# ROCKY MOUNTAIN 99'ers

# TIC TALK

VOL II, NO 13 DENVER, COLORADO USA SEPTEMBER 1984
Non-member Subscription Rate - $7.50 Annually Single Copy Price - 75 cents

## ELECTION OF OFFICERS

During the September meeting we will be electing the officers for the coming year. Your support of the elected officers will be needed for a strong and active User's Group. The nominations will be open for anyone who is willing to help. There will be some refreshments after the elections. Bring your questions (and maybe some answers) to this important meeting. In accordance with the By-Laws, the nominating committee has recommended the following individuals for your consideration:

| | |
|---|---|
| PRESIDENT | TED MICHELSEN |
| VICE PRESIDENT | PETE CROWELL<br>MIKE HOLMES<br>LLOYD MAPLE |
| SECRETARY/MEMBERSHIP | MARTHA WEEG |
| TREASURER | KEN MONSON |
| EDITOR | WAYNE LUEDTKA |
| LIBRARIANS | BOB & RICK GROSSART |

SPECIAL INTEREST GROUPS WILL HAVE COORDINATORS AS S.I.G. DECIDES

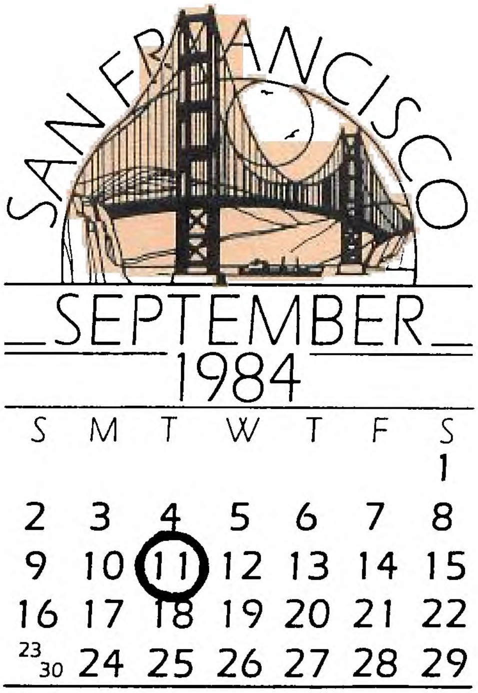
SEPTEMBER MEETING

SEPTEMBER 11

Jefferson County Fairgrounds

Auditorium 7:00 PM

6th Ave. West to Indiana Ave.



# PROGRAMING HINTS IN TI EXTENDED BASIC
## (PART I) by Ted Michelsen

With the extra programing features in TI Extended Basic it is possible to make very professional looking and performing programs. The use of the ACCEPT AT and DISPLAY AT along with the options allowed in such statements, such as: BEEP, SIZE, ERASE ALL and VALIDATE really make a program complete. The basic ACCEPT AT(row,Column) and DISPLAY AT(row, column) allow data input and printing anywhere on the screen. The options do the following:

BEEP -causes a beep to be make when the statement is executed, when data is displayed or ready to be input.

SIZE -allows the selection of data input or print field size and whether or not the field will be blanked before the data is printed or accepted. A minus sign in the size statement will cause the displayed value not to be erased when the ACCEPT AT statement is executed.

ERASE ALL -does the same as CALL CLEAR i.e. clears the entire screen.

VALIDATE -will only allow the input of the following types of data:

UALPHA -all uppercase alphabetic characters only.

DIGIT -only the numbers 0 through 9

NUMERIC -permits 0-9,".","+","-", and E for scientific notation.

"string expression" -only the characters listed by the string expression. The string expression must be enclosed in quotation marks.

