

Central Data 2650 Newsletter

Welcome to the first issue of our 2650 newsletter. I hope that the information contained within these 12 pages will help make your system more useful and enjoyable. Naturally, this being the first issue, there is not a lot of input from you — our customers. I hope that in future issues a major portion of the newsletter will be devoted to feedback from you.

I welcome anything that you send in, be it a suggestion, complaint, program, or an improvement that you have made to your system. Any letters that are sent in to me will get a reply, although we may not be able to print them all due to space limitations.

This issue contains the most requested information that we have along with two great BASIC programs written by Mike Herbach and a dumb terminal program sent in by Mike Kelley.

I hope that this newsletter promotes more interaction between users as well as between Central Data and its customers. To help this along we are going to devote a section of the newsletter to listing names and addresses

of users who want to talk to someone in their area or about a specific application that they are working on. If you are interested in letting other people get in touch with you to trade information, just fill out the coupon below and send it in to us. If you have any special interests, be sure to list them so that others with the same interests can get in touch with you.

I'm all for talking to others about my system. Please list my name and address in the newsletter as soon as possible.

Name _____

Address _____

City _____ State _____ Zip _____

Telephone (optional) _____

Special Interests _____

Person to Person Communication

We don't want you to forget that we are here to help you. Feedback from you helps us improve our products and manuals, so we're all for it!

Just a couple of suggestions for when you contact us:

1) If you write us, be sure to put your name *and address* on the letter. Sometimes we can't reply to a letter because we don't have a return address! Also, be sure to send all letters to our P.O. Box and all packages to our street address.

2) If you call, it is a good idea to make a list of things you want to talk about before the call. This helps things go a bit more quickly, which lets you keep some extra change at the end of the month.

3) Also if you call, it is usually good to know who to ask for. Talk to Jeff Peterson if you have technical questions about our RAM boards and Vicki Peterson if you're having problems getting your computer running or if you want to send in a board for repair. Judy Lowery can help you make an order or send you any literature that you need. Naturally, that leaves me with just about everything else.

4) Try to call between 9:00 AM and 12:00 noon or between 1:00 PM and 4:30 PM, central standard time.

If the person you want isn't in when you call, tell us what you need to know and chances are we can find someone to help you.

Letters

Dear Jeff;

Thank you again for all of your help over the past months. My unit seems to be working fine at this time. There is one problem which I mentioned to you in our last telephone conversation. The problem has to do with my inability to read other people's tapes — specifically those that I received from Mike Herbach. By adjusting R22 to a critical location I was able to load Mike's tapes. However, my unit would not load its own tapes with the setting of R22 adjusted for Mike Herbach's tapes. You are undoubtedly right that our two [cassette tape] units are running at different speeds. Mike suggested that a tape "standard" be established that would allow us to use any tape unit and adjust R22 and R21 for a "match" rather than a spec.

I hope this tape arrived in time for the first newsletter. If so here is a question for the 2650 world: has anyone developed either software or hardware to get black on white/white on black?

After these comments [see the program section . . .Ed.] is a program to allow the 2650 system to be used as a dumb terminal. The program is not as yet completely debugged, however, the intent is to be up on a 8080/Z-80 system within the next two weeks. If there is anyone out there with similar interest I would be more than willing to share my experiences and problems in developing what should (some day) be a fairly sophisticated system.

Anyone who would wish to make comments concerning the program both positively or negatively please do so. I would be particularly interested in any modifications to expand it into a full operating system for an intelligent terminal — intelligent multi cassette system — intelligent teletypewriter — etc. Send me a cassette with your ideas and comments. I will return it with any new info received.

Thank you all,
Mike Kelley
9951 Delco Ave.
Chatsworth, CA 91311

P.S. Here is the dumb terminal program [see the program section]. It has no comments as there is a lack of adequate memory to do so. They will be supplied to anyone interested.

Dear Mike:

It's good to see your dumb terminal program! Lots of people call to ask if their computer can be used as a terminal for a larger computer.

