



ROCKY MOUNTAIN 99ERS TIC TALK

VOL IV, NO 6 DENVER, COLORADO USA FEBRUARY 1986
Non-member Subscription Rate - \$7.50 Annually Single Copy Price - 75 cents

FROM THE EDITOR

Well, I seem to be having a little better luck with the Post Awful (sorry WW). At least everyone got a newsletter last time.

Many thanks to the ones that have offered to help with the newsletter. The most help you can give me is in the form of articles or programs to publish. No matter how insignificant you may think a tip or hint may be, there is always someone out there that doesn't know that particular thing. I know that I have learned a lot from these types of things from other newsletters.

I still have not been able to locate the files with all of the publications that support the TI. I am still trying to find it, though, and will print it when I do. I am also searching for an article out of another User group's newsletter on replacing the battery in the MiniMem cartridge. That also will be printed when found.

Thanks to Sherry Card for the Basic Tutorial at the February meeting. I think a lot of people got something from it (judging from the questions).

MARCH MEETING

March 11

Jefferson County Fairgrounds

Auditorium 7:00 PM

6th Ave. West to Indiana Ave.

March 1986						
S	M	T	W	T	F	S
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

MERGE FILE EDITOR
Makes Programming Easier

By Michael C. Amundsen
New Horizons, January 1986

TI EDITOR IS GOOD, BUT

In the time I have spent writing TI BASIC and XBASIC programs, I have come to appreciate the TI Line Editor built into the console. If all the home computers, TI's Line Editor is about the best I've worked with. Few computers offer the easy editing of a single line (typing NUM XXX or EDIT XXX and using arrow keys, etc.) or the global resequencing of program lines (great when you have to insert a line later) that the TI Line Editor has. In fact, in many machines, you need to use a word-processor to generate your original textfile for the basic programs (goodbye automatic line-numbers!).

There are some times when I could use some more flexibility than the current TI Editor offers, though. There are four editing actions that I often need, but are not allowed by the built-in console editor. They are: 1) delete a series of lines (say a whole subroutine); 2) copy a series of lines to another file for use in other programs; 3) move a series of lines to another area in the same program (for example, move all data statements to the end of the program); and 4) delete only the REM lines to save memory space once the program is completed.

To meet my needs for a more flexible editor (and my need to continue to write programs!), I wrote a program called MFE (Merge File Editor) that allows the editing actions I described above. This program works only on XBASIC's MERGE Format files and requires a disk drive, expansion Memory and, of course, the XB cartridge. Below is a run-down of the capabilities of this small, but powerful programming aid.

WHAT THE MFE CAN DO

The MFE is great for doing little "spot-editing" in your programs. It allows you to copy or delete any line or

sequence of lines in your program, delete only the comment lines, and resequence any line or group of lines including moving a group of lines from one part of the program to another. All these functions can be done on any BASIC or XBASIC program as long as it has been SAVED in XBASIC's MERGE format.

DELETE-ing Lines

If you suddenly realize that the subroutine you just wrote is a duplicate of some other lines in your program, you could use the built-in editor to erase each line, one at a time (and sit and wait around!) or you could use the MFE to do it all at once.

MFE asks you what the starting and ending lines to delete are and then creates a new program file with the offending lines removed.

COPY-ing Lines

I often discover that the subroutine I need has already been written in some other program. Instead of getting the printout and sitting at the console typing the thing in again, I just use the MFE to copy the desired lines from the original program into another file for use in my new project. This saves time, effort and reduces the chance of typing errors in transferring the routine.

Deleting REM Lines

I tend to write a lot of comments in my programs as I am designing them. It helps me remember where I am headed when I come back to the project later on. But these comments use up precious memory and need to be removed to improve the speed of the program. I use the MFE to delete all 'REM' and '!' comment lines from my completed programs.

RESEQUENCING Lines

This is by far the most handy of the MFE functions. It allows me to outline a specific set of lines (say 1050-2015) and to resequence them using any starting line number (say 3000).

