIMENT WITH CHANGES, THEN SEE THE COM
MENTS AT LINE 1390."
870 PRINT
880 PRINT"THE FUN IN PART 2 WILL BEGIN WHEN YOU GET";
890 PRINT" TO THE POINT WHERE YOUARE APPROACHING THE FILE STORAGE LIMITS OF THIS
5 FT. WAFER. INA REAL LIFE SITUATION AT THIS POINT, YOU WOULD BE TREADING ON
DANGEROUS GROUND, BUT HERE WE CAN EXPERIMENT AND WATCH WHAT HAP-PENS."
900 PRINT @ 903,"-:-:-:- THE SPACE BAR WILL TURN THE PAGE -:-:-:-";
910 I$=INKEY$
920 I$=INKEY$
930 IF I$="" THEN 920
940 IF I$=" " THEN CLS ELSE 920
950 PRINT"WORKING OUT TOWARDS THE END OF THE TAPE IS RISKY. ON SOME RARE OCCASI
ONS THE EOT/BOT SPLICE MARKER WON'T PROTECT YOU FROM OVER-WRITING THE FILE 0 MAR
KER AND THEN YOU ARE IN TROUBLE. IF YOU FAIL TO KEEP TRACK OF YOUR DATA USAGE,
";
960 PRINT" YOU COULD BE WIPING OUT FILES WITHOUT REALIZING IT."
970 PRINT
980 PRINT"JUST WHAT WILL HAPPEN WITH THESE DEMO. FILES WILL DEPEND ON VARIOU
S FACTORS. SOME OF THESE ARE: THE ACTUAL LENGTH OF YOUR 5 FT. WAFER; THE @NEWED
BYTE COUNT; HOW LONG ";
990 PRINT"THE ESF HAS BEEN ON AND RUNNING, ETC., ETC. I WILL STAY ALERT AND TRY
TO CATCH YOU WITH ERROR MESSAGES. IF EVERYTHING IS WORKING OK (I.E., THE EOT/BO
T MARKER IS PROTECTING FILE 1),";

```



```

1000 PRINT " PRESS " CHR$(34)"P" CHR$(34)" FOR PASS AND I'LL @NEW FOR YOU AND P
ASS ALONG SOME INTERESTING INFORMATION."
1010 PRINT @ 967,"-:-:-:- THE SPACE BAR WILL TURN THE PAGE -:-:-:-";
1020 I$=INKEY$
1030 I$=INKEY$
1040 IF I$="" THEN 1030
1050 IF I$=" " THEN CLS ELSE 1030
1060 PRINT
1070 PRINT
1080 PRINT"DURING THIS FUN PERIOD OF THE DEMO. I JUST MIGHT TAKE OFF AND @NEW
ON MY OWN. PLEASE UNDERSTAND THAT I AM NOT PROGRAMMED TO DO THIS BUT SOMETIMES WO
RKING WITH LOST DATA CAN BE SO FRUSTRATING THAT I JUST CAN'T TAKE IT ANYMORE."
1090 PRINT
1100 PRINT
1110 PRINT"NOW HIT A KEY AND LET'S WRITE, READ, AND DELETE SOME DATA FILES.

```

BE PREPARED FOR ANYTHING!";