The problem that you have with reading Mike's tapes is a hard one to solve. I recently got a tape from him which could not be read into our system at all . . . sometimes the two different tape recorders have such different speeds that it makes it impossible to send any data between the two machines.

Please keep us informed on any changes that you make to your program.

Dear Jeff,

Thanks for the update on your Editor/Assembler program for 2650.

The instructions at locations 32AD to 32B3 were omitted. However, I believe the following program will work much better. It stops at the bottom of the screen each time it is filled. Just hit any key for the next page. It uses the same locations your program uses.

```
3299 77 10 C3 0C 0B 9F 18 07 E7 0D 18 22 3F 03
      96 75 10 17 17 FF 3F 00 24 0C 0D B2 17
32C7 08 E2 F4 0F 3C 03 0F 3F 00 24 1B 55
3277 1F 32 99
2F24 3F 32 AD
```

Ed Young
PO Box 2111
Evansville, IN 47714

Dear Ed:

Thanks for your changes to the Editor/Assembler! It sure helps to be able to copy down some of those important values before they whiz by! The following bytes were indeed left out of the last change sheet:

```
32AD 3F 00 24 0C 0D B2 17
```

Hope that this helps out some of you people with no printer.

Jeff,

Below you will find a BASIC program to dump the TVT screen to a TTY at port 3. It's very slow (1 LPM), but does give a way to save a BASIC program's output. It's biggest value may be as seed for a machine language program used with a call command.

Am sending it along for what it's worth.

```
*      PROGRAM TO DUMP SCREEN TO TTY
9000 FOR A=4096 TO 4112
9010 FOR B=64 TO 1056 STEP 16
9020 G=B/16
9030 IF G=62 GOT09400
9040 C=A+B
9050 PEEK C,D
9055 IF D<32 LET D=D+64
9060 EXTOUT 3,D
9100 NEXT B
9110 NEXT A
9120 ERASE
9130 RETURN
9400 EXTOUT 3,13
9410 FOR P=1 TO 10
9420 NEXT P
9430 EXTOUT 3,10
9440 FOR W=1 TO 5
9450 NEXT W
9460 GOTO 9100
```

Dave Reinhart

Dear Dave:

Thanks for the program. I think that you're right in saying that it is better off done in machine language and then calling it from BASIC. An assembly language program could even abort sending all of the spaces at the end of short lines . . . which in itself could speed it up quite a bit. Anyone willing to whip up such a program is more than welcome to send it in!

Jeff:

The second issue of "2650 Computer User Notes" contained a list of changes to the Central Data Editor/Assembler. These changes have been superseded by the change notice Central Data sent out at the first of the year. However, two of my changes are not in that change notice and should still be added. Here they are, reset for the new assembler memory locations:

1) To have the cursor move when you push any key after a "BINARY OUTPUT" message, change 3179 to IF 33 15 and change 3315 to 3F 01 70, 3F 32 10, 1F 31 7C.

2) To have the cursor move when you push any key after a "DATA INPUT" message, change 2DBF to 1F 33 1E and change 331E to 3F 01 70, 3F 23 16, 1F 2D C2.

Mike Herbach
Signetics

Mike:

Thanks for the cassette which contained your letter and the two BASIC programs which are in this newsletter. I am afraid that I could not load the letter too reliably, so most of the last part is missing.

I hope that the above changes help some people, however, and I thought that printing part of your letter was better than printing none of it.

80 Chars/Line

A lot of people have called in asking how to get the full 80 characters per line on the screen. As most of you know, the supervisor program limits the number of characters per line to just 76. The reason for this is that if you get a hold of some slow RAM for use in the display section, the first four characters will be illegible because of timing problems.