This may not seem handy at first, but I have come to love this feature of MFE. Below are some examples of the use of resequencing to help improve programs:

1 - KEEPING THINGS NEAT

I like to keep things easy to ready and edit when I write a program. I try

to start all major routines with similar line numbers like 1000, 2000, 3000, etc. and I try to keep all line numbers in increments of 10.

When I am de-bugging, however, things get a bit messed up, discovering the need to add an extra line can mess up the line numbers, and using the TI editor to resequence can botch up my 1000, 2000, 3000 sections too!

I can use MFE to fix this, though. I can tell MFE to resequence lines 1000-1135 in increments of 10 (or 5, 20, etc.) starting at 1000. No other lines will be affected and every jump-reference (GOTO, GOSUB, etc.) will be adjusted if needed. Handy, eh?

2 - MOVING THINGS AROUND

The MFE can also move entire sections of code from one part of the program to another. How many times have you discovered you have just written some program code underneath an XBASIC Subprogram? The program won't run because all Subprograms must be at the end of the program code! How about when you wish you had put that subroutine at the end of the file instead or the middle? Or how about wanting to put all your DATA statements in one section instead of scattered throughout your program? Do you delete the code and write it all again in the proper place? Not if you have MFE.

With MFE you can move any line of code by just changing the starting address of the resequencing. For example, say I wanted to move the DATA statements now at lines 350-460 down to the end of the file at around 1500. All I need to do is tell MFE to resequence starting at 350 and ending at 460 and start the new line numbering at 1500 in increments of 10. MFE does the rest!

MFE DISK AVAILABLE

MFE has become a standard tool in my programming arsenal, and I highly recommend it for anyone who does a lot of BASIC and XBASIC programming.

A program disk including on-line instructions is available for \$5 by contacting:

Michael Amundsen c/o SubFile99
POB 533, Bowling Green, OH 43402
CIS: 71706,625 STC: T13361

TIPS FROM THE TIGERCUB

#31

Copyright 1986

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, only \$3.99 each plus \$1.50 per order for PPM. Entertainment, education, programmer's utilities. Descriptive catalog \$1.99, deductible from your first order.

Tips from The Tigercub, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, just \$15 postpaid. Tips from the Tigercub Vol. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just \$15 postpaid. Or, both for \$27 postpaid.

Nuts & Bolts (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. All for just \$19.95 postpaid.

Nuts & Bolts No. 2, another full disk of 100 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also \$19.95

postpaid, or both Nuts Bolts disks for \$37 postpaid.

Tigercub Full Disk Collections, just \$12 postpaid! Each of these contains either 5 or 6 of my regular \$3 catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - my own programs on these disks are greatly discounted from their usual price, and the public domain is a FREE bonus!

TIGERCUB'S BEST PROGRAMMING TUTOR
PROGRAMMER'S UTILITIES
BRAIN GAMES
BRAIN TEASERS
BRAIN BUSTERS!
MANEUVERING GAMES
ACTION GAMES
REFLEX AND CONCENTRATION
TWO-PLAYER GAMES
KID'S GAMES
MORE GAMES
WORD GAMES
ELEMENTARY MATH
MIDDLE/HIGH SCHOOL MATH
VOCABULARY AND READING
MUSICAL EDUCATION
KALEIDOSCOPES AND DISPLAYS

For descriptions of these send a dollar for my catalog!