Listed below is a short program that makes use of the ACCEPT AT and DISPLAY AT STATEMENTS to develop an address file.

```
90 DIM N$(20),T$(20),A$(20),
Z$(20),S$(20)
100 DISPLAY AT(1,1)ERASE ALL
:"DEMO OF DISPLAY AT"
110 DISPLAY AT(3,1):"NAME"
:: DISPLAY AT(4,1):"ADDRESS"
:: DISPLAY AT(5,1):"TOWN "
:: DISPLAY AT(5,16):"STATE"
:: DISPLAY AT(6,16):"ZIP"
120 FOR I=1 TO 20 :: ACCEPT
AT(3,6):N$(I):: ACCEPT AT(4,
9)SIZE(18):A$(I):: ACCEPT AT
(5,6)SIZE(9):T$(I)::ACCEPT A
T(5,23)SIZE(-2):S$(I)
125 ACCEPT AT(6,20)SIZE(5)
VALIDATE(NUMERIC):Z$(I)
130 IF N$(I)="" THEN 150
140 NEXT I
150 FOR J=1 TO I :: DISPLAY
AT(3,6)SIZE(19):N$(J)::DISP
LAY AT(4,6)SIZE(15):A$(J)::
DISPLAY AT(5,9)SIZE(9):T$(J)
152 DISPLAY AT(5,23)SIZE(5):
S$(J)
```

```
160 DISPLAY AT(6,23)SIZE(6):
Z$(J):: FOR K=1 TO 50 ::NEXT
K :: NEXT J
170 END
```

Now lets see what each statement does.

Statement 90 is the dimension statement which reserves space for 20 pieces of data for each variable.

Statement 100 erases the screen, then displays starting a row 1,column 1 DEMO OF DISPLAY AT, followed by NAME beginning at row 3 column 1,followed by ADDRESS beginning at row 4 column 1,followed by TOWN beginning at row 5 column 1,followed by STATE beginning at row 5 colunm 16, followed by ZIP beginning at row 6 column 16.

Statement 120 starts a FOR NEXT loop of 20 cycles in which data for variable N$(I), where I is the loop counter and increases by 1 at each cycle of the loop, is accepted beginning at row 3 column 6, followed by data being accepted beginning at row 4 column 9 for variable A$(I) but limited to 18 characters. Followed by data being accepted beginning at row 5 column 6 limited to 9 characters for the variable T$(I). In each of the above cases the allowed number of characters will be blanked out before the new data is accepted. The next step will accept data beginning at row 5 column but limited to 2 spaces for variable S$(I), but the last input will not be blanked as before, thus if the "ENTER" key is pressed the old value is re-entered for the new value. That is the same as including a default value in a program.

Statement 125 accepts data beginning at row 6 column 20 for variable Z$(I), but it is limited to 5 characters and only a numeric value will be accepted. If you attempt to enter a letter the computer will "honk" and not accept the character.

Statement 130 tests to see if null or no value was input for variable N$(I), which is done by pressing the "ENTER" key without first pressing another key for the prompt NAME. That will end the input of data.

Statement 140 just ends the loop.

Statement 150 starts a loop that will display the data just input. It displays the data in the correct space.

Statement 160 continues statement 150 and includes a delay loop, in this case 50 "do nothing" steps to slow the computer down so you can read the displayed values. This statement also contains the end of the loop started in statement 150.

Statement 170 ends the program.

By adding the following statements, we can save the data we input to either a disk or a tape.

```
162 OPEN #1:"DSK1.MAILLIST",INTERNAL ,OUTPUT,VARIAB LE 128
::REM FOR DISK STORAGE
(162 OPEN #1:"CS1",OUTPUT FIXED 128
FOR TAPE STORAGE )
160 FOR J=1 TO I::PRINT #1:N$(J),A$(J), T$(J),S$(J),
Z$(J)::NEXT J
```

In the future we will look at other useful extened basic statements.

## CAVE MAZE:
## AN ADVENTURE GAME

by Jon Todd

(reprinted from Wash DC UG -12/83)

You're lost in a subterranean labyrinth -- a maze of caves, each indistinguishable from the other. Suddenly, in the dim light, you see something sparkle! You reach for it and discover a faultless diamond! Pocketing the treasure, you continue your seemingly futile search for a way out.

Many adventure games require the player to negotiate a maze similar to the one just described. Although it seems complicated during play, the maze is very easy to program.