If you are sure about your RAMs, you can change the PROM to allow a full 80 characters per line on the screen. The following locations must be changed:

```
001D 00
005B 00
```

Also, the Editor/Assembler and BASIC need changes if you change the PROM. For the Editor/Assembler, change the following:

```
20E8 29
24DD 01
```

For BASIC, change the following addresses:

```
2061 29
23EC 01
```

Programs

Lunar Lander by Mike Herbach

* REAL TIME LUNAR LANDER GAME BY MIKE HERBACH 1-7-78

*

* INSERTS PROGRAM FOR KEYBOARD INPUT AT LOCATION CALCULATED FROM

* END OF AVAILABLE MEMORY

ERASE

PRINT%10 THIS IS A LUNAR LANDER SIMULATOR

INPUT'DO YOU WANT INSTRUCTIONS? PUSH 1 FOR YES.'Y

IF Y<>1 GOTO 10

ERASE

PRINT'YOU ARE THE COMMANDER OF A LUNAR LANDER MODULE AND ARE ABOUT TO
PRINT'LAND ON THE MOON. YOU MUST BURN THE AMOUNT OF FUEL THAT WILL BRING'
PRINT'YOU TO THE SURFACE WITH A FINAL SPEED OF 5 METERS PER SECOND OR LESS.'
PRINT'IF YOU DO NOT PUSH A NUMBER BUTTON ON YOUR COMPUTER CONSOLE '
PRINT'THE COMPUTER WILL AUTOMATICALLY STOP BURNING FUEL.'

PRINT

PRINT'A BURN OF 2 GRAMS IS NEEDED TO CANCEL THE EFFECT OF GRAVITY.'

PRINT

INPUT'PUSH "RETURN" TO START THE SIMULATION AND GOOD LUCK. COMMANDER!'Y

* PEEK, POKE AND DATA VALUES ARE DECIMAL EQUIVALENTS OF HEX NUMBERS.

10 PEEK 8196.M

LC=M*256-10

L=LC

* THIS IS A PROGRAM TO GET ANY KEYBOARD INPUT AND PUT THE

* NUMBER INTO A STORAGE LOCATION FOR FUTURE USE.

DATA 115,63,1.154,203,2,23,0

RESTORE

FOR T=1 TO 8

READ D

POKE L,D

L=L+1

NEXT T

ERASE

X=2

FOR I=1 TO 11

PRINT@I+1,65 #500-(I-1)*50

NEXT I

PRINT@12,50'-----'

20 D=500

F=100

T=0

S=-50

PRINT@1,5'TIME: '#T

PRINT@1,20'FUEL: '#F

PRINT@1,40'SPEED: '#S

PRINT@1,60'DISTANCE: '#D

```

PRINT@X,55'X'
PRINT@12,5
INPUT'READY? (PUSH RETURN)'Y
* ERASE 'READY....'
PRINT@13,5'
30 CALL LC
PEEK L.B
IF B>F B=F
IF F<=0 B=0
T=T+1
F=F-B
C=B-2
D=D+S+C/2
S=S-C
IF D<=0 GOTO 60
PRINT@1,10 #T
PRINT@1,25 #F
PRINT@1,47 #S
PRINT@1,69 #D
PRINT@X,55' '
X=12-INT(D/50+.5)
PRINT@X,55'X'
GOTO 30
60 IF S>-5 GOTO 70
PRINT@13,5'YOU HAVE CRASH LANDED AT A SPEED OF '#S' METERS PER SECOND.'
GOTO 100
70 PRINT@13,5'CONGRADULATIONS! YOU LANDED WITH A FINAL SPEED CF '#S' METERS
100 INPUT'DO YOU WANT TO TRY AGAIN? TYPE 1 FOR YES.'Y PFR SECOND.'
IF Y<>1 STOP
PRINT@X,55' '
* REPLACE '-' IF X WAS ON IT
IF X=12 PRINT@X,55'-----'
X=2
* ERASE TWO LINES ON BOTTOM OF SCREEN
PRINT@12,1
PRINT
PRINT
GOTO 20

```

Subscription Info.

Editor: Jeff Roloff.

The Central Data 2650 Newsletter is published every 8 weeks by Central Data Corporation. Any comments should be sent to Central Data Corp.; PO Box 2484, Station A; Champaign, IL 61820. The subscription rate is \$4.00 per year.

Please enter my subscription to the Central Data 2650 Newsletter. Enclosed is \$4.00.

Name _____

Address _____

City _____

State _____ Zip _____

Send to: Central Data Corp.; PO Box 2484, Station A; Champaign, IL 61820.