A few people have asked for a program that they could use to encode personal messages on a BBS. considering the current legal threats to BBS's, I doubt that a SysOp will allow coded messages, but here is a coder/decoder to create code that should be quite difficult to crack. First we need another of those programs that write a program -

```
100 !CODEPRINT by Jim Peters
on - creates a random code in
a MERGE format program COD
ESTRING to be MERGED into CO
DEMAKER
110 FOR J=1 TO 254 :: M$=M$&
CHR$(J):: NEXT J
120 FOR J=1 TO 254 :: RANDOM
```

```
IZE :: X=INT(RND*LEN(M$)+1):
: C=C&&SEG$(M$,X,1):: M$=SE
G$(M$,1,X-1)&SEG$(M$,X+1,LEN
(M$)):: NEXT J
130 OPEN #1:"DSK1.CODESTRING
",VARIABLE 163,OUTPUT :: PRI
NT #1:CHR$(0)&CHR$(1)&"C"&C
HR$(190)&CHR$(199)&CHR$(127)
&SEG$(C$,1,127)&CHR$(0)
140 PRINT #1:CHR$(0)&CHR$(2)
&"C2"&CHR$(190)&CHR$(199)&C
HR$(127)&SEG$(C$,120,127)&C
H R$(0)
150 PRINT #1:CHR$(0)&CHR$(3)
&"C"&CHR$(190)&"C"&CHR$(10
4)&"C2"&CHR$(0):: PRINT #1:
CHR$(255)&CHR$(255):: CLOSE
#1 :: END
```

And now the coder/decoder -
100 !TIGERCUB CODEMAKER writ
ten by Jim Peterson
110 !The MERGE format progra
m CODESTRING created by the
program CODEPRINT must be ME
RGEd into lines 1-3 of this
program

```
120 DIM A$(254):: DISPLAY AT
(3,6)ERASE ALL:"TIGERCUB COD
EMAKER" :: DISPLAY AT(12,1):
"Do you want to": "(1)Encode
":(2)Decode"
130 CALL KEY(0,K,ST):: IF K=
49 THEN 140 ELSE IF K=50 THE
N 290 ELSE 130
140 OPEN #1:"DSK1.CODE",VARI
ABLE 254,OUTPUT
150 DISPLAY AT(5,6)ERASE ALL
:"Type message in segments o
f": "not more than 254 charac
ters": "and Enter. When done,
type"
```

```
160 DISPLAY AT(9,1):"END and
Enter. Type slowly": "to avo
id skipped characters.": "Bac
kspace with FCTN S to": "corr
ect.": "Press any key"
170 CALL KEY(0,K,ST):: IF ST
=# THEN 170
180 CALL CLEAR :: CALL LONGA
CCEPT(0,M$):: IF M$="END" TH
EN 200
190 DISPLAY AT(20,1):"WAIT,
PLEASE - ENCODING"
200 FOR J=1 TO LEN(M$)
210 A$(ASC(SEG$(C$,J,1)))=SE
G$(M$,J,1)
220 NEXT J
230 FOR J=1 TO 254 :: RANDOM
IZE
```

```
240 IF A$(J)="" THEN A$(J)=C
HR$(INT(26*RND+65))
250 CODE$=CODE$&A$(J)
260 NEXT J :: PRINT CODE$
270 PRINT #1:CODE$ :: CODE$=
"" :: FOR J=1 TO 254 :: A$(J
)="" :: NEXT J :: GOTO 180
280 CLOSE #1 :: END
290 OPEN #1:"DSK1.CODE",VARI
ABLE 254,INPUT :: CALL CLEAR
:: DISPLAY AT(12,10):"DECOD
ING"
300 LINPUT :: CODE$ :: FOR J
=1 TO 254 :: M$=M$&SEG$(CODE
$,ASC(SEG$(C$,J,1)),1):: NEX
T J :: PRINT M$:: M$=""
310 IF EOF(1)<> THEN 300 ::
CLOSE #1 :: END
320 SUB LONGACCEPT(L,M$):: X
=# :: IF L<>0 THEN R=L ELSE
R=R+1
330 M$="" :: C=3 :: CH=140 :
CALL CHAR(140,RPT$(0,14)
&"FF")
340 CALL MCHAR(R,C,CH):: CH=
CH+5+(CH=160)*25 :: CALL KEY
(0,K,ST):: IF ST<1 THEN 340
350 IF K<>0 THEN 370 :: X=X-
1 :: C=C-1 :: IF C=2 THEN C=
30 :: R=R-1
360 M$=SEG$(M$,1,LEN(M$)-1):
GOTO 340
370 IF K=13 THEN 410
380 X=X+1 :: M$=M$&CHR$(K)::
CALL MCHAR(R,C,K):: IF X=25
4 THEN 410
390 C=C+1 :: IF C=31 THEN C=
3 :: R=R+1 :: IF R=25 THEN C
ALL CLEAR :: R=1
400 GOTO 340
410 R=# :: SUBEND
```