```

1120 I$=INKEY$
1130 IF I$="" THEN 1120
1150 CLS
1160 PRINT"ENTER FILE NUMBER YOU WISH WRITTEN. ";
1170 I$=INKEY$
1180 IF I$="" THEN 1170
1190 IF I$<"0" OR I$>"9" THEN 1170
1200 X=VAL(I$)
1210 IF C AND X>Z+1 THEN X=Z: GOTO 2470
1220 IF C=0 AND I$<>"1" THEN PRINT: PRINT: PRINT"SORRY. FIRST FILE ON TAPE MUST
BE A FILE #1. ALL FILES MUST BE SEQUENTIAL STARTING WITH #1. PLEASE TRY AGAIN.
"
1230 IF C=0 AND I$<>"1" THEN FOR T=1 TO 3000: NEXT: I$=INKEY$: GOTO 1150
1240 PRINT I$
1250 PRINT
1260 IF C AND X<Z AND X<>0 THEN 2550
1270 IF C<>1 THEN PRINT"I AM NOW LOOKING FOR THE BEGINNING OF TAPE MARKER.
(FILE 0 MARKER)" ELSE PRINT"I AM NOW LOOKING FOR THE END OF FILE";
1280 B=0
1290 IF A THEN PRINT Y-1 ELSE IF C THEN PRINT X-1
1300 IF A THEN 1650
1310 IF I$="1" THEN X$="A" ELSE IF I$="2" THEN X$="B" ELSE IF I$="3" THEN X$="C"
ELSE IF I$="4" THEN X$="D" ELSE IF I$="5" THEN X$="E" ELSE IF I$="6" THEN X$="F"
ELSE IF I$="7" THEN X$="G"
1320 @ OPEN X
1330 IF X>6 THEN 3270
1340 PRINT
1350 PRINT"THE FOLLOWING IS WHAT I WILL OUTPUT TO WAFER.
WATCH THE WRITE LIGHT."
1360 PRINT
1370 FOR T=1 TO 800
1380 NEXT
1390 FOR I%=0+X TO 9+X 'HERE AND THE READ LOOP AT
1680 IS WHERE YOU SHOULD
SUBSTITUTE AND EXPERIMENT
(THIS IS LINE 8080 IN THE
ACTUAL COMPUTER PROGRAM)
1400 @ PRINT I%
1410 PRINT USING "#####"; I%;

```

```
1420 NEXT I%
1430 PRINT
1440 PRINT TAB(17);
1450 FOR I=1 TO 30
1460 @ PRINT X$
1470 PRINT X$;
1480 NEXT I
1490 PRINT
1500 FOR I!=11*X TO 11*X+9
1510 @ PRINT I!
1520 PRINT USING"#####"; I!;
1530 NEXT I!
1540 PRINT
1550 FOR T=1 TO 500
1560 NEXT
1570 @ CLOSE
1580 FOR T=1 TO 500
1590 NEXT
```

```

1600 FOR I=1 TO 7
1610 PRINT @ 1023, ""
1620 NEXT
1630 PRINT @ 384, "I JUST WROTE THE ABOVE (THE FIRST WRITE) AND A LITTLE POSTAMBL
E (THE SECOND WRITE). THIS WILL TELL ME WHERE TO START THE NEXT
FILE."
1640 GOTO 1850

1650 @ OPEN Y
1660 PRINT
1670 PRINT "READ FILE" Y
1680 FOR I%=1 TO 10
1690 @ INPUT J%
1700 PRINT USING "*****"; J%;
1710 NEXT I%
1720 PRINT
1730 PRINT TAB(17);
1740 FOR I=1 TO 30
1750 @ INPUT X$
1760 PRINT X$;
1770 NEXT I
1780 PRINT
1790 FOR I!=1 TO 10
1800 @ INPUT J!
1810 PRINT USING "*****"; J!;
1820 NEXT I!
1830 PRINT
1840 @ CLOSE

1850 B=0
1860 IF A THEN PRINT: PRINT "YOU STILL ";: A=0 ELSE PRINT: PRINT "YOU NOW ";
1870 PRINT "HAVE" X;
1880 IF X=1 THEN PRINT "FILE"; ELSE PRINT "FILES";
1890 IF W THEN PRINT " ON THE WAFER. (MAYBE!!)": W=0 ELSE PRINT " ON THE WAFER."
1900 PRINT "DO YOU WISH TO (R)EAD, (W)RITE, (D)ELETE FILES OR (P)ASS?";
1910 Z=X
1920 H$=INKEY$
1930 IF B THEN C=0 ELSE C=1
1940 H$=INKEY$
1950 IF H$="" THEN 1940
1960 IF H$="D" THEN G$=" DELETE ": GOTO 2180
1970 IF H$="W" THEN G$=" WRITE ": GOTO 1150
1980 IF H$="R" THEN 2000
1990 IF H$="P" THEN CLS: GOTO 2730 ELSE 1940
2000 PRINT
2010 PRINT
2020 PRINT "WHICH FILE DO YOU WISH READ BACK IN AND HAVE DISPLAYED?";
2030 H$=INKEY$
2040 H$=INKEY$
2050 IF H$="" THEN 2040
2060 IF H$<"0" OR H$>"9" THEN 2040
2070 Y=VAL(H$)
2080 IF VAL(H$)>X+1 THEN 2140
2090 J$=I$
2100 I$=H$
2110 CLS
2120 A=1
2130 GOTO 1270

```



```

2140 CLS
2150 PRINT @ 192, "YOU ARE CALLING FOR A FILE THAT'S GREATER THAN ONE PLUS THE LA
STFILE NO. THEREFORE THERE IS NO END OF FILE MARKER FOR ME TO FIND AND I WILL RUN
FOREVER TRYING TO FIND IT.