Lower Case Write-Character Routine

This routine acts just like the WCHR routine in the supervisor. The only differences are that the character to be written must be passed in RO in this routine, and that this routine will select the lower case PROM when applicable.

```

0001 0000          PRNT
0002 0000          *
0003 0000          *
0004 0000          *
0005 0000          *   THIS ROUTINE ALLOWS THE USE OF A LOWER CASE PROM
0006 0000          *
0007 0000          *
0008 0000          *
0009 0000      EQ   EQU           0           POSSIBLE CONDITIONS
0010 0000      GT   EQU           1
0011 0000      LT   EQU           2
0012 0000      UN   EQU           3
0013 0000      R0   EQU           0           REGISTER EQUATES
0014 0000      R1   EQU           1
0015 0000      R2   EQU           2
0016 0000      R3   EQU           3
0017 0000      WC   EQU           8           WITH CARRY BIT OF PSL
0018 0000      CURSCR EQU        17FE
0019 0000      LFCR2 EQU        0029
0020 1600          ORG        1600
0021 1600          *
0022 1600          *
0023 1600          *
0024 1600 44 7F      WCHR      ANDI ,R0      7F           MASK OFF TOP BIT
0025 1602 F4 60          TMI ,R0      60           SEE IF IT IS A LOWER CASE
0026 1604 18 0A          BCTR,EQ      WCHR2        BRANCH IF SO
0027 1606 24 FF          EORI ,R0     FF           INVERT THE BYTE
0028 1608 F4 60          TMI ,R0      60           SEE IF IT IS A CONTROL CHAR
0029 160A 18 25          BCTR,EQ      WCHR3        BRANCH IF SO
0030 160C 24 FF          EORI ,R0     FF           CHANGE IT BACK TO NORMAL
0031 160E 44 3F          ANDI ,R0      3F           SELECT CHARACTER GENERATOR 1
0032 1610 CC 97 FE      WCHR2      STRA ,R0     *CURSOR      STORE THE CHARACTER
0033 1613 0C 17 FF          LODA ,R0     CURSOR+1      INCREMENT CURSOR POSTION
0034 1616 84 10          ADDI ,R0      10
0035 1618 CC 17 FF          STRA ,R0     CURSOR+1
0036 161B 77 08          PPSL          WC
0037 161D 20          EORZ ,R0
0038 161E 8C 17 FE          ADDA ,R0     CURSOR
0039 1621 CC 17 FE          STRA ,R0     CURSOR
0040 1624 75 08          CPSL          WC
0041 1626 F4 05          TMI ,R0      5
0042 1628 1C 00 29          BCTA ,EQ      LFCR2        IF END OF LINE, DO LINEFEED
0043 162B 04 1C          WCUR      LODI ,R0     '\ - 40      LOAD ASCII FOR CURSOR
0044 162D CC 97 FE          STRA ,R0     *CURSOR      STORE NEW CURSOR
0045 1630 17          RETC ,UN
0046 1631 24 FF          WCHR3      EORI ,R0     FF           INVERT THE BYTE
0047 1633 64 40          IORI ,R0     40           ADD ON THE BIT FOR LOWER CASE
0048 1635 1B 59          BCTR ,UN      WCHR2        WRITE IT

