

PHELPS GATES PAGE 31 LEVEL II PRINT#-1
 R. GORDON LLOYD PAGE 32 PATCH
 R. GORDON LLOYD PAGE 32 Z8080
 C. W. EVANS PAGE 36 AREA - - POLYGON
 PHELPS GATES PAGE 37 MERGE
 TED M. LAU PAGE 39 SUPERBAGELS

WE NEED MATERIAL FOR THE NEWSLETTER. REVIEWS OF HARDWARE
 & SOFTWARE, PROBLEMS, PROGRAMS... WHATEVER. PLEASE WRITE

I AM A UNIVERSITY PROFESSOR IN KOREA TEACHING MIS. I HAVE
 JUST ACQUIRED A TRS-80, LEVEL II, 16K WITH A MINI DISK. AND
 CURRENTLY I AM RUNNING TEST PROGRAMS WITH VARIOUS DEGREES OF
 SUCCESS. SINCE I AM THOUSANDS MILES AWAY FROM THE US, I AM IN
 NEED OF SOMEONE WHO DESIRES TO HELP ME IN RESEPT TO GENERAL
 ASPECTS OF THE SYSTEM OPERATION BY CORRESPONDENCE. I WILL BE
 HAPPY TO RECIPROCATATE ANY FAVOR OFFERED IN POSSIBLY EQUIVALENT
 VALUES.

DR. NAM-WON SUH
 SCHOOL OF MANAGEMENT
 KOREA UNIVERSITY
 SEOUL 132, KOREA

DEAR SIRs:

I WISH TO PURCHASE THE AUGUST & SEPTEMBER '78 ISSUES
 OF CLOUD MAGAZINE. IF ANYONE IN YOUR GROUP IS INTERESTED
 PLEASE CONTACT

MARK HARDENBURG
 1838 BRUNELLA AVE.
 PISCATAWAY, NJ 08854

CHANGE TO THE MAILLING LIST PROGRAM ON PAGE 22

LINE 176 IN STILL NOT RIGHT - IT SHOULD READ
 * * 176 LR=LOF(1)*3-5:IF LR<=0 THEN LR=0 * *

THE BITPICKER'S TOOLBOX

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

The built-in BASIC function RND returns what are known as
 "rectangularly distributed" random numbers, that is, any number
 between zero and one (if the argument is zero) or one and n (if
 the argument is a positive integer) has the same chance of being
 generated by the function as any other. For some purposes, you
 may want "normally distributed" random numbers, those that fit
 the infamous bell-shaped curve. The following subroutine will
 produce such numbers in the variable R. The integer variable
R% is used as a temporary loop-count.

5000 R=-6: FOR R%=0 TO 11: R=R+RND(0): NEXT R%: RETURN

The values returned by this subroutine have a mean of zero and a
 standard deviation of one.

G1A19

```

00010 ;THE LEVEL II PRINT#-1 STATEMENT LEAVES MUCH TO BE
00020 ;DESIRED -- THERE'S NO CHECKSUM, OVERHEAD IS HIGH
00030 ;(ESPECIALLY WITH SHORT DATA LISTS), AND YOU HAVE TO
00040 ;SPECIFY EVERY INDIVIDUAL DATA ITEM IN THE LIST.
00050 ;
00060 ;THE FOLLOWING ROUTINE WILL SAVE AN ENTIRE ARRAY TO TAPE
00070 ;SORRY, IT WON'T WORK WITH STRING ARRAYS (THEY'RE STORED
00080 ;DIFFERENTLY) -- CALL THE ROUTINE WITH USR (FIRST
00090 ;POKE 16526,171: POKE 16527,127) - SEE MANUAL PAGE 8/7.
00100 ;
00110 ;TO WRITE ARRAY "A" (ONE-DIMENSIONAL) ONTO TAPE:
00120 ;       POKE 32693,0: THIS TELLS THE ROUTINE TO WRITE
00130 ;       N=VARPTR(A(0))-5: N=USR(N)
00140 ;THE USR ARGUMENT TELLS THE ROUTINE WHERE THE ARRAY IS.
00150 ;
00160 ;TO READ ARRAY "A":
00170 ;       POKE 32693,37 :SWITCHES TO READ ROUTINE
00180 ;       N=VARPTR(A(0))-5
00190 ;       IF USR(N)=0 THEN PRINT"BAD LOAD" (OR WHATEVER)
00200 ;(USR WILL RETURN NON-ZERO VALUE IF THE TAPE LOADS OK)
00210 ;THE ARRAY READ MUST HAVE SAME TYPE AND DIMENSION AS THE
00220 ;ARRAY WRITTEN - OTHERWISE IT IS REDIMENSIONED, WITH
00230 ;DISASTROUS RESULTS TO THE READING PROGRAM.
00240 ;
00250 ;ONE CAUTION: REFERENCING A NEW VARIABLE WILL CAUSE THE
00260 ;ARRAY TO MOVE! (STORAGE IS REALLOCATED) -- "L=USR(N)"
00270 ;WILL CAUSE TROUBLE UNLESS L HAS BEEN PREVIOUSLY ASSIGNED
00280 ;(LIKELIKE 'INPUT"CASSETTE READY";A$', IF A$ HAS NOT BEEN
00290 ;REFERENCED PREVIOUSLY)
00300 ;
00310 ;FOR A 2-DIMENSIONAL ARRAY, N=VARPTR(A(0,0))-7
00320 ;FOR 3 DIMENSIONS, N=VARPTR(A(0,0,0))-9, ETC.
00330 ;
00340 ;       ORG 32683 ;CODE IS SET UP FOR 16K (SPECIFY
00350 ;MEMORY SIZE 32683), BUT IT IS RELOCATABLE AND YOU CAN
00360 ;PUT IT ANYWHERE IN AVAILABLE MEMORY
00370 ;       XOR A
00380 ;       CALL 530 ;TURN ON CASSETTE
00390 ;       CALL 2687 ;GET USR ARGUMENT(ARRAY LOCATION)
00400 ;       LD C,0 ;INITIALIZE CHECKSUM
00410 ;       JR INPUT ;READ/WRITE SWITCH (DEC 32693)
00420 ;       CALL 647 ;WRITE LEADER
00430 ;       LD E,(HL) ;LSB OF ARRAY BYTE COUNT IS AT
00440 ;       LD A,E ;VARPTR(A(0))-5
00450 ;       CALL 612 ;WRITE IT
00460 ;       INC HL ;NEXT BYTE IS MSB OF COUNT
00470 ;       LD D,(HL) ;<DE COUNTS NUMBER OF BYTES
00480 ;       LD A,D ;WRITTEN)
00490 ;       CALL 612 ;WRITE MSB OF COUNT
00500 ;       INC HL ;MOVE TO START OF DATA
00510 ;       LD A,(HL) ;SAVE ARRAY CONTENTS TO TAPE:
00520 ;       CALL 612 ;WRITE THE BYTE
00530 ;       INC HL ;MOVE TO NEXT BYTE
00540 ;       ADD A,C ;UPDATE THE CHECKSUM
00550 ;       LD C,A
00560 ;       DEC DE ;DONE YET?
00570 ;       LD A,D ;IF SO, D AND E ARE BOTH 0
00580 ;       OR E
00590 ;       JR NZ,WRITE
00600 ;       LD A,C ;WRITE CHECKSUM
00610 ;       CALL 612

```

```

7FD5 CDF801 00620 DONE CALL 504 ;TURN OFF RECORDER
7FD8 C39A0A 00630 JP 2714 ;RETURN TO BASIC
7FDB CD9602 00640 INPUT CALL 662 ;WAIT FOR SYNCH BYTE
7FDE CD3502 00650 CALL 565 ;GET BYTE COUNT LSB
7FE1 5F 00660 LD E,A
7FE2 CD3502 00670 CALL 565 ;BYTE COUNT MSB
7FE5 57 00680 LD D,A
7FE6 23 00690 INC HL ;MOVE UP TO START OF DATA
7FE7 23 00700 INC HL
7FE8 CD3502 00710 READ CALL 565 ;NOW READ THE DATA BYTES:
7FEB 77 00720 LD (HL),A
7FEC 23 00730 INC HL ;NEXT BYTE
7FED 81 00740 ADD A,C ;CHECKSUM IN C
7FEE 4F 00750 LD C,A
7FEF 1B 00760 DEC DE ;DE HOLDS COUNT
7FF0 7A 00770 LD A,D
7FF1 B3 00780 OR E ;ZERO IF DONE
7FF2 20F4 00790 JR NZ,READ
7FF4 CD3502 00800 CALL 565 ;GET THE CHECKSUM
7FF7 B9 00810 CP C ;IS IT RIGHT?
7FF8 28DB 00820 JR Z,DONE ;OK LOAD RETURNS NON-0 IN HL
7FFA 210000 00830 LD HL,0 ;BAD LOAD RETURNS ZERO
7FFD 18D6 00840 JR DONE
00850 ;
00860 ;

