

TRS-80 USERS GROUP	21	ZIPCODE SORTING PROGRAM
ROBERT G. LLOYD	22	MAILING LIST PROGRAM
C. W. EVANS	23	JOTTO
DAVID R. KIMBEL	24	12 DAYS OF CHRISTMAS
THOMAS W. DAVIS	26	CONVERSION SUBROUTINES
RAY JONES WA4AJY	27	SINE WAVE PROGRAM
RON MOEHLIS	28	ANIMATION
SHELDON KAUFMAN	29	MOVINGAVE

CHANGE TO THE MAILING PROGRAM ON PAGE 22

IF THIS IS THE FIRST RECORD FOR THIS FILE YOU MUST TAKE OUT
THE GOSUB TO 176 IN LINE # 32

LINE 176 IS NOT RIGHT - YOU NEED TO MAKE IT READ
** 176 LR=LOF(1)*3-5:IF LR<=0 THEN LR=1 **

PUBLICATION STATEMENT

THE TRS-80 NEWSLETTER IS PUBLISHED MONTHLY IN FAYETTEVILLE, NC BY THE TRS-80 USERS GROUP. SUBSCRIPTION RATES ARE \$15.00 IN THE US \$17.00 IN CANADA & 22.00 ELSEWHERE IN THE WORLD. BACK ISSUES ARE AVAILABLE FOR \$1.50 IN THE US, \$1.75 IN CANADA & \$2.00 ELSEWHERE THE FIRST ISSUE WAS PUBLISHED JANUARY 1978 (VOL. 1, NO. 1).

TRS-80 USERS GROUP, 7554 SOUTHGATE RD., FAYETTEVILLE, NC 28304
EDITOR/PUBLISHER - - R. GORDON LLOYD

```

1 REM TRS-80 USERS GROUP
2 REM FAYETTEVILLE, NC 28304
3 CLS:PRINT "TO STOP TYPE 'END' ELSE 'ENTER' ";:INPUT S$
4 IF S$="END" THEN RUN "MENU"
5 CLS:PRINT:GOTO 29
6 REM ** ZIPCODE SORTING ROUTINE **
7 PRINT
8 S=0:G=1
9 N1=N-1
10 X$="####":CLS:PRINT CHR$(23)
11 FOR I=0 TO N1
12 IF A(I)<A(I+1) THEN 18
13 Z=A(I)
14 PRINT@536,USING X$:Z
15 A(I)=A(I+1)
16 A(I+1)=Z
17 S=1
18 NEXT I
19 CLS
20 IF S=1 THEN 8

```

```

21 REM ** TEST & PRINT ROUTINE **
22 FOR I=0 TO N-2
23 PRINT A(I+2);G;
24 FOR J=0 TO N
25 IF (A(I+2)=VAL(F$(J)))AND(L(J)>0) THEN 57 ELSE NEXT
26 G=G+1:NEXT I
27 GOTO 31
28 REM ** READ DATA & DIM ROUTINE **
29 CLEAR 25000:LR=0
30 DIM A$(325),B$(325),C$(325),D$(325),E$(325),F$(325),A(325),L(325)
31 READ S$:PRINT@202,"NAME OF FILE TO BE SORTED ";S$
32 IF S$="END" THEN STOP
33 OPEN"R",1,S$
34 GOTO 65
35 REM ** SORT SET-UP ROUTINE **
36 N=LR-1:CLS:PRINT@202,"NUMBER OF FILES TO BE SORTED ";N
37 IF LRC20 THEN CLOSE:GOTO 31
38 LR=0
39 FOR I=0 TO N
40 LR=LR+1
41 GOSUB 53:PR=INT((LR-1)/3)+1
42 GET 1,PR
43 L(I)=LR
44 A$(I)=DA$
45 B$(I)=NA$
46 C$(I)=AD$
47 D$(I)=CT$
48 E$(I)=ST$
49 F$(I)=ZP$:A(I)=VAL(F$(I))
50 NEXT
51 CLOSE:GOTO 6
52 REM ** SUB-RECORD SUBROUTINE **
53 SR=LR-3*INT((LR-1)/3)-1
54 FIELD 1,SR*85 AS G$,7 AS DA$,25 AS NA$,30 AS AD$,15 AS CT$,3 AS ST$,5 AS ZP$
55 RETURN
56 REM ** LINE PRINT ROUTINE **
57 IF LEFT$(A$(J),4)="TEMP" THEN 63:REM ** TEST FOR NO PRINT **
58 LPRINT A$(J);CHR$(91);L(J);E$(J);CHR$(93)
59 LPRINT B$(J)
60 LPRINT C$(J)
61 LPRINT D$(J);E$(J);F$(J)
62 LPRINT CHR$(138);CHR$(138)
63 L(J)=0:GOTO 26
64 REM ** END OF FILE SUBROUTINE **
65 LR=LOF(1)*3-5:IF LR<=0 THEN LR=1
66 LR=LR+1:PRINT@460,"LOGICAL RECORD # =":LR
67 GOSUB 53:PR=INT((LR-1)/3)+1
68 GET 1,PR
69 ZP=VAL(ZP$)
70 IF ZP=0 THEN 36
71 PRINT@593,DA$:CHR$(91);LR;ST$:CHR$(93)
72 PRINT@657,NA$
73 PRINT@721,AD$
74 PRINT@785,CT$:ST$:ZP$
75 GOTO 66
76 REM ** DATA ROUTINE **
77 DATA "AK","AL","AR","AZ","CA","CO","CT","DC","DE","FL"
78 DATA "GA","HI","IA","ID","IL","IN","KS","KY","LA","MA"
79 DATA "ND","NE","MI","MN","MO","MS","MT","NC","ND","NE"
80 DATA "NH","NJ","NM","NV","OH","OK","OR","PA","RI"
81 DATA "SC","SD","TN","TX","UT","VA","VT","WA","WI","WV"
82 DATA "WY","AS","PR","END"

```

```

1 REM TRS-80 USERS GROUP
2 REM 7554 SOUTHGATE RD.
3 REM FAYETTEVILLE, NC
4 REM 28304
5 REM (919) 867-5822
6 REM FEB 1979 - ROBERT G. LLOYD
7 REM MAILING LIST PROGRAM
8 REM WRITE - READ - CHANGE - SEARCH
9 CLS
10 PRINT CHR$(23)
11 PRINT "MAKE SURE THAT DISK #"
12 PRINT "    2001 IS IN DRIVER # 1"
13 PRINT "    2002 IS IN DRIVER # 2"
14 PRINT
15 INPUT "PRESS ENTER ";O$
16 X=0:N=0
17 CLEAR 1000
18 CLS
19 CLOSE
20 PRINT@320,"ENTER FILE NAME OR STATE (TYPE 'END' TO STOP)";
21 LINEINPUT S$
22 IF S$="END" THEN END
23 PRINT:PRINT "    TYPE 1 TO WRITE"
24 PRINT "                2 TO READ"
25 PRINT "                3 TO CHANGE A FILE"
26 INPUT "                4 TO SEARCH A FILE ";N
27 OPEN"R",1,S$
28 CLS
29 ON N GOSUB 32,73,85,135
30 GOTO 16
31 REM ** WRITE ROUTINE **
32 GOSUB 176
33 LR=LR+1
34 GOSUB 184
35 PRINT@460,"NEXT RECORD NUMBER ";CHR$(94);LR;S$;" ";CHR$(93)
36 PRINT@0,"DATE - TYPE 'END' TO STOP " TAB(30);
37 LINEINPUT A$
38 J1=LEN(A$)
39 LSET DA$=A$
40 IF LEN(A$)>7 THEN 35
41 IF A$="END" THEN RETURN
42 PRINT "NAME" TAB(30);
43 LINEINPUT A$
44 J2=LEN(A$)
45 LSET NA$=A$
46 IF LEN(A$)>25 THEN 42
47 PRINT "ADD" TAB(30);
48 LINEINPUT A$
49 J3=LEN(A$)
50 LSET AD$=A$
51 IF LEN(A$)>30 THEN 47
52 PRINT "CITY" TAB(30);
53 LINEINPUT A$
54 J4=LEN(A$)
55 LSET CT$=A$
56 IF LEN(A$)>15 THEN 52
57 PRINT "STATE" TAB(30);S$
58 J5=LEN(S$)
59 LSET ST$=S$
60 IF LEN(S$)>2    57
61 PRINT "ZIP" TAB(30);
62 LINEINPUT A$
63 J6=LEN(A$)
64 LSET ZP$=A$
65 IF LEN(A$)>5 THEN 61