Here is a simple little game I call Cover-Up. Use the #1 joystick, try to cover the white square with the black square. Press the fire button to speed up, release it to slow down.

```
100 CALL CLEAR :: CALL CHAR(
96,RPT$(0,64)):: CALL SPRI
TE(0,96,5,92,124):: CALL MA
GNIFY(4):: CALL SPRITE(0,96
,16,100,100)
110 X=INT(20*RND)-INT(20*RND
):: Y=INT(20*RND)-INT(20*RND
):: CALL MOTION(0,2,X,Y):: T=
T+1 :: IF T=250 THEN 300
120 CALL JOYSPEED(1,1):: CAL
L COINC(0,0,0,0,0):: IF A=-1
```

```

THEN 130 ELSE 110
130 Z=Z+1 :: DISPLAY AT(1,1)
:Z :: CALL SOUND(-50,500,5):
: GOTO 120
300 CALL DELSPRITE(ALL):: DI
SPLAY AT(12,5):"YOUR SCORE I
S "&STR$(Z):: DISPLAY AT(20,
1):"PRESS ENTER TO PLAY AGAI
N"
310 CALL KEY$(#,K,S):: IF S=0
OR K<>13 THEN 310 :: T,Z=0
:: GOTO 100
21110 SUB JOYSPEED(N,A):: CA
LL JOYST(N,X,Y):: CALL KEY(N
,K,S):: S=S+K/9-1 :: S=S*AB
S(S)*0:: IF S>30 THEN S=30
21111 CALL MOTION(#A,-(Y*S),
X*S):: SUBEND

```

For a one-handed BREAK, if you can't reach FCTN and 4, try FCTN with J and the space bar together.

If you like to call BBS's, try the TIBBS Spirit of 99 BBS in Columbus, Ohio on (614)451-0800 and leave me a "hello!"

Probably useless info - holding down FCTN and CTRL together and typing 1, 2, 3 and 5 will give ASCII codes 145, 151, 133 and 140, which are the codes obtained from CTRL Q, W, E and T, the keys diagonally below the 1, 2, 3 and 5.

Occasionally someone sends me a program they have keyed in from my newsletter, and asks why it won't run, so I wrote this routine to help find the errors. It is also useful to check whether two copies of a program are identical, but only if they have not been resequenced.

100 !CHECKER by Jim Peterson
- to compare two programs and list all differing lines to the printer

```

110 DISPLAY AT(12,1)ERASE AL
L:"1st program DSK/filename?"
:"DSK" :: ACCEPT AT(13,4):F
14
120 DISPLAY AT(12,1)ERASE AL
L:"2nd program DSK/filename?"

```

```

:"DSK" :: ACCEPT AT(13,4):F
20
130 OPEN #1:"DSK"&F1$:INPUT
:: DIM M$(500),CH(500):: OPE
N #2:"PIO",VARIABLE 255 :: P
RINT #2:CHR$(15)
140 X=X+1 :: LINPUT #1:M$(X)
:: M$(X)=M$(X)&" " :: IF EOF
(1)<> THEN 140 :: CLOSE #1
:: OPEN #1:"DSK"&F2$:INPUT
150 IF EOF(1)=1 THEN 230 ::
LINPUT #1:X$ :: X$=X$&" "
160 FOR Y=1 TO X
170 IF X=M$(Y)THEN CH(Y)=1
:: GOTO 150
180 NEXT Y
190 P2=POS(X$," ",1):: P2%=S
EG$(X$,1,P2-1)
200 FOR Y=2 TO X :: P1=POS(M
$(Y)," ",1):: P1%=SEG$(M$(Y)
,1,P1-1)
210 IF P2%=P1% THEN PRINT #2
:"1st program = ";M$(Y):"2nd
program = ";X$ :: CH(Y)=1 :
: GOTO 150
220 NEXT Y :: PRINT #2:"2nd
program = ";X$ :: GOTO 150
230 FOR J=1 TO X :: IF CH(J)
=0 THEN PRINT #2:"1st progra
m = ";M$(J)
240 NEXT J
250 CLOSE #1 :: CLOSE #2