```

```

00870 ; INCIDENTALLY, YOU CAN ALSO USE THIS ROUTINE TO
00880 ; SAVE AND LOAD OBJECT FILES: POKE THE BYTE COUNT INTO THE
00890 ; TWO BYTES BEFORE THE FILE - FOR EXAMPLE:
00890 ; TO SAVE A 300-BYTE BLOCK AT 25610...
00900 ; POKE 32693,0: POKE 25608,44: POKE 25609,1: N=USR(25608)
00910 ; TO RELOAD IT:
00920 ; POKE 32693,37: IF USR(25608)=0 PRINT"CHECKSUM ERROR"
00930 ; TO SAVE STRING DATA, POKE THE ASCII VALUES INTO
00940 ; A BLOCK OF MEMORY AND SAVE THE BLOCK.
00950 ;
00960 ; IF YOU WANT TO POKE THIS IN FROM BASIC, HERE ARE THE
00970 ; DECIMAL VALUES FOR THE CODE (FROM 32683 TO 32766):
00980 ; 175, 205, 18, 2, 205, 127, 10, 14, 0, 24, 37, 205, 135, 2, 94, 123, 205,
00990 ; 100, 2, 35, 86, 122, 205, 100, 2, 35, 126, 205, 100, 2, 35, 129, 79, 27,
01000 ; 122, 179, 32, 244, 121, 205, 100, 2, 205, 248, 1, 195, 154, 10, 205,
01010 ; 150, 2, 205, 53, 2, 95, 205, 53, 2, 87, 35, 35, 205, 53, 2, 119, 35, 129,
01020 ; 79, 27, 122, 179, 32, 244, 205, 53, 2, 185, 40, 219, 33, 0, 0, 24, 214
01030 ; PHELPS GATES 6 CRESTWOOD TR. PK. - RT. 4
01040 ; CHAPEL HILL NC 27514 (919)-967-5133
01050 ; END ; 3/5/79

```

```

7FD5
00000 TOTAL ERRORS
READ 7FE8
DONE 7FD5
WRITE 7FC5
INPUT 7FDB

```

```

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 LINE.

```

```

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

```

```

1 DATA"NOP", "00", "00", "000", "00000000", "NOP", "0"
2 DATA"LXI B", "01", "01", "001", "00000001", "LD BC, NN", "2"
3 DATA"STAX D", "02", "02", "002", "00000010", "LD (BC), A", "0"
4 DATA"INX B", "03", "03", "003", "00000011", "INC BC", "0"
5 DATA"INR B", "04", "04", "004", "00000100", "INC B", "0"
6 DATA"DCR B", "05", "05", "005", "00000101", "DEC B", "0"
7 DATA"MVI B", "06", "06", "006", "00000110", "LD B, N", "1"
8 DATA"RLC", "07", "07", "007", "00000111", "RLCA", "0"
9 DATA"DAD B", "09", "09", "011", "00001001", "ADD HL, BC", "0"
10 DATA"LDAX B", "10", "0A", "012", "00001010", "LD A, (BC)", "0"
11 DATA"DCX B", "11", "0B", "013", "00001011", "DEC BC", "0"
12 DATA"INR C", "12", "0C", "014", "00001100", "INC C", "0"
13 DATA"DCR C", "13", "0D", "015", "00001101", "DEC C", "0"
14 DATA"MVI D", "14", "0E", "016", "00001110", "LD C, N", "1"
15 DATA"RRC", "15", "0F", "017", "00001111", "RRC", "0"
16 DATA"LXI D", "17", "11", "021", "00010001", "LD DE, NN", "2"
17 DATA"STAX D", "18", "12", "022", "00010010", "LD (DE), A", "0"
18 DATA"INX D", "19", "13", "023", "00010011", "INC DE", "0"
19 DATA"INR D", "20", "14", "024", "00010100", "INC D", "0"
20 DATA"DCR D", "21", "15", "025", "00010101", "DEC D", "0"
21 DATA"MVI D", "22", "16", "026", "00010110", "LD D, N", "1"
22 DATA"RAL", "23", "17", "027", "00010111", "RLA", "0"
23 DATA"DAD D", "25", "19", "031", "00011001", "ADD HL, DE", "0"
24 DATA"LDAX D", "26", "1A", "032", "00011010", "LD A, (DE)", "0"
25 DATA"DCX D", "27", "1B", "033", "00011011", "DEC DE", "0"
26 DATA"INR E", "28", "1C", "034", "00011100", "INC C", "0"
27 DATA"DCR E", "29", "1D", "035", "00011101", "DEC E", "0"
28 DATA"MVI E", "30", "1E", "036", "00011110", "LD E, N", "1"
29 DATA"RAR", "31", "1F", "037", "00011111", "RRA", "0"
30 DATA"LXI H", "33", "21", "041", "00100001", "LD HL, NN", "2"
31 DATA"SHLD", "34", "22", "042", "00100010", "LD (NN), HL", "2"
32 DATA"INX H", "35", "23", "043", "00100011", "INC HL", "0"
33 DATA"INR H", "36", "24", "044", "00100100", "INC H", "0"
34 DATA"DCR H", "37", "25", "045", "00100101", "DEC H", "0"
35 DATA"MVI H", "38", "26", "046", "00100110", "LD H, N", "1"
36 DATA"DAA", "39", "27", "047", "00100111", "DAA", "0"
37 DATA"DAD H", "41", "29", "051", "00101001", "ADD HL, HL", "0"
38 DATA"LHLD", "42", "2A", "052", "00101010", "LD HL, (NN)", "2"
39 DATA"DCX H", "43", "2B", "053", "00101011", "DEC HL", "0"
40 DATA"INR L", "44", "2C", "054", "00101100", "INC L", "0"
41 DATA"DCR L", "45", "2D", "055", "00101101", "DEC L", "0"
42 DATA"MVI L", "46", "2E", "056", "00101110", "LD L, N", "1"
43 DATA"CMA", "47", "2F", "057", "00101111", "CPL", "0"
44 DATA"LXI SP", "49", "31", "061", "00110001", "LD SP, NN", "2"
45 DATA"STA", "50", "32", "062", "00110010", "LD (NN), A", "2"
46 DATA"INX SP", "51", "33", "063", "00110011", "INC SP", "0"
47 DATA"INR M", "52", "34", "064", "00110100", "INC (HL)", "0"
48 DATA"DCR M", "53", "35", "065", "00110101", "DEC (HL)", "0"
49 DATA"MVI M", "54", "36", "066", "00110110", "LD (HL)", "1"
50 DATA"STC", "55", "37", "067", "00110111", "SCF", "0"
51 DATA"DAD SP", "57", "39", "071", "00111001", "ADD HL, SP", "0"
52 DATA"LDA", "58", "3A", "072", "00111010", "LD A, (NN)", "2"
53 DATA"DCX SP", "59", "3B", "073", "00111011", "DEC SP", "0"
54 DATA"INR A", "60", "3C", "074", "00111100", "INC A", "0"
55 DATA"DCR A", "61", "3D", "075", "00111101", "DEC A", "0"
56 DATA"MVI A", "62", "3E", "076", "00111110", "LD A, N", "1"
57 DATA"CMC", "63", "3F", "077", "00111111", "CCF", "0"
58 DATA"MOV B, B", "64", "40", "100", "01000000", "LD B, B", "0"
59 DATA"MOV B, C", "65", "41", "101", "01000001", "LD B, C", "0"
60 DATA"MOV B, D", "66", "42", "102", "01000010", "LD B, D", "0"