```

```

66 INPUT "IS THIS INFORMATION CORRECT (Y/N) ";X$
67 IF X$="N" GOSUB 188:GOTO 35
68 IF X$="Y" GOSUB 188:GOTO 70
69 GOTO 66
70 PUT 1,PR
71 GOTO 32
72 REM ** READ ROUTINE **
73 GOSUB 209
74 PRINT
75 INPUT "    INPUT THE # OF LABELS YOU NEED ";L
76 GOSUB 184
77 ZP=VAL(ZP$)
78 IF ZP=0 THEN RETURN
79 GOSUB 196
80 FOR I=1 TO L
81 GOSUB 202
82 NEXT I
83 GOTO 16
84 REM ** CHANGE ROUTINE **
85 CLS
86 GOSUB 209
87 CLS
88 GOSUB 119
89 X=1
90 PRINT@19,"THERE ARE 6 LINES OF DATA"
91 PRINT, "DATE", "NAME", "ADD"
92 PRINT, "CITY", "STATE", "ZIP"
93 PRINT "WHICH WOULD YOU LIKE TO CHANGE ";
94 LINEINPUT CH$
95 IF CH$="DATE" THEN 102
96 IF CH$="NAME" THEN 103
97 IF CH$="ADD" THEN 104
98 IF CH$="CITY" THEN 105
99 IF CH$="STATE" THEN 106
100 IF CH$="ZIP" THEN 107
101 GOTO 94
102 LINEINPUT "NEW DATE ";A1$:GOTO 108
103 LINEINPUT "NEW NAME ";A2$:GOTO 108
104 LINEINPUT "NEW ADD ";A3$:GOTO 108
105 LINEINPUT "NEW CITY ";A4$:GOTO 108
106 LINEINPUT "NEW STATE ";A5$:GOTO 108
107 LINEINPUT "NEW ZIP ";A6$
108 LSET DA$=A1$
109 LSET NA$=A2$
110 LSET AD$=A3$
111 LSET CT$=A4$
112 LSET ST$=A5$
113 LSET ZP$=A6$
114 INPUT "ANY MORE CHANGES TO THIS FILE (Y/N) ";YN$
115 CLS
116 IF YN$="Y" GOSUB 196:GOTO 90
117 PUT 1,PR
118 REM ** END OF FILE CHECK SUBROUTINE **
119 GOSUB 184
120 ZP=VAL(ZP$)
121 IF ZP=0 THEN RETURN
122 GOSUB 196
123 PRINT
124 A1$=DA$
125 A2$=NA$
126 A3$=AD$
127 A4$=CT$
128 A5$=ST$
129 A6$=ZP$
130 IF X=1 THEN 132

```

```

131 RETURN
132 INPUT "PRESS ENTER WHEN READY ";X$
133 GOTO 16
134 REM ** SEARCH ROUTINE **
135 A=0:B=0:C=0:D=0:E=0:H=0:LR=0:N=3
136 CLS
137 LR=LR+1
138 GOSUB 184
139 ZP=VAL(ZP$)
140 IF ZP=0 THEN CLOSE:RETURN
141 GOTO 152
142 CLS
143 GOSUB 196
144 PRINT "DO YOU NEED A HARD COPY (Y/N) "
145 PRINT "    TO MAKE A CHANGE (C) "
146 INPUT "    TO STOP (END) ";U$
147 IF U$="Y" GOSUB 202
148 IF U$="C" THEN CLS:GOTO 87
149 IF U$="END" THEN 16
150 CLS
151 GOTO 137
152 IF A=1 THEN 161
153 PRINT, "DATE", "NAME"
154 PRINT, "CITY", "ZIP":PRINT
155 LINEINPUT "WHAT WOULD YOU LIKE TO FIND ";H$
156 PRINT@226,"INPUT THE ";
157 IF H$="DATE" THEN A=1:H=1:GOTO 163
158 IF H$="NAME" THEN A=1:H=2:GOTO 165
159 IF H$="CITY" THEN A=1:H=3:GOTO 167
160 IF H$="ZIP" THEN A=1:H=4:GOTO 169
161 ON H GOTO 164,166,168,170
162 GOTO 152
163 LINEINPUT "DATE ";DT$
164 IF INSTR(DA$,DT$) THEN 142 ELSE 137
165 LINEINPUT "NAME ";NT$
166 IF INSTR(NA$,NT$) THEN 142 ELSE 137
167 LINEINPUT "CITY ";CA$
168 IF INSTR(CT$,CA$) THEN 142 ELSE 137
169 LINEINPUT "ZIP ";ZT$
170 IF ZP$=ZT$ THEN 142 ELSE 137
171 REM ** SUB-RECORD SUBROUTINE **
172 SR=LR-3*INT((LR-1)/3)-1
173 FIELD 1,SR*85 AS D$,7 AS DA$,25 AS NA$,30 AS AD$,
174 RETURN          15 AS CT$,3 AS ST$,5 AS ZP$
175 REM ** END OF FILE GET SUBROUTINE **
176 LR=LOF(1)*3-5
177 LR=LR+1
178 GOSUB 184
179 ZP=VAL(ZP$)
180 IF ZP=0 THEN LR=LR-1:RETURN
181 GOSUB 196
182 GOTO 177
183 REM ** PHYSICAL RECORD SUBROUTINE **
184 GOSUB 172
185 PR=INT((LR-1)/3)+1
186 GET 1,PR:RETURN
187 REM ** CLEAR INPUT SUBROUTINE **
188 PRINT@30,STRING$(J1,32);
189 PRINT@94,STRING$(J2,32);
190 PRINT@158,STRING$(J3,32);
191 PRINT@222,STRING$(J4,32);
192 PRINT@286,STRING$(J5,32);

```

(CONTINUED ON PAGE 30)

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

9806 Amber Trail
Sun City, AZ 85351
January 4, 1978

Dear Mr. Lloyd:

Inclosed is a cassette of my current JOTTO program. It simulates a paper-and-pencil wordgame copyrighted 1956. It consisted of a set of instructions and two pads of forms. It was played by two individuals or two teams. I inclose a blank form and set of instructions.

JOTTO was discussed in Calculators and Computers, March 1978 which included a BASIC PLUS program. Page 183 of BASIC Computer Games has a program for WORD which is similar but identifies the matching letters and their positions in the word as in HANGMAN.

My program conforms to the original JOTTO. It can be played using words, or if desired, 5-letter groups. The program randomly selects a vocabulary word using the INKEY\$ function to stop a continuous circulation of data words when the ENTER key is pressed. I learned this from Steve MacGregor, 3701 W. Wethersfield Rd., Phoenix, AZ 85029. He used it in an unpublished Level II, HANGMAN game which has outstanding graphics, very fast.