```

Here's a great idea that was printed and reprinted in several newsletters -

```

At the beginning of a
program that will run only
in Basic, add the lines -
1 IF PI=0 THEN (first line
of program)
2 PRINT "YOU ARE IN EXTENDED
BASIC": "THIS PROGRAM RUNS
ONLY IN BASIC"
3 STOP

```

The idea is that PI is a function in XBasic with the value of pi, but is just a variable name in Basic with an undefined value of 0.

The trouble is, it doesn't work! If PI is keyed in from Basic and saved, it is saved in token format as a variable name, and when loaded back into XBasic is still just a variable name. And if PI is saved from XBasic, it is tokenized as a function, loads back into Basic

as an unrecognized function and crashes! Can anyone come up with a way around that?

The above is the answer to the Challenge in Tips #30. Lines 100 and 110 were keyed in and saved from Basic, and loaded back into XBasic, then lines 120 and 130 were keyed in.

Here is a handy PEEK that hasn't been published as widely as most of them -

```

100 CALL INIT
110 CALL PEEK(8192,X)!Thanks
to Dale Loftis in the Orange
County US newsletter!
120 PRINT X !If X=32 you are
in Extended Basic; if X=165
you are in Basic with the
Editor Assembler or
MiniMemory module inserted.

```

And another 3-D sprite demo, just to make all the Apple polishers jealous. See if you can figure out how it works.

```

100 CALL CLEAR :: CALL SCREE
N(5):: CALL CHAR(100,RPT$("F
",64)):: CALL MAGNIFY(4):: F
OR S=5 TO 9 :: CALL COLOR(S,
16,1):: NEXT S
110 DISPLAY AT(3,3):"TIGERCU
B SPRITE SHUFFLE" !by Jim Pe
terson
120 DATA 70,116,2,75,121,7,6
9,124,11,70,115,16
130 FOR J=5 TO 8 :: READ P(J
,1),P(J,2),L(J):: CALL SPRIT
E(#J,100,L(J),P(J,1),P(J,2))
:: NEXT J :: W=45
140 DATA 3,6,7,8,8,3,6,7,7,8
,5,6,6,7,8,5
150 RESTORE 140 :: FOR Y=5 T
O 8 :: READ A,B,C,D
160 FOR J=1 TO W :: CALL LOC
ATE(#A,P(A,1)-J,P(A,2),#B,P(
B,1),P(B,2)-J,#C,P(C,1)+J,P(
C,2),#D,P(D,1),P(D,2)+J):: W
=9 :: NEXT J :: GOSUB 100
170 NEXT Y :: GOTO 150
180 FOR J=5 TO 7 :: CALL POS
ITION(#J,P(J+1,1),P(J+1,2))
: NEXT J :: CALL POSITION(100
,P(5,1),P(5,2))
190 T=L(8):: L(8)=L(7):: L(7
)=L(6):: L(6)=L(5):: L(5)=T
200 FOR J=5 TO 8 :: CALL SPR

```

```

ITE(#J-4,100,L(J),P(J,1),P(J
,2)):: NEXT J
210 FOR J=5 TO 8 :: CALL SPR
ITE(#J,100,L(J),P(J,1),P(J,2
)):: NEXT J :: CALL DELSPRIT
E(01,02,03,04):: RETURN

