

```

10 CLS : PRINT TAB(10) "<<< A RATIONAL TIC-TAC-TOE GAME >>>"
20 /
30 / (C) BY D. D. HINRICHS 1979
40 / FOR THE TRS-80 WITH LEVEL II BASIC
50 /
60 PRINT : DEFINT B-Z
70 A$="" : PRINT "DO YOU WANT INSTRUCTIONS (Y OR N) ?"
80 A$=INKEY$ : IF A$="" GOTO 80
90 IF A$="Y" GOTO 100 ELSE 310
100 CLS
110 PRINT "THIS PROGRAM PLAYS A VERY AGGRESSIVE GAME OF " ;
120 PRINT "TIC-TAC-TOE. "
130 PRINT "IF YOU MAKE ANY MISTAKE AT ALL, THE COMPUTER WILL " ;
140 PRINT "WIN THE GAME. "
150 PRINT "ONLY IF YOU PLAY A PERFECT GAME WILL YOU BE ABLE " ;
160 PRINT "TO GET A DRAW. "
170 PRINT "TO MAKE YOUR MOVE, PRESS THE DIGIT (1 THROUGH 9) " ;
180 PRINT "THAT"
190 PRINT "REPRESENTS THE BOARD CELL TO WHICH YOU WISH TO MOVE. "
200 PRINT "YOUR ENTRY WILL BE ADDED TO THE BOARD, AND THE " ;
210 PRINT "COMPUTER"
220 PRINT "WILL CALCULATE ITS RESPONSE. AT THE START, THE " ;
230 PRINT "COMPUTER"
240 PRINT "WILL RANDOMLY CHOOSE A CENTER, CORNER, OR SIDE " ;
250 PRINT "OPENING MOVE. "
260 PRINT
270 PRINT TAB(10) "COMPUTER MOVES ARE: X"
280 PRINT TAB(10) "YOUR MOVES ARE: O"
290 PRINT
300 INPUT "PRESS 'ENTER' TO START THE GAME" ; A$ : CLS
310 RANDOM / RANDOMIZE RND SEED
320 DATA 2,8,6,4,7,3,0,9,0,0, 3,7,6,9,4,7,3,1,4,9
330 DATA 6,4,8,2,1,9,0,7,0,0, 7,3,8,9,2,1,9,3,7,2
340 DATA 3,7,1,2,9,9,1,8,2,7, 1,9,8,2,4,6,0,3,0,0
350 DATA 1,9,7,4,3,9,1,6,3,4, 7,3,6,4,2,8,0,1,0,0
360 DATA 5,8,4,6,7,3,0,9,0,0, 5,8,9,1,4,6,0,7,0,0
370 DATA 5,8,3,1,7,1,3,5,9,8, 3,1,9,6,4,0,7,0,8,0
380 DATA 5,8,3,1,7,3,1,5,7,8, 1,3,5,8,9,0,0,0,0,0
390 DATA 1,5,7,4,6,3,6,5,7,4, 3,1,5,8,7,0,0,0,0,0
400 DATA 5,9,7,4,3,4,7,5,6,9, 7,4,9,5,8,9,5,7,8,4
410 DATA 3,2,9,5,6,5,9,2,8,3, 3,2,5,7,9,5,9,3,2,7
420 DATA 9,5,3,2,6,2,3,5,8,9, 7,4,5,3,9,3,2,5,9,7
430 DATA 3,2,7,4,5,7,4,3,5,2
440 / ENTRY POINT TO INITIALIZE FOR A NEW GAME
450 RESTORE / RESET DATA POINTER TO START OF DATA
460 F=INT(RND(0)*2) / SET FLAG TO 0 OR 1
470 E=0 : G=1 : H=0 : C1=10 : C2=10 : N=0
480 GOSUB 1540 / DRAW BOARD
490 FOR I=1 TO 9 : B(I)=0 : NEXT I / CLEAR TEST BOARD
500 C=INT(RND(0)*3) / SELECT INITIAL COMPUTER MOVE (0, 1, 2)
510 IF C=0 THEN C=5 / CENTER OPENING GAME
520 U=10-C
530 GOSUB 1490 / DISPLAY COMPUTER MOVE, GET 1ST USER MOVE
540 IF C=5 GOTO 1330 / CENTER OPENING GAME
550 IF C=2 GOTO 1070 / SIDE OPENING GAME
560 IF U=5 GOTO 790 / CORNER GAME, 1ST USER MOVE = 5
570 R=10*U+140 / CORNER GAME, 1ST USER MOVE NOT 5
580 IF U>5 THEN R=R-10
590 E=1 / REENTRY POINT FOR 5-UNIT RESPONSES

```

```

600 IF F=1 THEN R=R+5
610 F=1 REENTRY POINT FOR 7-UNIT RESPONSES
620 IF R=0 GOTO 660
630 FOR I=1 TO R
640 READ C INCREMENT DATA POINTER TO 1ST RESPONSE
650 NEXT I
660 REENTRY POINT FOR LOOP TO SELECT COMPUTER RESPONSES
670 F=(F-1)*(F-1) FLIP FLAG
680 READ C SELECT COMPUTER RESPONSE FROM LIBRARY
690 C2=C1 : C1=C : N=N+1
700 IF C=0 OR C=U GOTO 660 SPACER OR MATCH, TRY AGAIN
710 B(C)=1 : P=C PUT COMPUTER MOVE INTO CELL
720 IF C2=0 OR C=H GOTO 940 DRAW GAME
730 IF F=1 GOTO 1460 COMPUTER WINS
740 IF E=0 GOTO 760
750 IF N=5 GOTO 1460 COMPUTER WINS
760 GOSUB 1750 : GOSUB 1700 UPDATE BOARD
770 GOSUB 1370 ACCEPT NEXT USER MOVE
780 GOTO 660 LOOP TO SELECT NEXT RESPONSE
790 CORNER OPENING GAME, FIRST USER MOVE WAS 5
800 U=1 : GOSUB 1490 COMPUTER MOVE 9, GET 2ND USER MOVE
810 IF U=3 OR U=7 GOTO 1030
820 IF U=4 OR U=8 THEN G=0
830 GOSUB 1490 CALCULATE RESPONSE, GET 3RD MOVE
840 IF G=0 THEN C=3 ELSE C=7 SET UP TRIAL COMPUTER MOVES
850 IF C<>U GOTO 1460 TRIAL CORRECT, COMPUTER WIN
860 GOSUB 1490 OTHERWISE GET 4TH USER MOVE
870 IF G=0 GOTO 910
880 IF B(2)=0 THEN C=2 ELSE C=6 SET UP TRIAL COMPUTER MOVES
890 IF B(C)=0 GOTO 1460 TRIAL CORRECT, COMPUTER WIN
900 GOTO 930
910 IF B(4)=0 THEN C=4 ELSE C=8 SET UP TRIAL COMPUTER MOVES
920 IF B(C)=0 GOTO 1460 TRIAL CORRECT, COMPUTER WIN
930 C=10-U : P=C FINAL MOVE FOR DRAW GAME
940 GOSUB 1750 : GOSUB 1700 FINAL BOARD UPDATE
950 PRINT@ 896, "CONGRATULATIONS - YOU GOT A DRAW THAT GAME"
960 A$="" :PRINT"DO YOU WANT TO PLAY ANOTHER GAME (Y OR N) ?";
970 A$=INKEY$ : IF A$="" GOTO 970 ELSE CLS
980 IF A$="Y" GOTO 440
990 PRINT "SO LONG UNTIL NEXT TIME THEN"
1000 PRINT
1010 END
1020 CORNER OPENING GAME, 1ST USER MOVE 5, 2ND MOVE 3 OR 7
1030 IF U=3 THEN C0=4 ELSE C0=2 SET UP TRIAL COMPUTER MOVES
1040 GOSUB 1490 CALCULATE RESPONSE, GET 3RD USER MOVE
1050 C=C0 : IF U=C THEN C=C+4 ADJUST TRIAL COMPUTER MOVE
1060 GOTO 1460 COMPUTER WINS
1070 SIDE OPENING GAME REENTRY POINT
1080 IF U=8 GOTO 1150 1ST USER MOVE IS 8
1090 R=10*U+60 FIND RESPONSES FOR SIDE GAME
1100 IF U=1 THEN R=80
1110 IF INT(U/2)=U/2 GOTO 590 RETURN TO SELECT RESPONSES
1120 IF U>6 THEN E=1
1130 GOTO 610 RETURN TO SELECT RESPONSES
1140 SIDE OPENING GAME, 1ST USER MOVE = 8
1150 B(9)=1 : P=681 COMPUTER RESPONSE IS 9
1160 GOSUB 1700 : GOSUB 1370 UPDATE, GET 2ND USER MOVE
1170 IF U<4 GOTO 1230
1180 IF U>5 GOTO 1270
1190 B(3)=1 : P=41 COMPUTER RESPONSE IS 3

