Questdata

Volume 3 Issue # 3 ©

REVERSI

by

Werner Cirsovius

Stimulated by the program 'Othello' (published in Byte, Vol.2,No.10),I translated the Basic program into Cosmac assembly language. The program – running on the Netronics' based ELF II – uses a 4K byte RAM and the Netronics Video interface.

Program organization

The body of the program, including the message area, occupies pages 02 to 07 (Hex 0200 to 0711).

Beneath these pages, it uses:

Page 00 – as a working page, holding I/O linkages, message pointers as well as the variable field for the game values

Page 01 – as utility page, holding the initialization routine and a lot of subroutines called by the game

Page 08 – as I/O pages, holding the driver for the serial I/O Page 09

Page 0F – as the stack page

Subroutine handling will be performed by RCA's Standard Call and Return Technique. The work area (page 00) is accessed by a short subroutine, labelled 'GETROT' (at location 0141).

Table 1 shows the Cosmac registers as they are used by the game.

Utility programs

1– Computation of piece location:
The game board consists of 8 x 8 locations, which is represented by an array called board(8,8). Because of the algorithm for inspecting all locations for adjacent pieces in the way:board(I+i,J+j), with i,j = –1,0,+1, it is necessary to expand to 10 x 10 locations.

Whenever accessing the board with pointers i,j (values 0–9), the following formula has to be computed:

Access Address = Base Address of Board + I + J*10 (1)

When calling the routine, I is stored on stack while J is in the machine's accumulator.

2– string output:
There are two entry points for this routine. One for a direct and the other for the indirect load. In the direct load, the routine fetches a two byte address following the call. This address points to the string. In the indirect load, the routine fetches a one byte page 00 vector following the call. This vector points to a two byte address to the string. The indirect load eases changing the messages if required. The string characters will be printed until a NUL character is detected.

3– Hex/Decimal conversion:
For the display of the scores, it is necessary to convert hex values to decimal in ASCII format. Fortunately the biggest number is 64, so conversion is done in the following way:

- count tens by subtracting 10 from the number until result is less than zero
- adjust tens and units by adding ASCII-offset
- blank tens if zero

After conversion, the resulting two characters will be stored in a string, pointed to indirect by the page 00 vector following the call.

4– Keyboard input:
This routine first prints a question mark to

indicate the input mode. It then gets characters from keyboard into a char- acter buffer until carriage return detected. If the limitation of 12 characters will be exceeded, the routine overwrites the last character with a c.r. and returns.

5- Match routine:
In some cases only two possible inputs are valid (as Y or N for yes and no). The vector following this routine call points to a pair of match characters. If the player types the first one the Cosmac flag DF will be set. Typing the second character resets DF. On any other char- acter the keyboard input will be requested until a match is found.

The program body

This part is divided into the following parts:

0200 - 0289	Initialization of the game (Note that no rules may be printed, i removed this option from the ori- ginal program)
028A - 0325	Computer selecting the move
0326 - 0381	Computer performs the move
0382 - 042D	Player performing his move
042E - 046A	End of game handler
046B - 04DD	Subroutine score and update
04DE - 0505	Subroutine test neighbour
0506 - 054C	Subroutine print board

First, the game initializes some arrays, the game board and player defined options (such as kind of piece, best strategy, etc.). In the selecting mode of computers' move, first all locations of the game board will be examined. It advances to next location, if a location is occupied or if an unoccupied location has no opponent.

Whenever the computer finds an opponent, it looks for the numbers of pieces to flip. If any piece to flip, the computer decides for the best move comparing current count to previous count. After examination, the selected move will be performed really by flipping opponents to own pieces.

The player's part checks valid move (such as unoccupied location, adjacent etc.). If the player inputs 0, computer asks for forfeiting the move. If so, player's move will be skipped. The last part of the body is the end handler, which prints the winner and asks for a new game.

Machine transfers control to an address pointed to in location 'USADR' (000D) with Cosmac's P = X = 0, if no game is requested. This transfer address will be normally the start address of the Monitor from Netronics or Quest.

As mentioned above, this assembler version is the translation of Basic statements into Cosmac machine code. As an example of this transla- tion, let us compare the Basic version to assem- bler of the short subroutine, which checks if a location has a neighbour (labelled L2620 at 04DE to 0505)

```
Line  Basic text          Line  Assembler

2620  FOR I1=-1 TO 1     L2620: LDI -1;PLO WORK
2630  FOR J1=-1 TO 1                PHI WORK
2640  IF A(I+I1,J+J1)=T2
      THEN 2710           L2640:  ,GETZ,A.0(I-1)
                            GLO WORK;ADD;STR X
                             (I ON STK)
                            INC PZ;GHI WORK;ADD
                             (J  IN ACCU)
                            ,ARRAY
                            ,GETZ A.0(T2-1)
                            LDN TP;SM
                            BZ L2710
2650  NEXT J1               GHI WORK
                            ADI 1;PHI WORK
                            SMI 2
                            BNZ L2640
2660 NEXT T1                LDI -1;INC WORK
                            PHI WORK;GLO WORK
                            SMI 2
                            BNZ L2640
2670 F1=0:RETURN            ADI 0          (DF=0)
2710 F1=1:RETURN         L2710:  ,RTS      (DF=1)
```

Notes: 1-'GETZ' is a Cosmac subroutine call to a small program, which fetches the content of page 00 vector following the call. On return, the page 00 vec- tor register PZ contains address of vector+1. Also this register is designated as Cosmac X register.

2-'ARRAY' is a Cosmac subroutine, which computes the formula (1). On return register TP points to game board location A(I+I1,J+J1).

How to bring up the game

If all machine code is loaded, turn on the Cosmac. Hit the Return key for determining the Baud value of the serial I/O device for full duplex. Hit the Line Feed key for half duplex. Now the screen of the input device will be cleared by printing the Form Feed character (Hex 0C). Then the machine prints the first message and awaits the first input, indicating the input mode by a question mark. Of course, a lot of things may be different from other Cosmac users, so here is a detailed list of locations, which may be changed to interface to other systems.

0000 – 0002	Long branch to game start (0100)
0003 – 0005	Long branch to serial input driver
0006 – 0008	Long branch to serial output driver
0009 – 000A	Top of stack (Default:0FFF, top of 4K byte RAM)
000B – 000C	Address of initialization routine for serial I/O
000D – 000E	Address of system Monitor (Default: F000 for the Netronics Monitor)
000F	Cancel code for deleting last character from input routine (Default 08 for backspace)
0010 – 0011	Decision match character (Default:N (No) and Y (Yes))
0012 – 0013	Piece match characters (Default:X and O)
0014 – 0016	Board characters (Default:X and O)
0017 – 003E	Text pointers for string output
003F – 004A	Pointers for hex to decimal conversion

To change text, simply load it anywhere in free memory and store the start address of string into adequate vector (0017 – 003E). Do it in the same way for the conversion pointer (003E – 004A).

For I/O handling the processor line Q will be used as output line, while EF4 is used for the input line. If the input line is negated as in the Quest Super Elf, change locations as follows:

LOC	OLD	NEW
085B	3F	37
085D	37	3F
0868	3F	37
086B	3F	37
087A	37	3F
08BB	3F	37
08C2	3F	37
08D3	37	3F

Table 1

Cosmac registers used by the game:

Register	Label	Used as
0	PCO	Entry program counter
0	COMP	Forfeiting flag
1	WORK	Work register
2	X	Stack pointer
3	PC	Main program counter
6	LINK	Link register for SCRT
8	ARP	Relative board pointer
10	SAV	Work register
11	TP	Game board pointer
12	CP	Input character pointer
13	PZ	Page zero pointer
14	SP	String pointer
15	AC	Scratch register

```
0000 ;              0001 ..
0000 ;              0002 ..*******************
0000 ;              0003 ..THE GAME OF REVERSI
0000 ;              0004 ..*******************
0000 ;              0005 ..
0000 ;              0006 ..ORIGINAL CODE WRITTEN IN BASIC
0000 ;              0007 ..BY RICHARD O.DUDA
0000 ;              0008 ..(GAME CALLED 'OTHELLO')
0000 ;              0009 ..PUBLISHED IN BYTE VOL.2,NO.10
0000 ;              0010 ..
0000 ;              0011 ..WRITTEN FOR THE CDP 1802 BY W.CIRSOVIUS
0000 ;              0012 ..
0000 ;              0013 ..
0000 ;              0014 ..
0000 ;              0015 ..***REGISTER ASSIGNMENT***
0000 ;              0016 ..
0000 ;              0017 PCO=0
0000 ;              0018 COMP=0
0000 ;              0019 WORK=1
0000 ;              0020 X=2
0000 ;              0021 PC=3
0000 ;              0022 LINK=6
0000 ;              0023 ARP=8
0000 ;              0024 SAV=10
0000 ;              0025 TP=11
0000 ;              0026 CP=12
0000 ;              0027 PZ=13
0000 ;              0028 SP=14
0000 ;              0029 AC=15
0000 ;              0030 ..
0000 ;              0031 ..***MACRO DEFINITION***
0000 ;              0032 ..
0000 ;              0033 LDO=#9D
0000 ;              0034 CALL=#D4
0000 ;              0035 RTS=#D5
0000 ;              0036 SETZ=#D7
0000 ;              0037 ARRAY=#D9
0000 ;              0038 DELAY=#DC
0000 ;              0039 ..
0000 ;              0040 ..***PAGE 00 SET UP***
0000 ;              0041 ..
0000 C00100:        0042          LBR BEGIN       ..GO START
0003 C00BA0;        0043 INPUT:   LBR CHARI       ..INPUT LINKAGE
0006 C00BFD;        0044 OUTPUT:  LBR CHARO       ..OUTPUT LINKAGE
0009 0FFF;          0045 STKPNT:  ,#0FFF          ..STACK
000B 0850;          0046          ,A(INITO)       ..INITIALIZE
000D F000;          0047 USADR:   ,#F000          ..MONITOR LINKAGE
000F 08;            0048 CANCEL:  ,#08            ..CTRL H
0010 4E59;          0049 MTCH1:   ,T'NY'          ..MATCH CHARACTERS
0012 584F;          0050 MTCH2:   ,T'XO'
0014 582E4F;        0051 D#:      ,T'X.O'         ..BOARD CHARACTERS
0017 ;              0052 ..
0017 ;              0053 ..***TEXT POINTER***
0017 ;              0054 ..
0017 054C;          0055 TXTP:    ,A(TXT1)
0019 0583;          0056          ,A(TXT2)
001B 05A5;          0057          ,A(TXT3)
001D 05C5;          0058          ,A(TXT4)
001F 05D2;          0059          ,A(TXT5)
0021 05DE;          0060          ,A(TXT6)
0023 05F2;          0061          ,A(TXT7)
0025 0602;          0062          ,A(TXT8)
0027 061F;          0063          ,A(TXT9)
0029 0634;          0064          ,A(TXT10)
002B 0652;          0065          ,A(TXT11)
002D 065E;          0066          ,A(TXT12)
002F 0677;          0067          ,A(TXT13)
0031 0690;          0068          ,A(TXT14)
0033 06AA;          0069          ,A(TXT15)
0035 06CD;          0070          ,A(TXT16)
```

```
0037 06DE;
0039 06E8;
003B 06F2;
003D 06FD;
003F ;
003F ;
003F ;
003F 05FC;
0041 05FE;
0043 060B;
0045 0698;
0047 06B3;
0049 06C1;
004B ;
004B ;
004B ;
004B ;
004C ;
004D ;
0050 ;
0051 ;
0055 ;
0056 ;
0056 ;
0056 ;
0056 ;
0062 ;
0062 ;
0062 ;
0062 ;
0063 ;
006A ;
006B ;
0072 ;
0073 ;
0074 ;
0075 ;
0076 ;
0077 ;
0078 ;
0079 ;
007A ;
007B ;
007C ;
007D ;
007E ;
007F ;
0080 ;
0081 ;
0082 ;
0082 ;
0082 ;
0082 ;
00E5 ;
00E6 ;
0100 ;
0100 ;
0100 ;
0100 F800BD;
0103 F809AD;
0106 4DB2;
0108 4DA2;
010A 4DB3;
010C 0DA3;
010E F820A4;
0111 F832A5;
0114 F841A7;
0117 F84BA9;
```

```
0071          ,A(TXT17)
0072          ,A(TXT18)
0073          ,A(TXT19)
0074          ,A(TXT20)
0075 ..
0076 ..***CONVERSION POINTER***
0077 ..
0078 TXTPD:   ,A(TXT71)
0079          ,A(TXT71+2)
0080          ,A(TXTB1)
0081          ,A(TXT141)
0082          ,A(TXT151)
0083          ,A(TXT152)
0084 ..
0085 ..***I/O SAVE AREA***
0086 ..
0087 BDRT:    ORG *+1          ..BAUD VALUE
0088 LINTST:  ORG *+1          ..CHARACTER COUNT
0089 SAVI:    ORG *+3          ..SAVE AREA
0090 SAVIE:   ORG *+1
0091 SAVO:    ORG *+4
0092 SAVOE:   ORG *+1
0093 ..
0094 ..***CHARACTER BUFFER***
0095 ..
0096 BUFF:    ORG *+12
0097 ..
0098 ..***VARIABLE FIELD***
0099 ..
0100 F2:      ORG *+1
0101 I4A:     ORG *+7
0102 I4E:     ORG *+1
0103 J4A:     ORG *+7
0104 J4E:     ORG *+1
0105 C1:      ORG *+1
0106 H1:      ORG *+1
0107 N1:      ORG *+1
0108 C:       ORG *+1
0109 H:       ORG *+1
0110 B1:      ORG *+1
0111 T1:      ORG *+1
0112 T2:      ORG *+1
0113 I:       ORG *+1
0114 J:       ORG *+1
0115 I3:      ORG *+1
0116 J3:      ORG *+1
0117 S1:      ORG *+1
0118 S2:      ORG *+1
0119 S3:      ORG *+1
0120 ..
0121 ..***GAME BOARD***
0122 ..
0123 TAB:     ORG *+99
0124 TABE:    ORG *+1
0125          PAGE
0126 ..
0127 ..***INITIALISATION***
0128 ..
0129 BEGIN:   LDI A.1(STKPNT);PHI PZ
0130          LDI A.0(STKPNT);PLO PZ
0131          LDA PZ;PHI X     ..GET STACK
0132          LDA PZ;PLO X
0133          LDA PZ;PHI PC    ..GET NEW PC
0134          LDN PZ;PLO PC
0135          LDI A.0(CALROT);PLO 4    ..LOAD REGS
0136          LDI A.0(RETROT);PLO 5
0137          LDI A.0(GETROT);PLO 7
0138          LDI A.0(ARGET);PLO 9
```

```
011A 90B4B5B7B9;
011F D3;
0120 ;
0120 ;
0120 ;
0120 BFE2;
0122 9673;
0124 8673;
0126 83A6;
0128 93B6;
012A 46B3;
012C 46A3;
012E 9F301F;
0131 ;
0131 ;
0131 ;
0131 D3;
0132 BFE260;
0135 96B3;
0137 86A3;
0139 72A6;
013B F0B6;
013D 9F3031;
0140 ;
0140 ;
0140 ;
0140 D3;
0141 F800BD;
0144 43AD;
0146 ED72;
0148 3040;
014A ;
014A ;
014A ;
014A ;
014A ;
014A ;
014A D3;
014B E22252;
014E FEFEF4;
0151 FE60F4;
0154 FC82AB;
0157 90BB;
0159 304A;
015B ;
015B ;
015B ;
015B 46BE;
015D 46AE;
015F 306C;
0161 46AE;
0163 F800BE;
0166 4EBF;
0168 0EAE;
016A 9FBE;
016C 4E3292;
016F D40006;
0172 306C;
0174 ;
0174 ;
0174 ;
0174 ;
0174 ;
0174 F8FFAF;
0177 0D;
0178 FF0A1F;
0178 3378;
017D FC3ABF;
```

```
0139          GHI PC0;PHI 4;PHI 5;PHI 7;PHI 9
0140          SEP PC           ..START!!
0141 ..
0142 ..***SCRT CALL HANDLER***
0143 ..
0144 CALROT:  PHI AC;SEX X
0145          GHI LINK;STXD    ..PUSH LINK
0146          GLO LINK;STXD
0147          GLO PC;PLO LINK  ..GET PC
0148          GHI PC;PHI LINK
0149          LDA LINK;PHI PC  ..GET ROUTINE
0150          LDA LINK;PLO PC
0151          GHI AC;BR CALROT-1
0152 ..
0153 ..***SCRT RETURN HANDLER***
0154 ..
0155          SEP PC
0156 RETROT:  PHI AC;SEX X;IRX
0157          GHI LINK;PHI PC  ..GET RETURN
0158          GLO LINK;PLO PC
0159          LDXA;PLO LINK    ..POP LINK
0160          LDX;PHI LINK
0161          GHI AC;BR RETROT-1
0162 ..
0163 ..***PAGE 00 FETCH HANDLER***
0164 ..
0165          SEP PC
0166 GETROT:  LDI A.1(F2);PHI PZ
0167          LDA PC;PLO PZ    ..GET POINTER
0168          SEX PZ;LDXA      ..GET CONTENT
0169          BR GETROT-1
0170 ..
0171 ..***ARRAY POINTER COMPUTATION***
0172 ..
0173 ..COMPUTING POINTER BY:
0174 ..BASE+I+J*10
0175 ..
0176          SEP PC
0177 ARGET:   SEX X;DEC X;STR X        ..J
0178          SHL;SHL;ADD      ..J*5
0179          SHL;IRX;ADD      ..J*10+I
0180          ADI A.0(TAB);PLO TP      ..ADD BASE
0181          GHI PZ;PHI TP
0182          BR ARGET-1
0183 ..
0184 ..***STRING OUTPUT***
0185 ..
0186 STRFIX:  LDA LINK;PHI SP  ..GET POINTER
0187          LDA LINK;PLO SP  ..OF MESSAGE
0188          BR STRNG1
0189 STRING:  LDA LINK;PLO SP  ..GET POINTER
0190          LDI A.1(TXTP);PHI SP
0191          LDA SP;PHI AC    ..GET ADDRESS
0192          LDN SP;PLO SP
0193          GHI AC;PHI SP
0194 STRNG1:  LDA SP;BZ STROUT         ..PRINT TILL ZERO
0195          ,CALL,A(OUTPUT)
0196          BR STRNG1
0197 ..
0198 ..***CONVERT HEX TO DECIMAL***
0199 ..
0200 ..MAX HEX IS 99
0201 ..
0202 CNVHTA:  LDI @-1;PLO AC
0203          LDN PZ           ..GET HEX
0204 CNV1:    SMI 10;INC AC    ..GET TENS
0205          BPZ CNV1
0206          ADI @T'0'+10;PHI AC      ..UNITS
```

```
01B0 8FCE;          0207          GLO AC;LSZ        ..NO LEADING ZERO
0182 FC10;          0208          ADI @T'0'-T' '
0184 FC20AF;        0209          ADI T' ';PLO AC
01B7 46AD;          0210          LDA LINK;PLO PZ ..GET DESTINATION
0189 4DBC;          0211          LDA PZ;PHI CP
01BB 0DAC;          0212          LDN PZ;PLO CP
01BD 8F5C1C;        0213          GLO AC;STR CP;INC CP     ..STORE DECIMAL
0190 9F5C;          0214          GHI AC;STR CP
0192 D5;            0215 STROUT:  ,RTS
0193 ;              0216 ..
0193 ;              0217 ..***READ A LINE FROM KEYBOARD***
0193 ;              0218 ..
0193 8CFF56;        0219 BSCOD:   GLO CP;SMI A.0(BUFF)     ..TEST BEGINNING
0196 32A6;          0220          BZ NXTIN
0198 2C30A6;        0221          DEC CP;BR NXTIN ..IF NOT,ADJUST
019B F856AC;        0222 LININ:   LDI A.0(BUFF);PLO CP     ..POINT TO BUFFER
019E F800BC;        0223          LDI A.1(BUFF);PHI CP
01A1 F83F;          0224          LDI T'?'          ..PRINT KEY
01A3 D40006;        0225          ,CALL,A(OUTPUT)
01A6 D40003;        0226 NXTIN:   ,CALL,A(INPUT)   ..GET A CHARACTER
01A9 D70E;          0227          ,GETZ,A.0(CANCEL-1)
01AB 9FF3;          0228          GHI AC;XOR        ..TEST CANCEL
01AD 3293;          0229          BZ BSCOD
01AF 9F5C1C;        0230          GHI AC;STR CP;INC CP     ..STORE CHARACTER
01B2 FF0D;          0231          SMI #0D           ..TEST END
01B4 32BF;          0232          BZ CRCOD
01B6 8CFF62;        0233          GLO CP;SMI A.0(F2)       ..TEST IF FULL
01B9 3AA6;          0234          BNZ NXTIN
01BB 2C;            0235          DEC CP            ..IF SO,CLOSE
01BC F80D5C;        0236          LDI #0D;STR CP
01BF D4015B;        0237 CRCOD:   ,CALL,A(STRFIX) ..CLOSE LINE
01C2 01FA;          0238          ,A(CRLF)
01C4 F856AC;        0239          LDI A.0(BUFF);PLO CP     ..SET BEGINNING
01C7 D5;            0240          ,RTS
01CB ;              0241 ..
01CB ;              0242 ..***MATCH ROUTINE***
01C8 ;              0243 ..
01CB D4019B;        0244 MATCH:   ,CALL,A(LININ)   ..GET INPUT
01CB 3B1C0C;        0245          SKP;INC CP;LDN CP
01CE FF20;          0246          SMI T' '          ..SKIP BLANKS
01D0 32CC;          0247          BZ *-4
01D2 06ADED;        0248          LDN LINK;PLO PZ;SEX PZ   ..GET POINTER
01D5 0CF3;          0249          LDN CP;XOR        ..TEST FIRST
01D7 32E1;          0250          BZ FIRST
01D9 1D0CF3;        0251          INC PZ;LDN CP;XOR        ..TEST SECOND
01DC 3AC8;          0252          BNZ MATCH         ..RE-READ IF NEITHER
01DE FF00CB;        0253          SMI 0;LSKP
01E1 FC0016;        0254 FIRST:   ADI 0;INC LINK