```

Do you need some really REAL B16 letters on the screen? Just type your letter at the beep.

```

100 DIM X$(96):: CALL CLEAR
:: FOR CH=33 TO 89 STEP 8 ::
FOR A=0 TO 7 !REAL B16 LETT
ERS by Jim Peterson
110 CALL CHARPAT(CH+A,X$(CH+
A-32)):: CALL CHAR(CH+A,"0")
:: L$=L$&RPT$(CHR$(CH+A),3)
: NEXT A
120 FOR T=1 TO 3 :: R=R+1 ::
DISPLAY AT(R,4):L$ :: NEXT
T :: L$="" :: NEXT CH
130 CH$(1)=RPT$("0",16):: CH
$(2)=RPT$("F",16)
140 CALL SOUND(100,500,0)
150 CALL KEY$(#,CH,S):: IF S=
0 OR CH>96 THEN 150
160 CALL HEX_BIN(X$(CH-32),B
$):: FOR J=9 TO 64 :: CALL C
HAR(J+32,CH$(VAL(SEG$(B$,J,1
))+1))
170 NEXT J :: GOTO 140
180 SUB HEX_BIN(H$,B$):: HX$
="4123456789ABCDEF" :: B0$="
#####X0001X0001X0001X0001
1X011X0111X1000X10001X1010
011X1100X1101X110111X1111"
190 FOR J=LEN(H$)TO 1 STEP -
1 :: X$=SEG$(H$,J,1)
200 X=POS(HX$,X$,1)-1 :: T$=
SEG$(B0$,X$+1,4)&T0 :: NEXT
J :: B0=T0 :: T0="" :: SUBE
ND

```

Thought for the day. The excuses for piracy are exactly the same as the excuses for shoplifting, but you probably won't have to tell them to the judge - in this world, at least.

And that is almost

MEMORY FULL

Jim Peterson

SOME UNUSUAL COMPUTER LANGUAGES . . .

Copy provided by Rod Cook, Newark OH - Member OH-MI-TI

=====
Languages NOT included in the Commercial Language SIG or the Languages and Tools SIG

By Doug Bohrer, Bohrer and Company, Near Chicago and Ted A. Bear, NCA Corporation, In the Heart of Silicon Valley and A Usually Reliable Source, Digital Equipment Corporation, Somewhere in New England.

APL, BASIC, COBOL, FORTRAN, PASCAL, RPG...these programming languages are well known and (more or less) loved throughout the computer industry. There are numerous other languages, however, that are less well known yet still have ardent devotees. In fact, these little known languages, generally have the most fanatic admirers. For those who wish to know more about these obscure languages--and why they are obscure--we present the following catalogue.

C-
This language is named for the grade received by its creator when he submitted it as a class project in a graduate programming class. C- is best described as a "low level" programming language. In general, the language requires more C- statements than machine code instructions to execute a given task. In this respect it is very similar to COBOL.

DOG0
Developed at MIT (Massachusetts Institute of Obedience Training). DOG0 heralds a new era of computer literate pets. DOG0 commands include SIT, KEEL, STAY, PLAY_DEAD and ROLL_OVER. An innovative feature of DOG0 is "puppy graphics", a small cocker spaniel that occasionally leaves deposits as it travels across the screen.

FIFTH
FIFTH is a precise mathematical language in which the data types refer to quantities. The data types range from CC, DUNCE, SHOT and JIGGER to FIFTH (hence the name of the language), LITER, MAGNUM and BLOTTO. Commands refer to ingredients such as CHABLIS, CABERNET, GIN, VERMOUTH, VODKA, SCOTCH, BOURBON, CANADIAN, COORS, BUD, EVER-CLEAR and WHAT_EVERS_AROUND.

The many versions of the FIFTH language reflect the sophistication and financial status of its user. Commands in the ELITE dialect include VSOP, LAFITE and WAITERS_RECOMMENDATION. The GUTTER dialect commands include THUNDERBIRD, RIPPLE and HOUSE_RED. The GUTTER dialect is a particular favorite of frustrated FORTH programmers who end up using this language.

LAIDBACK

This language was developed at the Marin County Center for T'ai Chi.

Mellowness and computer programming (now defunct), as an alternative to the more intense atmosphere in nearby Silicon Valley.