```

```

1200 GOSUB 1700 : GOSUB 1370 UPDATE, GET 3RD USER MOVE
1210 IF U=1 THEN C=6 ELSE C=1 COMPUTER RESPONSE IS 1 OR 6
1220 GOTO 1460 COMPUTER WINS
1230 SIDE OPENING GAME, 1ST USER MOVE = 8, 2ND = 3 OR 1
1240 F=0 : H=6
1250 IF U=1 THEN F=1 : H=4 SET UP LAST 5 RESPONSES
1260 R=140 : GOTO 590 RETURN TO SELECT RESPONSES
1270 SIDE OPENING GAME, 1ST USER MOVE = 8, 2ND = 6 OR 7
1280 B(1)=1 : P=13 COMPUTER RESPONSE IS 1
1290 GOSUB 1700 : GOSUB 1370 UPDATE, GET 3RD USER MOVE
1300 IF U=3 THEN C=5 ELSE C=3 COMPUTER RESPONSE IS 3 OR 5
1310 GOTO 1460 COMPUTER WINS
1320 CENTER OPENING GAME REENTRY POINT
1330 R=U*10-10 FIND RESPONSES
1340 IF U>5 THEN R=R-10
1350 IF INT(U/2)=U/2 GOTO 590 ELSE 610 RETURN, GET RESPONSES
1360 SUBROUTINE TO ACCEPT AND CHECK NEW USER MOVE
1370 U=0 : PRINT@ 896, "YOUR MOVE IS ?";
1380 U=VAL(INKEY$) : IF U=0 GOTO 1390
1390 PRINT@ 896, "
1400 IF B(U)=1 GOTO 1440 ILLEGAL ENTRY
1410 B(U)=1 : P=U
1420 GOSUB 1750 : GOSUB 1650 UPDATE BOARD FOR USER MOVE
1430 RETURN
1440 PRINT "YOUR MOVE IS ILLEGAL. TRY AGAIN" : GOTO 1370
1450 ROUTINE FOR COMPUTER WINNING THE GAME
1460 P=C : GOSUB 1750 : GOSUB 1700
1470 PRINT@ 906, "***** I WON *****"
1480 GOTO 960
1490 CALCULATE & DISPLAY COMPUTER MOVE, GET USER MOVE
1500 C=10-U : B(C)=1 : P=C
1510 GOSUB 1750 : GOSUB 1700 : GOSUB 1370
1520 RETURN
1530 PRINT BIG BOARD
1540 CLS : PRINT@ 202, STRING$(41,CHR$(176))
1550 PRINT@ 522, STRING$(41,CHR$(176))
1560 FOR I=0 TO 12 : J=I*64+23
1570 PRINT@ J, CHR$(191);
1580 PRINT@ J+14, CHR$(191);
1590 NEXT I
1600 PRINT@ 20, 1 ; : PRINT@ 34, 2 ; : PRINT@ 48, 3 ;
1610 PRINT@ 276, 4 ; : PRINT@ 290, 5 ; : PRINT@ 304, 6 ;
1620 PRINT@ 596, 7 ; : PRINT@ 610, 8 ; : PRINT@ 624, 9 ;
1630 RETURN
1640 PRINT BIG 'O'
1650 PRINT@ P, CHR$(184); CHR$(135); STRING$(3,CHR$(131));
1660 PRINT@ P+64, CHR$(191); " "; CHR$(191); CHR$(139); CHR$(180);
1670 PRINT@ P+128, CHR$(139); CHR$(180); STRING$(3,CHR$(176));
1680 RETURN CHR$(184); CHR$(135);
1690 PRINT BIG 'X'
1700 PRINT@ P,CHR$(143); CHR$(176); " "; CHR$(176); CHR$(143);
1710 PRINT@ P+64, " "; CHR$(179); CHR$(140); CHR$(179); " ";
1720 PRINT@ P+128, CHR$(188); CHR$(131); " "; CHR$(131);
1730 RETURN CHR$(188);
1740 SET CELL POSITION TO PRINT BIG 'X' OR BIG 'O'
1750 ON P GOTO 1760,1770,1780,1790,1800,1810,1820,1830,1840
1760 P=13 : RETURN
1770 P=27 : RETURN
1780 P=41 : RETURN
1790 P=333 : RETURN
1800 P=347 : RETURN
1810 P=361 : RETURN
1820 P=653 : RETURN
1830 P=667 : RETURN
1840 P=681 : RETURN

```



MOBILE HOMES, INC.

EXECUTIVE OFFICE • 912 FIRST WICHITA NATIONAL BANK BUILDING • P. O. BOX 5003
PHONE 817/723-5523 • WICHITA FALLS, TEXAS 76307

J. WESLEY B. TAYLOR
Controller

15 May 1979
Mr. Gordon Lloyd, Publisher
TRS-80 Users Group Newsletter
7554 Southgate Road
Fayetteville, NC 28304

Re: LEVEL I Magazine on cassette tape
TapeTalk Magazine on cassette tape

Dear Mr. Lloyd:

I have previously sent to you photocopies of my correspondence with the publishers of LEVEL I magazine and TapeTalk Magazine. I have had no reply from either of these publications.

In my letters to them I have tried to be understanding of any problems they may have encountered in getting into production, asking only that they let me know the status of their respective ventures and when subscribers could look forward to receiving delivery under their subscription contracts.

Having had no reply from them, I can only assume that each of the publishers has perpetrated a fraud against their subscribers, using the United States mail as the medium for conducting the scheme. Therefore, I am today filing formal complaints against both of the named publishers, the complaints to be filed with the Attorney General of the State of California, the Chief of Postal Inspection having jurisdiction over the locale in which the publishers operate, the Federal Trade Commission, and the Internal Revenue Service.

I feel that I have been treated wrongfully and unfairly, that I have been defrauded of my money, and that I am entitled to redress and damages. I do not know how many others may be victims of this apparent fraudulent scheme, but I shall be pleased to have them join me in my complaint so they may recover any subscription payments they may have lost. I plan to vigorously push this matter before every possibly interested governmental agency and court of law to seek a satisfactory settlement of my claim against the party or parties responsible for this situation.

I hope that your publication, as well as other computer user/hobbyist periodicals, will feel free to quote me in my feeling and intentions in this matter.

Sincerely,

J. Wesley B. Taylor

JWBT/self

My personal mailing address:

J. Wesley B. Taylor
P.O. Box 4391

Wichita Falls, Texas 76308

Phone: Office 817 723-5523
Home 817 692-2687

618 Hal Greer Boulevard
Huntington, WVA 25701

Mr. Gordon Lloyd
TRS-80 Users Group
7554 Southgate Road
Fayetteville NC 28304

May 9, 1979

Dear Mr. Lloyd:

Some time ago I sent you photocopies of an editorial concept I was trying out, "DocuWare." I haven't had any reply. I suspect that this may be because of my references to World Power's distribution in your publication. Two days ago I discovered that World Power Systems was a huge fraud. Perry Pollock was arrested by the district attorney in Tucson last week. I had ordered a master control unit, made inquiries, and Monday found their phones unconnected.

I called California Digital who tipped me off. I then called the D. A. and then a couple of the slicks, and learned that World Power will probably turn into a huge case, having ripped off hundreds of people.

I won't be going ahead with DocuWare for awhile until I see what happens with what looks like will become a landmark case in personal-computer-based fraud. As I mentioned in my draft letter, regulation of the personal software field could well result.

Anyway I noticed your call for articles in the April 1979 issue (I got two pages 35 but no 33; could you send it?) and would like to contribute. I have my own typesetting equipment, so if you specify width of column you want submissions in, I can accommodate. A couple of Not So Basics are enclosed.