SORRY, I CANNOT ALLOW THAT FILE SEARCH.
PLEASE TRY AGAIN."
2160 A=1
2170 GOTO 1840
2180 CLS
2190 PRINT "WHICH FILE DO YOU WISH TO DELETE? (REMEMBER ALL FILES AFTER THAT
FILE NUMBER WILL BE DELETED ALSO) ";
2200 D$=INKEY$
2210 D$=INKEY$
2220 IF D$="" THEN 2210
2230 IF D$<"0" OR D$>"9" THEN 2210
2240 PRINT D$
2250 PRINT
2260 PRINT
2270 B=1
2280 IF D$="0" THEN D$="1": X=VAL(D$)-1 ELSE X=VAL(D$)-1: IF VAL(D$)>Z+1 THEN X=Z
: GOTO 2470
2290 IF VAL(D$)=Z+1 THEN A=1
2300 PRINT @ 525, "FROM FILE" VAL(D$)"ON UP";
2310 PRINT STRING$(2, 27)STRING$(13, 24);
2320 @ NEW VAL(D$)
2330 PRINT
2340 PRINT
2350 PRINT "THERE'S A NICE WAY TO FIND OUT HOW MUCH ROOM YOU HAVE LEFT ON A DATA
TAPE. JUST @NEW WITH A FILE NUMBER ONE HIGHER THAN THE LASTFILE YOU HAVE ON THE
WAFER."
2360 PRINT
2370 GOTO 1850
2380 CLS
2390 PRINT "ENTER FILE NUMBER YOU WISH WRITTEN. ";
2400 B=1
2410 I$=INKEY$
2420 I$=INKEY$
2430 IF I$="" THEN 2420
2440 IF I$<"2" OR I$>"9" THEN 2420
2450 PRINT I$
2460 GOTO 1320
2470 CLS
2480 PRINT @ 192, "YOU ARE ATTEMPTING TO" G$ "A FILE THAT'S GREATER THAN ONE PLUS
THE LAST FILE NO. THEREFORE THERE IS NO END OF FILE MARKER FOR ME TO FIND AND I
WILL RUN FOREVER TRYING TO FIND IT."
2490 PRINT
2500 PRINT "SORRY, I CANNOT ALLOW THAT" G$ "OPERATION TO BE";
2510 PRINT " ATTEMPTED.
PLEASE TRY AGAIN."
2520 PRINT
2530 A=1
2540 GOTO 1850
2550 CLS
2560 PRINT @ 256, "THIS IS THEORETICALLY POSSIBLE BUT NOT GOOD PRACTICE. THERE I
S ALWAYS THE CHANCE THAT THE NEXT HIGHER FILE COULD, UNDER SOME CIRCUMSTANCES,
BE OVERWRITTEN. DON'T CHANCE IT.";
2570 PRINT " IF YOU NEED TO REWRITE A PREVIOUS FILE, INITIALIZE A NEW WAFER AND

```

CREATE A SET OF NEW FILES."

2580 A=1

2590 X=Z

2600 GOTO 1840

2610 @ CLEAR ' ***** THIS IS LINE 8700 IN YOUR ACTUAL
COMPUTER PROGRAM *****

2620 PRINT

2630 PRINT "TAPE IS TOO SHORT FOR THIS FILE. I RECOMMEND THAT YOU ATTEMPT A " CHR
\$(34)"READ" CHR\$(34)" OF FILE 1. IN SOME CASES THIS ERROR CONDITION CAN WIPE O
UT THE FILE 0 MARKER. ";

2640 PRINT " IF THE TAPE WANTS TO RUN FOREVER, THEN YOU HAVE LOST THE MARKER.

IN THIS CASE, HIT THE BREAK KEY. I WILL @NEW THE WAFFER FOR YOU AND PASS YOU ALO
NG TO SOME FURTHER INFORMATION. WHENEVER YOU GET A TAPE TOO SHORT ERROR";

2650 PRINT " ALWAYS ENTER AN @CLEAR COMMAND. THIS WILL CLOSE THE FILE WITHOUT WRIT
ING AN END OF FILE MARKER LIKE AN @CLOSE WOULD. THIS GENERALLY WILL ALWAYS P
ROTECT YOU FROM OVER-WRITING THE FILE 0 MARKER. TO EXPERIMENT, CHANGE THE @CLEA
R IN LINE 2610 TO";

2660 PRINT " AN @CLOSE AND RERUN THE PROGRAM. WHEN YOU GET TO THE END OF THE TA