The cassette has multiple entries of both a 16K and 4K version. They are identical except that the 4K has less remarks and less vocabulary. You might wish to publish the 4K version and the user could use his own added vocabulary if he has 16K.

The program has a special feature—an optional standard opening of 6 words so chosen that the results of these 6 trials tell you the status of A,T,ME,ND,BL,FI,GH,OS. From there you need to guess the word.

If the user prefers to enter a secret word he can do so by RUN, BREAK, Type S="ABCDE, GOTO 740, ENTER.

For 4K, lines 50 and 60 are changed and the following are deleted: 100, 250-410, 491, 495, 650, and 660. The CLEAR memory is 332 for 4K and 11171 for 16K.

Yours very truly,


C. W. Evans

10 '12/19/78
20 'JOTTO WITH OPTIONAL STANDARD OPENING
30 'C. W. EVANS, 9806 AMBER TRAIL, SUN CITY, AZ 85351
40 '(602) 933-1616
50 'NEXT LINE SHOWS 16K CLEAR MEM
60 '11171
70 CLS
80 GOTO 500

100 /-----
110 ' ADD NEW DATA WORDS AS DESIRED

LAST DATA LINE MUST BE 490 DATA ""
120 DATA HORSE, FLASK, PLUMP, SLUMP, LYMPH, LAMPS, LUMPY, WHALE, FLING
130 DATA STAIR, LEAST, FEAST, VOLKS, PACES, FIGHT, NEVER, EVERY, SEVEN
140 DATA GHOST, BLAND, WIVES, XRAY, OTHER, RUSTY, BLANK, WASTE, WORDS
150 DATA BASIC, LULLS, NYMPH, SHALE, CLUNG, SLANG, GROAN, WRONG, GROWN
160 DATA AGAIN, WORTH, GAMES, SCORE, THOSE, THESE, GRIPE, UTTER, INCUR
170 DATA REARM, SUEDE, RIGOR, CLEAN, SINGE, SOLAR, BURLY, SORRY, BOOTY
180 DATA FENCE, LUNGE, OWNED, NOVEL, AFOOT, FOYER, TAKEN, CEASE, CHOKE
190 DATA MURKY, CHIDE, LOOSE, MOLDY, CHIME, EIGHT, TOKEN, WIPE, LUSTY
200 DATA BLAZE, DEITY, AISLE, FORTE, CREEK, GLADE, BLADE, UPPER, THREE
210 DATA YOUNG, CROWN, LITHE, HOUSE, OPIUM, ONION, FRAME, FLAME, CLOTH
220 DATA TOXIC, NEWLY, ABYSS, LATHE, ACRID, SHAKY, ORBIT, FABLE, CIVIL
230 DATA EMPTY, LOUSY, MOUSE, PLUSH, FLOOD, PLAYS, CHILD, NEWSY, FORTY
240 DATA PILOT, AMUSE, HAVOC, OPINE, HARPY, VIRUS, AUDIT, CHESS, APRON
250 DATA MERCY, ABIDE, AGING, BRAVE, WHITE, BLACK, GREEN, LEMON, FIORD
260 DATA ELUDE, SMASH, AWASH, YACHT, CLAIM, FRUIT, EAGLE, GLOAT, STOKE
270 DATA ALLOT, FROZE, SOAPY, DINER, SYNOD, QUEST, ALTAR, AWAKE, AWAKE
280 DATA MYTHS, HELIX, RAZOR, EXILE, STEAL, STALE, SLATE, BRING, UNCLE
290 DATA TALES, STEAM, TEAMS, MEATS, TAMES, MATES, PEALS, PALES, PLEAS
300 DATA LAPSE, LEAPS, RATES, TEARS, STARE, ASTER, TARES, TIMES, SMITE
310 DATA MITES, ITEMS, HEARS, SHARE, SHEAR, HARES, WIDER, WIRED, WEIRD
320 DATA ARROW, EASES, JEWEL, VIVID, TEXTS, EXERT, LATEX, FIRST, NINTH
330 DATA OASIS, DOPEY, JULEP, PAPAL, PEEVE, DISCS, MAMMA, QUEUE, TIARA
340 DATA ERROR, SEEPS, SOLOS, REFER, TEETH, CUBIC, DIODE, VERVE, ROTOR
350 DATA RADAR, ABACK, SEXES, EXIST, EXITS, RATIO, SASSY, TOPIC, QUEEN
360 DATA OXIDE, EQUAL, LEVEL, EXCEL, EPOCH, LOCAL, SALAD, PATIO, LEVEE
370 DATA PROPS, LADLE, ALONG, MOTTO, COCOA, OFTEN, TESTS, PSALM, PARRY
380 DATA RADIO, MINUS, GIVEN, PAPER, HANDS, STAND, VOICE, TOWER, WROTE
390 DATA SLOOP, POOLS, LOOPS, SORES, ROSES, MARCH, CHARM, SOUTH, SHOUT
400 DATA NAMES, MEANS, POSTS, STOPS, SPOTS, SNAIL, NAILS, RITES, TIRES
410 DATA TIERS, SWORD, SKILL, KILLS, PASTE, TAPES, SLAVE, SALVE, KITES
490 DATA""

491 /-----

495 ' INSTRUCTIONS

500 PRINTTAB(22)"J O T T O
510 PRINT
520 PRINT" THIS PROGRAM SIMULATES THE GAME JOTTO, COPYRIGHTED
530 PRINT"1956 BY THE JOTTO CORPORATION. THE COMPUTER SELECTS A
540 PRINT"SECRET 5-LETTER WORD AND YOU GUESS USING A SERIES
550 PRINT"OF TRIAL WORDS. THE COMPUTER TELLS YOU THE NUMBER OF JOTS IN
560 PRINT"EACH TRIAL. JOTS ARE THE NUMBER OF LETTERS IN THE TRIAL WORD
570 PRINT"THAT ARE IN THE SECRET WORD. WHEN YOU GUESS SECRET WORD
580 PRINT"THE COMPUTER SAYS J O T T O. LETTERS DUPLICATED IN THE
590 PRINT"SECRET WORD COUNT ONLY AS ONE JOT UNLESS THE TRIAL WORD
600 PRINT"ALSO CONTAINS DUPLICATIONS. THE OBJECT OF THE GAME IS TO
610 PRINT"FIND THE SECRET WORD WITH THE MINIMUM NUMBER OF TRIALS.
620 PRINT
630 PRINT"TO GIVE UP ON SECRET WORD, TYPE XXXXX
650 /-----

```

660 / MAIN PROGRAM
670 DEFINT H-N: DEFSTR S-U
680 PRINT"PRESS 'ENTER' TO START THE GAME"
690 / ENTER KEY BREAKS ENDLESS DATA CHAIN AND THEREBY
SELECTS THE SECRET WORD.
700 T=INKEY$
710 READ S
720 L=LEN(S):IF L=0 THEN RESTORE:GOTO 710
730 IF INKEY$="" THEN 710 / NOTE NO SPACE BETWEEN QUOTES
740 M=0:INPUT"DO YOU WANT A STANDARD OPENING? (Y/N) ";A$
742 / M = TRIAL NUMBER
745 /PRINT@1010,S: / THIS IS A TEST LINE
746 / =====
749 CLS
750 IF A$="N" THEN 840
760 IF A$<>"Y" THEN 740
780 U="AMEND":GOSUB 930
790 U="BLAME":GOSUB 930
800 U="BLAND":GOSUB 930
810 U="FIGHT":GOSUB 930
820 U="GHOST":GOSUB 930
830 U="FOIST":GOSUB 930
840 PRINT@0, "I HAVE A 5-LETTER SECRET WORD "
850 PRINT "TYPE A 5-LETTER TEST WORD ";
860 INPUT U : PRINT@0,CHR$(30)
870 IF LEN(U)>5 PRINT@256+M*64," 5 LETTERS PLEASE!": GOTO 850
880 /
890 GOSUB905
900 GOTO 850
905 IF S=U PRINT@192+M*64,M+1,U," 5 JOTS J O T T O":GOTO 1000
920 IF U="XXXXX" THEN 1070
930 H=0: /H=NUMBER OF JOTS
940 FOR I=1 TO 5: S(I)=MID$(S,I,1): NEXT
950 FOR J=1 TO 5: U(J)=MID$(U,J,1): NEXT
960 FOR I=1 TO 5
970 FOR J=1 TO 5
980 IF S(I) <> U(J) THEN 1020
990 H=H+1
1000 U(J)=" "
1010 GOTO 1030
1020 NEXT J
1030 NEXT I
1040 M=M+1
1050 PRINT@128+M*64,M,U,H:"JOTS"
1060 RETURN
1070 PRINT"MY WORD WAS ";S
1080 PRINT" HIT ENTER FOR ANOTHER GAME "
/ : INPUT A$: CLS: GOTO 680