I like your publication because it gets right down to some of the handiest tips on the TRS-80 ever. The keyboard disabling POKE by Michael Chertok was just what I was looking for. I would like to compile a list for the newsletter on several of handy POKES and PEEKS, the kind that aren't in the book but are great for particular applications.

Please let me hear from you .. thanks.

Chris Gundlach

enc.

NOT SO BASIC.

TIPS FOR TRICKS WITH LEVEL II BASIC
BY CHRIS GUNDLACH, 618 HAL GREER BOULEVARD, HUNTINGTON WVA

MACHINE-SPEED GRAPHICS IN BASIC

Yes, you can achieve machine-speed printing of your TRS-80 graphics. It's done by simply using some string functions in a clever way.

I won't try to describe drawing a graphic, only storing it and reprinting it. You probably have your own program that creates some kind of graphics with various SETs or POKEs.

Once you get your graphic onto the screen, determine where the image is positioned on the screen using the PRINT@ numbers on the worksheet. Add the PRINT@ location to 15360, and you'll know the memory locations.

Let's say, for instance, that you have a heading with some fancy boxes along with text. The heading is in the first four lines of the screen, which are PRINT@ positions 0 through 255. Now do this:

```
30000 FOR F = 15360 TO 15615
30010 A$ = A$+CHR$(PEEK(F))
30020 NEXT
```

```
Now clear the screen and
PRINT A$
```

How 'bout that? Your graphic heading returns at lightning speed. The description I've written above, of course, applies to your "discovery" of this technique. Once you see how it works, you can design routines like this into your program. You can, for instance, use a similar PEEK routine to give you all the ASCII numbers for the characters in your heading; then type those numbers into a DATA statement for another program. Then, when you initialize variables in the program, do a READ loop to concatenate everything into one string:

```
FOR F = 0 TO 255 : READ A : A$=A$+CHR$(A) : NEXT
```

(Put this in your own program, along with the appropriate DATA numbers.)

One disadvantage is that the 255-character graphic will occupy 255 bytes as a string variable; memory space is what you pay for speed. With some careful and logical studying of the ASCII codes in your Level II manual, however, you can replace large blank areas with tab codes. You can also design smaller pieces of graphics, such as boxes or dice. In order to make these variables printable at any location on the screen, you have to provide linefeeds and backspaces or other tricks to move the print position (the cursor) about. The following creates three bars:

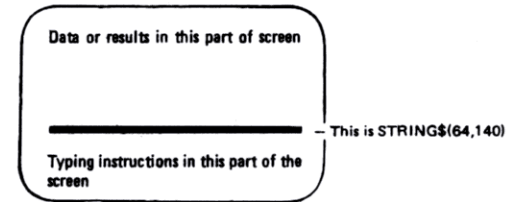
```
A$ = STRING$(16,191) + STRING$(2,26) + STRING$(16,24) + STRING$(16,191) + STRING$(2,26) +
STRING$(16,24) + STRING$(16,191)
```

Now just PRINT A\$ anywhere you want on the screen. The three parallel bars will come out fine and fast! STRING\$(2,26) is two downward linefeeds, and STRING\$(16,24) backs up the cursor 16 spaces for the STRING\$(16,191), the bar. This method allows you to use PRINT@ to locate boxes, spaceships, robots, or whatever. . . don't go closer to the edges than the "character" width, or it'll flop over to the next line. And, if you set up the STRING\$ as variables themselves, you can go crazy "typing" your pictures by just typing the variable! Have fun.

WHERE WAS I?

This trick will be useful to those of you designing word processors or terminal-like displays.

One useful screen format I've found, for instance, is:



While your program is printing data in the top area, of course, that's where the cursor is located. Suppose you wish to provide an INKEY\$ interrupt to allow the operator to input or change a piece of data, by typing a shift character, and then restore the cursor to its former location. The little routine below will let you do this:

[Your INKEY\$ response will jump your program to the following]

```
30000 CL=PEEK(16416) : CH=PEEK(16417)
30010 PRINT@ 768,"":INPUT "[your question or your print statement of instructions]
30020 POKE 16416, CL : POKE 16417, CH
30030 RETURN [you return to the main program routine that uses the
top of the screen]
```

Locations 16416-16417 contain the cursor location. A routine set up like the one above allows you to memorize the cursor location, then print in a different location, and then return to the former cursor position. This is very flexible because you do not have to anticipate where the cursor might have to go, nor do you need to know hex numbers. (This technique, by the way, is an in-Basic introduction to the memory-save and-recall routines you will learn to perform when you begin assembly language programming.)

This is a nice feature to include in graphic-ized games. You can provide for a shift-I to print instructions or remarks at 768 (or any other location), and then return to your main program with the cursor (and printing) resuming from where it left off. To erase the bottom of the screen, just PRINT@768,CHR\$(31) before you go back to the main program. CHR\$(31) is "clear to the end of the frame." It does.

MICRO MANAGEMENT SYSTEMS

MICRO-COMPUTER SPECIALIST
LARRY OWENS

★ 15% Discount: TRS-80's ★

RADIO SHACK DEALER - MINI MALL ✓ MSB
DOWNTOWN SHOPPING CENTER
CAIRO, GA. 31728 (912) 377-7120

After waiting over a year for someone else to come up with some practical programs for my Amway business, I finally have written my own. The most useful one is Level II 16K and holds the 4 prices of each of 290 items in memory; I just enter the quantity and stock number and the computer generates an order form, doing all multiplication, addition of 4 columns, and adds tax and handling. Really a terrific timesaver for verifying the many orders I process each week from my over-250 distributors. Also, another program keeps track of the sponsorship lines of these distributors, including an indication of those not renewed - in effect, it allows me to look at each branch of our Amway "family tree". Another program keeps track of the totals in a simple 12-column ledger bookkeeping program, and a fourth program generates a monthly gross profit summary.

All four programs are available on cassette (no disk or printer required) for \$24.95, including necessary documentation. Prices updated on cassette every January, May and September for \$10.

Fred Blechman K6UGT
7217 Bernadine Av.,
Canoga Park, CA 91307
(213) 346-7024

```

10 'THIS PROGRAM SHOWS <IN SLOW MOTION> HOW THE TRS-80 MEMORY-
15 'MAPPED KEYBOARD SCAN WORKS -- NOTICE WHAT HAPPENS WHEN YOU
20 'HOLD DOWN VARIOUS KEYS. THIS ALL GETS SORTED OUT BY THE
25 'INPUT ROUTINE IN ROM (AT 995 DEC.). FOR MORE DETAILS, SEE
30 'THE EXCELLENT ARTICLE BY WILLIAM BARDEN IN TRS-80 COMPUTING
35 'VOL 1, # 2
60 ' PHELPS GATES
70 ' 6 CRESTWOOD TR. PK. - RT. 4
80 ' CHAPEL HILL NC 27514
90 ' 5/13/79
100 DEFINTA-Z:CLS:S=14336:'START OF KB MATRIX
110 PRINT@0," LOC. CONTENTS BIT VALUE OF CONTENTS"
120 I=1
130 FORJ=1TO8
140 T=S+I:PRINT@J*64,T;:P=PEEK(T):PRINTP,
150 M=128:FORK=1TO8:'DISPLAY BIT VALUE OF CONTENTS
160 PRINT-((PANDM)<>0);:M=M/2
170 NEXT K
180 I=I*2:'COMPUTER SCANS S+1, S+2, S+4, S+8, ETC.
190 PRINT@J*64+46,"<---SCANNING";
200 PRINT@J*64-20,CHR$(30);
210 NEXT J
220 PRINT@556,CHR$(31):GOTO120

```

SAVE 10%, 15% OR MORE ON ALL COMPUTERS, PERIPHERALS, & SOFTWARE.

NO TAXES ON OUT-OF-STATE SHIPMENTS.

WARRANTIES WILL BE HONORED BY YOUR LOCAL 'RADIO SHACK' ® STORE.

VERBATIM DISKETTES \$3.50 EACH.