PE, YOU'LL NOTICE MOST OF THE TIME NOW YOU HAVE LOST YOUR PROTECTION."

2670 PRINT " *:::* PLAN YOUR DATA TRANSFERS CAREFULLY *:::*"

2680 PRINT " PRESS A KEY";

2690 A=1

2700 W=1

2710 X=X-1

2720 IF PEEK(14591) THEN FOR I=1 TO 14: PRINT @ 1023.: NEXT: PRINT @ 64, CHR\$(31)
: GOTO 1840 ELSE 2720

2730 @ NEW

2740 PRINT

2750 PRINT "BE ADVISED ----- EVERY TIME YOU OPEN AND CLOSE A FILE, SEVERAL HUND
RED BYTES OF TAPE SPACE IS USED JUST FOR OVERHEAD AND NOT IN THE STORAGE OF USEFUL
DATA INFORMATION. THIS OVERHEAD IS:"

2760 PRINT

2770 PRINT " A. APPROX. 395 BYTES IN WRITING THE FILE IDENTIFIERS AND
SPACE USED IN THE START UP AND SLOW DOWN TIMES OF THE DRIVE MOTOR."

2780 PRINT " B. 256 BYTES OF INTER-RECORD GAPS, I.E., EACH TIME THE
BUFFER";

2790 PRINT " EMPTIES, A 256 BYTE BUFFER GAP IS CREATED ON THE TAPE."

2800 PRINT

2810 PRINT " FOR A RULE OF THUMB IN FIGURING FILE STORAGE SPACE ON TAPE

... PRESS A KEY ...";

2820 IF PEEK(14591) THEN CLS ELSE 2820

2830 PRINT "DETERMINE THE TYPE OF DATA (INTEGER, SINGLE-PRECISION, STRING, ETC)
AND RECORD THE MEMORY OVERHEAD VALUE. (SEE LEVEL 11 MANUAL PAGE A/16 OR A/17 FOR
APPLICABLE VALUES). NOW ADD UP ALL THE VALUES FOR THE NUMBER OF DIFFERENT";

2840 PRINT " DATA ITEMS. ADD THIS RESULT TO THE 395 BYTE END OF FILE MARKER. NO
W ADD TO THIS TOTAL, 256 TIMES THE NUMBER OF BUFFER DUMPS. THE NUMBER OF BUFFER
DUMPS CAN BE DETERMINED BY:

BUFFER DUMPS = 1 + INT(NUMBER OF DATA ITEMS/256)

2850 PRINT "KEEP TRACK OF YOUR FREE TAPE BYTE SPACE. AS YOU APPROACH THE END OF TA
PE SPACE, PLAY IT SAFE AND CHRISTEN A NEW WAFFER."

2860 PRINT

2870 PRINT" FOR AN EXAMPLE IN FIGURING A RULE OF THUMB CASE

... PRESS A KEY ...";

2880 I\$=INKEY\$

2890 I\$=INKEY\$

2900 IF I\$="" THEN 2890

2910 CLS

2920 PRINT @ 0,"QUANTITY	TYPE	VAL. FROM MAN.	EXTENDED
--------------------------	------	----------------	----------

2930 PRINT @ 64, STRING\$(64, 45)

2940 PRINT @ 130,"10" TAB(17)"INTEGER" TAB(40)"2" TAB(59)"20"

2950 PRINT @ 194,"20" TAB(17)"STRING" TAB(40)"1" TAB(59)"20"

2960 PRINT @ 258,"10" TAB(17)"SNG.-PREC." TAB(40)"4" TAB(59)"40"

2970 PRINT @ 320,"-----" TAB(57)"-----"

2980 PRINT @ 386,"40" TAB(32)"TOTAL DATA ITEM BYTES ↑↑↑ 80"

2990 PRINT @ 484,"+ ONE BUFFER DUMP ↑↑↑ 256"

3000 PRINT @ 553,"+ EOF MARKER ↑↑↑ 395"

3010 PRINT @ 633,"-----"

3020 I\$=INKEY\$

3030 PRINT @ 640,"THESE 40 DATA ITEMS WILL USE APPROXIMATELY 731 BYTE
S"

3040 PRINT

3050 PRINT"AS YOU CAN SEE, DATA PROCESSING REQUIRES SERIOUS THOUGHT AND PREPA
RATION TO USE STORAGE SPACE EFFICIENTLY."

3060 PRINT

3070 PRINT" -:--:-- PRESS A KEY -:--:--";