```

Dumb Terminal Routine

by Mike Kelley

0001	0000			PRNT	
0002	0000			IFCR	EQU 24
0003	0000			GO	EQU 39F
0004	0000			FLAG	EQU 17E8
0005	0000			AMK	EQU 3F
0006	0000			CSD	EQU 10
0007	0000			DAD	EQU 11
0008	0000			RDRF	EQU 01
0009	0000			TDRE	EQU 02
0010	0000			KBS	EQU 80
0011	0000			SET	EQU 01
0012	0000			DELY	EQU 40
0013	0000			EMK	EQU 30
0014	0000			CRLE	EQU 05
0015	0000			LF	EQU 0A
0016	0000			CR	EQU 0D
0017	0000			BEL	EQU 07
0018	0000			DELE	EQU 7F
0019	0000			CRLS	EQU 13
0020	0000			CUR1	EQU 17FE
0021	0000			CUR2	EQU 17FF
0022	0000			RETU	EQU 83
0023	0000			SPCD	EQU 20
0024	0000			CUCD	EQU 1C
0025	0000			R0	EQU 0
0026	0000			R1	EQU 1
0027	0000			R2	EQU 2
0028	0000			R3	EQU 3
0029	0000			EQ	EQU 00
0030	0000			GT	EQU 01
0031	0000			LT	EQU 02
0032	0000			UN	EQU 03
0033	1510			ORG	1510
0034	1510	77	12	PPSL	12
0035	1512	04	11	LODI.R0	11
0036	1514	D4	10	WRTE.R0	CSD
0037	1516	20		EORZ.R0	
0038	1517	CC	17 E8	STRA.R0	FLAG
0039	151A	3F	15 91	BSTA.UN	CFSR
0040	151D	56	10 RSIP	REDE.R2	CSD
0041	151F	F6	01	TMI ,R2	RDRF
0042	1521	18	28	BCTR.FQ	DISP
0043	1523	0E	17 E8 RFLG	LODA.R2	FLAG
0044	1526	20		EORZ.R0	
0045	1527	F2		COMZ,R2	
0046	1528	1A	0C	BCTR.LT	FLGS
0047	152A	72		REDD.R2	
0048	152B	F6	80	TMI ,R2	KBS
0049	152D	1A	6E	BCTR,LT	RSIP
0050	152F	06	01	LODI.R2	SET
0051	1531	CF	17 E8	STRA.R2	FLAG

0052	1534	1B	67		BCTR,UN	RSIP
0053	1536	72		FLGS	REDD,R2	
0054	1537	F6	80		TMI ,R2	KBS
0055	1539	18	62		BCTR.EQ	RSIP
0056	153B	55	10	LP1	REDE,R1	CSD
0057	153D	F5	02		TMI ,R1	TDRE
0058	153F	1A	7A		BCTR,LT	LP1
0059	1541	D6	11		WRTE,R2	DAD
0060	1543	20			EORZ,R0	
0061	1544	CC	17	E8	STRA,R0	FLAG
0062	1547	05	40		LODI,R1	DELY
0063	1549	1B	30		BCTR,UN	DECD
0064	154B	05	40	DISP	LODI,R1	DELY
0065	154D	46	30		ANDI,R2	EMK
0066	154F	E6	00		COMI.R2	00
0067	1551	3D	15	A8	BSTA,GT	ERR
0068	1554	56	11		REDE,R2	DAD
0069	1556	46	3F		ANDI,R2	AMK
0070	1558	E6	0A		COMI.R2	LF
0071	155A	3C	00	24	BSTA,EQ	LFCR
0072	155D	E6	0D		COMI,R2	CR
0073	155F	1C	15	7B	BCTA.EQ	DECD
0074	1562	E6	05		COMI,R2	CRLE
0075	1564	3C	15	91	BSTA,EQ	CESR
0076	1567	E6	07		COMI.R2	BEI
0077	1569	1C	15	7B	BCTA.EQ	DECD
0078	156C	E6	7F		COMI,R2	DELE
0079	156E	1C	15	7B	BCTA,EQ	DECD
0080	1571	E6	13		COMI,R2	CRLS
0081	1573	1C	00	83	BCTA,EQ	RETU
0082	1576	02			LODZ,R2	
0083	1577	C3			STRZ,R3	
0084	1578	3F	03	9F	BSTA.UN	GO
0085	157B	FD	15	23	BDRA,R1	RFLG
0086	157E	72			REDD,R2	
0087	157F	F6	80		TMI ,R2	KBS
0088	1581	18	05		BCTR,EQ	HKBR
0089	1583	06	01		LODI,R2	SET
0090	1585	CF	17	E8	STRA,R2	FLAG
0091	1588	F6	10		REDE,R2	CSD
0092	158A	F6	01	HKBR	TMI ,R2	RDRF
0093	158C	1C	15	4B	BCTA,EQ	DISP
0094	158F	1F	6A		BCTR,UN	DECD
0095	1591	06	10		L DI,R2	10
0096	1593	CF	17	FE	STRA.R2	CUR1
0097	1596	06	11		LODI,R2	11
0098	1598	3F	00	24	BSTA,UN	LFCR
0099	159B	FA	7B		BDRR,R2	LP2
0100	159D	07	20		LODI,R3	SPCD
0101	159F	CF	97	FE	STRA,R3	*CUR1
0102	15A2	06	40		LODI,R2	40
0103	15A4	CE	17	FF	STRA.R2	CUR2
0104	15A7	17			RETC,UN	
0105	15A8	07	1C	EPR	LODI,R3	CUCD
0106	15AA	3F	03	9F	BSTA,UN	GO
0107	15AD	17			RETC,UN	