HENRY'S ENGINEERING CO.
P. O. BOX 629
PRINCE FREDERICK, MD 20678
(301) 855-6777 (301) 525-3142

FOR SALE

Level II 16K TRS-80 computer Many tapes (games-etc.) plus tape case. 15 Issues TRS-80 users group, 12 Issues BYTE magazine. Level II Manual, Level I Manual in Memory Information system Personal Finance Package. 8 Months old. Hardly Used. Excellent Condition!!! Little one on the way, need the money!!! Cash or Money order only. \$1100.00-Why wait? Call Garry Morgan After Five 404-554-4956 8am-12noon 404-554-3021

Dear Gordon,

What I need is not so much as a program but simply a poke statement that I could insert to command the program to run without typing in r u n and then enter example I had in mind-----

Load the cassette with a tape that has program after program on it. Manually you would once have to type in cload and from then on the computer would cload---shut off the recorder--and instead of returning a prompt and the word ready, it would run. Upon completing the program it would find as its last statement cload and start all over again. A simple problem?

Arthur Plante
489 Middle Rd.
Acushnet, MA 02743

T H E B I T P I C K E R ' S T O O L B O X

By Steve MacGregor, 3701 W Wethersfield, Phoenix, Arizona 85029

Here is a divide subroutine that will divide a 16-bit or 31-bit positive number by a 16-bit positive number, and give a 16-bit quotient and 16-bit remainder. To use it, call "DIV16" with the dividend in HL, or "DIV31" with the dividend in DE and HL (high-order 15 bits in DE, low-order 16 bits in HL; bit 7 of D is not used), and with the divisor in BC in either case. The quotient is returned in HL, and the remainder in DE. The AF, BC, IX, and IY registers are not affected, and are returned unaltered.

```

4980 11000000 DIV16 LD DE,0
4983 F5 DIV31 PUSH AF
4984 3E10 LD A,16
4986 29 D0 ADD HL,HL
4987 EB EX DE,HL
4988 ED6A ADC HL,HL
498A B7 OR A
498B ED42 SBC HL,BC
498D 3003 JR NC,D1
498F 09 ADD HL,BC
4990 1801 JR D2
4992 13 D1 INC DE
4993 EB D2 EX DE,HL
4994 3D DEC A
4995 20EF JR NZ,D0
4997 F1 POP AF
4998 C9 RET

```

The origin given here is 4980 because that was easy to type in with T-BUG, but the routine can be loaded anywhere, since it contains no absolute addresses.

```

10 'HERE'S A Z-80 DISASSEMBLER PROGRAM -- GIVES OPERANDS IN
20 'BOTH DECIMAL AND HEX (HANDY FOR POKING)
30 'FOR A TAPE OF A SLIGHTLY FANCIER VERSION, SEND TWO BUCKS
40 'TO COVER TAPE AND POSTAGE
50 ' PHELPS GATES
60 ' 6 CRESTWOOD TR. PK. - RT. 4
70 ' CHAPEL HILL, NC 27514
80 ' (919)-967-5193
90 ' 5/8/79
100 CLEAR50:DEFINTE=H:DEFSTRA=D:DIMA(97),VA(12),VV(23)
110 DATA 0,0,0,7,15,23,31,39,47,55,63,0,217:'1-BYTE INSTRUCTIONS
120 DATA B,C,D,E,H,L,(HL),A,NZ,Z,NC,C,PO,PE,P,M
130 DATA BC,DE,HL,SP,(IX+),(IX+),ADD,ADC,SUB,SBC,AND,XOR,OR,CP
140 DATA BC,DE,HL,AF,(HL),(IX),(IY),HL,IX,IY,RLC,RRC,RL,RR,SLA,SRA,X,SRL
150 DATA BIT,RES,SET,NOP,EX,DJNZ,RLCA,RRCA,RLA,RRR,DAA,CPL,SCF,CCF,RET,EXX
160 DATA JP,X,OUT,IN,EX,EX,DI,EI,NEG,RETN,IM 0,RETI,IM 1,IM 2,RRD,RLD,LDI,CPI
170 DATA INI,OUTI,LDD,CPD,IND,OUTD,LDIR,CPIR,INIR,OTIR,LDDR,CPDR,INDR,OTDR
180 DATA INC,DEC
190 FORN=0T012:READVA(N):NEXT:FORN=0T097:READA(N):NEXT
200 FORJ=0T023:READVV(J):NEXT:'2-BYTE INSTRUCTIONS
210 DATA68,69,70,77,86,94,103,111,160,161,162,163,168,169,170,171
220 DATA 176,177,178,179,184,185,186,187
230 A=STR$(N):INPUT"START(DEC. )";A:X=VAL(A):IFX>32767THENN=X-65536ELSEN=X
235 PRINT"TO STOP LISTING, PRESS 'X'"
240 I=0:T=0:C="":D=""
245 'SUBROUTINE AT LINE 300 DOES MOST OF THE DISASSEMBLY
250 GOSUB300:IFC="HL"THENC=A(37+I):'I=1 FOR "IX", 2 FOR "IY"
260 IFD="HL"THEND=A(37+I)
270 GOSUB870:PRINTTAB(17);B:IFC=""THEN280ELSEPRINTTAB(24);C;
275 IFD=""THENPRINT", ";D;
280 PRINTTAB(36);:FORX=NTON+P:P=PEEK(X):GOSUB860:PRINT" ";:NEXT:N=X:PRINT
290 IFINKEY#="X"THEN230ELSE250
300 I=0:B="LD":T=0:C="":D=C
310 P=PEEK(N+T):GOSUB780
320 ONFGOTO450,460,480
330 ONHGOTO370,380,410,420,420,430,440
340 B="JR":IFG<3THENB=A(G+51):IFG=0THENRETURN
345 IFG=1THENC="AF,AF":RETURN
350 T=T+1:U=PEEK(N+1):IFU>127THENU=U-256
360 X=N+U+2:GOSUB770:IF(G=2)OR(G=3)THEN840ELSEC=A(G+4):GOT0850
370 DV=A(J+16):IFE=0THENC=DY:GOT0720ELSEB="ADD":C="HL":D=DY:RETURN
380 DV="A":IFG>3THENIFG<6THENDY="HL"
390 IFG>3THENGOSUB740ELSEIFG>1THENDX="(DE)"ELSEDX="(BC)"
400 D=DY:IFE=0THEN840ELSEC=DY:GOT0850
410 B=A(96+E):C=A(16+J):RETURN
420 B=A(H+92):GOSUB790:GOT0840
430 GOSUB790:C=DX:T=T+1:GOSUB820:GOT0850
440 B=A(G+54):RETURN
450 IFP=118THENB="HALT":RETURN:ELSEGOSUB790:C=DX:GOSUB810:GOT0850
460 GOSUB810
470 B=A(G+22):C="A":IF(G<4)AND(G<2)THEN850ELSE840
480 ONHGOTO490,510,530,560,570,580,590:B="RET":C=A(G+8):RETURN
490 IF E ELSEB="POP":GOT0830
500 IFJ<2THENB=A(62+J):RETURN:ELSEIFJ=2THENB="JP":C=A(I+34):RETURN
505 C="SP":D="HL":RETURN
510 B="JP"
520 C=A(G+8):GOT0720
530 B=A(G+64):IFG>5THENRETURNELSEIFG=0THENGOSUB760:GOT0840
535 IFG=1THEN600