01E4 D5;            0255          ,RTS
01E5 ;              0256 ..
01E5 ;              0257 ..***FIX STRINGS***
01E5 ;              0258 ..
01E5 0D0A;          0259 BOARD:   ,#0D,#0A
01E7 20202020412042;0260          ,T'    A B C D E F G H'
01EE 2043204420452O;0260
01F5 462047204B;    0260
01FA 0D0A00;        0261 CRLF:    ,#0D,#0A,#00
01FD ;              0262 ..
01FD ;              0263 ..***STARTING THE GAME***
01FD ;              0264 ..
01FD ;              0265          PAGE
0200 D40161;        0266 MAIN:    ,CALL,A(STRING) ..GREETING
0203 17;            0267          ,A.0(TXTP)        ..ASK FOR WAIT
0204 D401C8;        0268          ,CALL,A(MATCH)
0207 10;            0269          ,A.0(MTCH1)
0208 D761;          0270          ,GETZ,A.0(F2-1)
020A 9DC7;          0271          ,LDO;LSNF         ..F2=0 IF NO
020C FB015D;        0272          LDI 1;STR PZ      ..F2=1 IF YES
020F 3B15;          0273          BNF L460
```

```
0211 D40161;        0274          ,CALL,A(STRING)
0214 19;            0275          ,A.0(TXTP+2)
0215 D40161;        0276 L460:    ,CALL,A(STRING) ..BEST STRATEGY?
0218 1B;            0277          ,A.0(TXTP+4)
0219 D401C8;        0278          ,CALL,A(MATCH)
021C 10;            0279          ,A.0(MTCH1)
021D D77F;          0280          ,GETZ,A.0(S2-1)
021F 9DC7;          0281          ,LDO;LSNF         ..S2=0 IF NO
0221 FB025D;        0282          LDI 2;STR PZ      ..S2=2 IF YES
0224 D774;          0283          ,GETZ,A.0(N1-1)
0226 FB0473;        0284          LDI 4;STXD        ..N1=4
0229 F67373;        0285          SHR;STXD;STXD     ..H1=C1=2
022C F673;          0286          SHR;STXD          ..SET ARRAY J4
022E 9D73;          0287          ,LDO;STXD         ..1,1,0,-1,-1,-1,0,1
0230 FBFF;          0288          LDI @-1
0232 737373;        0289          STXD;STXD;STXD
0235 9D73;          0290          ,LDO;STXD
0237 FB01;          0291          LDI 1
0239 7373;          0292          STXD;STXD
023B 737373;        0293          STXD;STXD;STXD    ..SET ARRAY I4
023E 9D73;          0294          ,LDO;STXD         ..0,-1,-1,-1,0,1,1,1
0240 FBFF;          0295          LDI @-1
0242 737373;        0296          STXD;STXD;STXD
0245 9D5D;          0297          ,LDO;STR PZ
0247 B0;            0298          PHI COMP          ..FLAG=0
0248 FB01A1;        0299          LDI 1;PLO WORK    ..W=+1
024B FBFFB1;        0300          LDI @-1;PHI WORK          ..B=-1
024E D7E4;          0301          ,GETZ,A.0(TABE-1)
0250 9D73;          0302 TABUP:   ,LDO;STXD         ..RESET GAME BOARD
0252 8DFF81;        0303          GLO PZ;SMI A.0(TAB-1)
0255 3A50;          0304          BNZ TABUP
0257 FBB9AD;        0305          LDI A.0(TAB+55);PLO PZ
025A 8173;          0306          GLO WORK;STXD     ..SET FOUR FIELDS
025C 915D;          0307          GHI WORK;STR PZ ..WITH COLOUR OF PIECES
025E FBAFAD;        0308          LDI A.0(TAB+45);PLO PZ
0261 9173;          0309          GHI WORK;STXD
0263 815D;          0310          GLO WORK;STR PZ
0265 D40161;        0311          ,CALL,A(STRING) ..X OR O?
0268 1D;            0312          ,A.0(TXTP+6)
0269 D401C8;        0313          ,CALL,A(MATCH)
026C 12;            0314          ,A.0(MTCH2)
026D D776;          0315          ,GETZ,A.0(H-1)
026F 3376;          0316          BDF HO            ..SKIP ON B
0271 9173;          0317          GHI WORK;STXD     ..H=R
0273 813079;        0318          GLO WORK;BR HO+3          ..C=W
0276 8173;          0319 HO:      GLO WORK;STXD     ..H=W
0278 915D;          0320          GHI WORK;STR PZ ..C=B
027A D40506;        0321          ,CALL,A(L3100)    ..PRINT BOARD
027D D40161;        0322          ,CALL,A(STRING)
0280 1F;            0323          ,A.0(TXTP+8)      ..WANNA START?
0281 D401C8;        0324          ,CALL,A(MATCH)
0284 10;            0325          ,A.0(MTCH1)
0285 3B91;          0326          BNF COMPS         ..MACHINE STARTS
0287 C00382;        0327          LBR HUMAN         ..HUMAN STARTS
028A ;              0328 ..
028A ;              0329 ..***MACHINE'S MOVE***
028A ;              0330 ..
028A D762;          0331 L1000:   ,GETZ,A.0(F2)     ..TEST IF WAIT
028C 3291;          0332          BZ COMPS
028E D4019B;        0333          ,CALL,A(LININ)
0291 D776BF;        0334 COMPS:   ,GETZ,A.0(C);PHI AC       ..GET COLOUR
0294 0DAF;          0335          LDN PZ;PLO AC
0296 D77D;          0336          ,GETZ,A.0(J3-1)
0298 9D7373;        0337          ,LDO;STXD;STXD    ..RESET LOOP COUNTS
029B FB01;          0338          LDI 1             ..SET LOOP VALUES
029D 7373;          0339          STXD;STXD
029F 8F73;          0340          GLO AC;STXD       ..SET COLOURS
02A1 9F73;          0341          GHI AC;STXD
02A3 FBFF5D;        0342          LDI @-1;STR PZ    ..SET FLAG
```

```
02A6 D77B;        0343 L1090:  ,GETZ,A.0(I)
02A8 520D;        0344         STR X;LDN PZ
02AA D9;          0345         ,ARRAY            ..GET POINTER(I,J)
02AB 083AFE;      0346         LDN TP;BNZ L1380          ..TEST IF OCCUPIED
02AE D404DE;      0347         ,CALL,A(L2620)  ..LOOK FOR NEIGHBOUR
02B1 3BFE;        0348         BNF L1380       ..NO,SKIP
02B3 93A0;        0349         GHI PC;PLO COMP
02B5 D4046B;      0350         ,CALL,A(L2820)  ..COUNT OPPONENTS
02B8 D77F;        0351         ,GETZ,A.0(S1)   ..ANY?
02BA 32FE;        0352         BZ L1380        ..NONE,SKIP
02BC D77B;        0353         ,GETZ,A.0(I)    ..LOOK FOR EDGE
02BE FF01;        0354         SMI 1
02C0 32C6;        0355         BZ L1220
02C2 FF07;        0356         SMI @8-1
02C4 3AC8;        0357         BNZ L1230
02C6 D77FF4;      0358 L1220:  ,GETZ,A.0(S1);ADD         ..THINK OF BONUS
02C9 2D5D;        0359         DEC PZ;STR PZ   ..FOR BEST STRATEGY
02CB D77C;        0360 L1230:  ,GETZ,A.0(J)
02CD FF01;        0361         SMI 1
02CF 32D5;        0362         BZ L1240
02D1 FF07;        0363         SMI @8-1
02D3 3ADA;        0364         BNZ L1260
02D5 D77FF4;      0365 L1240:  ,GETZ,A.0(S1);ADD
02D8 2D5D;        0366         DEC PZ;STR PZ
02DA D77BAF;      0367 L1260:  ,GETZ,A.0(B1);PLO AC     ..TEST FLAG
02DD FE33ED;      0368         SHL;BDF L1340
02E0 D77E;        0369         ,GETZ,A.0(S1-1)
02E2 8FF7;        0370         GLO AC;SM       ..TEST AGAINST NEW PIECES
02E4 3BED;        0371         BL L1340        ..NEW MOVE IS BETTER
02E6 3AFE;        0372         BNZ L1380       ..OLD MOVE IS BETTER
02E8 0274FE;      0373         LDN X;ADC;SHL   ..EQUAL,RANDOM DECISION
02EB 33FE;        0374         BDF L1380
02ED D77FAF;      0375 L1340:  ,GETZ,A.0(S1);PLO AC
02F0 D777;        0376         ,GETZ,A.0(B1-1)
02F2 8F5D;        0377         GLO AC;STR PZ   ..SET NEW MOVE TO CURRENT
.E
02F4 D77DBF;      0378         ,GETZ,A.0(I);PHI AC
02F7 4DAF;        0379         LDA PZ;PLO AC
02F9 9F5D;        0380         GHI AC;STR PZ   ..SAVE COORDINATES
02FB 1D8F5D;      0381         INC PZ;GLO AC;STR PZ
02FE D77C2D;      0382 L1380:  ,GETZ,A.0(J);DEC PZ
0301 FC015D;      0383         ADI 1;STR PZ    ..NOW TEST END OF LOOP
0304 FF09;        0384         SMI 9
0306 CA02A6;      0385         LBNZ L1090
0309 F80173;      0386         LDI 1;STXD
030C F45D;        0387         ADD;STR PZ      ..BUMP SECOND COUNT
030E FF09;        0388         SMI 9           ..TEST IF ALL CHECKED
0310 CA02A6;      0389         LBNZ L1090
0313 D77B;        0390         ,GETZ,A.0(B1)   ..ANY PIECE TO FLIP?
0315 321A;        0391         BZ L1420
0317 FE3B26;      0392         SHL;BNF L1480
031A D40161;      0393 L1420:  ,CALL,A(STRING) ..NO,FORFEIT MOVE
031D 21;          0394         ,A.0(TXTP+10)
031E 90;          0395         GHI COMP        ..TEST BREAK
031F CA042E;      0396         LBNZ L2190
0322 93B0;        0397         GHI PC;PHI COMP ..SET BREAK FLAG
0324 30B2;        0398         BR HUMAN        ..LET HUMAN PLAY
0326 ;            0399 ..
0326 ;            0400 ..***PERFORM THE MOVE***
0326 ;            0401 ..
0326 9DB0;        0402 L1480:  ,LDO;PHI COMP   ..RESET BREAK FLAG
0328 F83FAD;      0403         LDI A.0(TXTPD);PLO PZ
032B 4DBF;        0404         LDA PZ;PHI AC
032D 0DAF;        0405         LDN PZ;PLO AC
032F D77D;        0406         ,GETZ,A.0(I3)   ..CONVERT MOVE TO ASCII
0331 F930;        0407         ORI T'0'
0333 5F0D;        0408         STR AC;LDN PZ
0335 F94052;      0409         ORI @T'A'-1;STR X
0338 F841AD;      0410         LDI A.0(TXTPD+2);PLO PZ
033B 4DBF;        0411         LDA PZ;PHI AC
033D 0DAF;        0412         LDN PZ;PLO AC
033F 025F;        0413         LDN X;STR AC
0341 D40161;      0414         ,CALL,A(STRING) ..TELL MOVE
0344 23;          0415         ,A.0(TXTP+12)
0345 D77DBF;      0416         ,GETZ,A.0(I3);PHI AC
0348 0DAF;        0417         LDN PZ;PLO AC
034A D77B;        0418         ,GETZ,A.0(I)
034C 8F73;        0419         GLO AC;STXD     ..SET SELECTED MOVE
034E 9F5D;        0420         GHI AC;STR PZ
0350 9DA0;        0421         ,LDO;PLO COMP
0352 D4046B;      0422         ,CALL,A(L2820)  ..FLIP PIECES
0355 D77F52;      0423         ,GETZ,A.0(S1);STR X
0358 D772;        0424         ,GETZ,A.0(C1-1)
035A 02F4;        0425         LDN X;ADD       ..COMPUTE MACHINE'S SCORE
035C FC015D;      0426         ADI 1;STR PZ
035F D773;        0427         ,GETZ,A.0(H1-1)
0361 02F55D;      0428         LDN X;SD;STR PZ ..COMPUTE HUMAN'S SCORE
0364 D7752D;      0429         ,GETZ,A.0(N1);DEC PZ
0367 FC015D;      0430         ADI 1;STR PZ    ..BUMP PIECE COUNT
036A D77E;        0431         ,GETZ,A.0(S1-1)
036C D40174;      0432         ,CALL,A(CNVHTA)
036F 43;          0433         ,A.0(TXTPD+4)
0370 D40161;      0434         ,CALL,A(STRING) ..TELL FLIPPED PIECES
0373 25;          0435         ,A.0(TXTP+14)
0374 D40506;      0436         ,CALL,A(L3100)  ..OUTPUT BOARD
0377 D774;        0437         ,GETZ,A.0(H1)
0379 C2042E;      0438         LBZ L2190       ..TEST END OF GAME
037C 0DFF40;      0439         LDN PZ;SMI 64
037F C2042E;      0440         LBZ L2190
0382 ;            0441 ..
0382 ;            0442 ..***HUMAN'S MOVE***
0382 ;            0443 ..
0382 D776BF;      0444 HUMAN:  ,GETZ,A.0(C);PHI AC
0385 4D1D5D;      0445         LDA PZ;INC PZ;STR PZ    ..CHANGE COLOUR
0388 1D9F5D;      0446         INC PZ;GHI AC;STR PZ
038B D40161;      0447 L1720:  ,CALL,A(STRING) ..INPUT MOVE
038E 27;          0448         ,A.0(TXTP+16)
038F D4019B;      0449         ,CALL,A(LININ)
0392 D77B;        0450         ,GETZ,A.0(I)
0394 4CFF20;      0451         LDA CP;SMI T' '
0397 3294;        0452         BZ *-3          ..IGNORE BLANKS
0399 FF10;        0453         SMI @T'0'-T' '  ..TEST RANGE
039B 3B8B;        0454         BL L1720
039D FF09;        0455         SMI @T'8'-T'0'+1
039F 338B;        0456         BGE L1720
03A1 FC09AF;      0457         ADI @T'8'-T'0'+1;PLO AC ..CONVERT TO HEX
03A4 3AB9;        0458         BNZ L1820