```

61 DATA"MOV B, E", "67", "43", "103", "01000011", "LD B, E", "0"
62 DATA"MOV B, H", "68", "44", "104", "01000100", "LD B, H", "0"
63 DATA"MOV B, L", "69", "45", "105", "01000101", "LD B, L", "0"
64 DATA"MOV B, M", "70", "46", "106", "01000110", "LD B, (HL)", "0"
65 DATA"MOV B, A", "71", "47", "107", "01000111", "LD B, A", "0"
66 DATA"MOV C, B", "72", "48", "110", "01001000", "LD C, B", "0"
67 DATA"MOV C, C", "73", "49", "111", "01001001", "LD C, C", "0"
68 DATA"MOV C, D", "74", "4A", "112", "01001010", "LD C, D", "0"
69 DATA"MOV C, E", "75", "4B", "113", "01001011", "LD C, E", "0"
70 DATA"MOV C, H", "76", "4C", "114", "01001100", "LD C, H", "0"
71 DATA"MOV C, L", "77", "4D", "115", "01001101", "LD C, L", "0"
72 DATA"MOV C, M", "78", "4E", "116", "01001110", "LD C, (HL)", "0"
73 DATA"MOV C, A", "79", "4F", "117", "01001111", "LD C, A", "0"
74 DATA"MOV D, B", "80", "50", "120", "01010000", "LD D, B", "0"
75 DATA"MOV D, C", "81", "51", "121", "01010001", "LD D, C", "0"
76 DATA"MOV D, D", "82", "52", "122", "01010010", "LD D, D", "0"
77 DATA"MOV D, E", "83", "53", "123", "01010011", "LD D, E", "0"
78 DATA"MOV D, H", "84", "54", "124", "01010100", "LD D, H", "0"
79 DATA"MOV D, L", "85", "55", "125", "01010101", "LD D, L", "0"
80 DATA"MOV D, M", "86", "56", "126", "01010110", "LD D, (HL)", "0"
81 DATA"MOV D, A", "87", "57", "127", "01010111", "LD D, A", "0"
82 DATA"MOV E, B", "88", "58", "130", "01011000", "LD E, B", "0"
83 DATA"MOV E, C", "89", "59", "131", "01011001", "LD E, C", "0"
84 DATA"MOV E, D", "90", "5A", "132", "01011010", "LD E, D", "0"
85 DATA"MOV E, E", "91", "5B", "133", "01011011", "LD E, E", "0"
86 DATA"MOV E, H", "92", "5C", "134", "01011100", "LD E, H", "0"
87 DATA"MOV E, L", "93", "5D", "135", "01011101", "LD E, L", "0"
88 DATA"MOV E, M", "94", "5E", "136", "01011110", "LD E, (HL)", "0"
89 DATA"MOV E, A", "95", "5F", "137", "01011111", "LD E, A", "0"
90 DATA"MOV H, B", "96", "60", "140", "01100000", "LD H, B", "0"
91 DATA"MOV H, C", "97", "61", "141", "01100001", "LD H, C", "0"
92 DATA"MOV H, D", "98", "62", "142", "01100010", "LD H, D", "0"
93 DATA"MOV H, E", "99", "63", "143", "01100011", "LD H, E", "0"
94 DATA"MOV H, H", "100", "64", "144", "01100100", "LD H, H", "0"
95 DATA"MOV H, L", "101", "65", "145", "01100101", "LD H, L", "0"
96 DATA"MOV H, M", "102", "66", "146", "01100110", "LD H, (HL)", "0"
97 DATA"MOV H, A", "103", "67", "147", "01100111", "LD H, A", "0"
98 DATA"MOV L, B", "104", "68", "150", "01101000", "LD L, B", "0"
99 DATA"MOV L, C", "105", "69", "151", "01101001", "LD L, C", "0"
100 DATA"MOV L, D", "106", "6A", "152", "01101010", "LD L, D", "0"
101 DATA"MOV L, E", "107", "6B", "153", "01101011", "LD L, E", "0"
102 DATA"MOV L, H", "108", "6C", "154", "01101100", "LD L, H", "0"
103 DATA"MOV L, L", "109", "6D", "155", "01101101", "LD L, L", "0"
104 DATA"MOV L, M", "110", "6E", "156", "01101110", "LD L, (HL)", "0"
105 DATA"MOV L, A", "111", "6F", "157", "01101111", "LD L, A", "0"
106 DATA"MOV M, B", "112", "70", "160", "01110000", "LD (HL), B", "0"
107 DATA"MOV M, C", "113", "71", "161", "01110001", "LD (HL), C", "0"
108 DATA"MOV M, D", "114", "72", "162", "01110010", "LD (HL), D", "0"
109 DATA"MOV M, E", "115", "73", "163", "01110011", "LD (HL), E", "0"
110 DATA"MOV M, H", "116", "74", "164", "01110100", "LD (HL), H", "0"
111 DATA"MOV M, L", "117", "75", "165", "01110101", "LD (HL), L", "0"
112 DATA"HLT", "118", "76", "166", "01110110", "HALT", "0"
113 DATA"MOV M, A", "119", "77", "167", "01110111", "LD (HL), A", "0"
114 DATA"MOV A, B", "120", "78", "170", "01111000", "LD A, B", "0"
115 DATA"MOV A, C", "121", "79", "171", "01111001", "LD A, C", "0"
116 DATA"MOV A, D", "122", "7A", "172", "01111010", "LD A, D", "0"
117 DATA"MOV A, E", "123", "7B", "173", "01111011", "LD A, E", "0"
118 DATA"MOV A, H", "124", "7C", "174", "01111100", "LD A, H", "0"
119 DATA"MOV A, L", "125", "7D", "175", "01111101", "LD A, L", "0"
120 DATA"MOV A, M", "126", "7E", "176", "01111110", "LD A, (HL)", "0"

121 DATA"MOV A, A", "127", "7F", "177", "01111111", "LD A, A", "0"
122 DATA"ADD B", "128", "80", "200", "10000000", "ADD A, B", "0"
123 DATA"ADD C", "129", "81", "201", "10000001", "ADD A, C", "0"
124 DATA"ADD D", "130", "82", "202", "10000010", "ADD A, D", "0"
125 DATA"ADD E", "131", "83", "203", "10000011", "ADD A, E", "0"
126 DATA"ADD H", "132", "84", "204", "10000100", "ADD A, H", "0"
127 DATA"ADD L", "133", "85", "205", "10000101", "ADD A, L", "0"
128 DATA"ADD M", "134", "86", "206", "10000110", "ADD A, (HL)", "0"
129 DATA"ADD A", "135", "87", "207", "10000111", "ADD A, A", "0"
130 DATA"ADC B", "136", "88", "210", "10001000", "ADC A, B", "0"
131 DATA"ADC C", "137", "89", "211", "10001001", "ADC A, C", "0"
132 DATA"ADC D", "138", "8A", "212", "10001010", "ADC A, D", "0"
133 DATA"ADC E", "139", "8B", "213", "10001011", "ADC A, E", "0"
134 DATA"ADC H", "140", "8C", "214", "10001100", "ADC A, H", "0"
135 DATA"ADC L", "141", "8D", "215", "10001101", "ADC A, L", "0"
136 DATA"ADC M", "142", "8E", "216", "10001110", "ADC A, (HL)", "0"
137 DATA"ADC A", "143", "8F", "217", "10001111", "ADC A, A", "0"
138 DATA"SUB B", "144", "90", "220", "10010000", "SUB B", "0"
139 DATA"SUB C", "145", "91", "221", "10010001", "SUB C", "0"
140 DATA"SUB D", "146", "92", "222", "10010010", "SUB D", "0"
141 DATA"SUB E", "147", "93", "223", "10010011", "SUB E", "0"
142 DATA"SUB H", "148", "94", "224", "10010100", "SUB H", "0"
143 DATA"SUB L", "149", "95", "225", "10010101", "SUB L", "0"
144 DATA"SUB M", "150", "96", "226", "10010110", "SUB (HL)", "0"
145 DATA"SUB A", "151", "97", "227", "10010111", "SUB A", "0"
146 DATA"SBB B", "152", "98", "238", "10011000", "SBC A, B", "0"
147 DATA"SBB C", "153", "99", "231", "10011001", "SBC A, C", "0"
148 DATA"SBB D", "154", "9A", "232", "10011010", "SBC A, D", "0"
149 DATA"SBB E", "155", "9B", "233", "10011011", "SBC A, E", "0"
150 DATA"SBB H", "156", "9C", "234", "10011100", "SBC A, H", "0"
151 DATA"SBB L", "157", "9D", "235", "10011101", "SBC A, L", "0"
152 DATA"SBB M", "158", "9E", "236", "10011110", "SBC A, (HL)", "0"
153 DATA"SBB A", "159", "9F", "237", "10011111", "SBC A, A", "0"
154 DATA"ANA B", "160", "A0", "240", "10100000", "AND B", "0"
155 DATA"ANA C", "161", "A1", "241", "10100001", "AND C", "0"
156 DATA"ANA D", "162", "A2", "242", "10100010", "AND D", "0"
157 DATA"ANA E", "163", "A3", "243", "10100011", "AND E", "0"
158 DATA"ANA H", "164", "A4", "244", "10100100", "AND H", "0"
159 DATA"ANA L", "165", "A5", "245", "10100101", "AND L", "0"
160 DATA"ANA M", "166", "A6", "246", "10100110", "AND (HL)", "0"
161 DATA"ANA A", "167", "A7", "247", "10100111", "AND A", "0"
162 DATA"XRA B", "168", "A8", "250", "10101000", "XOR B", "0"
163 DATA"XRA C", "169", "A9", "251", "10101001", "XOR C", "0"
164 DATA"XRA D", "170", "AA", "252", "10101010", "XOR D", "0"
165 DATA"XRA E", "171", "AB", "253", "10101011", "XOR E", "0"
166 DATA"XRA H", "172", "AC", "254", "10101100", "XOR H", "0"
167 DATA"XRA L", "173", "AD", "255", "10101101", "XOR L", "0"
168 DATA"XRA M", "174", "AE", "256", "10101110", "XOR (HL)", "0"
169 DATA"XRA A", "175", "AF", "257", "10101111", "XOR A", "0"
170 DATA"ORA B", "176", "B0", "260", "10110000", "OR B", "0"
171 DATA"ORA C", "177", "B1", "261", "10110001", "OR C", "0"
172 DATA"ORA D", "178", "B2", "262", "10110010", "OR D", "0"
173 DATA"ORA E", "179", "B3", "263", "10110011", "OR E", "0"
174 DATA"ORA H", "180", "B4", "264", "10110100", "OR H", "0"
175 DATA"ORA L", "181", "B5", "265", "10110101", "OR L", "0"
176 DATA"ORA M", "182", "B6", "266", "10110110", "OR (HL)", "0"
177 DATA"ORA A", "183", "B7", "267", "10110111", "OR A", "0"
178 DATA"CP B", "184", "B8", "270", "10111000", "CP B", "0"
179 DATA"CP C", "185", "B9", "271", "10111001", "CP C", "0"
180 DATA"CP D", "186", "BA", "272", "10111010", "CP D", "0"

```

181 DATA"CMP E", "187", "BB", "273", "10111011", "CP E", "0"
182 DATA"CMP H", "188", "BC", "274", "10111100", "CP H", "0B"
183 DATA"CMP L", "189", "BD", "275", "10111101", "CP L", "0"
184 DATA"CMP M", "190", "BE", "276", "10111110", "CP (HL)", "0"
185 DATA"CMP A", "191", "BF", "277", "10111111", "CP A", "0"
186 DATA"RNZ", "192", "C0", "300", "11000000", "RET NZ", "0"
187 DATA"POP B", "193", "C1", "301", "11000001", "POP BC", "0"
188 DATA"JNZ", "194", "C2", "342", "11000010", "JP NZ", "2"
189 DATA"JMP", "195", "C3", "303", "11000011", "JP NN", "2"
190 DATA"CNZ", "196", "C4", "304", "11000100", "CALL NZ, NN", "2"
191 DATA"PUSH B", "197", "C5", "305", "11000101", "PUSH BC", "0"
192 DATA"ADI", "198", "C6", "306", "11000110", "ADD A, N", "1"
193 DATA"RST 0", "199", "C7", "307", "11000111", "RST 0", "0"
194 DATA"RZ", "200", "C8", "310", "11001000", "RET Z", "0"
195 DATA"RET", "201", "C9", "311", "11001001", "RET", "0"
196 DATA"JZ", "202", "CA", "312", "11001010", "JP Z, NN", "2"
197 DATA"CZ", "204", "CC", "314", "11001100", "CALL Z, NN", "2"
198 DATA"CALL", "205", "CD", "315", "11001101", "CALL NN", "2"
199 DATA"ACI", "206", "CE", "316", "11001110", "ADC A, N", "1"
200 DATA"RST 1", "207", "CF", "317", "11001111", "RST 0BH", "0"
201 DATA"RNC", "208", "D0", "320", "11010000", "RET NC", "0"
202 DATA"POP D", "209", "D1", "321", "11010001", "POP DE", "0"
203 DATA"JNC", "210", "D2", "322", "11010010", "JP NC, NN", "2"
204 DATA"OUT", "211", "D3", "323", "11010011", "OUT (N), A", "1"
205 DATA"CNC", "212", "D4", "324", "11010100", "CALL NC, NN", "2"
206 DATA"PUSH D", "213", "D5", "325", "11010101", "PUSH DE", "0"
207 DATA"SUI", "214", "D6", "326", "11010110", "SUB N", "1"
208 DATA"RST 2", "215", "D7", "327", "11010111", "RST 10H", "0"
209 DATA"RC", "216", "D8", "330", "11011000", "RET C", "0"
210 DATA"JC", "218", "DA", "332", "11011010", "JP C, NN", "2"
211 DATA"IN", "219", "DB", "333", "11011011", "IN A, (N)", "1"
212 DATA"CC", "220", "DC", "334", "11011100", "CALL Z, NN", "2"
213 DATA"SBI", "222", "DE", "336", "11011110", "SBC A, N", "1"
214 DATA"RST 3", "223", "DF", "337", "11011111", "RST 10H", "4"
215 DATA"RPO", "224", "E0", "340", "11100000", "RST P0", "0"
216 DATA"POP H", "225", "E1", "341", "11100001", "POP HL", "0"
217 DATA"JPO", "226", "E6", "342", "11100010", "JP P0, NN", "2"
218 DATA"XTHL", "227", "E3", "343", "11100011", "EX (SP), HL", "0"
219 DATA"COP", "228", "E4", "344", "11100100", "CALL P0, NN", "2"
220 DATA"PUSH H", "229", "E5", "345", "11100101", "PUSH HL", "0"
221 DATA"ANI", "230", "E6", "346", "11100110", "AND N", "1"
222 DATA"RST 4", "231", "E7", "347", "11100111", "RST 20H", "0"
223 DATA"RPE", "232", "E8", "350", "11101000", "RET PE", "0"
224 DATA"PCHL", "233", "E9", "351", "11101001", "JP (HL)", "0"
225 DATA"JPE", "234", "EA", "352", "11101010", "JP PE, NN", "2"
226 DATA"XCHG", "235", "EB", "353", "11101011", "EX DE, HL", "0"
227 DATA"CPE", "236", "EC", "354", "11101100", "CALL PE, NN", "2"
228 DATA"XRI", "238", "EE", "356", "11101110", "XOR N", "1"
229 DATA"RST 5", "239", "EF", "357", "11101111", "RST 20H", "0"
230 DATA"RP", "240", "F0", "360", "11110000", "RET P", "0"
231 DATA"POP PSW", "241", "F1", "361", "11110001", "POP AF", "0"
232 DATA"JP", "242", "F2", "362", "11110010", "JP P, NN", "2"
233 DATA"DI", "243", "F3", "363", "11110011", "DI", "0"
234 DATA"CP", "244", "F4", "364", "11110100", "CALL P, NN", "2"
235 DATA"PUSH PSW", "245", "F5", "365", "11110101", "PUSH AF", "0"
236 DATA"ORI", "246", "F6", "366", "11110110", "OR N", "1"
237 DATA"RST 6", "247", "F7", "367", "11110111", "RST 30H", "0"
238 DATA"RM", "248", "F8", "370", "11111000", "RET M", "0"
239 DATA"SPHL", "249", "F9", "371", "11111001", "LD SP, HL", "0"
240 DATA"JM", "250", "FA", "372", "11111010", "JP M, NN", "2"