```

MR. LLOYD,

THIS PROGRAM WILL DISPLAY THE WORDS TO <THE 12 DAYS OF CHRISTMAS>. YOU MAY WANT TO PUBLISH IT IN THE NEWSLETTER. I WOULD LIKE TO GET IN TOUCH WITH THE WRITER OF THE <TRS-80 JUKEBOX> AS IT WOULD BE INTERESTING TO ADD THE MUSIC <VIA AM RADIO> TO THIS PROGRAM. I THINK IT COULD BE DONE IN THE FOR-NEXT LOOPS AFTER EACH LINE OF WORDS. I APPOLIGIZE FOR NOT HAVING USED LPRINT'S IN MY LINES. THE ONLY CHANGE I CAN THINK OF TO LET IT RUN IN LEVEL 1 WOULD BE TO REMOVE THE PRINT CHR\$(23).

KEEP UP THE GREAT WORK ON THE NEWSLETTER.

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

```

10 X=0
50 X=X+1
55 CLS
60 ONK GOSUB 71,72,73,74,75,76,77,78,79,80,81,82
65 GOTO100
71 D$="FIRST":RETURN
72 D$="SECOND":RETURN
73 D$="THIRD":RETURN
74 D$="FOURTH":RETURN
75 D$="FIFTH":RETURN
76 D$="SIXTH":RETURN
77 D$="SEVENTH":RETURN
78 D$="EIGHTH":RETURN
79 D$="NINETH":RETURN
80 D$="TENTH":RETURN
81 D$="ELEVENTH":RETURN
82 D$="TWELVETH":RETURN
100 PRINT CHR$(23):PRINT"ON THE ";D$;" DAY OF CHRISTMAS
MY TRUE LOVE SENT TO ME":FORT=@T01400:NEXTT
105 FOR D=@T01 STEP-1
110 OND GOSUB1000,2000,3000,4000,5000,6000,7000,8000,9000,10000,11000,12000
120 NEXT D
800 GOTO50
900 STOP
1000 IFX>1THEN PRINT "
AND A PARTRIGE IN A PEAR TREE "ELSEPRINT"
A PARTRIGE IN A PEAR TREE"
1010 FORT=@T00800:NEXTT
1020 IFX<11CLS:PRINTCHR$(23):PRINT"
ONE
MORE
TIME !
1025 FORT=@T0700:NEXTT
1030 RETURN
2000 PRINT"
TWO TURTLE DOVES":FORT=@T0600:NEXTT:RETURN
3000 PRINT"
THREE FRENCH HENS":FORT=@T0600:NEXTT:RETURN
4000 PRINT"
FOUR CULLY BIRDS":FORT=@T0600:NEXTT:RETURN
5000 PRINT"
FIVE GOLDEN RINGS":FORT=@T01400:NEXTT:RETURN
6000 PRINT"
SIX GEESE A-LAYING":FORT=@T0700:NEXTT:RETURN
7000 PRINT"
SEVEN SWANS A-SWIMMING":FORT=@T0700:NEXTT:RETURN
8000 PRINT"
EIGHT MAIDS A-MILKING":FORT=@T0700:NEXTT:RETURN
9000 PRINT"
NINE LADIES DANCING":FORT=@T0700:NEXTT:RETURN
10000 PRINT"
TEN LORDS A-LEAPING":FORT=@T0700:NEXTT:RETURN
11000 PRINT"
ELEVEN PIPERS PIPEING":FORT=@T0700:NEXTT:RETURN
12000 PRINT"
TWELVE DRUMMERS DRUMMING":FORT=@T0700:NEXTT:RETURN

```

Gentlemen:

In one of your issues (I think it was Vol. 1 No. 4) was a letter from a TRS-80 user who told of using a lowpass filter to help overcome difficulty in loading cassettes.

The idea is good, but his suggested values for the capacitors may insert too much impedance in the audio output line from the CTR. I think he had suggested .01 mfd.

Capacitors in series, as required in a T-format filter, have a total effective capacity of less than the value of any one of the series capacitors. In this instance the effective capacity would be .005 mfd.

The formula for capacitive reactance is $X_C = \frac{1}{6.28 \times f \times C}$

where X_C is capacitive reactance in ohms
6.28 is the same as 2 pi
f is the frequency in Hertz
C is the capacity in farads

I built a lowpass filter, using your reader's schematic, but substituted capacitors having a value of 2.2 mfd each. The result is quite satisfactory.

To give me sharper (vernier-type) adjustment on the audio output level I also inserted a 200 ohm pot in the line, with an oversized knob on it. Using an AM pocket radio resting on the cable between my keyboard unit and the expansion interface, I monitor the output from the cassette. I adjust the volume control on the CTR to the low point where the sound of the data transfer, as heard on the radio, begins to distort, and I note that position on the volume control knob. Next I adjust the control to the high point where the sound again begins to distort and note that position. I then re-set that control to a midpoint between the two positions where the distortion was noted. I then use my external 200 ohm

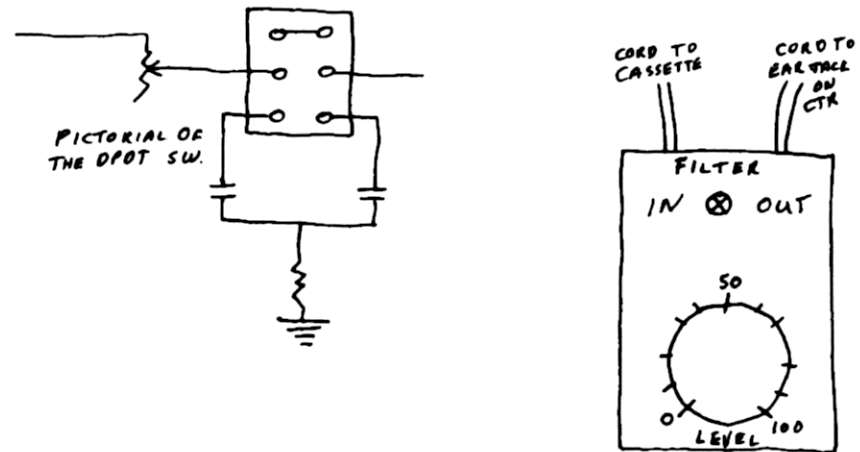
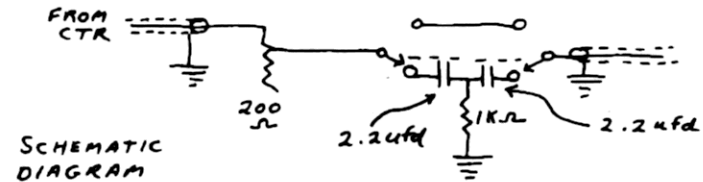
pot to make any fine-tuning adjustment of the level that may be necessary to get the tape to load. Try this method. You'll be surprised how many cassettes which may have been a problem to CLOAD may now be loaded with absolutely no difficulty.