3080 I\$=INKEY\$

3090 IF I\$="" THEN 3080

3100 S\$="RULE OF THUMB PROGRAM"

3110 CLS

3120 PRINT @ 342,;

3130 IF SS=0 THEN PRINT"(1) RERUN DEMO.

(2) QUIT

(3) " S\$: PRINT @ 727,"[CHOOSE A NUMBER": I\$=INKEY\$ ELSE
PRINT"(1) RERUN DEMO.

(2) QUIT": I\$=INKEY\$

3140 I\$=INKEY\$

3150 IF I\$="" THEN 3140

3160 IF I\$="1" THEN A=0: B=0: C=0: SS=0: W=0: X=0: Y=0: Z=0: I\$="": S\$="": GOTO
1150

3170 IF I\$="2" THEN CLS ELSE IF I\$="3" AND SS=0 THEN 3350 ELSE 3140

3180 CLEAR 50

3190 POKE 16396, 201

3200 ON ERROR GOTO 0

3210 CLS

3220 END

3230 PRINT @ 395,"YOU HAD A PARITY ERROR ON A READ OPERATION

TRY AGAIN"

3240 FOR T=1 TO 900

3250 NEXT

3260 GOTO 1840

3270 @ CLEAR

3280 CLS

3290 PRINT"AS YOU KNOW, I AM WRITING INDIVIDUAL FILES USING INTEGERS, SNG. PREC.

NUMBERS, AND STRINGS. WITH THE AMOUNT OF BYTE SPACE THESE RECORDS USE, ALONG WITH THE PROTECTIVE GAPS BETWEEN EACH FILE, YOU SHOULD NOT BE GETTING THIS MANY";

3300 PRINT"FILES ON A 5 FT. WAFER. YOU ARE OBVIOUSLY WRITING OVER YOUR EXISTING FILES AND WOULD BE GETTING SOME STRANGE RESULTS IF YOU TRIED TO READ THEM.

UNDER CERTAIN CIRCUMSTANCES, WHEN YOU'RE PRINTING";

3310 PRINT" DATA OUT NEAR THE END OF THE TAPE, A TAPE TOO SHORT ERROR WON'T PROTECT YOU, AND YOUR PRINT OPERATION WILL SLIDE RIGHT BY THE EOT/BOT MARKER.RESULT

**** DISASTER **** . --- WHAT TO DO? --- IN THIS CASE PRESS THE BREAK KEY";

3320 PRINT". I'LL @NEW THE WAFER FOR YOU AND PASS YOU ALONG TO THE NEXT PART OF THE PROGRAM. -- MORAL -- KEEP DATA AWAY FROM THE END OF THE TAPE. WAFERS ARE CHEAP COMPARED TO YOUR LOST TIME."

3330 PRINT" -:-:-:- PRESS THE BREAK KEY -:-:-:-";

3340 IF PEEK(14400)=4 THEN CLS: GOTO 2730 ELSE 3340

3350 CLS

3360 I=0

3370 B=256

3380 O=395

3390 INPUT"HOW MANY INTEGERS";I

3400 INPUT"HOW MANY SINGLE-PRECISIONS";SP

3410 INPUT"HOW MANY DOUBLE PRECISIONS";DP

3420 INPUT"HOW MANY STRING CHARACTERS";ST

3430 D=2*I+4*SP+8*DP+ST

3440 E=I+SP+DP+ST

3450 TS=D+O+256*(1+INT(D/256))

3460 IF E=0 THEN TS=0: Z\$="" ELSE Z\$=" APPROXIMATELY"

3470 PRINT

3480 PRINT"THESE" E"DATA ITEMS"

3490 PRINT"WILL EAT UP" Z\$TS"BYTES OF TAPE SPACE."

3500 PRINT

3510 PRINT"(M)ENU OR (F)IGURE MORE DATA SPACE?"

3520 I\$=INKEY\$

3530 I\$=INKEY\$

3540 IF I\$="" THEN 3530

3550 IF I\$="F" THEN I=0: SP=0: DP=0: ST=0: D=0: E=0: TS=0: GOTO 3350

3560 IF I\$="M" THEN SS=1: GOTO 3110 ELSE 3530

***** STRINGY-FLOPPY DATA I/O DEMONSTRATION ROUTINES *****

----- ADVANCED (A COMPLETE FILE PROCESSING PROGRAM) -----

0 CLS:PRINTCHR\$(23):PRINT@392,"A FILE PROCESSING DEMO.
M. N. KIDDER":FORT=1T01000:NEXT

BY