```

```

241 DATA"EI", "251", "FB", "373", "11111011", "EI", "0"
242 DATA"CM", "252", "FC", "374", "11111100", "CALL M, NN", "2"
243 DATA"CPI", "254", "FE", "376", "11111110", "CP N", "1"
244 DATA"RST 7", "255", "FF", "377", "11111111", "RST 30H", "0"
245 DATA"0", "08", "08", "0", "0", "0", "0"
246 DATA"0", "16", "10", "0", "0", "0", "0"
247 DATA"0", "24", "18", "0", "0", "0", "0"
248 DATA"0", "32", "20", "0", "0", "0", "0"
249 DATA"0", "40", "28", "0", "0", "0", "0"
250 DATA"0", "48", "30", "0", "0", "0", "0"
251 DATA"0", "56", "38", "0", "0", "0", "0"
252 DATA"0", "102", "66", "0", "0", "0", "0"
253 DATA"0", "203", "CB", "0", "0", "0", "0"
254 DATA"0", "217", "D9", "0", "0", "0", "0"
255 DATA"0", "221", "DD", "0", "0", "0", "0"
256 DATA"0", "237", "ED", "0", "0", "0", "0"
257 DATA"0", "253", "FD", "0", "0", "0", "0"
500 DATA"SHLD", "34", "22", "042", "00100010", "LD (NN), HL", "2"
510 DATA"STA", "50", "32", "062", "00110010", "LD (NN), A", "2"

```

```

520 PRINT "INPUT THE STARTING ADDRESS FOR YOUR PROGRAM (DECIMAL) "
525 INPUT "DECIMAL 57344 - HEX E000 "; B
530 A=0-65536
540 PRINT:PRINT"IF YOU ARE ASKED FOR AN OPERAND, YOU MUST INPUT THAT"
550 PRINT:"INFO IN HEX, 8 BIT (EX. 5D) OR 16 BIT (EX. A05E)":PRINT
560 PRINT:LINEINPUT "INPUT THE 0000 SOURCE CODE ('X' TO STOP) "; A$
570 IF A$="X" THEN END
580 PRINT
590 READ B$, C$, D$, E$, F$, G$, H$
600 IF A$=B$ THEN 620
610 GOTO 590
620 PRINT "200          0000          HEX OCT DECIMAL BINARY"
630 PRINT G$:TAB(12); B$:TAB(24); D$:TAB(29); E$:TAB(36); C$:TAB(43)
640 FOR Z=1 TO 8
650 PRINT MID$(F$, Z, 1); " ";
660 NEXT Z:RESTORE
670 POKE A, VAL(C$)
680 A=A+1:PRINT:H=VAL(H$)
690 IF H=0 THEN 900
700 IF H=1 THEN 720
710 IF H=2 THEN 770
720 PRINT:INPUT"INPUT A 8 BIT HEX NUMBER "; D1$
730 RESTORE
740 READ B$, C$, D$, E$, F$, G$, H$
750 IF D1$=D0$ POKE A, VAL(C$):A=A+1:GOTO 860
760 GOTO 740
770 PRINT:INPUT"INPUT A 16 BIT HEX NUMBER "; D2$
780 RESTORE
790 READ B$, C$, D$, E$, F$, G$, H$
800 IF RIGHT$(D2$, 2)=D0$ POKE A, VAL(C$):A=A+1:GOTO 820
810 GOTO 790
820 RESTORE
830 READ B$, C$, D$, E$, F$, G$, H$
840 IF LEFT$(D2$, 2)=D0$ POKE A, VAL(C$):A=A+1:GOTO 860
850 GOTO 830
860 IF MID$(G$, 4, 4)="(NN)" THEN G1$=LEFT$(G$, 4):G2$=MID$(G$, 7, 4):GOTO 960
870 IF MID$(G$, 5, 3)="(N)" THEN G1$=LEFT$(G$, 5):G2$=MID$(G$, 7, 3):GOTO 900
880 IF RIGHT$(G$, 3)="(N)" THEN G1$=LEFT$(G$, 5):GOTO 970
890 IF RIGHT$(G$, 4)="(NN)" THEN G$=LEFT$(G$, LEN(G$)-4):GOTO 950
900 IF RIGHT$(G$, 2)="NN" THEN G$=LEFT$(G$, LEN(G$)-2):GOTO 940
910 IF RIGHT$(G$, 1)="N" THEN G$=LEFT$(G$, LEN(G$)-1)
920 PRINT:LPRINT D$; " "; D1$:TAB(15); G$:D1$:GOTO 990