03A6 D40161;      0459         ,CALL,A(STRING) ..O DETECTED,FORFEITING?
03A9 29;          0460         ,A.0(TXTP+18)
03AA D401C3;      0461         ,CALL,A(MATCH)
03AD 10;          0462         ,A.0(MTCH1)
03AE 3B8B;        0463         BNF L1720       ..NO,GET NEXT
03B0 90;          0464         GHI COMP        ..IF SO,TEST END
03B1 CA042E;      0465         LBNZ L2190
03B4 93B0;        0466         GHI PC;PHI COMP ..SET BREAK FLAG
03B6 C0028A;      0467         LBR L1000       ..GET MACHINE'S MOVE
03B9 4CFF0D;      0468 L1820:  LDA CP;SMI #0D  ..TEST CR
03BC 328B;        0469         BZ L1720
03BE FF13;        0470         SMI @T' '-#0D
03C0 32B9;        0471         BZ L1820        ..IGNORE BLANKS
03C2 FF0C;        0472         SMI @T','-T' '  ..TEST SEPARATOR
03C4 32B9;        0473         BZ L1820
03C6 FF15;        0474         SMI @T'A'-T','  ..TEST RANGE
03C8 3B8B;        0475         BL L1720
03CA FF08;        0476         SMI @T'H'-T'A'+1
03CC 338B;        0477         BGE L1720
```

```
03CE FC09;        0478          ADI @T'H'-T'A'+2          ..CONVERT TO HEX
03D0 738F5D;      0479          STXD;GLO AC;STR PZ        ..SET COORDINATES
03D3 521D0D;      0480          STR X;INC PZ;LDN P7
03D6 D9;          0481          ,ARRAY              ..GET POINTER(I,J)
03D7 0B32E0;      0482          LDN TP;BZ L1910     ..TEST IF EMPTY
03DA D40161;      0483          ,CALL,A(STRING)     ..NO,TELL IT
03DD 2B;          0484          ,A.0(TXTP+20)
03DE 308B;        0485          BR L1720
03E0 D404DE;      0486 L1910:   ,CALL,A(L2620)      ..TEST NEIGHBOUR
03E3 33EB;        0487          BDF L1970
03E5 D40161;      0488          ,CALL,A(STRING)     ..NO,TELL IT
03E8 2D;          0489          ,A.0(TXTP+22)
03E9 308B;        0490          BR L1720
03EB 93A0;        0491 L1970:   GHI PC;PLO COMP     ..SET FLAG
03ED D4046B;      0492          ,CALL,A(L2820)      ..GET OPPONENTS
03F0 D77F;        0493          ,GETZ,A.0(S1)       ..TEST IF ANY
03F2 3AFA;        0494          BNZ L2030
03F4 D40161;      0495          ,CALL,A(STRING)     ..NONE,TELL IT
03F7 2F;          0496          ,A.0(TXTP+24)
03F8 308B;        0497          BR L1720
03FA 9DB0;        0498 L2030:   ,LDO;PHI COMP
03FC D77E;        0499          ,GETZ,A.0(S1-1)
03FE D40174;      0500          ,CALL,A(CNVHTA)     ..CONVERT NUMBER
0401 45;          0501          ,A.0(TXTPD+6)
0402 D40161;      0502          ,CALL,A(STRING)     ..TELL NUMBER
0405 31;          0503          ,A.0(TXTP+26)
0406 9DA0;        0504          ,LDO;PLO COMP
0408 D4046B;      0505          ,CALL,A(L2820)      ..FLIP PIECES
040B D77F52;      0506          ,GETZ,A.0(S1);STR X
040E D773;        0507          ,GETZ,A.0(H1-1)
0410 02F4;        0508          LDN X;ADD           ..COMPUTE HUMAN'S SCORE
0412 FC015D;      0509          ADI 1;STR PZ
0415 D772;        0510          ,GETZ,A.0(C1-1)
0417 02F55D;      0511          LDN X;SD;STR PZ     ..COMPUTE MACHINE'S SCORE
041A D7752D;      0512          ,GETZ,A.0(N1);DEC PZ
041D FC015D;      0513          ADI 1;STR PZ        ..BUMP PIECE COUNT
0420 D40506;      0514          ,CALL,A(L3100)      ..OUTPUT BOARD
0423 D773;        0515          ,GETZ,A.0(C1)
0425 322E;        0516          BZ L2190            ..TEST END OF GAME
0427 1D0D;        0517          INC PZ;LDN PZ
0429 FF40;        0518          SMI 64
042B CA028A;      0519          LBNZ L1000
042E ;            0520 ..
042E ;            0521 ..***END OF GAME***
042E ;            0522 ..
042E D773;        0523 L2190:   ,GETZ,A.0(H1-1)
0430 D40174;      0524          ,CALL,A(CNVHTA)     ..CONVERT SCORES
0433 47;          0525          ,A.0(TXTPD+8)
0434 D772;        0526          ,GETZ,A.0(C1-1)
0436 D40174;      0527          ,CALL,A(CNVHTA)
0439 49;          0528          ,A.0(TXTPD+10)
043A D40161;      0529          ,CALL,A(STRING)     ..TELL TOTAL SCORES
043D 33;          0530          ,A.0(TXTP+28)
043E D773F5;      0531          ,GETZ,A.0(C1);SD          ..LOOK FOR WINNER
0441 324B;        0532          BZ L2290            ..AHA,A TIE
0443 3351;        0533          BPZ L2310           ..HUMAN WINS
0445 D40161;      0534          ,CALL,A(STRING)
0448 35;          0535          ,A.0(TXTP+30)
0449 3055;        0536          BR L2510
044B D40161;      0537 L2290:   ,CALL,A(STRING)
044E 37;          0538          ,A.0(TXTP+32)
044F 3055;        0539          BR L2510
0451 D40161;      0540 L2310:   ,CALL,A(STRING)
0454 39;          0541          ,A.0(TXTP+34)
0455 D40161;      0542 L2510:   ,CALL,A(STRING)     ..ASK FOR A NEW GAME
0458 3B;          0543          ,A.0(TXTP+36)
0459 D401CB;      0544          ,CALL,A(MATCH)
045C 10;          0545          ,A.0(MTCH1)
045D C30215;      0546          LBDF L460           ..OK
0460 D40161;      0547          ,CALL,A(STRING)     ..TELL BYE,BYE
0463 3D;          0548          ,A.0(TXTP+38)
0464 D70DB0;      0549          ,GETZ,A.0(USADR);PHI PC0          ..GET TRANSFE
.R ADDRESS
0467 0DA0;        0550          LDN PZ;PLO PC0
0469 D0E0;        0551          SEP PC0;SEX PC0     ..END WITH P=X=0
046B ;            0552 ..
046B ;            0553 ..***SUBROUTINE SCORE AND UPDATE***
046B ;            0554 ..
046B D77E;        0555 L2820:   ,GETZ,A.0(S1-1)
046D 9D5DBC;      0556          ,LDO;STR PZ;PHI CP        ..RESET SUM
0470 F863AC;      0557          LDI A.0(I4A);PLO CP       ..POINT TO CHECKER TA
.BLE
0473 4CA1;        0558 L2840:   LDA CP;PLO WORK
0475 8CFC07;      0559          GLO CP;ADI 7
0478 AD0DB1;      0560          PLO PZ;LDN PZ;PHI WORK
047B D77A;        0561          ,GETZ,A.0(I-1)      ..SET LOOP
047D 81F4A8;      0562          GLO WORK;ADD;PLO ARP
0480 521D;        0563          STR X;INC PZ
0482 91F4B8;      0564          GHI WORK;ADD;PHI ARP
0485 D9;          0565          ,ARRAY              ..GET POINTER(I,J)
0486 D7B0;        0566          ,GETZ,A.0(S3-1)
0488 9D5D;        0567          ,LDO;STR PZ         ..RESET SUM
048A D779;        0568          ,GETZ,A.0(T2-1)
048C 0BF3;        0569          LDN TP;XOR          ..LOOK FOR OPPONENT
048E 3AD8;        0570          BNZ L3070           ..NO,SKIP
0490 D781;        0571 L2910:   ,GETZ,A.0(S3)
0492 FC01;        0572          ADI 1
0494 2D5D;        0573          DEC PZ;STR PZ       ..BUMP COUNT
0496 E28852;      0574          SEX X;GLO ARP;STR X
0499 81F4A8;      0575          GLO WORK;ADD;PLO ARP      ..COMPUTE DIRECTION
049C 9852;        0576          GHI ARP;STR X       ..FOR FURTHER SEARCH
049E 91F4B8;      0577          GHI WORK;ADD;PHI ARP
04A1 885298;      0578          GLO ARP;STR X;GHI ARP
04A4 D9;          0579          ,ARRAY
04A5 D77B;        0580          ,GETZ,A.0(T1-1)
04A7 0B32D8;      0581          LDN TP;BZ L3070     ..TEST IF EMPTY
04AA F33A90;      0582          XOR;BNZ L2910       ..TEST OWN PIECE
04AD D78152;      0583          ,GETZ,A.0(S3);STR X
04B0 D77E;        0584          ,GETZ,A.0(S1-1)
04B2 02F45D;      0585          LDN X;ADD;STR PZ          ..TOTAL COUNT
04B5 803AD8;      0586          GLO COMP;BNZ L3070        ..TEST IF TO FLIP
04B8 D77BA8;      0587          ,GETZ,A.0(I);PLO ARP      ..GET COORDINATES
04B8 0DB8;        0588          LDN PZ;PHI ARP
04BD D781;        0589          ,GETZ,A.0(S3)
04BF AA1A;        0590          PLO SAV;INC SAV     ..GET COUNT
04C1 D779BA;      0591          ,GETZ,A.0(T1);PHI SAV
04C4 885298;      0592 L3030:   GLO ARP;STR X;GHI ARP
04C7 D9;          0593          ,ARRAY              ..GET VECTOR OF PIECE
04C8 9A5B;        0594          GHI SAV;STR TP      ..SET PIECE
04CA 8852;        0595          GLO ARP;STR X
04CC 81F4A8;      0596          GLO WORK;ADD;PLO ARP      ..COMPUTE NEW COORDIN
.ATES
04CF 9852;        0597          GHI ARP;STR X
04D1 91F4B8;      0598          GHI WORK;ADD;PHI ARP
04D4 2A8A;        0599          DEC SAV;GLO SAV     ..TEST END OF FLIPPING
04D6 3AC4;        0600          BNZ L3030
04D8 8CFF6B;      0601 L3070:   GLO CP;SMI A.0(J4A)       ..TEST END OF LOOP
04DB 3A73;        0602          BNZ L2840
04DD D5;          0603 L2710:   ,RTS
04DE ;            0604 ..
04DE ;            0605 ..***TEST NEIGHBOUR***
04DE ;            0606 ..
04DE F8FF;        0607 L2620:   LDI @-1
04E0 A1B1;        0608          PLO WORK;PHI WORK         ..SET LOOP
04E2 D77A;        0609 L2640:   ,GETZ,A.0(I-1)
04E4 81F452;      0610          GLO WORK;ADD;STR X
04E7 1D91F4;      0611          INC PZ;GHI WORK;ADD
```

```
04EA D9;                0612          ,ARRAY              ..POINT TO PIECE
04EB D779;              0613          ,GETZ,A.0(T2-1)
04ED 0BF7;              0614          LDN TP;SM           ..TEST OPPONENT
04EF 32DD;              0615          BZ L2710            ..YEAH,FOUND
04F1 91;                0616          GHI WORK
04F2 FC01B1;            0617          ADI 1;PHI WORK      ..CONTINUE LOOP
04F5 FF02;              0618          SMI 2
04F7 3AE2;              0619          BNZ L2640
04F9 F8FF11;            0620          LDI 0-1;INC WORK
04FC B1B1;              0621          PHI WORK;GLO WORK
04FE FF02;              0622          SMI 2
0500 CA04E2;            0623          LBNZ L2640          ..TEST TOTAL END
0503 FC00D5;            0624          ADI 0,RTS
0506 ;                  0625 ..
0506 ;                  0626 ..***PRINT BOARD***
0506 ;                  0627 ..
0506 D4015B;            0628 L3100:   ,CALL,A(STRFIX) ..PRINT HEAD
0509 01E5;              0629          ,A(BOARD)
050B F801A1;            0630          LDI 1;PLO WORK
050E F820;              0631 L3130:   LDI T' '
0510 D40006;            0632          ,CALL,A(OUTPUT) ..BLANK
0513 81F930;            0633          GLO WORK;ORI T'0'
0516 D40006;            0634          ,CALL,A(OUTPUT) ..PRINT #LINE
0519 F820;              0635          LDI T' '
051B D40006;            0636          ,CALL,A(OUTPUT)
051E F801B1;            0637          LDI 1;PHI WORK
0521 F820;              0638 L3150:   LDI T' '
0523 D40006;            0639          ,CALL,A(OUTPUT)
0526 815291;            0640          GLO WORK;STR X;GHI WORK
0529 D9;                0641          ,ARRAY              ..GET PIECE
052A EBF815;            0642          SEX TP;LDI A.0(D$+1)
052D F4AD0D;            0643          ADD;PLO PZ;LDN PZ         ..GET ASCII
0530 D40006;            0644          ,CALL,A(OUTPUT)
0533 91FC01;            0645          GHI WORK;ADI 1   ..TEST INNER LOOP
0536 B1FF09;            0646          PHI WORK;SMI 9
0539 3A21;              0647          BNZ L3150
053B D4015B;            0648          ,CALL,A(STRFIX) ..CLOSE LINE
053E 01FA;              0649          ,A(CRLF)
0540 1181;              0650          INC WORK;GLO WORK         ..TEST LOOP
0542 FF093A0E;          0651          SMI 9;BNZ L3130
0546 D4015B;            0652          ,CALL,A(STRFIX)
0549 01FA;              0653          ,A(CRLF)
054B D5;                0654          ,RTS
054C ;                  0655 ..