```

```

540 D="HL":IFG=4THENC="(SP)":RETURN:ELSEIFG=5THENC="DE":RETURN
550 C="A":X=PEEK(N+1):GOSUB770:GOSUB750:T=T+1:GOT0850
560 B="CALL":GOT0520
570 IFE=0THENB="PUSH":GOT0830:ELSEIFJ=2THEN630
575 IFJ<0THENI=(J+1)/2:T=T+1:GOT0310ELSEB="CALL":GOT0730
580 X=PEEK(N+1):GOSUB770:T=T+1:GOT0470
590 B="RST":X=G+8:GOSUB770:GOT0840
600 T=T+1:IFITHENP=PEEK(N+3)ELSEP=PEEK(N+1)
610 GOSUB780:IFFTHEN620ELSEB=A(40+G):GOSUB810:GOT0840
620 B=A(47+F):X=G:GOSUB770:C=DX:GOSUB810:GOT0850
630 T=T+1:FORL=0T023:IFVV(L)=PEEK(N+1)THENB=A(L+72):RETURN:ELSENEXT
640 P=PEEK(N+1):GOSUB780:IFF<>1THEN710ELSEIFH=7THEN690
645 ONHGOTO660,670,680
650 IFH<3THEN710ELSEB="IN":C="A":GOSUB790:GOSUB750:GOT0850
660 B="OUT":D="A":GOSUB790:GOSUB750:GOT0840
670 B=A(25-2+E):C="HL":D=A(16+J):RETURN
680 GOSUB740:DV=A(16+J):IFETHENC=DY:GOT0850:ELSED=DY:GOT0840
690 DX="A":IFETHENDY="R"ELSEDY="I"
700 C=DY:IFG<2THEN850ELSED=DY:GOT0840
710 B="?":RETURN
720 GOSUB760:GOT0850
730 GOSUB760:GOT0840
740 GOSUB760
750 DX="("+DX+")":RETURN
760 T=T+2:X=PEEK(N+T-1)+256+PEEK(N+T):TH=N+T
770 DX=STR$(X-(X<0)*65536):DX=MID$(DX,2):RETURN
780 F=P/64:G=(PRAND56)/8:H=PRAND7:J=G/2:E=GAND1:RETURN:'INSTRUCTION FIELDS
790 L=G
800 DX=A(L):IFL<6THENRETURNELSEIFI=0THENRETURN
805 DX=A(I+19)+MID$(STR$(PEEK(N+2)),2)+":T=T+1:RETURN
810 L=H:GOT0800
820 X=PEEK(N+T):GOT0770
830 C=A(30+J):RETURN
840 C=DX:RETURN
850 D=DX:RETURN
860 Q=P/16:P=P-16+Q:PRINTCHR$(Q+48-7*(Q>9));CHR$(P+48-7*(P>9));
865 RETURN
870 P=PEEK(VARPTR(N)+1):GOSUB860:P=PEEK(VARPTR(N)):GOSUB860
880 PRINTN-65536+(N<0);TAB(12);:RETURN

```

TRS-80 QUALITY SOFTWARE

LEVEL I. AND LEVEL II.		LEVEL II	
#1.	IDM-I CASSETTE DATA BASE \$20.	#11.	WORD-I WORD PROCESSOR \$25.
#2.	INV-I INVENTORY CONTROL \$20.	#15.	MAIL-I NAME AND ADDRESS \$25.
#3.	STOCK-I SECURITY INFO. \$10.	#16.	SORT-I SORT UTILITY \$10.
#4.	BANK-I CHECK BALANCE \$10.	#17.	STAT-I STATISTICS \$10.
#5.	FINANCE-I STOCK-I & BANK-I \$15.	#18.	KEY-I KEY-ACCESS \$10.
		#19.	SALE-I SALE ANALYSIS \$10.
		#20.	UTIL-I SORT-I & KEY-I \$16.
DISKETTE.			
#12.	MAIL-III MAILING LIST \$35.		
#14.	WORD-III WORD PROCESSOR \$35.		
#21.	INV-III INVENTORY CONTROL \$35.		
#22.	KEY-III KEY RANDOM ACCESS \$15.		

MICRO ARCHITECT
96 DOTHAM ST.
ARLINGTON, MA 02174

PRICE INCLUDES POSTAGE, CASSETTER & DOCU.

Dear Gordon;

In Vol 1 No 4 of the Newsletter, page 7 you placed a letter I sent to you in regard to the Practical Auot-mation #DMPT-6 Line Printer and #UP-72 Interface.

I received from CLOAD Magazine, the address of PM (Trap Falls Rd, Shelton, CT. 06484) and I sent a letter, requesting information about adaption of this unit to the TRS-80. I got the answer of, "we dont sell to the individual, only to dealers". I didnt even ask this question.

I was informed though that Digital Group and the Integral Data Systems were the agents for this unit and they sell computers and I dont think they would be in the mood to supply informtaion to a TRS-80 user.

Maybe they dont realize the potential sales to the TRS-80 users of this small, 8", and economical unit.

Even Radio Shack hasnt even answered several of my letters in regard to this unit....

Maybe if enough of the TRS-80 Users Group wrote to R/S and PM about this unit they would wake up to the fact that there is intrest in this unit and at least let us know.

One more thing I might mention. The bologna that R/S puts out that it only takes about 6 weeks to get a computer after ordering is a lot of bologna. I ordered a Level II, 16k unit 14 Feb 78 and havnt received mine yet, but was told that it would be sent around the 15th of May 78, (I guess they ment)but I havnt seen it. There are plenty of Level I's available. So, fortunately, I have a local R/S store manager who got a Level I for me to learn on until mine arrives.

I am pleased with the new format of the TRS-80 Users Group Newsletter, center folded, page numbered and book form and we are getting many interesting tiems to read and programs to play with.

Keep up the good work.

Thanks

MEL BUSCH

10909 Carr Rd.

Jeffersonville, OH.

43128

Dear Gordon,

The TRS-80 USERS Have an Amateur Radio Net which meets on the air each Sunday at 1900 Hours GMT for one to two hours. The frequency is 14.342 MHZ and is coordinated on West Coast by myself.

WA6YKH BILL,

and on the East Coast by WD8SAS, also BILL.

We now have 92 members of the net including members from Mexico and Canada.

You may use this info anyway you desire.

Sincerely,

W. J. Myatt
14721 Candeda Place
Tustin, CA 92680

201 N. Broadmore St.
Plentywood, MT 59254
6 June 1978

TRS-80 Users Group
7554 Southgate Rd.
Fayetteville, NC 28304

```
1Ø INPUT F
2Ø PRINT F;
3Ø IF F PRINT "TRUE": GOTO 1Ø
4Ø PRINT "FALSE": GOTO 1Ø
```

F can be any expression. The false branch will be executed if $F \geq 0$ and $F < 1$. But be careful; a bug causes anything $\geq .9999997$ and < 1 to be printed as 1.0000005. Still, if you input anything ≥ 1 (that includes 1.0000005) the true branch will be excuted. Also, all negative numbers result in a true branch.

THEN and T. are completely unnecessary, though they are nice to have. If you want to go to another line, you can just say GOTO or G. If you want to execute some other statement just say it and omit THEN.

Other people have come up with these ways to change to 32 characters per line; >P.#"2" (rewind)
1Ø INPUT "EEE" >CL.
2Ø GOTO 1Ø >1ØCS.
(Craig H. Reinhardt) >R.
(Timothy Loos)

With both of these the keyboard is locked up. Sometimes, by playing with the on/off switch, I can get mine in double width mode. The keyboard is unlocked so you can program in that mode. Sometimes you'll have to push CLEAR a couple times first. Type exactly as you would in normal mode.

Anyone out there got hardware brains? I need a circuit to attach to the "card slot for an RS-232 interface -- or whatever!" in the expansion interface. The circuit should produce acceppable output and accept input from an RS-232 interface in a KSR-43 TTY. If anyone thinks they can help, write me.

The Newsletter is great!! To E.O. Tuck: Your sub-routines are super! Yes, a lot of us have the bug in INT.

Sincerely,

William R. Trowbridge
William R. Trowbridge

Dr. Jack W. Crenshaw
2114 Cecille Drive SW
Huntsville, Ala. 35803

Dear Bob:

Since my article on T-BUG, I've found out how the jump out of BASIC works. Thought I would share it with you. Incidentally, I didn't figure out myself. The information comes from two sources: Barry Lewis of Washington, D.C. and James Falk of Sea Cliff, NY.

The trick is super simple. When the BASIC interpreter reads a tape, it executes a subroutine call. This means that the CPU pushes the current address (inside the monitor) onto the stack. The address is popped off of the stack when the subroutine return is encountered. To branch out into some user program, all you have to do is to make sure your program overlays the stack area, with the desired address replacing the one BASIC put there. When the tape finishes reading, the return causes this address to be popped off, and presto...

The magic address is 41FE-41FF. Incidentally, Barry says that you don't have to reload all of T-BUG when you accidentally get back to BASIC. All you need is the branch. It should be possible to write a tape with just the start address in 41FE-41FF. I tried that, but it didn't work. Apparently, T-BUG needs some extra initialization on a restart. Anyone have any ideas?

Here's some other random info: There is a character generator chip... a MCM 6670. The RAM chips are NEC uPD416D or Mostek MK4116. (Courtesy Les Logan, Norfolk, Va.) George Kowalski has a disassembled listing of Level I BASIC. Write him at 1333 Holmes Dr., Colorado Springs, Colo. 80909.