```
200 ' *****
    * THIS WILL ILLUSTRATE A COMPLETE FILE PROCESSING PROGRAM *
    *****

210 CLS: CLEAR 500      ' PROVIDE STRING SPACE
215 PRINT "DATA I/O DEMONSTRATION PROGRAM
    (LIST PROGRAM FOR COMMENTS)"
220 ON ERROR GOTO 3000 ' THIS IS NEEDED TO TRAP DATA ERRORS
230 POKE 16396,23      ' DISABLE THE BREAK KEY. THE BREAK KEY
                        ' IS THE MEANS USED TO STOP THE DRIVE
                        ' WHEN A FILE CANNOT BE FOUND.

240 @CLEAR              ' USED TO RESET ANY OPEN FILE CONDITION
                        ' AND TO VERIFY THAT THE DATA I/O PROGRAM
                        ' IS AVAILABLE (SEE LINE 3010 ).

250 PRINT "CREATING A FILE" ' WE WILL GENERATE A TEST FILE
                        ' THIS IS WHERE YOU WOULD CREATE YOUR FILE

260 DEFSTR S,K          ' OUR FILE WILL HAVE ONE STRING FIELD
270 DEFINT I,J,N,X,Y    ' AND ONE INTEGER FIELD.
280 DIM S(26)           ' OUR FILE WILL HAVE 26 RECORDS IN ARRAYS
290 DIM I(26)           ' EACH CONTAINING 1 STRING AND 1 INTEGER
300 I=ASC("A")-1        ' SET START GENERATION CHARACTER
310 FOR J=1 TO 26        ' GENERATE THE FILE
320 S(J)=STRING$(4,I+J)' AAAA, BBBB, CCCC, ....
330 I(J)=J              ' 1, 2, 3, ...
340 NEXT J              ' REPEAT THE GENERATION LOOP
350                     ' NOW WE ARE READY TO WRITE THE FILE
360 GOSUB 1000           ' GO SEE IF THERE IS A REFLECTIVE STICKER
    I=0 YES
                        I=1 NO
                        I=2 BREAK KEY WAS HIT

370 IF I=0 PRINT "REMOVE WAFER" ' IF THERE IS A WAFER WITH A
                        REFLECTIVE STICKER IN THE DRIVE, BE
                        SAFE AND REMOVE IT.

380                     ' AN EXTRA STEP THAT CAN PREVENT
                        DISTROYING A PROGRAM OR DATA FILE.

390 GOSUB 1000           ' SEE IF WAFER WAS REMOVED
400 IF I=2 THEN 2000     ' IF THE BREAK KEY WAS HIT, EXIT
410 IF I=0 THEN 390      ' IF THE WAFER WAS NOT REMOVED THEN WAIT
420 PRINT "INSERT WAFER WITH A REFLECTIVE STICKER"
430 GOSUB 1000           ' SEE IF WAFER WAS INSERTED
440 IF I=2 THEN 2000     ' IF THE BREAK KEY WAS HIT, EXIT
450 IF I=1 THEN 430      ' WAIT FOR WAFER TO BE INSERTED
460 PRINT "ENTER FILE NUMBER "; ' WE CAN HAVE MORE THAN 1 FILE ON
                        A WAFER.
```