Numbers Game by Mike Herbach

* "NUMBERS" GAME MODIFIED FROM "KILOBAUD" FOR CENTRAL DATA SYSTEM
 * BY MIKE HERBACH

```

ERASE
DIM A(10).B(10)
N1=1
N2=2
N3=3
N4=4
N5=5
N6=6
N7=7
N8=8
N9=9
N10=10
PRINT 'THIS IS THE NUMBERS GAME'
INPUT 'DO YOU WANT INSTRUCTIONS? PUSH 1 FOR YES.'Y
IF Y<>1 GOTO 200
ERASE
PRINT 'I WILL THINK OF A NUMBER WITH THE AMOUNT OF DIGITS'
PRINT 'YOU ASK FOR. THEN YOU GUESS THAT MANY DIGITS AND'
PRINT 'ENTER EACH ONE WITH RETURN. THEN I WILL SCORE YOUR'
PRINT 'GUESS WITH TWO NUMBERS. THE FIRST NUMBER WILL BE THE'
PRINT 'NUMBER OF DIGITS YOU GUESSED CORRECTLY IN THE CORRECT'
PRINT 'POSITION. THE SECOND WILL BE THE NUMBER OF DIGITS THAT'
PRINT 'ARE THE CORRECT VALUE BUT IN THE WRONG POSITION.'
PRINT
PRINT 'ENTER A NUMBER LARGER THAN 9 TO QUIT.'
PRINT
200 INPUT 'HOW MANY DIGITS ? (1 TO 10)'N
IF N<1 GOTO 200
IF N>10 GOTO 200
FOR I=1 TO 10
A(I)=I-1
NEXT I
FOR I=1 TO N
K=INT(RND(10))+1
T=A(I)
A(I)=A(K)
A(K)=T
NEXT I
K=0
400 ERASE
PRINT 'TYPE #N DIGITS, NONE ALIKE'
410 GOTO (1100+N)
450 IF B(N1)>9 GOTO 800

```

```
I=0
J=0
K=K+1
FOR E=1 TO N
FOR F=1 TO N
IF A(E)=B(F) GOSUB 1000
NEXT F
NEXT E
IF I=N GOTO 900
PRINT / ' #I #J
GOTO 410
800 PRINT 'SORRY TO SEE YOU QUIT'
PRINT 'THE NUMBER WAS:'
FOR I=1 TO N
PRINT /#A(I)
NEXT I
PRINT
GOTO 910
900 PRINT 'CORRECT IN '#K' TRYS'
910 INPUT 'DO YOU WANT TO PLAY AGAIN? (1 OR 0)' ' X
IF X=1 GOTO 200
PRINT 'GOODBY. HOPE YOU HAD FUN.'
STOP
1000 IF E=F LET I=I+1
IF E<>F LET J=J+1
RETURN
1101 INPUT B(N1)
GOTO 450
1102 INPUT B(N1),B(N2)
GOTO 450
1103 INPUT B(N1),B(N2),B(N3)
GOTO 450
1104 INPUT B(N1),B(N2),B(N3),B(N4)
GOTO 450
1105 INPUT B(N1),B(N2),B(N3),B(N4),B(N5)
GOTO 450
1106 INPUT B(N1),B(N2),B(N3),B(N4),B(N5),B(N6)
GOTO 450
1107 INPUT B(N1),B(N2),B(N3),B(N4),B(N5),B(N6),B(N7)
GOTO 450
1108 INPUT B(N1),B(N2),B(N3),B(N4),B(N5),B(N6),B(N7),B(N8)
GOTO 450
1109 INPUT B(N1),B(N2),B(N3),B(N4),B(N5),B(N6),B(N7),B(N8),B(N9)
GOTO 450
1110 INPUT B(N1),B(N2),B(N3),B(N4),B(N5),B(N6),B(N7),B(N8),B(N9),B(N10)
GOTO 450
```