From various sources, I have received various partial schematics of the circuit. No complete one as yet.

Barry Lewis claims to have modified the tape I/O software to speed it up. Apparently the speed is controlled by a counter which is easily changed. Barry is running at 1200 baud! Someone else has got a Teletype interfaced to the unit.

I suppose one of the questions many of us have been asking is how much support we can expect from Tandy, with respect to using the unit for things it was not designed for. I recently read an interview with Chuck Peddle of Commodore, in which he flatly stated the they would never release schematics of the PET. Since Tandy hasn't either, it was looking bad for the good guys. No more. I understand that Tandy expects to be offering full schematics soon. Not only that, the Assembler comes with a list of all usable entry points in both Level I and Level III! Take that, Commodore!

INPUT AND RUN 'PATCH'-BEFORE YOU START YOUR PRINTER AND
YOU WILL GET 60 LINES ON A PAGE. AT THE
END OF YOUR RUN TYPE 'LPRINT CHR\$(12)'
AND SET THE PRINTER ON THE NEXT PAGE.

```
100 A=40960-65536
110 READ B
120 IF B=300 THEN 200
130 POKE A, B
135 A=A+1
140 GOTO 110
150 DATA 245, 197, 213, 229, 33, 41, 64, 126, 254, 60, 204, 20
160 DATA 160, 225, 209, 193, 241, 195, 141, 5, 62, 0, 50, 41, 64
170 DATA 6, 6, 5, 8, 62, 10, 205, 59, 0, 8, 200, 195, 27, 160, 300
200 POKE 16424, 61:POKE 16425, 0
210 POKE 16422, 0:POKE 16423, 160
```

R.G. Lloyd
TRS-80 Users Group
7554 Southgate Rd.
Fayetteville, N.C.
28304

Dear Gordon,

Here are some thoughts on C.P. Tigla's problem of retaining and displaying data after he has used the INPUT# or PRINT# functions.

I haven't tried it, but I think that if the alphanumeric variables that are being used were substituted by the a(x) array that these variables could be displayed with a simple loop program as long as the memory had not been powered down after loading.

```
Something like: 10 For X = 1 to B - 1 (PP 223 line 70)
                 20 P. A(X),A(X+60)
                 30 Next X
```

By using A(X) as the array for temp and A(x+60) as the humidity array you could call up at will any part of this information.

I use this program to display the variables in a business program I am writing to work as a diagnostic.

If there is ANYBODY that can tell me how to hook up an ASCII Selectric, an RO-33, or any 20 mil or RS 232C device without going to the terribly expensive interface connector I would greatly appreciate it. Call me collect weekends at 817-441-8052 or use the address below. I would gladly pay a reasonable amount for information OR a working unit. I can use screen-read...or whatever..I just need the ability to get it on 8 1/2 by 11 forms on regular paper.

Also I am interested in anyone's business programs that are useful in 16K memory without disc or Level II (yet).

Has anyone done anything yet on anykind of process control of external devices? This is another area that I am interested in, but my level of understanding is lacking....

You have a fine newsletter...keep up the good work!

Pete Charlton
491 Elbow Ct.
Weatherford, Tx. 76086
817-441-8052/8045



Z80	8080	HEX	OCT	DECIMAL	BINARY
NOP	NOP	00	000	00	0 0 0 0 0 0 0 0 0
LD BC, NN	LXI B	01	001	01	0 0 0 0 0 0 0 0 1
LD (BC), A	STAX B	02	002	02	0 0 0 0 0 0 0 1 0
INC BC	INC B	03	003	03	0 0 0 0 0 0 0 1 1
INC B	INR B	04	004	04	0 0 0 0 0 0 1 0 0
DEC B	DCR B	05	005	05	0 0 0 0 0 0 1 0 1
LD B, N	MVI B	06	006	06	0 0 0 0 0 0 1 1 0
RLCA	RLC	07	007	07	0 0 0 0 0 0 1 1 1
ADD HL, BC	DAD B	09	011	09	0 0 0 0 0 1 0 0 1
LD A, (BC)	LDAX B	0A	012	10	0 0 0 0 0 1 0 1 0
DEC BC	DCX B	0B	013	11	0 0 0 0 0 1 0 1 1
INC C	INR C	0C	014	12	0 0 0 0 0 1 1 0 0
DEC C	DCR C	0D	015	13	0 0 0 0 0 1 1 0 1
LD C, N	MVI D	0E	016	14	0 0 0 0 0 1 1 1 0
RRC	RRC	0F	017	15	0 0 0 0 0 1 1 1 1
LD DE, NN	LXI D	11	021	17	0 0 0 0 1 0 0 0 1
LD (DE), A	STAX D	12	022	18	0 0 0 0 1 0 0 1 0
INC DE	INX D	13	023	19	0 0 0 0 1 0 0 1 1
INC D	INR D	14	024	20	0 0 0 0 1 0 1 0 0
DEC D	DCR D	15	025	21	0 0 0 0 1 0 1 0 1
LD D, N	MVI D	16	026	22	0 0 0 0 1 0 1 1 0
RLA	RAL	17	027	23	0 0 0 0 1 0 1 1 1
ADD HL, DE	DAD D	19	031	25	0 0 0 0 1 1 0 0 1
LD A, (DE)	LDAX D	1A	032	26	0 0 0 0 1 1 0 1 0
DEC DE	DCX D	1B	033	27	0 0 0 0 1 1 0 1 1
INC E	INR E	1C	034	28	0 0 0 0 1 1 1 0 0
DEC E	DCR E	1D	035	29	0 0 0 0 1 1 1 0 1
LD E, N	MVI E	1E	036	30	0 0 0 0 1 1 1 1 0
RRA	RAR	1F	037	31	0 0 0 0 1 1 1 1 1
LD HL, NN	LXI H	21	041	33	0 0 0 1 0 0 0 0 1
LD (NN), HL	SHLD	22	042	34	0 0 0 1 0 0 0 0 1 0
INC HL	INX H	23	043	35	0 0 0 1 0 0 0 0 1 1
INC H	INR H	24	044	36	0 0 0 1 0 0 0 1 0 0
DEC H	DCR H	25	045	37	0 0 0 1 0 0 0 1 0 1
LD H, N	MVI H	26	046	38	0 0 0 1 0 0 0 1 1 0
DAA	DAA	27	047	39	0 0 0 1 0 0 0 1 1 1
ADD HL, HL	DAD H	29	051	41	0 0 0 1 0 1 0 0 0 1
LD HL, (NN)	LHLD	2A	052	42	0 0 0 1 0 1 0 0 1 0
DEC HL,	DCX H	2B	053	43	0 0 0 1 0 1 0 0 1 1
INC L	INR L	2C	054	44	0 0 0 1 0 1 0 1 0 0
DEC L	DCR L	2D	055	45	0 0 0 1 0 1 0 1 0 1
LD L, N	MVI L	2E	056	46	0 0 0 1 0 1 0 1 1 0
CPL	CMA	2F	057	47	0 0 0 1 0 1 0 1 1 1
LD SP, NN	LXI SP	31	061	49	0 0 0 1 1 0 0 0 0 1
LD (NN), A	STA	32	062	50	0 0 0 1 1 0 0 0 1 0
INC SP	INX SP	33	063	51	0 0 0 1 1 0 0 0 1 1
INC (HL)	INR M	34	064	52	0 0 0 1 1 0 0 1 0 0
DEC (HL)	DCR M	35	065	53	0 0 0 1 1 0 0 1 0 1
LD (HL)	MVI M	36	066	54	0 0 0 1 1 0 0 1 1 0
SCF	STC	37	067	55	0 0 0 1 1 0 0 1 1 1
ADD HL, SP	DAD SP	39	071	57	0 0 0 1 1 1 0 0 0 1
LD A, (NN)	LDA	3A	072	58	0 0 0 1 1 1 0 0 1 0
DEC SP	DCX SP	3B	073	59	0 0 0 1 1 1 0 0 1 1
INC A	INR A	3C	074	60	0 0 0 1 1 1 0 0 0 0
DEC A	DCR A	3D	075	61	0 0 0 1 1 1 0 0 0 1
LD A, N	MVI A	3E	076	62	0 0 0 1 1 1 0 0 1 0
CCF	CMC	3F	077	63	0 0 0 1 1 1 0 0 1 1
LD B, B	MOV B, B	40	100	64	0 0 1 0 0 0 0 0 0 0
LD B, C	MOV B, C	41	101	65	0 0 1 0 0 0 0 0 0 1
LD B, D	MOV B, D	42	102	66	0 0 1 0 0 0 0 0 1 0