```

```

940 PRINT:LPRINT D$;" ";RIGHT$(D2$,2);" ";LEFT$(D2$,2);TAB(15);G$;D2$;GOTO 990
950 PRINT:LPRINT D$;" ";RIGHT$(D2$,2);" ";LEFT$(D2$,2);TAB(15);G$;"(";D2$;")":GOTO 990
960 PRINT:LPRINT D$;" ";RIGHT$(D2$,2);" ";LEFT$(D2$,2);TAB(15);G1$;D2$;G2$;GOTO 990
970 PRINT:LPRINT D$;" ";D1$;TAB(15);G1$;"(";D1$;")":GOTO 990
980 PRINT:LPRINT D$;" ";D1$;TAB(15);G1$;D1$;G2$
990 RESTORE:D1$="":D2$="":GOTO 560

```

IF YOU LIKE TO USE SHORT UTILITY ROUTINES (<1.5K OR LESS) WRITTEN IN 8080 MNEMONIC - HERE IS THE PROGRAM FOR YOU. (28080)

SAMPLE RUN * * *

INPUT THE STARTING ADDRESS FOR YOUR PROGRAM (DECIMAL)
DECIMAL 57344 - HEX E000 57344

IF YOU ARE ASKED FOR AN OPERAND, YOU MUST INPUT THAT
INFO IN HEX. 8 BIT (EX. 5D) OR 16 BIT (EX. A85E)

INPUT THE 8080 SOURCE CODE (<'X' TO STOP) LDA

```

Z80      8080      HEX OCT DECIMAL BINARY
LD A,(NN) LDA      3A 072 58 0 0 1 1 1 0 1 0

```

```
3A 03 05      LD A,(0503)
```

INPUT THE 8080 SOURCE CODE (<'X' TO STOP) ADI

```

Z80      8080      HEX OCT DECIMAL BINARY
ADD A,N  ADI      C6 306 198 1 1 0 0 0 1 1 0

```

```
C6 D4      ADD A,D4
```

INPUT THE 8080 SOURCE CODE (<'X' TO STOP) STAX D

```

Z80      8080      HEX OCT DECIMAL BINARY
LD (BC),A STAX D  02 002 02 0 0 0 0 0 0 1 0

```

```
02      LD (BC),A
```

INPUT THE 8080 SOURCE CODE (<'X' TO STOP) SHLD

```

Z80      8080      HEX OCT DECIMAL BINARY
LD (NN),HL SHLD  22 042 34 0 0 1 0 0 0 1 0

```

```
22 6E A4      LD (A46E),HL
```

INPUT THE 8080 SOURCE CODE (<'X' TO STOP) X

RSM-1 V-2.1 (C) 1978
SMALL SYSTEM SOFTWARE
COMMAND?

D E000 E00F
E000: 3A 03 05 C6 D4 02 22 6E A4 00 00 00 00 00 00 00
COMMAND?

MARCH 25, 1979

DEAR BOB,

ONE OF THE FILE COMMANDS OF THE DISK BASIC IS MERGE. THIS IS A HANDY COMMAND TO HAVE. IT ALLOWS YOU TO MERGE TWO PROGRAMS TOGETHER OR TO KEEP A FILE OF SUBROUTINES THEN MERGE THEM WITH A NEW PROGRAM AS YOU WRITE. THIS WILL SAVE YOU TIME AND MAKE THOSE LONG PROGRAMS EASY.

YOU CAN DO THE SAME THING WITH LEVEL II BASIC USING THE PEEK, AND POKE COMMANDS. THE FIRST THING YOU MUST DO IS EXTEND LEVEL II RESERVED MEMORY TO PROTECT THE FIRST PROGRAM, THEN LOAD YOUR SECOND PROGRAM OR SUBROUTINE. AFTER LOADING BOTH PROGRAMS THEN YOU JUST RESET THE ENDING ADDRESS OF THE RESERVED MEMORY, AND YOUR PROGRAMS ARE MERGED.

THE MEMORY ADDRESS 16548-9 HOLDS THE ENDING ADDRESS OF LEVEL II RESERVED MEMORY, AND 16635-6 THE ENDING ADDRESS OF ARRAYS (?). AFTER YOU LOAD THE FIRST PROGRAM TYPE (< WITHOUT LINE NUMBER, OR SPACE) A=PEEK(16635):B=PEEK(16636):?A,B AND THEN ENTER. THIS WILL NOT BE A VALID ADDRESS IF YOU RAN THE PROGRAM. IF YOU NEED TO RUN THE PROGRAM YOU MUST FIRST DROP THE MEMORY USED TO STORE SIMPLE VARIABLES AND ARRAYS. THIS MAY BE DONE BY GOING INTO THE EDIT MODE. THIS WILL GIVE YOU THE ENDING ADDRESS OF YOUR PROGRAM PLUS THE MEMORY USED BY THE PEEK STATEMENT (< 9 BYTES). NOW TYPE POKE 16548,(A-9) AND POKE 16549,B. REMEMBER THE VALUES USED FOR THE PEEK AND POKE STATEMENTS ARE IN DECIMAL FORM, HOWEVER THEY REPRESENT A HEXIDECIMAL NUMBER. IF THE VALUE OF A IS LESS THAN 9 YOU MUST REDUCE B BY 1 AND INCREASE A BY 256.

YOU MAY NOW LOAD YOUR SECOND PROGRAM, OR SUBROUTINE. THE FIRST LINE OF THE SECOND PROGRAM MUST BE LARGER # THAN THE LAST LINE OF FIRST PROGRAM. THE FEB. 79 NEWSLETTER HAS A GOOD RENUMBERING PROGRAM ON PAGE 20 BY PHELPS GATES.

THE LAST ITEM IS TO RESET THE RESERVED MEMORY. THIS IS DONE BY TYPING POKE 16548,233 AND POKE 16549,66. AND YOUR PROGRAMS ARE MERGED AND REDY TO RUN OR CSAVE.

SINCERELY

JAMES LIMKEMANN
7672 BRANCH WOOD CTR.
FAYETTEVILLE NC. 28304
(919) 867-6286

TRS-80 QUALITY SOFTWARE

LEVEL I, AND LEVEL II.

#1. IDM-I CASSETTE DATA BASE	\$20.
#2. INV-I INVENTORY CONTROL	\$20.
#3. STOCK-I SECURITY INFO.	\$10
#4. BANK-I CHECK BALANCE	\$10.
#5. FINANCE-I STOCK-I & BANK-I	\$15.

LEVEL II

#11. WORD-I WORD PROCESSOR	\$25.
#15. MAIL-I NAME AND ADDRESS	\$25.
#16. SORT-I SORT UTILITY	\$10.
#17. STAT-I STATISTICS	\$10.
#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 DOTHAN ST.
ARLINGTON, MA 02174

PRICE INCLUDES POSTAGE, CASSETTER & DOCU.

GENERAL LEDGER FOR TRS-80

THE PROGRAMS CAN GENERATE SEVERAL REPORTS.

- 1 PRINT OUT FOR JOURNAL
- 2 TRIAL BALANCE.
- 3 BALANCE SHEET.
- 4 PROFIT AND LOSS STATEMENT.
- 5 PRINT A CHARTS OF ACCOUNTS.

YOU CAN LIST THE ENTRIES TO THE DISPLAY BY REFERENCE #, LIST ALL ENTRIES LIST 80-90 TO THE PRINTER. YOU CAN ADD OR DELETE ACCOUNTS. YOU CAN POST ALL ENTRIES TO THE YEAR TODATE COLOUMNS AT THE END OF A PERIOD AVAILBLE TO RUN ON ONE DRIVE THE FILES ARE ON DRIVE(1), THE PROGRAMS ARE ON DRIVER(0). THIS ALLOWS APPRO. 83. K STORAGE ON ONE DESKETTE.

ONE DISK SYSTEM \$75.00 W/1.50 P/H.

TWO DISK SYSTEM \$80.00 WE PAY P/H.

ALLOW 2 WEEKS FOR SHIPMENT.

PLEASE SEND MONEY ORDER TO T. H. BAILEY SR.

T. H. BAILEY SR.
P. O. BOX 5319
AUGUSTA, GA. 30906

Dear Mr. Lloyd:

An inquiry on my SURVEY program published in the Volume 2, #1 Newsletter led to the development of the inclosed 16K Level II program. It computes the area of irregular shaped lots from dimensions shown on plats. The dimensions are used to find arbitrary coordinates of significant points. These coordinates are entered into the program to calculate the areas. When one or more circular segments are involved, the radii must also be entered. The program also computes the angles and azimuths of the lines in radians. Azimuths are counterclockwise from east. The program could be modified to convert angles and azimuths to degrees.

The inclosed sketch shows the results of the use of the program on a sample plat. It illustrates the use of interior, exterior and supplementary angles for the circular segments.

If those interested will send me a cassette and self-addressed stamped envelope (28¢), I will send them a copy of either SURVEY, AREA, or both. Also, if requested, PASSBOOK from Volume 1, #7; SKETCH from Volume 2, #1; and JOTTO, Volume 2, #3. I also have PASSBOOK and SURVEY in Level I. If I should get swamped with requests, there might be some delay.

The AREA program will load in 4K but must be abbreviated by deleting remarks in order to have enough memory to run.