054C ;                  0656 ..***MESSAGES***
054C ;                  0657 ..
054C 0D0A202A2A2052;0658 TXT1:    ,#0D,#0A,T' ** REVERSI **'
0553 455645525334920;0658
055A 2A2A;              0658
055C 0D0A53484F554C;0659          ,#0D,#0A,T'SHOULD I WAIT'
0563 4420492057414 9;0659
056A 54;                0659
056B 0D0A4245464F52;0660          ,#0D,#0A,T'BEFORE MAKING MY MOVE',#00
0572 45204D414B494E;0660
0579 47204D59204D4F;0660
0580 564500;            0660
0583 4F4B2C494E5055;0661 TXT2:    ,T'OK,INPUT RETURN IF'
058A 542052455455 2;0661
0591 4E204946;          0661
0595 0D0A492053484F;0662          ,#0D,#0A,T'I SHOULD GO',#0D,#0A,#00
059C 554C4420474F0D;0662
05A3 0A00;              0662
05A5 53484F554C4420;0663 TXT3:    ,T'SHOULD I PLAY MY BEST',#0D,#0A
05AC 4920504C415920;0663
05B3 4D592042455354;0663
05BA 0D0A;              0663
05BC 53545241544547;0664          ,T'STRATEGY',#00
05C3 5900;              0664
05C5 57414E4E41205B;0665 TXT4:    ,T'WANNA X OR O',#00
05CC 204F52204F00;      0665
05D2 57414E4E412053;0666 TXT5:    ,T'WANNA START',#00
05D9 5441525400;        0666
05DE 4920464F524645;0667 TXT6:    ,T'I FORFEIT MY MOVE',#0D,#0A,#00
05E5 4954204D59204D;0667
05EC 4F56450D0A00;      0667
05F2 49204D4F564520;0668 TXT7:    ,T'I MOVE TO '
05F9 544F20;            0668
05FC 492C4A0D0A00;      0669 TXT71:  ,T'I,J',#0D,#0A,#00
0602 49224C4C204745;0670 TXT8:    ,T'I"LL GET '
0609 5420;              0670
060B 5858204F462059;0671 TXT81:   ,T'XX OF YOUR PIECES',#0D,#0A,#00
0612 4F555220504945;0671
0619 4345530D0A00;      0671
061F 594F5552204D4F;0672 TXT9:    ,T'YOUR MOVE--(ROW,COL)',#00
0626 56452D2D28524F;0672
062D 572C434F4C2900;0672
0634 41524520594F55;0673 TXT10:   ,T'ARE YOU FORFEITING',#0D,#0A
063B 20464F52464549;0673
0642 54494E470D0A;      0673
0648 594F5552205455;0674          ,T'YOUR TURN',#00
064F 524E00;            0674
0652 4F43435550494 5;0675 TXT11:  ,T'OCCUPIED!',#0D,#0A,#00
0659 44210D0A00;        0675
065E 594F5520415245;0676 TXT12:   ,T'YOU ARE NOT NEXT TO ME',#0D,#0A,#00
0665 204E4F54204E45;0676
066C 585420544F204D;0676
0673 450D0A00;          0676
0677 495420444F4553;0677 TXT13:   ,T'IT DOESN"T FLANK A ROW',#0D,#0A,#00
067E 4E22542046 4C41;0677
0685 4E4B204120524F;0677
068C 570D0A00;          0677
0690 594F5520474554;0678 TXT14:   ,T'YOU GET '
0697 20;                0678
0698 5858204F46204D;0679 TXT141:  ,T'XX OF MY PIECES',#0D,#0A,#00
069F 59205049454345;0679
06A6 530D0A00;          0679
06AA 594F5520484156;0680 TXT15:   ,T'YOU HAVE '
06B1 4520;              0680
06B3 585820414E4420;0681 TXT151:  ,T'XX AND I HAVE '
06BA 4920484156452 0;0681
06C1 5858205049454 3;0682 TXT152:  ,T'XX PIECES',#0D,#0A,#00
06C8 45530D0A00;        0682
06CD 4920574F4E2054;0683 TXT16:   ,T'I WON THAT ONE',#0D,#0A,#00
06D4 484154204F4E45;0683
06DB 0D0A00;            0683
06DE 412054945212 1;0684 TXT17:   ,T'A TIE!!',#0D,#0A,#00
06E5 0D0A00;            0684
06E8 594F5520574F4E;0685 TXT18:   ,T'YOU WON',#0D,#0A,#00
06EF 0D0A00;            0685
06F2 57414E54204147;0686 TXT19:   ,T'WANT AGAIN',#00
06F9 41494E00;          0686
06FD 5448414E4B5320;0687 TXT20:   ,T'THANKS FOR PLAYING',#0D,#0A,#00
0704 464F5220504C41;0687
070B 59494E470D0A00;0687
0712 ;                  0688          PAGE
0800 ;                  0689          ORG $+80
0850 ;                  0690 ..
0850 ;                  0691 ..***I/O INITIALISATION***
0850 ;                  0692 ..
0850 D74B;              0693 INIIO:   ,GETZ,A.0(LINTST-1)
0852 9D73;              0694          ,LDO;STXD         ..RESET COUNT
0854 AFA1;              0695          PLO AC;PLO WORK ..CLEAR REGS
0856 938C;              0696          GHI PC;PHI CP
0858 F891AC;            0697          LDI A.0(DELROT);PLO CP  ..LOAD ROUTINE
085B 3F5B375D;          0698          BN4 $;B4 $        ..WAIT FOR KEY
085F F803;              0699          LDI 3
```

```
0861 FF01;          0700 SHDEL:  SMI 1            ..DETERMINE BAUD VALUE
0863 3A61;          0701         BNZ SHDEL        ..COUNTING THE PULSE
0865 8F3A6B;        0702         GLO AC;BNZ WTN4
0868 3F6D1F;        0703         BN4 WTN4+2;INC AC
086B 3F72;          0704 WTN4:   BN4 DELRT        ..TEST END OF PULSE
086D 11FB07;        0705         INC WORK;LDI 7
0870 3061;          0706         BR SHDEL
0872 2121;          0707 DELRT:  DEC WORK;DEC WORK
0874 81;            0708         GLO WORK
0875 F901B1;        0709         ORI 1;PHI WORK
0878 DC0C;          0710         ,DELAY,12
087A 378091;        0711         B4 EV;GHI WORK   ..DETERMINE DUPLEX
087D FAFEB1;        0712         ANI #FE;PHI WORK
0880 DC26;          0713 EV:     ,DELAY,38
0882 915D;          0714         GHI WORK;STR PZ
0884 F80C;          0715         LDI #0C
0886 D40006;        0716         ,CALL,A(OUTPUT) ..CLEAR SCREEN
0889 C00200;        0717         LBR MAIN         ..ENTER GAME
088C ;              0718 ..
088C ;              0719 ..***I/O DELAY***
088C ;              0720 ..
088C DCDCDC;        0721         SEP CP;SEP CP;SEP CP
088F DCD3;          0722         SEP CP;SEP PC
0891 91F6A1;        0723 DELROT: GHI WORK;SHR;PLO WORK   ..HALF COUNT
0894 2143;          0724         DEC WORK;LDA PC
0896 FF013A96;      0725         SMI 1;BNZ *-2    ..COUNT DOWN VALUE
089A 8132BC;        0726         GLO WORK;BZ DELROT-5
089D 233094;        0727         DEC PC;BR DELROT+3
08A0 ;              0728 ..
08A0 ;              0729 ..***CHARACTER INPUT***
08A0 ;              0730 ..
08A0 D74F;          0731 CHARI:  ,GETZ,A.0(SAVIE-1)
08A2 9173;          0732         GHI WORK;STXD    ..PUSH REGS
08A4 8173;          0733         GLO WORK;STXD
08A6 9C73;          0734         GHI CP;STXD
08A8 8C5D;          0735         GLO CP;STR PZ
08AA 93BC;          0736         GHI PC;PHI CP
08AC F891AC;        0737         LDI A.0(DELROT);PLO CP   ..LOAD ROUTINE
08AF D74BB1;        0738         ,GETZ,A.0(BDRT);PHI WORK          ..GET BAUD VA
.LUE
08B2 F880BF;        0739 INSIN:  LDI #80;PHI AC
08B5 37B53FB7;      0740         B4 *;BN4 *       ..WAIT FOR KEY
08B9 DC02;          0741         ,DELAY,2         ..NOW ASSEMBLE CHARACTER
08BB 3FB7;          0742         BN4 *-4
08BD C491F6;        0743 INSKPP: NOP;GHI WORK;SHR
08C0 33C7;          0744         BDF *+7          ..TEST ECHO
08C2 3FC6;          0745         BN4 *+4
08C4 7B387A;        0746         SEQ;SKP;REQ      ..SET LINE
08C7 C4DC07;        0747         NOP,DELAY,7
08CA C4C4;          0748         NOP;NOP
08CC 9FF6BF;        0749         GHI AC;SHR;PHI AC        ..SHIFT BITS
08CF 33DB;          0750         BDF BYTRDY       ..END IF BIT OUT
08D1 F980;          0751         ORI #80
08D3 37BD;          0752         B4 INSKPP        ..BIT=0
08D5 BF30BE;        0753         PHI AC;BR INSKPP+1       ..BIT=1
08D8 7A32B2;        0754 BYTRDY: REQ;B7 INSIN     ..RE-READ ON NUL
08DB D70E;          0755         ,GETZ,A.0(CANCEL-1)
08DD 9FF3;          0756         GHI AC;XOR       ..TEST CANCEL
08DF 32EC;          0757         BZ IOUT
08E1 D74B;          0758         ,GETZ,A.0(LINTST-1)
08E3 9FFB0D;        0759         GHI AC;XRI #0D   ..TEST END
08E6 3AF7;          0760         BNZ LINCNT
08E8 9D5D;          0761         ,LDO;STR PZ      ..IF SO,CLEAR COUNT
08EA DC40;          0762         ,DELAY,64
08EC D74DAC;        0763 IOUT:   ,GETZ,A.0(SAVI);PLO CP   ..POP REGS
08EF 4DBC;          0764         LDA PZ;PHI CP
08F1 4DA1;          0765         LDA PZ;PLO WORK
08F3 0DB1;          0766         LDN PZ;PHI WORK
08F5 9FD5;          0767         GHI AC,RTS
08F7 9F;            0768 LINCNT: GHI AC
08F8 D40967;        0769         ,CALL,A(CHKCHR) ..BUMP COUNT
08FB 30EC;          0770         BR IOUT
08FD ;              0771 ..
08FD ;              0772 ..***CHARACTER OUTPUT***
08FD ;              0773 ..
08FD D40967;        0774 CHARO:  ,CALL,A(CHKCHR) ..BUMP COUNT
0900 D754;          0775         ,GETZ,A.0(SAVOE-1)
0902 9173;          0776         GHI WORK;STXD    ..PUSH REGS
0904 8173;          0777         GLO WORK;STXD
0906 9C73;          0778         GHI CP;STXD
0908 8C73;          0779         GLO CP;STXD
090A 8B5D;          0780         GLO TP;STR PZ
090C F891AC;        0781         LDI A.0(DELROT);PLO CP   ..LOAD ROUTINE
090F F808BC;        0782         LDI A.1(DELROT);PHI CP
0912 D74BB1;        0783         ,GETZ,A.0(BDRT);PHI WORK
0915 9FABE2;        0784         GHI AC;PLO TP;SEX X
0918 FB0DC6;        0785         XRI #0D;LSNZ     ..TEST CR
091B 9D5D;          0786         ,LDO;STR PZ      ..ZERO COUNT IF CR
091D 9FFB0A;        0787         GHI AC;XRI #0A   ..TEST LF
0920 3A25;          0788         BNZ *+5
0922 F88BCB;        0789         LDI #80+11;LSKP          ..IF SO,INSERT NUL C
.UNT
0925 F80BAF;        0790         LDI 11;PLO AC
0928 7B8B52;        0791         SEQ;GLO TP;STR X
092B DC07;          0792 NXTBIT: ,DELAY,7
092D 2FF5;          0793         DEC AC;SD        ..DO THE TIMING
092F F07652;        0794         LDX;SHRC;STR X
0932 3336;          0795         BDF *+4
0934 7BC87AC4;      0796         SEQ;LSKP;REQ;NOP         ..ACTIVATE LINE
0938 BFFA0FC4;      0797         GLO AC;ANI #0F;NOP
093C 3A2B;          0798         BNZ NXTBIT       ..TEST ALL BITS DONE
093E 8FFCFB;        0799         GLO AC;ADI #FB   ..TEST REMAINING NULS
0941 AF3B4C;        0800         PLO AC;BNF TSTHOM
0944 FF1B;          0801         SMI #1B
0946 324C;          0802         BZ TSTHOM
0948 9DAB;          0803         ,LDO;PLO TP
094A 302B;          0804         BR NXTBIT-3
094C 9FFB0C;        0805 TSTHOM: GHI AC;XRI #0C ..TEST FORM FEED
094F 3A5A;          0806         BNZ OOUT
0951 F890AF;        0807         LDI #90;PLO AC
0954 DC07;          0808         .DELAY,7
0956 2F8F;          0809         DEC AC;GLO AC    ..DELAY
0958 3A54;          0810         BNZ *-4
095A D751AB;        0811 OOUT:   ,GETZ,A.0(SAVO);PLO TP   ..POP REGS
095D 4DAC;          0812         LDA PZ;PLO CP
095F 4DBC;          0813         LDA PZ;PHI CP
0961 4DA1;          0814         LDA PZ;PLO WORK
0963 0DB1;          0815         LDN PZ;PHI WORK
0965 9FD5;          0816         GHI AC,RTS
0967 ;              0817 ..
0967 ;              0818 ..***BUMP COUNT***
0967 ;              0819 ..
0967 73;            0820 CHKCHR: STXD             ..PUSH CHARACTER
0968 D74C2D;        0821         ,GETZ,A.0(LINTST);DEC PZ
096B FC015D;        0822         ADI 1;STR PZ     ..BUMP
096E FF20;          0823         SMI 32           ..AND TEST
0970 3A7E;          0824         BNZ CHOUT
0972 9D5D;          0825         ,LDO;STR PZ
0974 FB0D;          0826         LDI #0D          ..IF LINE FULL,CLOSE IT
0976 D40900;        0827         .CALL,A(CHARO+3)
0979 F80A;          0828         LDI #0A
097B D40900;        0829         .CALL,A(CHARO+3)
097E 1202;          0830 CHOUT:  INC X;LDN X      ..POP CHARACTER
0980 D5;            0831         ,RTS
0981 ;              0832         END
```