```

470 K=INKEY$:IF K="" THEN 470 'WAIT FOR A KEY TO BE HIT
480 IF ASC(K)=2 THEN 2000 'IF BREAK KEY, EXIT
490 IF K<"1" OR K >"9" THEN 470 'ONLY ACCEPT 1 - 9
500 PRINT K 'DISPLAY THE VALID KEY
510 X=VAL(K) 'SAVE THE FILE NUMBER FOR VERIFY
520 PRINT"OPENING FILE "K 'KEEP USER INFORMED
530 @OPEN X 'OPEN THE FILE
540 PRINT"WRITING FILE"
550 FOR I=1 TO 26 'SET UP LOOP
560 @PRINT S(I),I(I) 'WRITE A RECORD
570 NEXT I 'REPEAT THE LOOP
580 @PRINT "END",9999 'WRITE AN ENDING RECORD SO WE CAN
KNOW WHERE TO STOP WHEN READING BACK
590 @CLOSE 'CLOSE THE FILE
600 PRINT"FILE CLOSED" 'INFORM USER
610 PRINT"WRITE VERIFY"; 'VERIFY THE FILE?
620 GOSUB 6000 'FIND OUT IF YES OR NO
640 IF NO THEN 850 'GO LIST FILE
650 'VERIFY THE FILE
660 PRINT"OPENING FILE"X"FOR VERIFICATION"
670 @OPEN X 'OPEN SAME FILE
680 PRINT"VERIFYING FILE"
690 J=0 'INITIALIZE ARRAY NUMBER
700 J=J+1 'BEGINNING OF LOOP
710 @INPUT S,I 'READ ONE RECORD
720 IF S="END" AND I=9999 THEN 820 'TEST FOR END.
THIS IS ONLY ONE WAY OF KNOWING THE END OF A
FILE. IF THE NUMBER OF RECORDS IN A FILE IS
KNOWN BEFORE THE FIRST RECORD IS WRITTEN,
730 'A BETTER WAY IS TO FIRST WRITE A FIELD
CONTAINING THE NUMBER OF RECORDS AND USE IT IN
A LOOP TO READ THE FILE IN.
740 IF S=S(J) AND I=I(J) THEN 700 'IF O.K. READ NEXT RECORD
750 PRINT"VERIFY ERROR FOR RECORD"J 'BAD COMPARE
760 @CLOSE 'CLOSE THE FILE
770 PRINT"WANT TO RE-WRITE THE FILE"
780 GOSUB 6000 'FIND OUT IF YES OR NO
800 IF YES THEN 350 'GO WRITE THE FILE AGAIN
810 GOTO 840 'ELSE GO LIST THE FILE
820 PRINT"FILE VERIFIED CORRECTLY" 'FILE IS O.K.
830 @CLOSE 'CLOSE FILE
840 PRINT"FILE CLOSED"
850 PRINT"LIST FILE";
860 GOSUB 6000 'FIND OUT IF YES OR NO
880 IF NO THEN 2000 'EXIT
890 @OPEN X 'OPEN FILE FOR LIST
900 @INPUT S,I 'READ A RECORD
910 IF I=9999 THEN 940 'TEST FOR END
920 PRINT S,I, 'DISPLAY RECORD
930 GOTO 900 'GO READ NEXT RECORD
940 @CLOSE 'CLOSE FILE
950 PRINT"PRESS SPACE BAR";:I$=INKEY$ EXIT
960 I$=INKEY$:IFI$=""THEN960ELSEIFI$=" "THEN 2000 ELSE960
1000 I=INP(240) AND 1 'TEST FOR REFLECTIVE STICKER
1010 IF INKEY$=CHR$(2) THEN I=2 'TEST FOR BREAK KEY
1020 RETURN

```

```

2000                                ' END OR RERUN DEMO.
2010 CLS:PRINT@342,"(1) RERUN DEMO.

                                (2) QUIT":I$=INKEY$      ' MENU
2020 I$=INKEY$:IF I$="" THEN 2020 ELSE IF I$="1" THEN CLS:GOTO 460 ELSE IF I$="2
' THEN CLS ELSE 2020
2030 POKE 16396,201      ' ENABLE BREAK KEY
2040 ON ERROR GOTO 0      ' RESET ERROR TRAP
2050 CLEAR 50:END        ' RESET STRING SPACE AND GOODBYE
2090 END                  ' STOP! YOU' RE ALL THROUGH