Sincerely,

PS I used a double throw, double pole switch so I could switch the filter in or out of the circuit without having to unplug any cords. I mounted the filter, the switch and the 200 ohm level adjustment in a Radio Shack minibox, all foil-lined to eliminate any possible RF leakage.

WB5TSS

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



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.

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.

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.

MICRO ARCHITECT
96 DOTHAM ST.
ARLINGTON, MA 02174

PRICE INCLUDES POSTAGE, CASSETTER & DOCU.

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.

Dear Bob --

A couple of errors seem to have crept into my BASIC renumbering program (Newsletter, vol. 2, #2, p. 20) in the editing process. Lines 390, 400, and 410 should be combined into a single line -- otherwise the program crashes, since POKing 16548-9 in effect deletes the rest of the program. Also, in this line (in 400 in the printed version), for "D=256*HH" read "D=256*HH".

I'd be interested in seeing notes in the Newsletter on the contents of Level II Reserved RAM locations -- so far, I've discovered the following:

16548-9	address of bottom of program text
16633-4	" " " " simple variables
16635-6	" " " " arrays
16637-8	" " " " free memory
16616-7	" " " " stack
16544-5	" " " " string space
16561-2	" " " " top of string space (MEMORY SIZE)
16607-8	entry point for object tape loaded under SYSTEM
16554-6	RND (0) -- 16555 is RANDOM seed
16722-868	vectors to Disk Basic (L3) statements
16429-31	vector to DOS (5000H)
16512-25	binary/decimal conversion routine (self-altering code)
16688-93	holds converted decimal number
16435-7	vector to alternate KB driver

Does anyone know of a complete list?

The following ROM subroutines may come in handy: RST 24D compares HL and DE and sets flags (A is altered). CALL 900D waits for KB input (ASCII value is placed in A, other registers unaltered).

Sincerely,

Chiefs
H. Phelps Gates
6 Crestwood Tr. Pk. - Rt. 4
Chapel Hill NC 27514
(919)-967-5193

CONVERSION SUBROUTINES
BY
THOMAS W. DAVIS

9999 END

```
10000 ' DECIMAL TO HEX DEC%=# TO CONVERT 0-255 RETURNS HEX$
10010 Z1%= DEC%/16: Z2%= DEC% AND 15
10020 IF Z1%<10 THEN Z1%=Z1%+48 ELSE Z1%=Z1%+55
10030 IF Z2%<10 THEN Z2%=Z2%+48 ELSE Z2%=Z2%+55
10040 HEX$=CHR$(Z1%)+CHR$(Z2%)
10050 RETURN

10100 'HEX TO DECIMAL HEX$=# TO CONVERT 00-FF RETURNS DEC%
10110 DEC%=ASC(RIGHT$(HEX$,1)): IF DEC%>=65 THEN DEC%=DEC%
-55 ELSE DEC%=DEC%-48
10120 Z%=ASC(MID$(HEX$,1,1)): IF Z%>=65 THEN Z%=Z%-55 ELSE
Z%=Z%-48
10130 DEC%=DEC%+Z%*16
10140 RETURN
```

```
10200 'DECIMAL TO OCTAL DEC%=# TO CONVERT 0-511 RETURNS OCT$
10210 Z1%=DEC%/64: Z2%=(DEC% AND 56)/ 8: Z3%=DEC% AND 7
10220 OCT$= RIGHT$(STR$(Z1%),1) + RIGHT$(STR$(Z2%),1) +
RIGHT$(STR$(Z3%),1)
10230 RETURN

10300 'OCTAL TO DECIMAL OCT$=# TO CONVERT 0-777 RETURNS DEC%
10310 DEC%=VAL(RIGHT$(OCT$,1)): DEC%=DEC%+VAL(MID$(OCT$,2,1))
*8: DEC%=DEC%+VAL(MID$(OCT$,1,1))*64
10320 RETURN
```

Dear Fellow Users:

I have just recently purchased a Level II 4k TRS-80 and have had a problem that I think you should know about. It seems that something is wrong about the way the 4k chips (MCM6604) initialize and the READ statement doesn't work. It doesn't advance the READ pointer and reads the 1st data item over and over.

There are two fixes for this:

1. Software--When you power up do a POKE 16553, 255.
2. Hardware--Replace Bit 7 (chip Z13) with a Mostek MK4015. I don't know why this fixes it but that's what the Radio Shack Repair Center did and it works. (MUST BE MAGIC!!)

I need some help obtaining some info. What ports are assigned to the printer and how is handshaking taken care of? Can wait states of longer than 1 msec be used in the TRS-80 without losing mem due to refresh? What port is assigned to cassette #2? Where can I get a schematic of the TRS-80 Expansion Interface unit?

I am a computer technician with the US Navy and am interested in any hardware addition or modification to the TRS-80. Anyone who would like to communicate with me about hardware would be greatly appreciated. I am inclosing a group of subroutines I have written to convert from decimal ↔ octal and from decimal ↔ Hex. I use these in a memory inspect and change program. They only work for #'s of 0-255 DEC but the method can be used with larger #'s.

My address is;

T. W. Davis
1227 Mesa Rd.
Mare Island, CA 94592

THE BITPICKER'S TOOLBOX

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

A fast way to generate an 8-bit random number is to use the two-byte instruction LD A,R (ED5F hex). Caution is advised, however. If you use this many times without some random-length delay, such as waiting for an operator response, you can get a non-random sequence.

THE BITPICKER'S TOOLBOX

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

On most computers with both logical and arithmetic shifts in their instruction sets, an arithmetic left shift differs from a logical left shift only in the effect upon the condition flags: the arithmetic shift should set the "overflow" flag if the sign bit changed as a result of the shift. However, the SLA instructions on the Z-80 are misnamed, since the P/V flag is set or reset according to the parity of the result. In case you ever need a genuine arithmetic shift, the instruction ADD A,A will do just fine, although it works only on the A-register. Also, the instruction ADC A,A has the same effect, but in addition copies the C-flag to bit zero of the A-register. This feature is useful in linked shifts.

HERES A LISTING AND RUN OF AN INTERESTING LITTLE PROGRAM, NOT MY OWN, THAT PRINTS TO THE LINE PRINTER A SINE WAVE, USING AS ONE ELEMENT THE DATE AND AS THE NEXT THE TIME OF DAY FROM THE CLOCK IN TRS-80. A REAL LINE PRINTERS TRUE THROUGHPUT CAN EASILY BE DETERMINED, BY SIMPLY LOOKING AT THE TIME TAKEN TO DO THE TAB STOPS. IN THIS CASE, WITH A TELETYPE MACHINE AT 60 WORDS PER MINUTE, THE THROUGHPUT IS RATHER HUMOUROUS, AT BEST.