Q*BUG

In this column, we will concentrate on the creation of additional two byte "Shorthand" commands similar to the "PR" command for "PRINT". We will also shorten some of the existing command words and move the printer driver routines to another location. This will free up 45 memory locations in the command table which we will use for the "Shorthand" commands.

By now, you are, hopefully, familar with the Statement command table which presently runs from location 0500 thru 06D2. A rather slow printer driver routine runs from location 06D3 thru 06FF.

Our first task is to move the printer driver routine to work page 0000. It will start at location 0050 and run thru location 007F.

For those with a monitor operating system with a block move function, this will be a simple operation:

New address:	0050
Start address of block to move	06D3
End address of block to move	06FF

This block move can also be made with a small Basic program such as:

```
10 A=(@0050)
20 B=(@06D3)
30 C=PEEK (B)
40 POKE (A,C)
50 B=B+1:A=A+1
60 IF B<(@0700) GOTO 30
```

In either case, after you have moved the routine, you must correct three branch addresses within the relocated routine. These are:

Location	Old byte	New byte
0062	E4	61
0067	F2	6F
006E	EE	6B

Fill location 006D, 006E, and 006F with "C4".

Now, change the following address locations to reflect the change in the location of the printer driver routine:

Location	Old address	New address
00E9	06D3	0050
0754	06F3	0070
12F8	06F3	0070

Incidentally, if you look at the printer driver routine, you will see that it actually consists of two separate routines. The second routine, which is now located at 0070 thru 007C, serves to pick up the address at location 00E9 and 00EA on work page 0000. Since we previously "froze" work page 0000 by eliminating the initialization routine at location 1800, we only have to change the work page static data.

Now, if you are running a Basic program which contained the command PLIST, Super will stuff "00 50" to location 358F and 3590 on work page 3500. This is the "Output Hook" for Supers' printer output and will now read "D4 00 50 D5". The PLIST routine will call this by doing a "D4 35 8F D5". Although our moving the printer driver routine opened up 45 bytes of memory, we will shorten some Statement names to free more room in the command table. Each new two byte Shorthand command requires four bytes in the command table and we will be adding 13 new commands.

The statement names we will shorten are:

Old	New
FIXED	FIX
SFMON	SF
FDMON	FD
PSAVE	SAVE
PLOAD	LOAD
DSAVE	D/S
DLOAD	D/L
PLIST	P/L
TOUT	T/O
POUT	P/O
TRACE	TR
RENUMBER	RE#

The shortening of these names will open an additional 26 bytes of memory in the command table. This gives us room for the new commands plus some room for future expansion. My choice of words, to shorten or assign a Shorthand command to, are strictly personal and you can follow my lead or change as you see fit.

The Shorthand commands I assigned are:

Statement	Shorthand
PRINT	PR
(Already exists)	
GOTO	GT
INPUT	IP

LIST	LI
GOSUB	GS
RETURN	RT
WAIT	WA
NEXT	NX
DATA	DA
READ	RD
NEW	NN
RUN	RR
BYE	BB
HELP	HH

(These are the words I use most often but you may have other favorites. You make the final decision)

I tried to keep each Shorthand command as meaningful as possible, and, at the same time, followed the constraint of not using two letters that are the same as the first two letters of another command (RE for RETURN would conflict with RESTORE). Finally, since I am a two fingered typist, I tried to keep the keys to be pressed as close together as possible.

In a previous column, I mentioned that several Statement tokens are unused. For the new command "HELP", I chose to use the first unused token "A9". For ease in finding the name in the command table, we will place it in the proper sequence in the table (the table is organized in token number sequence).

If you decide to make these changes, you will end up rewriting over a page and a half of the program. You really need a good CRT/terminal monitor operating system. If you do not have one, contact the folks at QUEST ELECTRONICS. They can supply you with a dandy and the cost is far less than the grief of punching in the changes with the hex keypad.

The command table entries at locations 0500 thru 0564 will not be changed. What follows is an annotated listing of the balance of the command table from location 0565 thru 06FF. In the case of a shortened name, I have enclosed the deleted part of the name in paranenthsis:

Location	Code	Comment
0565	64 46 49 D8 91	FIX(ED)
056A	25 50 4F 4B C5 92	POKE
0570	63 53 C6 93	SF(MON)
0574	63 46 C4 94	FD (MON)
0578	64 4D 45 CD 95	MEM
057D	67 44 45 46 49 4E D4 96	DEF INT
0585	65 53 41 56 C5 97	(P) SAVE
058B	65 4C 4F 41 C4 98	(P) LOAD
0591	26 44 45 46 55 D3 99	DEFUS
0598	24 45 4F D0 9A	EOP
059D	65 44 41 54 C1 9B	DATA
05A3	65 52 45 41 C4 9C	READ
05A9	28 52 45 53 54 4F 52 C5 9D	RESTORE
05B2	24 45 4F C4 9E	EOD
05B7	24 43 4C C4 9F	CLD
05BC	64 44 2F D3 A0	D/S(AVE)
05C1	64 44 2F CC A1	D/L (OAD)
05C6	66 45 4E 49 4E D4 A2	ENINT
05CD	67 44 49 53 49 4E D4 A3	DISINT
05D5	64 50 2F CC A4	P/L (IST)
05DA	64 49 2F CF A5	I/O
05DF	64 54 2F CF A6	T/O(UT)
05E4	63 54 D2 A7	TR (ACE)
05E8	65 43 41 4C CC A8	CALL
05EE	65 48 45 4C D0 A9	HELP
05F4	64 50 2F CF AA	P/O (UT)
05F9	64 4F 55 D4 AB	OUT
05FE	64 42 59 C5 AD	BYE
0603	65 45 58 49 D4 AE	EXIT
0609	64 52 45 A3 B1	RE# (NUMBER)
060E	04 53 49 CE D4	SIN
0613	04 43 4F D3 D5	COS
0618	02 A8 D6	(
061B	04 41 54 CE D8	ATN
0620	04 45 58 D0 D9	EXP
0625	04 4C 4F C7 DA	LOG
062A	04 53 42 D3 DB	SQR
062F	04 49 4E D4 DC	INT
0634	05 50 45 45 CB DD	PEEK
063A	04 41 42 D3 DE	ABS
063F	04 52 4E C4 DF	RND
0644	04 55 53 C2 E0	USR
0649	05 49 4E 55 CD E1	INUM
064F	05 46 4E 55 CD E3	FNUM
0655	04 41 53 C3 E4	ASC
065A	04 4C 45 CE E5	LEN
065F	04 53 47 CE E7	SGN
0664	04 49 4E D0 E9	INP
0669	03 50 C9 EB	PI
066D	05 43 48 52 A4 B9	CHR$
0673	05 4D 49 44 A4 BA	MID$
0679	02 DE BB	
067C	04 54 41 C2 BC	TAB
0681	03 3E BD BD	> =
0685	03 3C BD BE	< =
0689	03 3C BE BF	< >
068D	65 53 54 45 D0 C0	STEP
0693	63 54 CF C1	TO
0697	02 AC C2	,
069A	02 BB C3	;
069D	02 A9 C4	)
06A0	65 54 48 45 CE C5	THEN
06A6	02 BC C6	<
06A9	02 BE C7	>
06AC	02 AB C8	+
06AF	02 AD C9	–
06B2	02 AA CA	*
06B5	02 AF CB	/
06B8	02 BD CC	=
06BB	02 BA CD	:
06BE	63 47 D4 87	GT (GOTO)
06C2	63 49 D0 89	IP (INPUT)
06C6	63 4C C9 8A	LI (LIST)
06CA	63 47 D3 8B	GS(GOSUB)
06CE	63 52 D4 8C	RT (RETURN)

```
06D2    63 57 C1 8D          WA(WAIT)
06D6    63 4E D8 90          NX (NEXT)
06DA    63 44 C1 9B          DA(DATA)
06DE    63 52 C4 9C          RD (READ)
06E2    63 4E CE 82          NN (NEW)
06E6    63 52 D2 83          RR (RUN)
06EA    63 42 C2 AD          BB (BYE)
06EE    FF FF
06F0 to 06FF - fill with "FF"
```

When we previously established the "HELPP" routine, we used either the SFMON or FDMON word location in the command table for "HELPP". The above listing returns SFMON or FDMON to their proper place in the command table. We must now return the addresses for SFMON or FDMON in the execution table to their original state. Memory locations 0726 thru 0729 should be corrected, where necessary, to:

Location	Code
0726	0C
0727	AC
0728	0C
0729	A9

The "HELP" routine uses token A9 which "points" to location 0752. Change location 0752 in the execution table to 00 and location 0753 to 10. HELP is now a valid statement command executing at location 0010. You may have to change the byte at location 003F in the actual HELP routine to keep Super from splitting too many words. I am using "3D" and have only the word "PEEK" split. Experiment until you find a value that satisfies you. You can do this in the direct execution mode by POKEing the value you wish to try with:

POKE(@003F,??) (?? = your value) : HELP

Finally, make a new master Super program tape. Don't forget to include work page 0000 on your tape.