Memory Test

This program tests any memory block of your system. All you have to do to use it is load it in starting at address 1510H and set the two bytes labeled STARTT and ENDT to the high bytes of the starting and ending addresses. For example, if you wanted to test memory between 2000H and 5FFFH, you would change STARTT to 20H and ENDT to 60H. Then just execute at address 1510H, and the test will begin. If any errors are found, they are displayed address first, then the data written to the location, and finally the data read from the location.

0001	0000			PRNT		
0002	0000		*			
0003	0000		*			
0004	0000		*			
000F	0000		*	MEMORY TEST		
0006	0000		*			
0007	0000		*			
0008	0000		*			
0009	0000			EQ	EQU	0
0010	0000			GT	EQU	1
0011	0000			LT	EQU	2
0012	0000			UN	EQU	3
0013	0000			R0	EQU	0
0014	0000			R1	EQU	1
0015	0000			R2	EQU	2
0016	0000			R3	EQU	3
0017	0000			HXOT	EQU	006A
0018	0000			LFCR	EQU	0024
0019	0000			WCHR	EQU	0396
0020	0000			RETU	EQU	0083
0021	0000		*			
0022	0000		*			
0023	0000		*			
0024	1510			ORG		1510
0025	1510	1B 02		BCTR, UN	START	BRANCH OVER POINTERS
0026	1512	00		STARTT	RES	HIGH BYTE OF START ADDRESS
0027	1513	00		ENDT	RES	HIGH BYTE OF END ADDRESS+1
0028	1514	04 00		START	LODI, R0	0
0029	1516	93			LPSL	SETUP PSL
0030	1517	3F 00 24			BSTA, UN	LFCR
0031	151A	20		STRT	EORZ, R0	DO LINEFEED AT BEGINNING
0032	151B	09 75			LODR, R1	GET READY TO CLEAR LOW BYTE OF
0033	151D	C9 29			STRR, R1	STARTT
0034	151F	C8 28			STRR, R0	RAMPTR
0035	1521	05 FF			LODI, R1	RAMPTR+1
0036	1523	8F 01		WRT	ADDI, R1	ZERO LOW BYTE OF POINTER
0037	1525	01			LODZ, R1	FF
0038	1526	06 00			LODI, R2	SETUP STARTING DATA
0039	1528	CF F5 48		WSTL	STRA, R2	ADD 1 TO STARTING DATA
0040	152B	84 01		WRTL	ADDI, R0	1
0041	152D	98 02			BCFR, EQ	PUT IT IN R0 FOR USE
0042	152F	84 01			ADDI, R0	SETUP INDEX REGISTER
0043	1531	DA 75		WTSKP	BIRF, R2	0
0044	1533	72			REDD, R2	*RAMPTR, I
0045	1534	9E 00 83			BCFA, LT	1
0046	1537	0A 0F			LODR, R2	INCREMENT DATA
0047	1539	86 01			ADDI, R2	WTSKP
0048	153B	CA 0B			STRR, R2	IF ZERO DONT BRANCH
0049	153D	EE 15 13			COMA, R2	1
0050	1540	98 64			BCFR, EQ	DONT LET ZERO BE STORED
0051	1542	0A 4E			LODR, R2	DO 256 TIMES
0052	1544	CA 02			STRR, R2	READ THE KEYBOARD
0053	1546	1B 02			BCTR, UN	RETU
						RETURN IF KEY IS PRESSED
						INCREMENT POINTER
						1
						RAMPTR
						ENDT
						SEE IF END OF TEST AREA
						IF NCT, BRANCH
						STARTT
						SETUP FOR START OF TEST AREA
						RAMPTR
						RD
						START READ LOOP