THE FORMAT HAS BEEN DONE FOR 64 CHARACTERS, SO SIMPLY CHANGE THE "LPRINT" S TO "PRINT" S TO DO IT ON THE DISPLAY.
RAY, WA4AJY
226 VALLEY RD.
LAWRENCEVILLE, GA. 30245

XXX CONVENTIONS USED IN ASCII BAUDOT CONVERSIONS XXX

SINCE THE BAUDOT MACHINE HAS NO "GREATER THAN", "LESS THAN", ETC. CHARACTERS, CERTAIN CONVENTIONS HAVE BECOME STANDARDIZED FOR USE BETWEEN THESE SYMBOL SETS. HERE IS AN OUTLINE CHART OF THOSE CONVENTIONS;

XX ASCII XX	XX BAUDOT XX
GREATER THAN	GT
LESS THAN	LT
EQUALS	EQ
PLUS SIGN	AMPERSAND
ASTERISK	X (DENOTES MULTIPLY)

AS ONE BECOMES FAMILIAR WITH THESE SUBSTITUTIONS, COMPUTER CODING IS NO HARDER TO READ THAN WITH THE OTHER CHARACTER SET, BUT MAY REQUIRE SOME GETTING USED TO.

```

110 FOR T EQ 0 TO 40 STEP.25
120 A EQ INT(26 & 25 X SIN(T))
130 LPRINT TAB(A);
140 IF B EQ 1 THEN 130
150 LPRINT LEFT$(TIMES,3)
160 B EQ 1
170 GOTO 200
180 LPRINT RIGHT$(TIMES,3)
190 B EQ 0
200 NEXT
210 GOTO 110
    
```

```

02/04/79
19:58:39
02/04/79
19:58:56
02/04/79
19:59:17
02/04/79
19:59:38
02/04/79
19:59:58
02/04/79
20:00:14
02/04/79
20:00:27
02/04/79
20:00:36
02/04/79
20:00:41
02/04/79
20:00:45
02/04/79
20:00:50
02/04/79
20:00:57
02/04/79
20:01:09
02/04/79
20:01:24
02/04/79
20:01:43
02/04/79
20:02:04
02/04/79
20:02:25
02/04/79
20:02:44
02/04/79
20:02:59
02/04/79
20:03:10
02/04/79
20:03:17
02/04/79
20:03:21
02/04/79
20:03:25
02/04/79
20:03:31
02/04/79
20:03:40
02/04/79
20:03:53
    
```


HI GORDON,
 APPRECIATE YOUR USERS GROUP-KEEP UP THE GOOD WORK.
 COPY OF LETTER SENT DIRECT TO HIM FYI
 TO DON FIELDING (PAGE W VOLUME 1 ISSUE 10)

CENTRONIX PLUG	TRS-80 IFC	DESCRIPTION
		LINE PRINTER DATA STROBE LO FOR 1.5MS PER DATA WORD
1	1	
19	2	GND
2	3	DATA 0 L. S. B.
20	4	GND
3	5	D1
21	6	GND
4	7	D2
22	8	GND
5	9	D3
23	10	GND
6	11	D4
24	12	GND
7	13	D5
25	14	GND
8	15	D6
26	16	GND
9	17	D7 MSB
27	18	GND
10	19	N/C
28	20	GND
11	21	BUSY TO IFC LO-READY TO PRINT HI-BUSY-HOLD
29	22	GND
12	23	OUT OF PAPER LO-READY HI-TROUBLE-WAIT
30	24	GND
13	25	UNIT SELECT HI-UNIT SELECTED-OK LO-WAIT
31	26	PRIME +5V THRU 4.7KOHM RESISTOR
14	27	GND
32	28	FAULT HI-OK LO-TROUBLE-WAIT
15	29	NOT CONN.
33	30	N/C
16	31	GND
34	32	N/C
17	33	GND
35	34	GND

WHEN TRS-80 IS OUT OF DATA---IT SENDS (HOLDS) AT ASCII FOR CARRIAGE RETURN.
 IT SENDS NO LINE FEES COMMANDS---SO DATA WORD SHOULD BE TAPED (DECODED)
 FOR CARRIAGE RETURN=LINE FEED.

I AM CURRENTLY INTERRFACING TO AN AMPEREX (SURPLOS)
 20 CHARACTER DOT MATRIX PRINTER... AFTER WHICH I EXPECT TO PROGRESS TO
 A FRIDEN FLEXOWRITER.

ENCLOSED IS A SAMPLE OF AMPEREX UNIT.

PAT CLINCH
 5318 TUSCARAMAS RD.
 BETHESDA, MD 20816

I CONTINUE TO GET A LOT OF INQUIRIES INTO THE USE OF
 BAUDOT TELETYPE MACHINES, MODEL 15, 23, 19 ETC. WITH
 THE TRS-80. I AM USING A MODEL 23 AND THE RADIO SHACK
 EDT/ASM (WITH A PATCH) TO WRITE AND EDIT THIS TEXT. I
 ALSO HAVE SOFTWARE TO ACTIVATE THE BASIC "LLIST" AND
 "LPRINT" COMMANDS TO BAUDOT MACHINES. A SHORT "SYSTEM"
 TAPE LOAD AND TWO "POKES" WILL PUT THE BAUDOT MACHINE ON
 LINE FROM BASIC JUST AS THOUGH IT WERE A CENTRONICS,
 WHICH IS WHAT THE COMPUTER THINKS IT IS. I HAVE THE TAPE
 PUNCH AND READ FEATURE ON MY 23 WHICH ALLOWS THE TRS-80
 TO PUNCH OUT "LLIST" COMMANDS TO PAGE AND TAPE FOR REAL
 SECURITY ON THOSE "I DONT WANT TO GO THROUGH THAT AGAIN"
 PROGRAMS. THE SOFTWARE THAT WORKS WITH THE EDT/ASM ALSO
 MAKES A DIRECT PATCH FOR IT TO ANSWER THE "H" AND "T"
 COMMANDS JUST AS IF IT WERE A LINE PRINTER. PRETTY NICE
 FOR A \$100.00 MACHINE. FOR THE HAMS, THIS ALLOWS US TO
 USE THE RADIO AS A GIANT "TIMESHARE" NETWORK, AND MAKES
 POSSIBLE DIRECT CORE DUMPS FROM THE TRS-80, EITHER IN
 BASIC OR ASSEMBLY SOURCE CODE STRAIGHT ONTO THE AIR.
 I ALSO HAVE THE PATCH TO MAKE THE RSM SERIES PRINT OUT
 ALL THEIR COMMANDS TO A BAUDOT MACHINE. THE MODIFIED
 RADIO SHACK EDT/ASM TAPE AND THIS SOFTWARE DRIVER MAKE A
 HIGHLY SOPHISTICATED TEXT EDITOR, RIVALING THE BETTER
 COMMERCIAL SYSTEMS IN EVERYTHING BUT PRINTER SPEED.
 THE SOFTWARE REQUIRES A DECODED PORT, DECIMAL 15 AND
 BIT NUMBER 0, ONLY. I CAN SUPPLY A BOX TO SUIT THIS
 NEED WITH DOUBLE OPTO-ISOLATORS AND TWO SOFTWARE CON-
 TROLLED RELAYS FOR GENERAL CONTROL APPLICATIONS. THIS BOX
 BOTH READS FROM AND WRITES TO THE TRS-80 AT PORT 15.
 INQUIRIES ARE ANSWERED PROMPTLY, OR WRITE ME FOR WHAT
 HELP YOU MAY NEED IN THIS REGARD.