Yours very truly,

C. W. Evans
C. W. Evans

```
10 'AREA---POLYGON +/- CIRCULAR SEGMENTS
20 'BY C. W. EVANS, (602) 933-1616
30 '9806 AMBER TRAIL, SUN CITY, AZ 85351
40 '3/3/79
50 'RUN 1000 FOR TEST
```

```
100 '-----INSTRUCTIONS-----
120 CLS
130 PRINT"THIS PROGRAM COMPUTES THE AREA OF A POLYGON.
140 PRINT" +/- CIRCULAR SEGMENTS FROM THE COORDINATES OF THE VERTICES.
150 PRINT"THE VERTICES ARE NUMBERED, IN SEQUENCE,
160 PRINT" CLOCKWISE AROUND THE FIGURE. ":PRINT
170 PRINT"X AND Y COORDINATES, SEPARATED BY A COMMA, BUT WITHOUT
180 PRINT" PARENTHESES, ARE ENTERED IN SEQUENCE. ":PRINT
190 PRINT"IT MAY BE CONVENIENT TO ARRANGE COORDINATES
200 PRINT" SO THAT ONE LINE OF A FIGURE LIES ON THE X AXIS.
210 PRINT:PRINT"USUALLY NEGATIVE COORDINATES SHOULD BE AVOIDED.
220 PRINT:PRINT
230 INPUT"HIT ENTER WHEN READY";Z$
240 '-----ENTER COORDINATES-----
250 CLS:INPUT"HOW MANY POINTS ARE THERE ON THE POLYGON? ";N
260 P=1:PRINT
270 PRINT"WHAT ARE THE COORDINATES OF POINT # ";P;" ";
280 INPUT X(P),Y(P)
290 IF P=N THEN 320
300 P=P+1
310 GOTO 270
320 '-----CHANGE COORDINATES?-----
330 PRINT:INPUT"ENTER NUMBER OF ANY POINT WHOSE COORDINATES
ARE TO BE CHANGED. ENTER 99 IF NONE";P
340 IF P=99 THEN X(0)=X(N):Y(0)=Y(N):
X(N+1)=X(1):Y(N+1)=Y(1):GOTO 400
350 PRINT"WHAT ARE COORDINATES OF POINT # ";P;:INPUT X,Y
360 X(P)=X:Y(P)=Y
370 GOTO 330

400 '-----COMPUTE ANGLES-----
410 PI=3.14159' Z=DIRECTION OF LINE, W=INTERIOR ANGLE
420 FOR P=0 TO N
430 A=X(P):E=Y(P)
440 B=X(P+1):F=Y(P+1)
450 Z=ATN( (F-E)/(B-A+.000001) )
460 IF ADB THEN Z=Z+PI' FOR QUADRANTS II OR III
470 IF Z<0 THEN Z=2*PI+Z
480 IF Z>2*PI THEN Z=Z-2*PI
490 Z(P)=Z
500 NEXT P
510 FOR P=1 TO N
520 W(P)=Z(P)-Z(P-1)+PI
530 IF W(P)<0 THEN W(P)=W(P)+2*PI
540 IF W(P) >= 2*PI THEN W(P)=W(P)-2*PI
550 NEXT P
560 '-----COMPUTE AREA OF POLYGON-----
570 K=0' K=AREA
580 FOR P= 1 TO N
590 D(P)=(X(P)+X(P+1))*(Y(P)-Y(P+1))' D=2 TIMES AREA DIFF
600 K=K+D(P)/2
610 NEXT
620 '-----PRINT TABLE-----
630 CLS:PRINT" COORDINATES OF POLYGON--Z=DIRECTION, W=ANGLE":PRINT
640 PRINT" P X Y Z W
650 FOR P=1 TO N
```

```

660 PRINT P; X(P),Y(P),Z(P),W(P)
670 NEXT
680 PRINT " AREA = ";K
690 /-----CIRCULAR SEGMENTS-----
700 INPUT"CIRCULAR SEGMENT INVOLVED?? (Y/N) ";S$
710 IF S$="N" PRINT:PRINT " FOR ANOTHER LOT--";GOTO 230
720 INPUT"WHAT IS THE POINT NUMBER OF THE CENTER ";P
730 INPUT"WHAT IS THE RADIUS ";R
740 INPUT"IS ANGLE INTERIOR, EXTERIOR, OR SUPPLEMENTARY
(TYPE I, E, OR S) ";W$
750 IF W$="I" THEN 790
760 IF W$="E" THEN W(P)=2*PI-W(P): GOTO 790
770 IF W$="S" THEN W(P)=PI-W(P): GOTO 790
780 GOTO 740
790 PRINT"THE SEGMENT ANGLE, ARC, AND AREA ARE: ";W(P);W(P)*R;W(P)*R*R/2
800 INPUT"IS THE SEGMENT AREA ADDED TO THE POLYGON (Y/N)";S$
810 IF S$="Y" THEN J=1: GOTO 830
820 IF S$="N" THEN J=-1 ELSE 800
830 K=K+J*W(P)*R*R/2
840 PRINT"THE ADJUSTED AREA IS: ";K
850 PRINT"IS ANOTHER ";: GOTO 700
1000 /-----INPUT FOR TEST-----
1010 X(1)=0:Y(1)=195
1020 X(2)=99.38:Y(2)=195
1030 X(3)=160:Y(3)=160
1040 X(4)=0:Y(4)=100
1050 P=4:N=4:GOTO 330

```

9806 Amber Trail
Sun City, AZ 85351
(602) 933-1616
March 26, 1979

TRS-80 Users Group Newsletter
7554 Southgate Road
Fayetteville, NC 28304

Hey,

it was great to see my letter in your publication, but I now find I have one small error in terminology: not just once, but twice, I referred to the circuit as a "lowpass" filter, when in fact it is a highpass filter.

The TRS-80 responds better to the higher frequency tone, and that's one of the main purposes of the filter, to be sure that the highs are passed and the lows suppressed.

Oh well, I guess everyone is entitled to build one Edsel in his lifetime.

Sincerely and 73,

Wesley

WB5TSS
COL. J. WESLEY B. TAYLOR
P. O. Box 4391
Wichita Falls, Texas 76308

```

29740 'EVER WISH YOU COULD LOAD A LEVEL II BASIC PROGRAM WITHOUT ERASING THE PROGRAM
29750 'THAT'S ALREADY IN THE COMPUTER -- TO ADD A UTILITY SUBROUTINE AT THE END
29760 'OF A PROGRAM, FOR EXAMPLE?
29765 '
29770 'ORDINARILY, OF COURSE, CLOAD WILL OVERWRITE THE RESIDENT PROGRAM -- THE
29780 'COMPUTER READS THE TAPE INTO MEMORY STARTING AT THE ADDRESS STORED IN
29790 '16548-9, AND THIS IS NORMALLY 17129. BUT IT WILL ADD THE TAPE PROGRAM
29800 'ONTO THE END IF YOU POKE THE END ADDRESS OF THE RESIDENT PROGRAM (THE CONTENTS
29810 'OF 16633-34, LESS 2) INTO 16548-9, AS FOLLOWS:
29815 '
29820 '(1) POKE 16548,PEEK(16633)-2: POKE 16549,PEEK(16634) -- IF THE PROGRAM
29830 'HAPPENS TO END AT A PAGE BOUNDARY, YOU'LL GET A 2FC ERROR, AND IN THAT
29840 'CASE, POKE 16548,PEEK(16633)+254: POKE 16549,PEEK(16634)-1
29845 '
29850 '(2) NOW TYPE NEW (TO RESET POINTERS) AND CLOAD THE PROGRAM TO BE ADDED
29860 '(ITS LINE NUMBERS MUST ALL BE > THE NUMBERS OF THE RESIDENT PROGRAM)
29865 '
29870 '(3) POKE 16548,233: POKE 16549,66 -- THIS RESTORES THE FIRST PART OF
29880 'THE PROGRAM. YOU CAN NOW LIST THE MERGED PROGRAM, BUT IT WON'T RUN UNLESS
29890 'YOU CALL 6904 (IN THE ROM), TO FIX UP LINE POINTERS:
29900 '(4) POKE 16526,248: POKE 16527,26: X=USR(0)
29901 '
29905 ' 2/24/79
29910 ' H PHELPS GATES
29920 ' 6 CRESTWOOD TR. PK. - RT. 4
29930 ' CHAPEL HILL, NC 27514 (919)-967-5193
29940 'THE FOLLOWING IS A HANDY DEBUGGING SUBROUTINE WHICH LISTS ALL LEVEL II
29950 'SIMPLE VARIABLES IN A PROGRAM, WITH THEIR VALUES. SINCE LEVEL II ASSIGNS
29960 'VARIABLE SPACE AT RUN-TIME, THIS SUBROUTINE WON'T PRINT ANYTHING UNTIL
29970 'AFTER THE PROGRAM RUNS -- YOU CAN LOAD THIS ROUTINE WITH THE PROCEDURE
29980 'DESCRIBED ABOVE
29990 END
30000 XZ!=0: XZ1=0: XZ2=0: XZ3=""
30010 'EXPLICIT DECLARATION OVERRIDES ANY DEF STATEMENTS IN MAIN PROGRAM
30020 VZ!(3)=VARPTR(XZ#): VZ!(2)=VARPTR(XZ%): VZ!(4)=VARPTR(XZ!): VZ!(8)=VARPTR(XZ@)
30030 FORNZ!=PEEK(16633)+256*PEEK(16634) TO PEEK(16635)+256*PEEK(16636)-1: TZ!=PEEK(NZ!)
30040 'THIS SUBROUTINE USES VARIABLES WHOSE SECOND LETTER IS Z, AND SUCH
30050 'VARIABLES ARE SUPPRESSED FROM LISTING. TO LIST ALL, EDIT THIS LINE
30060 IFPEEK(NZ!+1)=90 THEN 30110 ELSE PRINTCHR$(PEEK(NZ!+2));CHR$(PEEK(NZ!+1));
30070 'THE CONTENTS OF THE VARIABLE ARE POKED INTO THE DUMMY VARIABLE (XZ)
30080 'OF THE APPROPRIATE TYPE, WHICH IS THEN PRINTED
30085 PRINTID$(CHR$(PEEK(NZ!+1));CHR$(PEEK(NZ!+2));TZ!,1); " ";
30090 FORKZ!=@TZ!-1: POKEVZ!(TZ!)+KZ!,PEEK(NZ!+3+KZ!): NEXT
30100 IFTZ!=3THENPRINTXZ%,ELSEIFTZ!=2THENPRINTXZ%,ELSEIFTZ!=4THENPRINTXZ!,ELSEPRINTXZ@,
30110 NZ!=NZ!+TZ!+2: NEXTNZ!: PRINT: RETURN