LD B, E	MOV B, E	43	103	67	0 0 1 0 0 0 0 0 1 1
LD B, H	MOV B, H	44	104	68	0 0 1 0 0 0 0 0 1 0 0
LD B, L	MOV B, L	45	105	69	0 0 1 0 0 0 0 0 1 0 1
LD B, (HL)	MOV B, M	46	106	70	0 0 1 0 0 0 0 0 1 1 0
LD B, A	MOV B, A	47	107	71	0 0 1 0 0 0 0 0 1 1 1
LD C, B	MOV C, B	48	110	72	0 0 1 0 0 0 1 0 0 0 0
LD C, C	MOV C, C	49	111	73	0 0 1 0 0 0 1 0 0 0 1
LD C, D	MOV C, D	4A	112	74	0 0 1 0 0 0 1 0 0 1 0
LD C, E	MOV C, E	4B	113	75	0 0 1 0 0 0 1 0 0 1 1
LD C, H	MOV C, H	4C	114	76	0 0 1 0 0 0 1 0 0 0 0
LD C, L	MOV C, L	4D	115	77	0 0 1 0 0 0 1 0 0 0 1
LD C, (HL)	MOV C, M	4E	116	78	0 0 1 0 0 0 1 0 0 0 1 0
LD C, A	MOV C, A	4F	117	79	0 0 1 0 0 0 1 0 0 0 1 1
LD D, B	MOV D, B	50	120	80	0 0 1 0 0 1 0 0 0 0 0 0
LD D, C	MOV D, C	51	121	81	0 0 1 0 0 1 0 0 0 0 0 1
LD D, D	MOV D, D	52	122	82	0 0 1 0 0 1 0 0 0 0 1 0
LD D, E	MOV D, E	53	123	83	0 0 1 0 0 1 0 0 0 0 1 1
LD D, H	MOV D, H	54	124	84	0 0 1 0 0 1 0 0 0 1 0 0
LD D, L	MOV D, L	55	125	85	0 0 1 0 0 1 0 0 0 1 0 1
LD D, (HL)	MOV D, M	56	126	86	0 0 1 0 0 1 0 0 0 1 0 1 0
LD D, A	MOV D, A	57	127	87	0 0 1 0 0 1 0 0 0 1 0 1 1
LD E, B	MOV E, B	58	130	88	0 0 1 0 0 1 0 0 0 0 0 0
LD E, C	MOV E, C	59	131	89	0 0 1 0 0 1 0 0 0 0 0 1
LD E, D	MOV E, D	5A	132	90	0 0 1 0 0 1 0 0 0 0 1 0
LD E, E	MOV E, E	5B	133	91	0 0 1 0 0 1 0 0 0 0 1 1
LD E, H	MOV E, H	5C	134	92	0 0 1 0 0 1 0 0 0 1 0 0
LD E, L	MOV E, L	5D	135	93	0 0 1 0 0 1 0 0 0 1 0 1
LD E, (HL)	MOV E, M	5E	136	94	0 0 1 0 0 1 0 0 0 1 0 1 0
LD E, A	MOV E, A	5F	137	95	0 0 1 0 0 1 0 0 0 1 0 1 1
LD H, B	MOV H, B	60	140	96	0 0 1 0 0 0 0 0 0 0 0 0
LD H, C	MOV H, C	61	141	97	0 0 1 0 0 0 0 0 0 0 0 1
LD H, D	MOV H, D	62	142	98	0 0 1 0 0 0 0 0 0 0 0 1 0
LD H, E	MOV H, E	63	143	99	0 0 1 0 0 0 0 0 0 0 0 1 1
LD H, H	MOV H, H	64	144	100	0 0 1 0 0 0 0 0 0 0 0 1 0
LD H, L	MOV H, L	65	145	101	0 0 1 0 0 0 0 0 0 0 0 1 0 1
LD H, (HL)	MOV H, M	66	146	102	0 0 1 0 0 0 0 0 0 0 0 1 0 1 0
LD H, A	MOV H, A	67	147	103	0 0 1 0 0 0 0 0 0 0 0 1 0 1 1
LD L, B	MOV L, B	68	150	104	0 0 1 0 0 0 0 0 0 0 0 0 0 0
LD L, C	MOV L, C	69	151	105	0 0 1 0 0 0 0 0 0 0 0 0 0 1
LD L, D	MOV L, D	6A	152	106	0 0 1 0 0 0 0 0 0 0 0 0 0 1 0
LD L, E	MOV L, E	6B	153	107	0 0 1 0 0 0 0 0 0 0 0 0 0 1 1
LD L, H	MOV L, H	6C	154	108	0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0
LD L, L	MOV L, L	6D	155	109	0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1
LD L, (HL)	MOV L, M	6E	156	110	0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 0
LD L, A	MOV L, A	6F	157	111	0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 1 1
LD (HL), B	MOV M, B	70	160	112	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
LD (HL), C	MOV M, C	71	161	113	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1
LD (HL), D	MOV M, D	72	162	114	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0
LD (HL), E	MOV M, E	73	163	115	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1
LD (HL), H	MOV M, H	74	164	116	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
LD (HL), L	MOV M, L	75	165	117	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
HALT	HLT	76	166	118	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
LD (HL), A	MOV M, A	77	167	119	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1
LD A, B	MOV A, B	78	170	120	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
LD A, C	MOV A, C	79	171	121	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
LD A, D	MOV A, D	7A	172	122	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
LD A, E	MOV A, E	7B	173	123	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1
LD A, H	MOV A, H	7C	174	124	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0
LD A, L	MOV A, L	7D	175	125	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
LD A, (HL)	MOV A, M	7E	176	126	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 0
LD A, A	MOV A, A	7F	177	127	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1
ADD A, B	ADD B	80	200	128	0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