RAY, WA4AJY
 226 VALLEY RD.
 LAWRENCEVILLE, GA. 30245

```

100 REM * TRS-80 ANIMATION - BY RON MOEHLIS *
110 DIM X(42),Y(42):RANDOM
120 FOR I=1 TO 14:READ X(I),Y(I):NEXT I
130 CLS:FOR I=1 TO 120:SET(I,16):NEXT I
140 FOR I=1 TO 120:SET(I,37):NEXT I
150 FOR I=16 TO 37:SET(12,I):NEXT I
160 PRINT@217,"TURTLE RACE"
170 FOR I=1 TO 14:SET(X(I),Y(I)):NEXT I
180 FOR J=1 TO 2:FOR I=1 TO 14
190 K=I+14*J:X(K)=X(I):Y(K)=Y(I)+J*6
200 SET(X(K),Y(K)):NEXT I,J
210 FOR I=1 TO 1000:NEXT I
220 R=RND(3):R=R-1:T=14*R
230 K=1+T:SET(X(K),Y(K)):SET(X(K+3),Y(K+3))
240 FOR I=K TO 4+T:RESET(X(I),Y(I)):NEXT I
250 FOR I=K TO 4+T:X(I)=X(I)+1:SET(X(I),Y(I)):NEXT I
260 FOR I=5+T TO 8+T:X(I)=X(I)+1:SET(X(I),Y(I)):NEXT I
270 RESET(X(9+T),Y(9+T)):RESET(X(12+T),Y(12+T))
280 FOR I=9+T TO 14+T:X(I)=X(I)+1:SET(X(I),Y(I)):NEXT I
290 IF X(14+T)=120 THEN 310
300 GOTO 220
310 PRINT@936,"WINNER WAS #":R+1
320 RESTORE:FOR I=1 TO 3000:NEXT I:GOTO 120
330 DATA 1,19,1,20,1,21,1,22,2,20,2,21,3,20
340 DATA 3,21,4,19,4,20,4,21,4,22,5,20,5,21
  
```


Sheldon Kaufman
910 W. Elm St.
Wheaton, Ill. 60187

I have written a program and am submitting it to the TRS-80 Users Group Newsletter which computes moving averages of stock market prices. This program, MOVINGAVE, uses the closing weekly prices of 3 to 5 stocks, stores them in the array A(I), and computes three different moving averages for each stock. A moving average is simply the average price over a definite time period; it "moves" along with the calendar. Obviously this Newsletter is not the place to discuss how these averages can be used in stock market analysis --- you should read a text on that subject. The use of a computer to create a data base and to perform repetitive calculations is almost essential in this field.

In my program MOVINGAVE the time periods for averaging are set by the variables R, S, and T in line 100 to 5, 15, and 40 weeks. Also in line 100 the number of stocks is set by P (=5) and the maximum number of weeks of data by Q (=72). These may be changed as desired by the user. Since I have a Level I TRS-80, the program is written in Level I BASIC, but I have used only a few abbreviations, namely P. for PRINT, P.A. for PRINT AT, and IN. for INPUT, to make it more understandable and easier to convert to Level II.

The data structure is as follows: A(0), A(1), and A(2) contain the month, day, and year of the first entry; A(3) and A(4) are reserved for possible future use. The stock prices are stored in A(5) - A(364), and A(365) - A(369) is a buffer area. For 5 stocks the prices of Stock 1 are stored in A(5), A(10), ... A(360), and those of Stock 2 in A(6), A(11), etc. If only 4 or 3 stocks are used, the intervals are 4 and 3, respectively. To start out, store the date of the first week in A(0) - A(2), set A(3) - A(364) equal to zero, and RUN. The computer asks "ADD MORE?", you type "Y", and then enter the data as indicated.

The names of the stocks are specified in the DATA statement, line 90 (up to 9 letters each). Line 100 sets the control variables P, Q, R, S, T. Lines 110-210 list the entire set of stock prices for a quick review (you can "freeze" the display using the ↑ key during this listing). Lines 220-470 add new data to the array; when the array is filled the latest week pushes out the oldest week and updates the starting date in A(0) - A(2). (Lines 410-470). When all new data has been added a zero entry exits to the computation section, lines 500-790. The date, price, and 3 averages for each stock are listed, either the most recent 13 weeks (one screenful) or everything possible (starting at week 40). The program could be changed to a daily basis by changing the subroutine starting at line 800, but remember to skip weekends and holidays.

A special routine handles stock splits (lines 310-390). All prices for that stock are adjusted to the new shares in order to compute the averages properly. To do this, you enter instead of new weekly prices the following quantities: -L (a flag to indicate a stock split); the number of the stock being split (1-5); and the split factor (2 for a 2-for-1 split, 1.5 for a 3-for-2 split, etc.). The computer then prints a message telling you what it proposes to do (line 350) and asks approval. Typing "N" cancels the command, in case of error.

Originally I saved the data on cassette by using the PRINT# command, for example

```
1100 FOR I = 0 TO 72 : J = 5*I
1110 PRINT# A(J);";";A(J+1);";";A(J+2);";";A(J+3);";";A(J+4)
1120 NEXT I
```

The data would then be read the following week using INPUT# in a similar loop. However, this takes about 7 minutes for reading and another 7 minutes for writing, because of the inefficient way these tape operations store data, with each record having a long tape leader. For example, if you are following 50 stocks (not a very large number) you spend 140 minutes just watching the tape move! Even with Level II with twice the tape speed, that is too long a time.

I have solved this problem by writing a machine language program to write and read the entire array in one large record, similar to the way the CSAVE command writes a program. Using this program it only takes 40 seconds to write or read 365 data items, about ten times faster than before. This program, which I call MEMDUMP, is written for Level I and loads using CLOAD (you don't need T-BUG). It will write to cassette any or all of the memory, starting from A(0). In fact it can write the current values of all the variables A-Z, the string variables A\$ and B\$, and itself as well! The resulting tape can be read either by MEMDUMP, or if you prefer, simply using CLOAD. If any members of the Users Group would like this program, I will send them a cassette with MEMDUMP on both sides for \$5.00.

For an additional \$2.00 I will include the MOVINGAVE program as listed here, and also in the abbreviated version. The latter has all REM statements omitted, uses all legal abbreviations in Level I and other tricks to squeeze it into the minimum memory so that it runs in a 4K system (like mine) with space for 72 weeks X 5 stocks. Complete instructions will be included.