The following program in TI Basic generates a 12-room maze (one of the rooms is actually a testing location, where you are either allowed to leave the maze or are forced to go back). As in all good mazes, the rooms are identical in appearance and are interconnected by winding passages. The result is that when one leaves a cave heading south, he may find himself in a cave actually located to the north. He may even find himself back in the same cave he just left. Since the caves are identical, the adventurer must somehow mark the cave if he is to recognize it on a later visit.

Traditional maze-solving involves leaving objects in each cave which are recognized and possibly retrieved on a return visit. Cave maze presents an easier approach. One of the rooms contains a piece of chalk that may be used to write with. The WRITE/DRAW section handles the creation of your messages, and lines 420-430 will show you what message (if any) you wrote on any previous visits.

As in September's epic "Farmer's Dilemma," all direction values and messages are stored in arrays. Thus OB(1) is the location of object #1, the chalk, and OB$(1) is the corresponding name of the chalk. OBDES$(1) is the supplemental description of chalk, which is printed if we type LOOK CHALK as a command (see LOOK section). MSG$(R) is the message you write with the chalk for each room (R). And, as was explained in the August newsletter, N(R), S(R), E(R), and W(R) represent the room numbers located north, south, east, and west of the current room (R).

Notice that in most sections that affect objects (i.e., GET, DROP, LOOK) the program compares the last 3 letters of your object command SEG$(V2$,LEN(V2$)-2,3) with the last 3 letters of the object name SEG$(OB$(A),LEN(OB$(A))-2,3) to determine which object you mean. This allows you to type GET DIAMOND or GET BLUE-WHITE DIAMOND and have the desired result.

### PROGRAMING TIPS
### (From Tri Cities UG)

Here is an easy way to get rid of those print statements whenever you want to put a bunch of words up on the screen by using the colon (:). The colon acts like a carriage return on a typewriter. Here is an example:

```
  100 CALL CLEAR
  110   PRINT  "THIS  IS  A
DEMONSTRATION OF"
  120 PRINT  "HOW  TO  STACK  A
BUNCH OF"
  130 PRINT "LINES TOGETHER."
  140 PRINT
  150 PRINT
  160 PRINT "USING THE COLON IS
EASY"
  170 PRINT
  180 PRINT
  190 PRINT
  200 GOTO 200
```