```

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) 535-3142

Dear Gordon:

Enclosed is my renewal for another year of the TRS-80 Users Group Newsletter. I preferred the previous booklet format, but I suppose I'll get used to the current loose-leaf format. The main problem is that now all issues cannot be filed together.

I greatly appreciated the BASIC renumbering program by H. Phelps Gates in the February issue, but it does contain one small error: in line 400, the PRINT statement D=256*EH should be D-256*EH

For my longer programs, I have increased space for the program to be renumbered from 10701 bytes to 12701 bytes by leaving out all REMarks and spaces, and by making a few other minor changes. To do this, change statements in line 50:

DIR T(1000) to DIR T(800)
ST=22000 to ST=20000

Also change statements in line 390:

POKE 21743,0 to POKE19743,0
POKE 16548,240 to POKE16548,32
POKE 16549,84 to POKE16549,77

Then to run, enter manually:

POKE 16548,32 (instead of POKE 16548,240)
POKE 16549,78 (instead of POKE 16549,85)
POKE 19999,0 (instead of POKE 21999,0)
NEW (same, except ?NEW gives 12701)

The rest of the operation is the same as originally.

Sincerely,



Delmer D. Hinrichs
2116 S. E. 377th Ave.
Washougal, Washington 98671

a currently running (error-free) program can place you back into control with a Poke 16405,1. Also don't hit reset or else your program will be lost since you'll have to power down. In another Poke you can break the link to your video monitor. Poke 16413,1. Keyboard is still in control but there is no current print-out on the monitor. To regain the display, put in your program (or type carefully) Poke 16413,7.

4. How you can tell if a key is currently being pressed on your TRS-80 Level II
This is an excellent feature for real time games, doodle programs, etc. Unlike INKEY\$, which only records the initial depression of a key, this technique can be used to tell if someone is currently pressing a particular key. This would allow you to write Software like... Holding down the left arrow will move a paddle to the left, holding down the right arrow will move a paddle to the right and not pressing any keys will keep the paddle stable. Not wanting to take up more space in Gordon's excellent newsletter and being a money-starved capitalist in need of an expansion interface, I am selling a full report on how to access and use this hidden feature, as well as some software using the technique for a mere \$2.00. Please send check or money order to...
Michael Chertok
20 Butternut Dr.
Pearl River, N.Y. 10965

Dear Gordon and Readers,

I have some interesting tips and techniques for Level II users.

1. Make your own 3-d graphic characters
First design your character using the boxes on the video display worksheet. Then figure out the graphics character code for each print position (2x3 area) across the screen. You can do this using the method described by Steve Mac Gregor in Newsletter #10. When you've printed all the characters you want on the current line, put in the control code for a down cursor (#26), and a few back cursors (#24), until you reached the desired position on that line. My program shows the technique more clearly.
2. Important for noisemakers
According to official Radio Shack releases, flicking your relay may be hazardous to your TRS-80. Don't do it!
3. Your TRS-80 can now hibernate
Turn your keyboard off to the world. Just Poke 16405,2. But remember, only

```
10 DIS: CLEAR 500
20 FOR K=1 TO 76: READ A: M$=M$+CHR$(A): NEXT K
30 PRINT M$
100 DATA 32,130,164,32,32,152,129,32,26,24
110 DATA 24,24,24,24,24,24,32,32,191,187
120 DATA 183,191,32,32,26,24,24,24,24,24
130 DATA 24,24,32,32,152,191,191,164,32,32
140 DATA 26,24,24,24,24,24,24,24,32,130
150 DATA 32,151,171,32,129,32,26,24,24,24
160 DATA 24,24,24,24,24
170 DATA 32,130,140,133,138,140,129,32
```

TRS-80 BUS CONNECTORS FOR HOME-BREW INTERFACES. TOP QUALITY VIKING AND 3-M. GOLD PLATED CONTACTS. SOLDERLESS TYPE TERMINATED IN 18" 40 CONDUCTOR RIBBON CABLE \$9.95. SOLDER TAIL TYPE \$6.95. PRICES POSTPAID. NY RESIDENTS ADD 6% TAX. APPLIED INVENTION, RD. 2, RT. 21, HILLSDALE, NY 12529. INQUIRE ABOUT OUR PARALLEL PRINTER INTERFACE FOR LEVEL II.

YOU OR YOUR SUBSCRIBERS MAY BE INTERESTED IN A WAY OF IMPRESSING FRIENDS WHEN RUNNING TRS-80 MICROCHESS. THE FOLLOWING MOVES WILL BEAT THE MACHINE IN 6 MOVES:

1-KING'S PAWN TO KP-2 E2-E4 1-KING'S PAWN TO KP-2 E7-E5
2-KB TO QB-4 F1-C4 2-QP TO P2 D7-D5
3-KB TO QKNT-3 C4-B3 PAWN TAKES PAWN D5-E4
KBP TO KBP-3 F2-F3 4-PAWN TAKES PAWN E4-F3
5-QUEEN TAKES PAWN D1-F3 5-KP TO KP-5 E5-E4
6-QUEEN TAKES PAWN F3-F7 CHECKMATE!!

THESE MOVES CHECKMATE MICROCHESS WITHOUT REGARD AS TO WHAT LEVEL (1, 2, OR 3) IT IS IN. I HAVE GREAT SPORT WHEN SHOWING MY TRS-80 TO CHESSPLAYING FRIENDS BECAUSE THEY DON'T BELIEVE THE PROGRAM IS AS SOPHISTICATED AS IT IS. THEY WILL IMMEDIATELY SIT DOWN TO WHIP THE MACHINE, AND 30 MINUTES LATER ARE PULLING THEIR HAIR OUT.

I ENJOY THE NEWSLETTER IMMENSELY, BUT COULD YOU HELP WITH RUNNING MACHINE LANGUAGE PROGRAMS OR SUBROUTINES WITHOUT USING DEBUG OR OTHER ASSEMBLERS?

THE NECESSARY STEPS ARE NOT TOO CLEAR IN THE LEVEL II MANUAL.

HAVE YOU INTERFACED ANYTHING OTHER THAN RADIO SHACK EQUIPMENT TO YOUR TRS-80 IF SO, I WOULD LIKE TO HEAR OF YOUR EXPERIENCES. PLEASE INCLUDE: TYPE OF EQUIPMENT, BRAND NAME, COSTS, INTENDED USES, PROBLEMS, DOWNTIME, GENERAL SPECIFICATIONS, ETC.

IF A REPRESENTATIVE SAMPLE OF INFORMATION IS RECEIVED FOR A PARTICULAR PIECE OF EQUIPMENT, I WILL COMPILE THE DATA INTO AN EQUIPMENT REVIEW FOR THE NEWSLETTER. WE COULD ALL BENEFIT FROM YOUR EXPERIENCE!

SEND INFORMATION TO:

DAVID R. KIMBEL
RT 9 BOX 650
FAYETTEVILLE, NC 28301

THE BITPICKER'S TOOLBOX

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

The following one-line program will cause LPRINT statements to behave like PRINT statements:

```
FOR I=16421 TO 16428: POKE I,PEEK(I-8): NEXT I
```

This is useful for developing programs to run on a TRS-80 with a printer when you don't have one yourself, or for running programs with LPRINT statements without editing them all to PRINT. It works by copying the Video Device Control Block into the area for the Line Printer Control Block, so that whenever BASIC makes a call to the printer driver, it gets the video driver instead. In effect, you have two cursors on the screen at the same time. However, it is not advisable to do both PRINT and LPRINT, as the automatic scrolling caused when one cursor reaches the end of the screen will scroll the whole screen without affecting other cursor. Also, a CLS command will not reset the "LPRINT" cursor to the upper left corner, as it does the "PRINT" cursor. Try this some time to see how it looks, and get a feel for these limitations.

Dear Gordon,

Enclosed is a generalized version of the code deduction game variously called "Mastermind", "Bulls and Cows", etc., for Level 1, 4K. The player can vary the degree of difficulty by setting the code generation parameters. The program in turn gives a measure of the degree of difficulty by computing the number of non-unique codes that could be generated. It is interesting to see how many guesses on average it takes to break the code as a function of the difficulty.