P.S. My MOVINGAVE program just does the calculations --- if you lose money on the stock market, don't blame me!

```
5 REM *** MOVINGAVE ***
6 REM *** A PROGRAM TO COMPUTE MOVING AVERAGES OF 3-5 STOCKS.
7 REM COPYRIGHT 1979, SHELDON KAUFMAN
8 REM 910 W. ELM ST., WHEATON, IL 60187.
9 REM DATA STATEMENT 90 HAS NAMES OF STOCKS.
10 REM STATEMENT 100 SETS VALUES OF VARIABLES P, Q, R, S, T.
11 REM VARIABLES: P=NO. OF STOCKS; Q=MAX. NO. OF ENTRIES.
12 REM R, S, T = 3 AVERAGING TIMES.
13 REM USES WEEKLY STOCK PRICES.
14 REM ---- FOR LEVEL 1 BASIC CHANGE ALL @ TO AT ----
15 DATA STOCK 1, STOCK 2, STOCK 3, STOCK 4, STOCK 5
16 P=5: Q=72: R=5: S=15: T=40
17 CLS: V=1: N=0: E=INT(50/P)
18 GOSUB 1000: G=64: L=5: GOSUB 950
19 FOR K=1 TO Q: IF A(L)=0 THEN 210
20 GOSUB 1050: F=E+2: FOR Z=0 TO P-1
21 PRINTTAB(F);A(L+Z);
22 F=F+E: NEXT Z: PRINTTAB(63)
23 IF G=960 GOSUB 950
24 L=L+P: G=G+64: IF G>960 G=960
25 GOSUB 800: NEXT K
26 M=(L-5)/P: PRINT@, " ";
27 INPUT"ADD MORE?";X: IF X=N THEN 500
28 REM ADD MORE DATA. 0 TO EXIT. -1 FOR STOCK SPLIT.
```

```

240 IF G>896 GOSUB 950: G=896
250 PRINT@G, " ": GOSUB 1050: F=G+E+2
260 FOR Z=0 TO P-1: PRINT@F, " ";
270 INPUT@L+Z): IF A(L)=0 THEN 500
280 IF (Z=2)*(A(L)<0) THEN 310
290 F=F+E: NEXT Z: G=G+64
300 IF G>896 PRINT@G, " ": G=896: GOSUB 950
305 GOTO 400
310 REM ADJUST FOR STOCK SPLIT
320 B=A(L+1)-1: C=A(L+2): RESTORE
330 FOR D=0 TO B: READ A$: NEXT D
340 FOR D=L TO L+4: A(D)=0: NEXT D: PRINT@G, " "
350 PRINT@G, " ADJUSTING "; A$; " FOR"; C: "FOR 1 SPLIT. OK";
360 INPUTX: IF X=N THEN 390
370 B=B+5: FOR J=1 TO M: A(B)=INT(100*A(B)/C)/100
380 B=B+P: NEXT J
390 PRINT@G: GOSUB 1050: GOTO 250
400 IF L<P*Q+P M=(L-5)/P+1: L=L+P: GOSUB 800: GOTO 250
410 REM PUSH NEW DATA ONTO ARRAY; ADVANCE START DATE 1 WEEK.
420 FOR K=P TO L-1: A(K)=A(K+P): NEXT K
430 FOR K=L TO L+P-1: A(K)=0: NEXT K
440 B=U: C=V: D=W: U=A(0): V=A(1): W=A(2)
450 GOSUB 800: A(0)=U: A(1)=V: A(2)=W
460 U=B: V=C: W=D: GOSUB 800: GOTO 250
470 U=B: V=C: W=D: GOSUB 800: GOTO 250
500 REM COMPUTE AVERAGES. H=NO. OF LINES TO PRINT.
505 REM M=NO. OF WEEKS IN ARRAY; E=NO. OF TIMES DATE ADVANCES.
510 INPUT"LAST 13 WEEKS(1) OR ALL(2)"; Z: H=M-T+1: RESTORE
520 IF H<1 PRINT"INSUFFICIENT DATA": END
550 IF Z=2 E=M-Q+T-1: L=0: GOTO 580
560 L=M-T-12: IF L<0 L=0
570 E=M-13: IF H>13 H=13
580 IF E<-1 E=-T-1
590 FOR K=0 TO P-1: CLS: A=0: B=0: C=0: GOSUB 1000
610 FOR I=1 TO E: GOSUB 800: NEXT I: READ A$: PRINTTAB(20), A$
630 PRINT@67, "WEEK"; TAB(20); "PRICE"; TAB(29); R; "WK";
640 PRINTTAB(39); S; "WK"; TAB(49); T; "WK"
650 G=128: J=L*P+K+5
660 FOR I=1 TO T: C=C+A(J): J=J+P: NEXT I
670 J=J-S*P: FOR I=1 TO S: B=B+A(J): J=J+P: NEXT I
680 J=J-R*P: FOR I=1 TO R: A=A+A(J): J=J+P: NEXT I
690 F=A(J-P): FOR I=1 TO H: GOSUB 1050
700 PRINTTAB(19); F; TAB(29); INT(100*A/R)/100;
710 PRINTTAB(39); INT(100*B/S)/100; TAB(49); INT(100*C/T)/100
720 F=A(J): A=A+F-A(J-R*P)
730 B=B+F-A(J-S*P): C=C+F-A(J-T*P)
740 GOSUB 800: J=J+P: G=G+64: IF (G<960)+(I=H) THEN 760
750 PRINT@955, " ": INPUTB$: G=960: PRINT@955, " "
760 NEXT I: IF K=P-1 THEN 780
770 INPUT"NEXT STOCK"; X: IF X=N THEN 790
780 NEXT K: INPUT"FINISHED"; X: IF X=N THEN 500
790 END
800 REM SUBROUTINE TO ADVANCE DATE 1 WEEK
805 REM U, V, W = MONTH, DATE, YEAR
810 V=V+7: IF U=2 THEN 870
820 IF (U=4)+(U=6)+(U=9)+(U=11) THEN 850
830 IF V<32 THEN 920
840 V=V-31: U=U+1: GOTO 920
850 IF V<31 THEN 920
860 V=V-30: U=U+1: GOTO 920
870 IF W-4*INT(W/4)=0 THEN 900
880 IF V<29 RETURN
890 V=V-28: U=3: RETURN
900 IF V<30 RETURN
910 V=V-29: U=3: RETURN

```

```

920 IF U<13 RETURN
930 U=1: W=W+1: RETURN
950 REM SUBROUTINE TO PRINT STOCK NAMES
960 RESTORE: F=E+3: PRINT@0, " WEEK";
970 FOR Z=1 TO P: READ A$: PRINTTAB(F); A$:
980 F=F+E: NEXT Z: RETURN
1000 REM SUBROUTINE TO SET START DATE
1010 U=A(0): V=A(1): W=A(2): RETURN
1050 REM SUBROUTINE TO PRINT DATE
1060 PRINT@G, U;: PRINT@G+3, "/";: PRINT@G+4, V;
1070 PRINT@G+7, "/";: PRINT@G+8, W;
1080 RETURN

```

FOUR NEW UTILITY PROGRAMS FOR LEVEL II TRS-80'S

LINKUP....A machine language program that inhibits line merges and allows any number of BASIC programs to be loaded, one after the other. Keep a separate tape of subroutines; load and use them with a program already in the machine. Put several games on one tape and run them all without reloading A very useful utility \$14.95

UNNEW....As you probably suspected, the command "NEW" really doesn't wipe out your whole program. This short machine language program recalculates address links and resets pointers so that your program is restored and rises again from the ashes like the legendary Phoenix \$9.95

NEWNUM....Changes line numbers within your basic program and makes it look like you did it right the first time. Makes nice even increments between line numbers and rennumbers your GOSUB and GOTO statements at the same time. This little bugger will even catch an ON ERROR GOTO statement and put in the new line number. Useful with LINKUP above \$9.95

4BASE....This BASIC program will convert the 5 digit number that you input in Hex, Decimal, Octal or Binary into the equivalent value in the other three bases. Use with LINKUP as a subroutine \$4.95

All programs are LEVEL II 4K or 16K cassette tape. 16K versions of the machine language programs will be shipped unless you tell us otherwise. Sorry, they don't work in LEVEL I. Please include \$1.50 for postage and handling. Texas residents add 6% state tax. If you've read this far, you lucky devil, you can save almost 10 bucks.

..... Order all four programs for only \$29.95

Abcor Inc.
P.O. box 58216
Houston, TX. 77058

(FROM PAGE 22)

```

193 PRINT@350, STRING$(J6, 32)
194 RETURN
195 REM ** VIDEO DISPLAY SUBROUTINE **
196 PRINT@593, DA$; CHR$(94); LR; ST$; CHR$(93)
197 PRINT@657, NA$
198 PRINT@721, AD$
199 PRINT@785, CT$; ST$; ZP$
200 RETURN
201 REM ** LABEL-LPRINT SUBROUTINE **
202 LPRINT DA$; CHR$(91); LR; ST$; CHR$(93)
203 LPRINT NA$
204 LPRINT AD$
205 LPRINT CT$; ST$; ZP$
206 LPRINT CHR$(138); CHR$(138)
207 RETURN

```

```

208 REM ** RECORD # INPUT SUBROUTINE **
209 PRINT@192,
210 PRINT "INPUT THE LR # FOR THE FILE YOU ARE WORKING ON ";
211 INPUT LR
212 RETURN

```