POSSIBLE CONDITIONS

REGISTER DEFINITIONS

```

0054 1548 *
0055 1548 *
0056 1548 *
0057 1548 00 00 RAMPTR RES 2
0058 154A *
0059 154A *
0060 154A *
0061 154A 01 RD LODZ,R1 LOAD THE STARTING DATA VALUE
0062 154B C3 STRZ,R3 SAVE IT IN R3
0063 154C 06 00 RSTL LODI,R2 0 SETUP INDEX REGISTER
0064 154E 0E F5 48 RDLP LODA,R2 *RAMPTR,I GET NEXT DATA BYTE
0065 1551 E3 COMZ,R3 COMPARE TO WHAT IT SHOULD BE
0066 1552 B8 23 BSFR,EQ ERROR IF NOT THE SAME, GOTO ERROR
0067 1554 87 01 ADDI,R3 1 INCREMENT DATA
0068 1556 98 02 BCFR,EQ RDSKP DONT ALLOW ZERO AGAIN
0069 1558 87 01 ADDI,R3 1
0070 155A DA 72 RDSKP BIRR,R2 RDLP DO THIS LOOP 256 TIMES
0071 155C 72 REDD,R2 READ KEYBOARD
0072 155D 9E 00 83 BCFA,LT RETU RETURN IF KEY IS PRESSED
0073 1560 0A 66 LODR,R2 RAMPTR INCREMENT RAM POINTER
0074 1562 86 01 ADDI,R2 1
0075 1564 CA 62 STRR,R2 RAMPTR
0076 1566 EF 15 13 COMA,R2 ENDT SEE IF END OF TEST AREA
0077 1569 98 61 BCFR,EQ RSTL IF NOT, BRANCH
0078 156B 0E 15 12 LODA,R2 STARTT REDO THE WHOLE THING AGAIN
0079 156E CA 58 STRR,R2 RAMPTR
0080 1570 1F 15 23 BCTA,UN WRT
0081 1573 *
0082 1573 *
0083 1573 *
0084 1573 00 00 00 TMP0 RES 4
0085 1577 *
0086 1577 *
0087 1577 *
0088 1577 C8 7A ERROR STRR,R0 TMP0 SAVE THE REGISTERS
0089 1579 C9 79 STRR,R1 TMP0+1
0090 157B CA 78 STRR,R2 TMP0+2
0091 157D CB 77 STRR,R3 TMP0+3
0092 157F 0A 47 LODR,R2 RAMPTR LOAD THE HIGH BYTE
0093 1581 3F 00 6A BSTA,UN HXOT WRITE IT
0094 1584 0A 6F LODR,R2 TMP0+2 WRITE THE LOW BYTE
0095 1586 3F 00 6A BSTA,UN HXOT
0096 1589 07 20 LODI,R3 20 WRITE A SPACE
0097 158B 3F 03 96 BSTA,UN WCHR
0098 158E 0A 66 LODR,R2 TMP0+3 WRITE THE DATA WRITTEN
0099 1590 3F 00 6A BSTA,UN HXOT
0100 1593 07 20 LODI,R3 20 WRITE A SPACE
0101 1595 3F 03 96 BSTA,UN WCHR
0102 1598 0A 59 LODR,R2 TMP0 WRITE THE DATA READ
0103 159A 3F 00 6A BSTA,UN HXOT
0104 159D 3F 00 24 BSTA,UN LFCR
0105 15A0 08 51 LODR,R0 TMP0 RESTORE THE REGISTERS
0106 15A2 09 50 LODR,R1 TMP0+1
0107 15A4 0A 4F LODR,R2 TMP0+2
0108 15A6 0B 4E LODR,R3 TMP0+3
0109 15A8 17 RETC,UN RETURN

```

Central Data Corporation
 PO Box 2484, Station A
 1207 North Hagan
 Champaign, IL 61820
 (217) 359-8010