- Now use the colon to stack them up and save memory:

```
  100 CALL CLEAR
  110   PRINT   "THIS   IS   A
DEMONSTRATION OF":"HOW TO STACK
A BUNCH OF":"LINES  TOGETHER.":
: :"USING THE COLON IS EASY"
  120 PRINT : : : 130 GOTO 130
```

```
100 REM ****************
110 REM *  CAVE  MAZE  *
120 REM ****************
130 REM   BY JON TODO
140 REM   SEP 16,1983
150 REM
160 CALL CLEAR
170 DIM MSG$(12),N(12),S(12),E
(12),W(12)
180 GOSUB 1740
190 R=8
200 REM *****************
210 REM PRINT ROOM, STATUS
220 REM *****************
230 CALL CLEAR
240 IF R=3 THEN 1940
250 PRINT "YOU ARE IN:": :"A C
AVE"
260 IF R<>9 THEN 280
270 PRINT :"THERE IS A DOOR TO
 THE NORTH"
280 IF R<>6 THEN 300
290 PRINT :"THERE IS LIGHT COM
ING FROM  THE NORTH AND FROM T
HE WEST"
300 PRINT :"YOU ARE CARRYING:"
310 FOR A=1 TO 5
320 IF OB(A)<>99 THEN 340
330 PRINT "A "&OB$(A)
340 NEXT A
350 PRINT : :"YOU CAN SEE:"
360 FOR A=1 TO 6
370 IF OB(A)<>R THEN 400
380 PRINT :"A "&OB$(A)
390 GOTO 420
400 NEXT A
410 PRINT :"NO OBJECTS"
420 IF MSG$(R)="" THEN 470
430 PRINT :"SOMEONE DREW ON TH
E WALL:": :"'"&MSG$(R)&"'"
440 REM ****************
450 REM PARSER
460 REM ****************
470 V2$=""
480 PRINT
490 INPUT "COMMANO? ":V$
500 FDR A=1 TO LEN(V$)
510 IF SEG$(V$,A,1)=" " THEN 5
50
520 NEXT A
530 V1$=V$
540 GOTO 600
550 V1$=SEG$(V$,1,A-1)
560 V2$=SEG$(V$,A+1,LEN(V$))
570 REM ****************
580 REM DIRECTION HANDLING
590 REM ****************
600 IF (V1$="GO")+(V1$="WALK")
=0 THEN 620
610 V1$=V2$
620 IF (V1$="NORTH")+(V1$="SOU
TH")+(V1$="EAST")+(V1$="WEST")
=0 THEN 840
```

```
630 X=R
640 IF V1$<>"NORTH" THEN 710
650 IF R<>9 THEN 700
660 IF LOCK=1 THEN 700
670 PRINT :"THE DOOR IS LOCKED
"
680 OB(6)=9
690 GOTO 470
700 R=N(R)
710 IF V1$<>"SOUTH" THEN 730
720 R=S(R)
730 IF V1$<>"EAST" THEN 750
740 R=E(R)
750 IF V1$<>"WEST" THEN 770
760 R=W(R)
770 IF R<>0 THEN 230
780 R=X
790 PRINT :"YOU CAN'T GO THAT
WAY"
800 GOTO 440
810 REM **************
820 REM GET ROUTINE
830 REM **************
B40 IF V1$<>"GET" THEN 1010
850 IF LEN(V2$)<4 THEN 960
860 FOR A=1 TO 5
870 IF SEG$(V2$,LEN(V2$)-2,3)<
>SEG$(OB$(A),LEN(OB$(A))-2,3)T
HEN 950
880 IF OB(A)<>99 THEN 910
890 PRINT :"YOU'VE ALREADY GOT
 THE ";OB$(A)
900 GOTO 440
910 IF OB(A)<>R THEN 960
920 PRINT :"YOU'VE GOT THE ";O
B$(A)
930 OB(A)=99
940 GOTO 440
950 NEXT A
960 PRINT :"THERE IS NO ";V2$;
" HERE"
970 GOTO 440
9B0 REM ****************
990 REM DROP ROUTINE
1000 REM ****************
1010 IF V1$<>"DROP" THEN 1150
1020 IF LEN(V2$)<4 THEN 1100
1030 FOR A=1 TO 5
1040 IF SEG$(V2$,LEN(V2$)-2,3)
<>SEG$(OB$(A),LEN(OB$(A))-2,3)
THEN 1090
1050 IF OB(A)<>99 THEN 1090
1060 PRINT :"OK, YOU DROPPED T
HE ";OB$(A)
1070 OB(A)=R
1080 GOTO 440
1090 NEXT A
1100 PRINT :"YOU DON'T HAVE TH
E ";V2$
1110 GOTO 440
1120 REM ****************
1130 REM DRAW/WRITE ROUTINE