BAGELS

```
10 REM            BAGELS PROGRAM
20 REM
30 REM            Adapted by Fred Hannan
40 REM
50 REM
60 REM        Bagels is a simple but mind stimulating program that
70 REM        I have been playing since my Tiny Basic days. I have
80 REM        updated it for each version of Basic that I acquired
90 REM        but the program retains its simplicity.
100 REM
110 REM        I must confess that I did not write the original version,
120 REM        but there have been many versions published. The roots
130 REM        of my version are lost in antiquity.
140 DEFINT Z: CLS
150 PRINT TAB(20);"BAGELS"
160 PRINT TAB(20);"======"
170 PRINT
180 PRINT "I WILL THINK OF A THREE DIGIT NUMBER (100 TO 999),"
190 PRINT "THEN YOU TRY TO GUESS WHAT THE NUMBER IS."
200 PRINT
210 PRINT "FOR EACH CORRECT DIGIT IN THE RIGHT LOCATION,"
220 PRINT "I WILL PRINT 'FERMI'."
230 PRINT : PRINT "FOR EACH CORRECT DIGIT IN THE WRONG LOCATION, "
240 PRINT "I WILL PRINT 'PICO'."
250 PRINT : PRINT "IF NO DIGITS ARE CORRECT, I WILL PRINT 'BAGELS'."
260 INPUT "READY TO PLAY? IF SO, PRESS 'RETURN' KEY."X$
270 CLS
280 A=RND(9)+1
290 B=RND(10)
300 C=RND(10)
310 P=0
320 PRINT "PLEASE GUESS A THREE DIGIT NUMBER (100-999)."
330 GOTO 350
340 PRINT "YOUR GUESS (#";(P+1);")"
350 INPUT G
360 IF G>999 GOTO 320
370 IF G<100 GOTO 320
380 M=0:N=0:P=P+1:H=G/100
390 IF H=A THENM=M+1
400 IF H<>B GOTO 420
410 IF H=B THENN=N+1
420 IF H<>C GOTO 440
430 IF H=C THENN=N+1
440 I=G-(H*100)
450 I=I/10
460 IF I<>A GOTO 480
470 IF I=A THENN=N+1
480 IF I<>C GOTO 500
490 IF I=C THENN=N+1
500 IF I=B THENM=M+1
510 Z=G/10
520 J=G-(Z*10)
530 IF J<>A GOTO 550
540 IF J=A THENN=N+1
550 IF J<>B GOTO 570
560 IF J=B THENN=N+1
570 IF J=C THENM=M+1
580 IF M<3 GOTO 650
590 PRINT A;B;C;" IS CORRECT."
600 PRINT "CONGRATULATIONS!!! YOU
    GUESSED IT IN ";P;" TRIES."
610 PRINT : INPUT "PLAY AGAIN"Q$
620 IF Q$=" " GOTO 700
630 IF Q$<>"YES" GOTO 700
640 IF Q$="YES" GOTO 270
650 IF M<>0 PRINT "FERMI ";M;
    " PLACE(S)."
660 IF N<>0 PRINT "PICO ";N;
    " PLACE(S)."
670 IF M+N=0 PRINT "BAGELS"
680 PRINT
690 GOTO 340
700 PRINT "GOODBYE"
710 CLS
```

BEATLE SONGS

by

Don Stevens

Here are some Beatle songs written so that the Elf Super Sound Program can play them. They can be played using the equal tempered scale, but they sound better when played in a just scale. There are many possible just scales for any given key. By definition, a just scale in the key of C has the frequencies of C, D, E, F, G, A, B being proportional to 1, 9/8, 5/4, 4/3, 3/2, 5/3, 15/8. Five of the tones in the octave have not been specified. We choose a complete just scale with the frequencies of successive tones starting with the key tone being proportional to 1, 16/15, 9/8, 6/5, 5/4, 4/3, 45/32 3/2, 8/5, 5/3, 9/5, 15/8.

The Beatle's music sounds better in this just scale (or the proper key) because this is closer to what the Beatles created; they did not use (exactly) the tempered scale. The table gives divisor lists for this just scale in all 12 keys. The divisors for the key of D# (tone 4) are in the 4th row, the divisors for the key of A (tone 10) are in the 10th row, etc. "Eleanor Rigby" and "Obladi Oblada" sound best in the key of A and "Penny Lane" sounds best in the key of D#.

Table of Divisor Lists

1	34F7	31A7	2F14	2C23	2A5F	27B9	25AA	234F	211A	1FC7	1D6D	1C3F
2	363C	32D9	2FAB	2D32	2A5F	28AD	2622	242B	21E6	1FC7	1E82	1C3F
3	34F7	32D9	2FAB	2CB0	2A5F	27B9	2622	23C0	21E6	1FC7	1DCB	1C9A
4	35A0	31A7	2FAB	2CB0	29E5	27B9	253E	23C0	2184	1FC7	1DCB	1BEE
5	34F7	32D9	2F14	2D32	2A5F	27B9	25AA	234F	21E6	1FC7	1E21	1C3F
6	34F7	31A7	2FAB	2C23	2A5F	27B9	253E	234F	211A	1FC7	1DCB	1C3F
7	363C	32D9	2FAB	2DC3	2A5F	28AD	2622	23C0	21E6	1FC7	1E82	1C9A
8	35A0	32D9	2FAB	2CB0	2AE7	27B9	2622	23C0	2184	1FC7	1DCB	1C9A
9	363C	32D9	3036	2D32	2A5F	28AD	25AA	242B	21E6	1FC7	1E21	1C3F
10	34F7	32D9	2FAB	2D32	2A5F	27B9	2622	234F	21E6	1FC7	1DCB	1C3F
11	34F7	31A7	2FAB	2CB0	2A5F	27B9	253E	23C0	211A	1FC7	1DCB	1BEE
12	35A0	32D9	2FAB	2DC3	2AE7	28AD	2622	23C0	2252	1FC7	1E82	1C9A

Obladi Oblada

00	35	0D	01	02	35	0D	01	02	35	0D	01	02	35	0D	01	02	
10	35	0D	01	02	35	0F	15	0F	B4	0F	A4	0F	15	10	01	02	00
20	15	2C	01	0F	45	15	01	02	45	0D	01	02	45	0D	01	02	10
30	45	0D	01	02	45	0D	01	02	45	0F	35	0F	15	0F	B4	3A	20
40	01	2C	65	15	01	02	65	0D	01	02	65	0D	01	02	65	0D	30
50	01	02	65	0D	01	02	65	0F	45	0F	35	0F	45	0F	65	0F	40
60	01	04	B5	20	01	02	B5	0F	65	0F	45	0F	35	0E	01	02	50
70	35	0E	45	0F	35	0F	15	0F	45	0F	35	0F	15	0D	B4	30	60
80	01	0F	B4	0F	35	0F	65	1D	B4	0F	35	0F	65	1D	B4	0F	70
90	35	0F	65	2B	B5	2C	01	0F	65	1D	45	0F	35	0F	45	0F	80
A0	35	0F	15	0F	B4	42	01	0F	B4	0F	35	0F	65	1D	B4	0F	90
B0	35	0F	65	1D	B4	0F	35	0F	65	2B	B5	25	01	0F	65	1D	A0
C0	45	0F	35	0F	45	0F	35	0F	15	0F	B4	35	01	49	B4	0F	B0
D0	45	0F	65	0F	85	1D	65	0F	85	0F	B5	0F	01	0F	45	0F	
E0	85	1D	B4	1D	35	1D	01	4A	B4	0F	45	0F	65	0F	85	0F	
F0	01	0D	65	0F	85	0F	B5	0F	01	0D	45	0F	85	2C	01	0D	
00	B4	0F	35	0E	01	02	35	0D	01	02	35	0E	45	1D	35	1C	
10	01	02	35	0F	15	3A	01	40	85	0F	01	02	85	0C	01	02	
20	85	0C	B5	10	01	02	B5	0E	85	05	B5	1C	01	2C	85	0F	
30	65	0F	45	0F	35	0F	15	0F	B4	0F	01	FF	00				

Eleanor Rigby

55	2D	75	0B	85	0B	A5	16	85	16	75	16	55	0B	C4	11	
A4	85	B4	38	01	44	B4	0B	A4	0B	C4	0B	B4	16	54	22	
B4	0B	A4	0B	C4	0B	35	16	25	0B	C4	0B	25	16	C4	0B	
A4	0B	C4	16	A4	0B	B4	0B	A4	38	01	2D	B4	0B	A4	0B	
C4	0B	15	22	C4	16	01	0B	B4	0B	A4	0B	C4	0B	B4	16	
54	22	B4	0B	A4	0B	C4	0B	35	16	25	0B	C4	0B	25	16	
C4	0B	A4	0B	C4	16	A4	0B	B4	0B	A4	38	01	2D	B4	0B	
A4	0B	C4	0B	15	22	C4	16	01	0B	A4	16	B4	0B	A4	22	
C4	0B	B4	16	54	22	01	16	54	0B	55	22	C4	0B	A4	16	
B4	16	54	38	01	2D	A4	16	B4	0B	A4	22	C4	0B	B4	16	
54	22	01	16	54	0B	85	22	55	0B	C4	16	A4	16	B4	38	
01	FF	00														

Penny Lane

00	B4	0F	15	0F	35	0F	15	0F	B4	0F	A4	0F	B4	0F	A4	0F
10	B4	0F	64	0F	B4	0F	64	0F	44	2C	64	0F	B4	0F	15	0F
20	35	0F	15	0F	B4	0F	A4	0F	B4	0F	64	0F	B4	0F	94	5B
30	01	0F	64	0F	B4	0F	15	0E	01	02	15	0E	B4	0E	01	02
40	B4	16	01	02	B4	1C	15	0F	25	4A	01	0F	B4	0F	15	0F
50	25	1D	B4	0F	15	1D	01	80	B4	0F	15	0F	35	0F	15	0F
60	B4	0F	A4	0F	B4	0F	A4	0F	B4	0F	64	0F	B4	0F	64	0F
70	44	2C	64	0F	B4	0F	15	0F	35	0F	15	0F	B4	0F	A4	0F
80	B4	0F	64	0F	B4	0F	B4	0F	94	4A	01	0F	B4	0F	15	0E
90	01	02	15	0E	B4	0E	01	02	B4	0E	01	02	B4	0E	01	02
A0	B4	1D	15	0F	25	2C	01	2C	B4	0F	15	0F	25	1D	B4	0F
B0	15	1D	01	1D	15	0F	B4	0F	B4	1D	01	3B	15	0F	25	0F
C0	45	2C	25	0F	15	0F	25	0F	45	3B	25	0F	15	0F	B4	1D
D0	94	B4	01	76	35	0F	45	0F	65	2C	45	0F	35	0F	45	1D
E0	65	3B	45	0F	35	0F	15	1D	B4	60	01	FF	00			

QUESTDATA
P.O. Box 4430
Santa Clara, CA 95054

Publisher.....Quest Electronics
Editor.....Paul Messinger
QBUG Editor.....Fred Hannan
Proof Reading.....Judy Pitkin
Production.....John Larimer

COMBINATION

by

Gilbert Hemmer

This is a challenging game requiring only one page of memory. The computer "thinks" of a 4 digit, non-repeating, non-zero hex number. Your task is to determine the 4 digits, in their proper order. The computer will give clues to help determine the number.