The center was ideal for programmers who liked to soak in hot tubs while they worked. Unfortunately few programmers could survive there because the center outlawed Pizza and Coca-Cola in favor of Tofu and Perrier.

Many mourn the demise of LAIDBACK because of its reputation as a gentle and non-threatening language since all error messages are in lower case. For example, LAIDBACK responded to syntax errors with the message:

"I hate to bother you, but I just can't relate to that. Can you find the time to try it again?"

LITMP

This otherwise unremarkable language is distinguished by the absence of an "s" in its character set. Programmers and users must substitute "th". LITMP is said to be useful prothething liht. This language was developed in San Francisco.

REAGAN

This language was also developed in California, but is now widely used in Washington, D.C. It is the current subset of the international bureaucratic language known as DOUBLESPEAK. Commands include REVENUE_ENHANCEMENT, STOCKMAN, CAP_WEINBERGER, MALCOMB_BALDRIDGE, CABINET, CHIP_WOOD, LAXALT and SCENARIO. WATT and BURFORD have been removed from the commands while there is a current effort to add MEESE.

The operating systems used is NEW_RIGHT and the designated memory is THE_RANCH. The compile SCENARIO is a compile with NANCY followed by a link with BOMZO resulting in a SMOOZE. COMMIES (program bugs) are removed with the GRANADA command.

A REAGON program commences with LANDSLIDE and terminates with SENILITY.

RENE

Named after the famous French philosopher and mathematician Rene DesCaters, RENE is a language used for artificial intelligence. The language is being developed at the Chicago Center of Machine Politics and Programming under a grant from the Jane Byrne Victory Fund. A spokesman described the language as "Just as great as dis (sic) great city of ours."

The center is very pleased with progress to date. They say

they have almost succeeded in getting a VAX to think. However, sources inside the organization say that each time the machine fails to think it ceases to exist.

SATRE

Named after the late existential philosopher, SATRE is an extremely unstructured language. Statements in SATRE have no purpose; they just are. Thus SATRE programs are left to define their own functions. SATRE programmers tend to be boring and depressing and are no fun at parties.

SIMPLE

SIMPLE is an acronym for Sheer Idiot's Monopurpose Programming Linguistic Environment. This language, developed at Hanover College for Technological Misfits, was designed to make it impossible to write code with errors in it. The statements are, therefore, confined to BEGIN, END and STOP. No matter how you arrange the statements, you can't make a syntax error.

SLOBOL

SLOBOL is best known for the speed, or lack of it, of the compiler. Although many compilers allow you to take a coffee break while they compile, the SLOBOL compiler allows you to travel to Columbia to pick the coffee. Forty-three programmers are known to have died of boredom sitting at their terminals while waiting for a SLOBOL program to compile.

VALGOL

From its modest beginnings in Southern California's San Fernando Valley, VALGOL is enjoying a dramatic surge of popularity across the industry.

VALGOL commands include REALLY, LIKE, WELL and Y&KNOW. Variables are assigned with the =LIKE and =TOTALLY operators. Other operators include the California Booleans, AX and NOWAY. Repetitions of code are handled in FOR - SURE loops.

Here is a sample program:

```

      LIKE, Y&KNOW (I MEAN) START
IF PIZZA      =LIKE BITCHEN AND
GUY           =LIKE TUBULAR AND
VALLEY GIRL  =LIKE GRODY&MAX(FERSURE)##2
THEN
      FOR I =LIKE 1 TO ONTHAYBE 100
      DO:NAH - (DITTY##2)
      BARF(I) = TOTALLY GROSS(OUT)
SURE

      LIKE BAG THIS PROGRAM
      REALLY
      LIKE TOTALLY (Y&KNOW)
      INSURE
      GOTO THE HALL

```

VALGOL is characterized by its unfriendly error messages. For example, when the user makes a syntax error, the interpreter displays the message:

GAG ME WITH A SPOON!!