3000 PRINT                ' IF SYNTAX ERROR ON @CLEAR, EXIT
3010 IF ERR=2 AND ERL=240 PRINT"DATA I/O PGM NOT LOADED":
RESUME 2030
3020 IF ERR<>42 THEN 2030 ' END IF NOT DATA I/O ERROR
3030 IF ERL<>530 THEN 3150 ' OPEN ERROR ON WRITE
3040 IF INKEY$<>CHR$(2) THEN 3130 ' BRANCH IF NOT BREAK KEY
3050 PRINT"BREAK KEY HIT. FILE START NOT FOUND"
3060 IF X<>1 THEN 3110 ' BRANCH IF NOT FILE 1
3070 PRINT"WANT TO CERTIFY WAFER";
3080 GOSUB 6000          ' FIND OUT IF YES OR NO
3090 IF NO THEN 3110    ' SKIP
3100 @NEW                ' CERTIFY WAFER
3110 PRINT"ENTER NEW FILE NUMBER ";
3120 RESUME 470          ' GO TRY WRITE AGAIN
3130 PRINT"DATA ERROR OPENNING FILE"X
3140 RESUME 460          ' RETRY WRITE
3150 PRINT"DATA I/O ERROR - ";
3160 @CLEAR              ' CLEAR OPEN CONDITION
3170 IF ERL<560 OR ERL>590 THEN 3230 ' ERROR WRITING FILE
3180 PRINT"PROBABLY TAPE TOO SHORT"
3190 PRINT"WANT TO TRY AGAIN";
3200 GOSUB 6000          ' YES OR NO
3210 IF YES RESUME 350 ' GO WRITE THE FILE
3220 RESUME 610          ' TRY VERIFY
3230 IF ERL<>670 AND ERL<>890 THEN 3300 ' OPEN ERROR
3240 PRINT"OPEN ERROR ON READ - ";
3250 IF INKEY$=CHR$(2) THEN PRINT"BREAK KEY HIT"
3260 PRINT: PRINT"WANT TO RETRY";
3270 GOSUB 6000          ' YES OR NO
3280 IF NO RESUME 2030 ' EXIT
3290 IF ERL=670 THEN RESUME 670 ELSE RESUME 890 ' RETRY OPEN
3300 IF ERL<700 OR ERL>900 RESUME 2030 ' UNKNOWN DATA I/O ERROR
3310 PRINT"BAD DATA ON READ"
3320 PRINT"WANT TO RETRY";
3330 GOSUB 6000          ' YES OR NO
3340 IF NO RESUME 2030 ' EXIT
3350 IF ERL=710 RESUME 650 ' RETRY VERIFY
3360 IF ERL=900 RESUME 890 ' RETRY LIST
3370 PRINT"ERROR WAS ON CLOSE"
3380 IF ERL=940 RESUME 2030 ' EXIT
3390 RESUME 850          ' GOTO LIST

6000 PRINT"? Y/N";      ' ROUTINE TO READ & SET YES OR NO
6010 K=INKEY$
6020 IF K="Y" THEN YES=-1: NO=0: GOTO 6050
6030 IF K="N" THEN YES=0: NO=-1: GOTO 6050
6040 GOTO 6010

```


6050 PRINT CHR\$(8) CHR\$(8) CHR\$(8) K

6060 RETURN

7000 A FILE MANAGEMENT SYSTEM IS AVAILABLE FROM EXATRON FOR \$19.95 THAT USES THESE PRINCIPALS. IT IS DESIGNED TO RUN ON A 16K OR GREATER SYSTEM AND USES A STRINGY FLOPPY TO SAVE FILES.

7010 THE FILE MANAGEMENT SYSTEM PROGRAM (CALLED FMS) IS WRITTEN IN BASIC AND HAS THE FOLLOWING FEATURES:

- * CAN BE USED FOR NAME AND ADDRESS FILES OR MOST OTHER INFORMATION OF A SIMILAR TYPE.

7020 * IT DYNAMICALLY ALLOCATES MEMORY BASED ON THE FILE DEFINITION, SO THE NUMBER OF RECORDS IS RELATED TO THE AMOUNT OF DATA IN A RECORD.

7030 * THE NUMBER & TYPES OF FIELDS (UP TO 9) ARE USER DEFINED

7040 * RECORDS CAN BE ADDED, CHANGED, DELETED, SEARCHED, SORTED, LISTED, PRINTED, TOTALED, SAVED AND LOADED.

7050 * FIELDS CAN BE FORMATED FOR PRINTING. FIELDS CAN BE PLACED IN ORDER ANYWHERE ON A LINE, ON A NEW LINE, OR NOT PRINTED. NUMERIC FIELDS CAN BE LINED UP WITH OR WITHOUT A DOLLAR SIGN AND/OR DECIMAL POINT.