1140 REM ****************
```

```
1150 IF (V1$="DRAW")+(V1$="WRI
TE")=0 THEN 1290
1160 IF OB(1)<>99 THEN 1240
1170 IF V2$<>"" THEN 1210
1180 PRINT "WHAT DO YOU WANT T
O ";V1$;"?"
1190 INPUT MSG$(R)
1200 GOTO 1220
1210 MSG$(R)=V2$
1220 PRINT :"OK, YOU'VE JUST W
RITTEN:": :"'"&MSG$(R)&"'"": :"
AND ERASED ANY OLD MESSAGE"
1230 GOTO 440
1240 PRINT :"YOU DON'T HAVE AN
YTHING TO":V1$&" WITH"
1250 GOTO 440
1260 REM ****************
1270 REM * LOOK ROUTINE *
1280 REM ****************
1290 IF V1$<>"LOOK" THEN 1480
1300 FOR A=1 TO 5
1310 IF V2$="" THEN 1430
1320 IF SEG$(V2$,LEN(V2$)-2,3)
<>SEG$(OB$(A),LEN(OB$(A))-2,3)
THEN 1360
1330 IF (OB(A)=99)+(OB(A)=R)=0
 THEN 1400
1340 PRINT :"YOU SEE:": :OBDES
$(A)
1350 GOTO 440
1360 NEXT A
1370 IF V2$<>"CAVE" THEN 1400
1380 PRINT :"YOU SEE:": :"A DI
MLY LIT CAVE; IT LOOKS  JUST L
IKE ALL THE OTHERS!"
1390 GOTO 440
1400 IF (V2$="DOOR")+(V2$="SLO
T")=0 THEN 1430
1410 PRINT :"A SIGN SAYS: 'TO
PASS NORTH YOU MUST INSERT A C
OIN'"
1420 GOTO 440
1430 PRINT :"I DON'T SEE A ";V
2$
1440 GOTO 440
1450 REM ***************
1460 REM * INSERT COIN *
1470 REM ***************
1480 IF V1$<>"INSERT" THEN 167
0
1490 IF R=9 THEN 1520
1500 PRINT :"THERE'S NO SLOT H
ERE"
1510 GOTO 440
1520 IF (V2$="SILVER COIN")+(V
2$="DIME")+(V2$="COIN")=0 THEN
 1620
1530 IF OB(3)=99 THEN 1560
1540 PRINT :"YOU DON'T HAVE A
";V2$
1550 GOTO 440
1560 PRINT :"OK"
1570 N(9)=6
```

```
1580 LOCK=1
1590 PRINT "THE DOOR IS OPEN"
1600 OB(3)=0
1610 GOTO 440
1620 PRINT "YOU CAN'T"
1630 GOTO 440
1640 REM ****************
1650 REM UNKNOWN COMMAND
1660 REM ****************
1670 PRINT :"I DON'T KNOW HOW
TO ";V1$
1680 GOTO 440
1690 REM ****************
1700 REM INITIALIZATION
1710 REM
1720 REM OBJECT LOCATION
       AND NAME
1730 REM ****************
1740 FOR A=1 TO 6
1750 READ OB(A),OB$(A)
1760 NEXT A
1770 DATA 2,PIECE OF CHALK,11,
GOLDEN NEDALLION,12,SILVER COI
N,7,BLUE-WHITE DIAMOND,4,SPARK
LING RUBY,0,SLOT BY THE DOOR
1780 FOR A=1 TO 5
1790 READ OBDES$(A)
1800 NEXT A
1810 DATA ORDINARY CHALK,REAL
GOLD!,A 1952 DINE,ABOUT 2 CARA
TS WORTH!,A REAL GEM!
1820 REM ****************
1830 REM ROOM DESCRIPTIONS,
       DIRECTION VALUES
1840 REM ****************
1B50 FOR A=1 TO 12
1B60 READ N(A),S(A),E(A),W(A)
1870 NEXT A
1880 DATA 1,4,4,1,1,5,5,4,0,6,
0,0,0,7,0,4,2,8,0,4,3,0,0,5
1890 DATA 7,10,8,7,5,11,9,7,0,
12,0,8,8,5,11,0,8,11,8,10,8,12
,12,1
1900 RETURN
1910 REM **************
1920 REM END ROUTINE
1930 REM **************
1940 IF (OB(2)=99)+(OB(4)=99)+
(OB(5)=99)=-3 THEN 2000
1950 PRINT :"YOU HAVEN'T YET F
OUND ALL OFTHE TREASURES.": :"
YOU MUST GO BACK!"
1960 FOR DELAY=1 TO 1000
1970 NEXT DELAY
1980 CALL CLEAR
1990 GOTO 250
2000 PRINT :"YOU HAVE FOUND AL
L OF THE   TREASURES AND HAVE
ESCAPED  THE MAZE!"
2010 PRINT :"GOODBYE"
2020 END
```