Notes on Superbagels:

Variables used:

G Collects each guess as an integer (11 - 999999)
W Counts digits of correct value, wrong position
B Counts correct digits
A(1)-A(6) Digits of the code to be guessed
A(7)-A(12) Digits of the current guess
A(13)-A(18) 0 if code digit is available to be matched
1 if code and guess digits match
2 if code digit has been matched by value
A(19) "B" value of guess #1
A(20) "W" value of guess #1
A(21) Integer guess #1
A(22)-A(24) ... results of guesses #2 thru #96 (eomem)

Logic employed:

100-120 Sets the randomizer
210-260 Collects the game options. Last half of 260 checks that enough digit values (C) are available for the number of digits (D) chosen if duplicates are not allowed (E=1)
270 Computes C to the D power
280 Computes C factorial divided by D factorial
300-330 Generates the target code. If dupls are not allowed but the current digit is found to be one (flag F=1), the current digit is rejected
440-450 Range checks the guess for the right number of digits (G must be between 10 to D power and 10 to D minus 1)
460-490 Stores the guess, breaks the integer into digits, range checks the digits, and counts the correct digits (B)
510-535 If the code was not guessed, each remaining guess digit is compared to each remaining code digit, with matched code digits being marked against further matching
1000-1070 Displays previous guesses and results

Ted M. Hau
Syntropic Systems
St. Louis, Mo.

```

10 N=1:V=2:CLS
20 INPUT"WELCOME TO SUPERBAGELS. INSTRUCTIONS (Y/N) ";I
30 ON I GOTO 100
40 PRINTTAB(5);"SUPERBAGELS IS A DEDUCTION GAME IN WHICH YOU ARE TO"
45 PRINT"ATTEMPT TO GUESS A STRING OF DIGITS (OR CODE) GENERATED BY THE"
50 PRINT"PROGRAM. THE PROGRAM WILL INFORM YOU AFTER EACH GUESS HOW MANY"
55 PRINT"OF THE DIGITS IN YOUR GUESS ARE IN THE PROPER POSITION (P)."
```

```

60 PRINT"AND HOW MANY DIGITS ARE OF THE PROPER VALUE (V) BUT IN THE"
65 PRINT"WRONG POSITION. YOU MAY REVIEW YOUR PREVIOUS GUESSES OR"
70 PRINT"GIVE UP AT ANY TIME."
75 PRINT TAB(5);"YOU MAY INCREASE THE DIFFICULTY OF THE GAME BY INCREAS-"
80 PRINT"ING THE NUMBER OF POSITIONS IN THE CODE, BY INCREASING THE"
85 PRINT"NUMBER OF VALUES AVAILABLE FOR EACH POSITION (0 IS NOT USED)."
```

```

90 PRINT"AND BY ALLOWING THE SAME VALUE IN MORE THAN ONE POSITION."
95 PRINT TAB(5);"HAPPY DEDUCING!"
100 INPUT"TO RANDOMIZE THE PROGRAM, ENTER A NUMBER BETWEEN 1-100 ";K
110 IF (K<1)+(K>100) GOTO 100
120 FOR I=1 TO K:J=RND(32767):NEXT I
200 CLS
210 PRINTAT0:PRINTAT0;:INPUT"ENTER MAXIMUM DIGIT VALUE (2-9) ";C
220 C=INT(C):IF (C<2)+(C>9) GOTO 210
230 PRINTAT64:PRINTAT64;:INPUT"ALLOW DUPLICATE DIGIT VALUES (Y/N) ";E
240 IF (E<1)+(E>2) GOTO 230
250 PRINTAT120:PRINTAT120;:INPUT"ENTER NUMBER OF DIGITS (2-6) ";D
260 D=INT(D):IF (D<2)+(D>6)+(E=1)*(C<D)) GOTO 250
270 P=1:IF E=2 FOR I=1 TO D:P=P*C:NEXT I
280 IF E=1 FOR I=1+C-D TO C:P=P*I:NEXT I
290 PRINT"ONE OF ";P;" POSSIBLE VALUES IS NOW BEING GENERATED":I=1:A=1
300 A(I)=RND(C):IF (E=2)+(I=1) GOTO 330
310 F=0:FOR J=1 TO I-1:IF A(J)=A(I) F=1
320 NEXT J:IF F=1 GOTO 300
330 IF D>I THEN I=I+1:GOTO 300
400 W=0:B=0:FOR I=1 TO D:A(I+12)=0:NEXT I
410 PRINT"REVIEW (1), GIVE UP (0), OR GUESS ";A;" ";:INPUT G
420 IF G<0 GOTO 410
430 IF G=0 PRINT"VALUE WAS ";:FOR I=1 TO D:PRINT A(I);:NEXT I:PRINT:GOTO 950
435 IF (G=1)+(A<1) GOSUB 1000:PRINT:GOTO 410
440 J=1:FOR I=1 TO D:J=10*J:NEXT I:IF G=J GOTO 410
450 J=J/10:IF G<J GOTO 410
460 A(10+3*A)=G:G=G/J:FOR I=1 TO D:A(I+6)=INT(G)
470 IF (A(I+6)<1)+(A(I+6)>C) F=1
480 IF A(I)=A(I+6) B=B+1:A(I+12)=1
490 G=10*(G-INT(G)+.000001):NEXT I:IF F=1 GOTO 440
500 IF B=0 PRINT"YOU HAVE DEDUCED THE VALUE IN ";A;" GUESSES":GOTO 950
510 FOR I=1 TO D:J=1:IF A(I+12)=1 GOTO 540
520 IF (A(J)=A(I+6))*(A(J+12)=0) W=W+1:A(J+12)=2:GOTO 540
530 J=J+1:IF J>D GOTO 540
535 GOTO 520
540 NEXT I:PRINT TAB(46);"P=";B;TAB(54);"V=";W
550 A(16+3*A)=B:A(17+3*A)=W:A=A+1:GOTO 400
950 INPUT"WANT TO PLAY AGAIN (Y/N) ";I
960 IF I=1 PRINT"GLAD YOU ENJOYED SUPERBAGELS. PLAY AGAIN SOON.":END
970 GOTO 200
1000 CLS:FOR I=1 TO 3:PRINTAT(10*I-12);"# GUESS P V";:NEXT I
1010 X=60:Z=0:FOR I=1 TO A-1
1020 IF X<42 X=5:Z=Z+64
1030 PRINT AT(X+2),I;
1040 PRINT AT(X+2+3),A(18+3*I);
1050 PRINT AT(X+2+10),A(16+3*I);
1060 PRINT AT(X+2+12),A(17+3*I);
1070 X=X+10:NEXT I:RETURN

```

Dear Bob,

Although still waiting for my first copy of your newsletter to arrive, I have however had the chance to read a friend's copy. I learnt quite a bit and in return thought you be interested in the following tip. My problem was a large game program that took up all but 300 bytes of my 4K. I needed to include playing instructions but had no more room. Answer; use my T-Bug to record the instructions in machine language. Here's how to do it:

First get 'READY' on screen then type CLEAR to move it to the top. Type a line number followed by 15 spaces, then type your instructions. Avoid spilling over into the next line. You have 12 lines at your disposal. on the 13th cursor(↓), Load your T-Bug tape and type CLOAD.

Now with T-Bug in memory, all the first 15 spaces of each line vanish (now you see why we have to use 15 spaces).

Place a clean tape in your machine and ready to record. Then type P 300/ 3FFF. The video portion of your memory (the instructions you have on your screen) now get saved to tape in machine language.

Next step is to return to BASIC, load your games program into memory and get ready to OSAVE it. Load the 'Instruction' tape into the machine, and dump your basic game onto the end of it. (I stopped my instruction tape before the end of the Trailer and saved my game from there on.

When this new tape is finished, read it back into the TRS-80 and PRESTO. The instructions are printed onto the screen just as they read from the tape - they take up no room in RAM. On the finish of these, another 'READY' is printed and you can load your game as normal. That's it folks ..

By the way, whilst in T-Bug, have a look at the Video portion of the RAM. By loading the ASCII set into this area they will be printed directly onto the screen. For instance, starting at 3001 (15 spaces in from left margin) I deposited all the Graphic set to see what they look like. 2A is the asterisk, 9 is the start of Graphics and so on as per the codes in the books. Had a lot of fun but have yet to find an easy way of getting the codes from keyboard and putting them onto the screen.

Hope this will be of some use and thanks for all the help you newsletter has been to me

Sincerely

Les. FINCH.
(V12BBD)

128A BOORALIE RD.,
DUFFYS FOREST
N.S.W. 2084 AUSTRALIA

***** MBUG NEWS *****
MONTEREY BAY USERS' GROUP FOR THE TRS-80
WILLIAM S. PITT, EDITOR

P. O. BOX 6H, PACIFIC GROVE, CA 92950
THE COMPUTER NUTS IN PACIFIC GROVE ARE LUCKY TO HAVE
A GROUP LIKE MBUG TO EXCHANGE PROGRAMS & HARDWARE WITH.
KEEP UP THE GOOD WORK AND KEEP THE NEWSLETTERS COMING.
YOUR EDITOR
R. GORDON LLOYD

DUES: \$6.00 PAYABLE TO ALLAN HANDFORTH
210 E. ROMIE LANE, APT. B
SALINAS, CA 93901