After loading the program, start the game by placing the computer in the Run mode. Press Input and EE will be displayed when the computer is ready for your entry. Enter the first 2 digits, press Input and they will be displayed. Enter the next 2 digits and press Input again. They will be displayed for a short time and then the clue will be displayed. The upper half of the clue tells the number of digits which match and are in the same location in the number. The lower half of the clue tells how many of the other digits you chose are contained in the computer number. The challenging part is trying to determine which digits they are. Continue making guesses until the exact number is determined. With a correct guess, the Q light will come on and the number of guesses it took will be displayed. Press Input again for another game.

Here is what a sample guess might be like:

Computer # : 4A19

Your Guess : 1A93

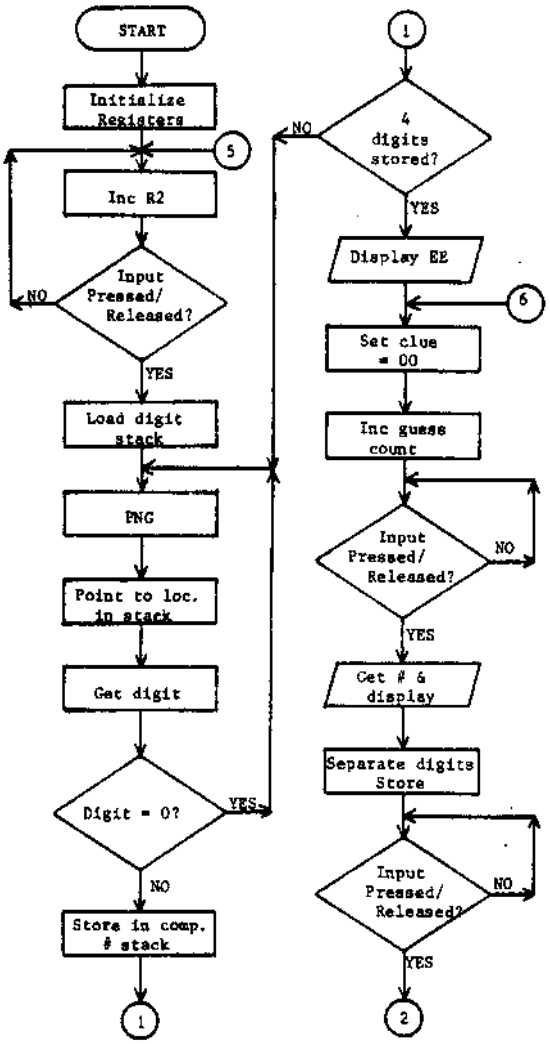
Clue display: 12

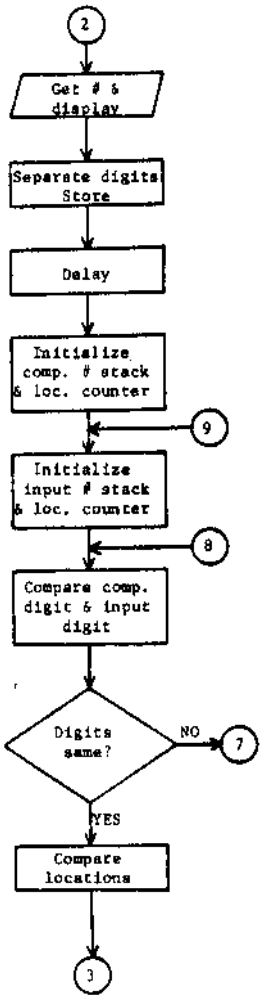
Program Operation

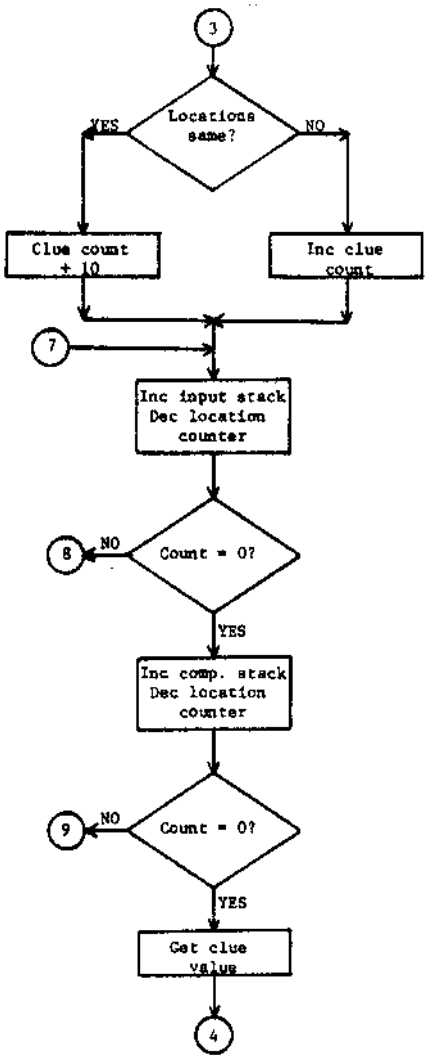
The major obstacle in writing this program was determining a way to select 4 non-repeating digits. To do this, I used the Pseudorandom Number Generator (PNG) described in Questdata (Volume #2, Issue #7). First, a digit stack in locations F0-FF is loaded with hex digits 00-0F. The PNG is initialized by continually incrementing R2 at the start of the program until Input is pressed. The low order number obtained from the PNG is in R3 and this number is OR'd with F0 so that R3 points to one of the digit stack locations. If the digit obtained from the stack is 00, the program goes back to the PNG to get another number. If the digit is not 00, it is stored in the computer number stack and 00 is stored in the digit stack so that digit cannot be selected again. This is repeated until the 4 computer digits are selected.

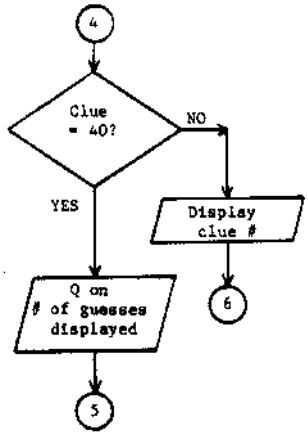
The player then makes his selection, 2 digits at a time, and enters them into the computer. The digits are separated and placed into the input number stack. The remainder of the program compares the input digits to each of the computer digits. If any matches occur, 10 is added to the clue register if the locations also match, or it is incremented by 01 if they are in different locations. The clue is displayed unless the number has been guessed at which time Q is turned on and the number of guesses it took is displayed.

Have fun trying to guess the combination.









Register Assignments

R0 - Main Program

R1 - Temporary Stack

R2 - PNG

R3 - Computer Digit Generation

R5 - Temporary Counter

R6 - Guess Count

R7 - Clue Storage

R8 - Computer Digit Location Counter

R9 - Input Digit Location Counter

RD - Input Number Stack

RE - Computer Number Stack

Addr	Code	Opcode	Comments
0000	F8 00	LDI	Initialize
0002	B1 B3	PHI	registers
0004	BD BE	PHI	
0006	F8 EF	LDI	Temporary
0008	A1	PLO	stack
0009	F8 E0	LDI	Computer #
000B	AE	PLO	stack
000C	F8 00	LDI	Guess
000E	A6	PLO	counter
000F	12	INC	Initialize
0010	82	GLO	PNG
0011	32 0F	BZ	(cannot be 00)
0013	3F 0F	BN4	Input pressed
0015	37 15	B4	released
0017	E3	SEX	Load
0018	F8 FF	LDI	digit
001A	A3	PLO	stack
001B	F8 0F	LDI	
001D	A5	PLO	
001E	85	GLO	
001F	73	STXD	
0020	32 25	BZ	
0022	25	DEC	
0023	30 1E	BR	
0025	7A	REQ	Reset Q
0026	F8 04	LDI	Set digit
0028	A8	PLO	count
0029	E1	SEX	
002A	92	GHI	Pseudorandom
002B	FE	SHL	Number
002C	51	STR	Generator
002D	FE	SHL	
002E	F3	XOR	
002F	FE	SHL	
0030	82	GLO	
0031	7E	SHLC	
0032	A2 A3	PLO	
0034	92	GHI	
0035	7E	SHLC	
0036	B2	PHI	
0037	83	GLO	Get low #
0038	F9 F0	ORI	Point to
003A	A3	PLO	digit stack
003B	03	LDN	Get digit
003C	32 29	BZ	Check if = 00
003E	5E	STR	Store in comp. # stack
003F	1E	INC	Inc stack
0040	F8 00	LDI	Load 00 into
0042	53	STR	digit stack

Addr	Code	Opcode	Comments
0043	28	DEC	Check if
0044	88	GLO	4 digits
0045	3A 29	BNZ	loaded
0047	F8 EE	LDI	Display
0049	51	STR	EE
004A	64	OUT	
004B	21	DEC	
004C	F8 00	LDI	Set clue
004E	A7	PLO	counter
004F	16	INC	Inc guess counter
0050	3F 50	BN4	Input pressed
0052	37 52	B4	released
0054	6C	INP	Store #
0055	64	OUT	Display #
0056	21	DEC	
0057	F6 F6	SHR	Separate high
0059	F6 F6	SHR	digit
005B	5E	STR	Store in input stack
005C	1E	INC	& inc
005D	01	LDN	Get #
005E	FA 0F	ANI	separate low digit
0060	5E	STR	store in input stack
0061	1E	INC	& inc
0062	3F 62	BN4	Input pressed
0064	37 64	B4	released
0066	6C	INP	Store #
0067	64	OUT	Display #
0068	21	DEC	
0069	F6 F6	SHR	Separate high
006B	F6 F6	SHR	digit
006D	5E	STR	Store in input stack
006E	1E	INC	& inc
006F	01	LDN	Get #
0070	FA 0F	ANI	separate low digit
0072	5E	STR	store in input stack
0073	F8 50	LDI	Delay
0075	B5	PHI	
0076	25	DEC	
0077	95	GHI	
0078	3A 76	BNZ	
007A	F8 E0	LDI	Computer #
007C	AE	PLO	stack
007D	F8 04	LDI	Computer # location
007F	A8	PLO	counter
0080	F8 E4	LDI	Input #
0082	AD	PLO	stack
0083	F8 04	LDI	Input # location
0085	A9	PLO	counter
0086	EE	SEX	

Addr	Code	Opcode	Comments
0087	0D	LDN	Compare
0088	F5	SD	digits
0089	32 97	BZ	
008B	1D	INC	Inc input # stack
008C	29	DEC	Check if all
008D	89	GLO	input #s
008E	3A 86	BNZ	compared
0090	1E	INC	Inc comp # stack
0091	28	DEC	Check if all
0092	88	GLO	computer #s
0093	3A 80	BNZ	compared
0095	30 A7	BR	Br to clue display
0097	E1	SEX	Check if
0098	88	GLO	digit
0099	51	STR	location
009A	89	GLO	same
009B	F5	SD	
009C	32 A1	BZ	
009E	17	INC	Inc clue counter
009F	30 8B	BR	Br
00A1	87	GLO	Add 10
00A2	FC 10	ADI	to clue
00A4	A7	PLO	counter
00A5	30 8B	BR	Br
00A7	E1	SEX	Check if
00A8	87	GLO	clue
00A9	51	STR	counter
00AA	FF 40	SMI	= 40
00AC	32 B2	BZ	
00AE	64	OUT	Display clue
00AF	21	DEC	count
00B0	30 4C	BR	Return for new guess
00B2	7B	SEQ	Turn Q on
00B3	86	GLO	Display
00B4	51	STR	guess
00B5	64	OUT	count
00B6	21	DEC	
00B7	30 06	BR	Return for new game