ADD A, C	ADD C	81	201	129	1 0 0 0 0 0 0 1
ADD A, D	ADD D	82	202	130	1 0 0 0 0 0 1 0
ADD A, E	ADD E	83	203	131	1 0 0 0 0 0 1 1
ADD A, H	ADD H	84	204	132	1 0 0 0 0 1 0 0
ADD A, L	ADD L	85	205	133	1 0 0 0 0 1 0 1
ADD A, (HL)	ADD M	86	206	134	1 0 0 0 0 1 1 0
ADD A, A	ADD A	87	207	135	1 0 0 0 0 1 1 1
ADC A, B	ADC B	88	210	136	1 0 0 0 1 0 0 0
ADC A, C	ADC C	89	211	137	1 0 0 0 1 0 0 1
ADC A, D	ADC D	8A	212	138	1 0 0 0 1 0 1 0
ADC A, E	ADC E	8B	213	139	1 0 0 0 1 0 1 1
ADC A, H	ADC H	8C	214	140	1 0 0 0 1 1 0 0
ADC A, L	ADC L	8D	215	141	1 0 0 0 1 1 0 1
ADC A, (HL)	ADC M	8E	216	142	1 0 0 0 1 1 1 0
ADC A, A	ADC A	8F	217	143	1 0 0 0 1 1 1 1
SUB B	SUB B	90	220	144	1 0 0 1 0 0 0 0
SUB C	SUB C	91	221	145	1 0 0 1 0 0 0 1
SUB D	SUB D	92	222	146	1 0 0 1 0 0 1 0
SUB E	SUB E	93	223	147	1 0 0 1 0 0 1 1
SUB H	SUB H	94	224	148	1 0 0 1 0 1 0 0
SUB L	SUB L	95	225	149	1 0 0 1 0 1 0 1
SUB (HL)	SUB M	96	226	150	1 0 0 1 0 1 1 0
SUB A	SUB A	97	227	151	1 0 0 1 0 1 1 1
SBC A, B	SBB B	98	238	152	1 0 0 1 1 0 0 0
SBC A, C	SBB C	99	231	153	1 0 0 1 1 0 0 1
SBC A, D	SBB D	9A	232	154	1 0 0 1 1 0 1 0
SBC A, E	SBB E	9B	233	155	1 0 0 1 1 0 1 1
SBC A, H	SBB H	9C	234	156	1 0 0 1 1 1 0 0
SBC A, L	SBB L	9D	235	157	1 0 0 1 1 1 0 1
SBC A, (HL)	SBB M	9E	236	158	1 0 0 1 1 1 1 0
SBC A, A	SBB A	9F	237	159	1 0 0 1 1 1 1 1
AND B	ANA B	A0	240	160	1 0 1 0 0 0 0 0
AND C	ANA C	A1	241	161	1 0 1 0 0 0 0 1
AND D	ANA D	A2	242	162	1 0 1 0 0 0 1 0
AND E	ANA E	A3	243	163	1 0 1 0 0 0 1 1
AND H	ANA H	A4	244	164	1 0 1 0 0 1 0 0
AND L	ANA L	A5	245	165	1 0 1 0 0 1 0 1
AND (HL)	ANA M	A6	246	166	1 0 1 0 0 1 1 0
AND A	ANA A	A7	247	167	1 0 1 0 0 1 1 1
XOR B	XRA B	A8	250	168	1 0 1 0 1 0 0 0
XOR C	XRA C	A9	251	169	1 0 1 0 1 0 0 1
XOR D	XRA D	AA	252	170	1 0 1 0 1 0 1 0
XOR E	XRA E	AB	253	171	1 0 1 0 1 0 1 1
XOR H	XRA H	AC	254	172	1 0 1 0 1 1 0 0
XOR L	XRA L	AD	255	173	1 0 1 0 1 1 0 1
XOR (HL)	XRA M	AE	256	174	1 0 1 0 1 1 1 0
XOR A	XRA A	AF	257	175	1 0 1 0 1 1 1 1
OR B	ORA B	B0	260	176	1 0 1 1 0 0 0 0
OR C	ORA C	B1	261	177	1 0 1 1 0 0 0 1
OR D	ORA D	B2	262	178	1 0 1 1 0 0 1 0
OR E	ORA E	B3	263	179	1 0 1 1 0 0 1 1
OR H	ORA H	B4	264	180	1 0 1 1 0 1 0 0
OR L	ORA L	B5	265	181	1 0 1 1 0 1 0 1
OR (HL)	ORA M	B6	266	182	1 0 1 1 0 1 1 0
OR A	ORA A	B7	267	183	1 0 1 1 0 1 1 1
CP B	CMP B	B8	270	184	1 0 1 1 1 0 0 0
CP C	CMP C	B9	271	185	1 0 1 1 1 0 0 1
CP D	CMP D	BA	272	186	1 0 1 1 1 0 1 0
CP E	CMP E	BB	273	187	1 0 1 1 1 0 1 1
CP H	CMP H	BC	274	188	1 0 1 1 1 1 0 0
CP L	CMP L	BD	275	189	1 0 1 1 1 1 0 1
CP (HL)	CMP M	BE	276	190	1 0 1 1 1 1 1 0

CP A	CMP A	BF	277	191	1 0 1 1 1 1 1 1
RET NZ	RNZ	C0	300	192	1 1 0 0 0 0 0 0
POP BC	POP B	C1	301	193	1 1 0 0 0 0 0 1
JP NZ	JNZ	C2	342	194	1 1 0 0 0 0 0 1
JP NN	JMP	C3	303	195	1 1 0 0 0 0 0 1
CALL NZ, NN	CNZ	C4	304	196	1 1 0 0 0 0 1 0
PUSH BC	PUSH B	C5	305	197	1 1 0 0 0 0 1 0
ADD A, N	ADI	C6	306	198	1 1 0 0 0 0 1 1
RST 0	RST 0	C7	307	199	1 1 0 0 0 0 1 1
RET Z	RZ	C8	310	200	1 1 0 0 0 1 0 0
RET	RET	C9	311	201	1 1 0 0 0 1 0 0
JP Z, NN	JZ	CA	312	202	1 1 0 0 0 1 0 1
CALL Z, NN	CZ	CC	314	204	1 1 0 0 0 1 0 0
CALL NN	CALL	CD	315	205	1 1 0 0 0 1 0 1
ADC A, N	ACI	CE	316	206	1 1 0 0 0 1 1 0
RST 00H	RST 1	CF	317	207	1 1 0 0 0 1 1 1
RET NC	RNC	D0	320	208	1 1 0 0 1 0 0 0
POP DE	POP D	D1	321	209	1 1 0 0 1 0 0 0
JP NC, NN	JNC	D2	322	210	1 1 0 0 1 0 0 1
OUT (N), A	OUT	D3	323	211	1 1 0 0 1 0 0 1
CALL NC, NN	CNC	D4	324	212	1 1 0 0 1 0 1 0
PUSH DE	PUSH D	D5	325	213	1 1 0 0 1 0 1 0
SUB N	SUI	D6	326	214	1 1 0 0 1 0 1 1
RST 10H	RST 2	D7	327	215	1 1 0 0 1 0 1 1
RET C	RC	D8	330	216	1 1 0 0 1 1 0 0
JP C, NN	JC	DA	332	218	1 1 0 0 1 1 0 1
IN A, (N)	IN	DB	333	219	1 1 0 0 1 1 0 1
CALL Z, NN	CC	DC	334	220	1 1 0 0 1 1 0 0
SBC A, N	SBI	DE	336	222	1 1 0 0 1 1 1 0
RST 10H	RST 3	DF	337	223	1 1 0 0 1 1 1 1
RST PO	RPO	E0	340	224	1 1 0 0 0 0 0 0
POP HL	POP H	E1	341	225	1 1 0 0 0 0 0 1
JP PO, NN	JPO	E6	342	226	1 1 0 0 0 0 1 0
EX (SP), HL	XTHL	E3	343	227	1 1 0 0 0 0 1 1
CALL PO, NN	COP	E4	344	228	1 1 0 0 0 1 0 0
PUSH HL	PUSH H	E5	345	229	1 1 0 0 0 1 0 1
AND N	ANI	E6	346	230	1 1 0 0 0 1 1 0
RST 20H	RST 4	E7	347	231	1 1 0 0 0 1 1 1
RET PE	RPE	E8	350	232	1 1 0 0 1 0 0 0
JP (HL)	PCHL	E9	351	233	1 1 0 0 1 0 0 1
JP PE, NN	JPE	EA	352	234	1 1 0 0 1 0 1 0
EX DE, HL	XCHG	EB	353	235	1 1 0 0 1 0 1 1
CALL PE, NN	CPE	EC	354	236	1 1 0 0 1 1 0 0
XOR	XRI	EE	356	238	1 1 0 0 1 1 1 0
RST 20H	RST 5	EF	357	239	1 1 0 0 1 1 1 1
RET P	RP	F0	360	240	1 1 0 0 1 0 0 0
POP AF	POP PSW	F1	361	241	1 1 0 0 1 0 0 1
JP P, NN	JP	F2	362	242	1 1 0 0 1 0 0 1
DI	DI	F3	363	243	1 1 0 0 1 0 0 1
CALL P, NN	CP	F4	364	244	1 1 0 0 1 0 1 0
PUSH AF	PUSH PSW	F5	365	245	1 1 0 0 1 0 1 1
OR N	ORI	F6	366	246	1 1 0 0 1 0 1 1
RST 30H	RST 6	F7	367	247	1 1 0 0 1 0 1 1
RET M	RM	F8	370	248	1 1 0 0 1 0 0 0
LD SP, HL	SPHL	F9	371	249	1 1 0 0 1 0 0 1
JP M, NN	JM	FA	372	250	1 1 0 0 1 0 0 1
EI	EI	FB	373	251	1 1 0 0 1 0 0 1
CALL M, NN	CM	FC	374	252	1 1 0 0 1 0 0 0
CP N	CPI	FE	376	254	1 1 0 0 1 0 1 0
RST 30H	RST 7	FF	377	255	1 1 0 0 1 0 1 1
LD (NN), HL	SHLD	22	042	34	0 0 1 0 0 0 1 0
LD (NN), A	STA	32	062	50	0 0 1 0 0 0 1 0