# QUICK COPYer PROGRAM REVIEW

"Why didn't TI think of that?" was my first reaction after using QUALITY SOFTWARE'S new QUICK COPYer program. Our D.C.User Group wizzes Larry Hughes and Oscar Farah have taken a great idea and refined it.

ALL important Disk programs should be "backed-uped". If you have a program or data that you "must keep" then you better have more than one copy of it for sooner or later a "bug" will bite your bytes and all will be lost. (Just ask past club Pres. Bill Whitmore what happens to mailing list data when no back-ups are made.) Let me emphasize again: should you have any program which you want to keep you better make 1 or 2 copies.

I have a Disk Drive and of course the Disk Manager Module from TI. This Manager does have a back-up feature under the "Disk Commands" menu. Have you used it? Have you watched the way the Disk Manager works? With 2 Disk Drives (one the Master, the other the Copy) the Manager will read a single filename or up to 11.5Kbytes of data, whichever is less, then write them to the Copy. This process will continue until all 90K is copied--if the Disk is filled. You will watch Disk 1 spin-read, then Disk 2 spin-write, etc. If you diskette happens to have 50 filenames then this process will keep these two drives bouncing back and forth for quite a while.

With only one Disk Drive things can get really hectic for now you must physically insert the Master--read, then take it out, insert the Copy--write, etc. again with only a single filename at a time or up to 11.5K bytes whichever comes first. I shudder to think of the fun one can have with a diskette with 20,30 or 127 filenames to copy.

Now comes QUALITY SOFTWARE's QUICK COPYer program to our rescue. With a 32k RAM attached to our system, why not fill it up with data from our Master Disk and then read it out to our Copy Disk? That is exactly what QUICK COPYer does. It allows us to fill up our machine with bytes then copy that data. Thus we are not limited to a single filename or only 11.5K chunks at a time. Thus an entirely filled diskette (all 360 sectors) can be copied in only 3 passes or less if single-sided, 6 passes or less if double-sided. Just thinks of the savings:

1. Time--especially if you have only one Disk Drive
2. Disk Drives--not all that wear and tear stopping, starting, searching,etc.
3. Nerves--insert Master, insert copy, click, whirl, buzz, etc.
4. Program--we need to save and back-up our data, some are not doing this with the Disk Manager Module because of all the hastle involved.

QUICK COPYer comes in three versions: 1.for Mini-Memory, 2.for Editor/Assembler, and 3.for Extended Basic. The difference being that Mini-Memory versions will copy a 90K disk in 3 passes or less--the other versions in 4 passes or less.

As I said in the beginning, why didn't TI think to let us use the memory we had available to help speed up our back-up procedures? Thanks Larry and Oscar for giving us another QUALITY SOFTWARE* product. I like it. I like it. (Notice by back-ups.)

Rev. Keith G. Koch
Washington DC Users Group
Reprinted with permission

* QUALITY 99 SOFTWARE, 1884 Columbia Rd. #500, Washington, DC 20009 202-667-3574
($39.95, plus $2 shipping & handling. Specify which module.)
Also reviewed in Enthusiast '99, January 1984, page 51.

## Autoloader for XBASIC discs

With TI's new pricing on the PES with 32k ram and disc we expect this XBASIC utility will be of interest to many more people than in the past. This program was published in 99'er some time ago. It utilizes three XBASIC functions, which are not available when using the console only. The functions are:

- Automatic loading of a disc file named LOAD on start-up
- Ability to have one program load and run another
- Ability to 'POKE' information directly into memory

```
100 REM ********************
110 REM * GENERAL PURPOSE *
120 REM *  DISKETTE MENU  *
130 REM *  BY  A. KLUDGE  *
140 REM ********************
150 OPTION BASE 1 :: DIM PG$(20):: CALL CLEAR :: CALL CHARPAT(97,A$):: IF SEG$(A
$,7,8)="38447C44" THEN 180
160 DISPLAY AT(15,2):"abcdefghijklmnopqrstuvwxyz"
170 FOR I=65 TO 90 :: CALL CHARPAT(I,A$):: B$="0000"&SEG$(A$,1,4)&BEG$(A$,7,4)&B
EG$(A$,13,4):: CALL CHAR(I+32,B$):: NEXT I :: CALL KEY(5,I,I)
180 IMAGE ##
190 DISPLAY AT(1,9)ERASE ALL:"DISKETTE MENU" :: DISPLAY AT(12,6):"DISK? (1-3): 1
" :: ACCEPT AT(12,19)BIZE(-1)VALIDATE("123"):D$
200 D$="DSK"&D$&"." :: OPEN #1:D$,INPUT ,RELATIVE,INTERNAL :: INPUT #1:N$,A,A,A
:: DISPLAY AT(1,8)ERABE ALL:SEG$(D$,1,3)&"    "&N$;:: I=0
210 FOR X=1 TO 20 :: I=I+1 :: IF I>127 THEN K=X :: GOTO 280
220 INPUT #1:P$,A,B,B
230 IF LEN(P$)=0 THEN 260
240 IF ABS(A)<>5 THEN 220
250 DISPLAY AT(X+2,10):USING 180:X :: DISPLAY AT(X+2,14):P$ :: PG$(X)=P$ :: NEXT
 X
260 DISPLAY AT(X+2,10):USING 180:X :: DISPLAY AT(X+2,14):"EXT-BASIC" :: DISPLAY
AT(X+4,2):"CHOICE? 1"
270 ACCEPT AT(X+4,10)SIZE(-2)VALIDATE(DIGIT):K
280 IF K=X THEN CALL CLEAR :: CLOSE #1 :: END
290 IF K<1 OR K>20 OR LEN(PG$(K))=0 THEN 260
300 CLOSE #1
310 CALL INIT :: CALL PEEK(-31952,A,B):: CALL PEEK(A*256+B-65534,A,B):: C=A*256+
B-65534 :: A$=D$&PG$(K):: CALL LOAD(C,LEN(A$))
320 FOR I=1 TO LEN(A$):: CALL LOAD(C+I,ASC(SEG$(A$,I,1))):: NEXT I :: CALL LOAD(
C+I,0)
330 RUN "DSKX.1234567890"
```

### ((((( DISPLAY ADS )))))

```
        10 in X 7.5 in - $30.00    ALL DISPLAY ADDS must be camera ready
RATES: 5.5 in X 7.5 in - $16.00    and must be received before the 15th
         3 in X 7.5 in -  $9.00    of the month and accompanied by a
```

check made out to the ROCKY MOUNTAIN 99ers P.O. Box 3400, Littleton, CO 80161. Since the Club is a non-profit organization all money collected for advertizing goes toward the publishing costs of this newsletter.

==========================================================

### ((((( WANT AD RATES )))))

MEMBERS - FREE (25 word max) We must have your add by the 15th of the month to assure insertion in the next issue. Call 979-6677 or mail to Box 3400 Littleton, CO 80161. NON-MEMBERS must use DISPLAY ADS!

Rocky Mountain 99'ers

# TIC TALK

This publication is printed monthly for the benifit of the membership of the Rocky Mountain 99'ers Computer Club. The Club and the paper are not for the benifit nor backed by any commercial enterprize. Both are non-profit in nature and are for the sole purpose of computer education. Any fees collected are used to defray any cost to maintain the organization. Neither the paper nor the Club have any affiliation with Texas Instruments. Any statements published in this paper are not necessarily the opinion of the membership.

## \`\`\` OFFICERS and CHAIRMEN ´´´

| | | |
|---|---|---|
| PRESIDENT | RON KUSESKI | 444-1797 |
| VICE PRESIDENT | TED MICHELSEN | 986-3513 |
| SECRETARY | MARTHA WEEG | 320-5589 |
| TREASURER | KEN MONSON | 233-1788 |
| EDITOR | LLOYD MAPLE | 979-6677 |
| LIBRARIAN | PETE CROWELL | 750-5949 |
| MEMBERSHIP | MARTHA WEEG | 320-5589 |
| PROGRAM CHAIRMAN | TED MICHELSEN | 986-3513 |
| EDITOR/ASSEMBLER..SIG | MIKE HOLMES | 751-7945 |
| TI FORTH.........SIG | PETE CROWELL | 750-5949 |
| MULTIPLAN........SIG | BEN KRAMER | 237-1056 |
| THE STAR BOARD....BBS | | 455-3113 |

* * ROCKY MOUNTAIN 99ers * *
P.O. Box 3400
Littleton, CO 80161

**FIRST CLASS**

********************************
* Do you see stars on the label *
* this means your membership is *
* now due. Send in your renew- *
* al today so you don't miss a *
* single issue of TIC-TALK!!! *
********************************