(This article first appeared in the APL SIG newsletter THE SPECIAL CHARACTER SET (D. BOHRER, EDITOR) and has gained steam ever since.) I took it from the Cleveland area newsletter 2/86 (who took it from the Northwest Ohio 99er News 1/86. David

This is a list of the tokens that we at during the February meeting. As I said then, you don't always get them to work, but the instances where they DO work make it worth knowing them. It's always worth a try! Just use the CTRL key with each letter or character. (IE- CTRL F=GOTO)

A=ELSE	B=::	C=!
D=IF	E=GO	F=GOTO
G=GOSUB	H=RETURN	I=DEF
J=DIM	K=END	L=FOR
M=LET	N=BREAK	O=UNBREAK
P=TRACE	Q=UNTRACE	R=INPUT
S=DATA	T=RESTORE	U=RANDOMIZE
V=NEXT	W=READ	X=STOP
Y=DELETE	Z=REM	1=TO
2=STEP	3=,	4=;
5=:	6=)	7=(
8=OPTION	9=OPEN	0=THEN
==CALL	/=AND	:=PRINT
.=ON		

Please take note that I used "=". You have to disable the "QUIT" key to use this. To do this, type the following: CALL INIT :: CALL LOAD(-31806,16) and then just hit ENTER. You can only "QUIT" with BYE or CALL LOAD(-31803,35). Just remember to type CALL INIT every time you use a CALL LOAD.

TEXAS INSTRUMENTS SALE

TI PRO Systems

TI Pro 156K, 1 disk drive, 3 Plane Graphics, monochrome monitor, and TI 850 Printer: includes Lotus 123, new Wordstar 3.3, Multiplan, MS DOS
\$1200

TI PRO 256K, 10 Meg TI Hard disk, 3 Plane Graphics, monochrome monitor, TI Modem and TI 850 Printer: includes Lotus 123, CPM 86, UCSD Pascal, new Wordstar 3.3, Multiplan, EasyWriterII, TTY Communications, and MS DOS
\$2600

TI PRO 64K, 2 disk drives, 3 Plane Graphics, and TI 850 Printer: includes new Wordstar 3.3, Multiplan, MS DOS (No Monitor)
\$900

TI PRO Software Specials (NEW)

WORDSTAR 3.3	\$65	(also IBM and CPM Versions Available)
MULTIPLAN	\$65	
EASYWRITER II	\$65	
TTY Communication	\$45	

BPI BUSINESS ACCOUNTING SYSTEM

6 Volume Set	\$400
Separate Volumes	\$100

InfoCom Games \$20

TI 850 Printers

.03, 150 cps, Bi-directional with Tractor, Serial and Parallel, Near Letter Quality and Double Strike, Epson Compatible Graphics
\$230

TI-99/4A

NEW STILL IN TI FACTORY CARTONS

TI P Code (Pascal) Card with Software \$200
TI 32K Memory Card \$80
TI BS232 and Parallel Card \$50 - SOLD

CALL KENT 694 8797 (DAYS) 797 6739 (EVES)

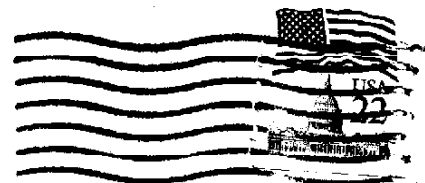
Rocky Mountain 99'ers

TIC TALK

This publication is printed monthly for the benefit of the membership of the Rocky Mountain 99'ers Computer Club. The Club and the paper are not for the benefit nor backed by any commercial enterprise. 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.....	COLEMAN EWING.....	795-0692
VICE PRESIDENT.....	JOE MILLER.....	458-5600
SECRETARY.....	BILL WARREN.....	341-7297
TREASURER.....	KEN MARBLE.....	986-0371
EDITOR.....	DAVID OWEN.....	458-7315
LIBRARIAN.....	DEAN SCHNEEBECK.....	795-2522
MEMBERSHIP.....	BILL WARREN.....	341-7297
PROGRAM CHAIRMAN.....	JOE MILLER.....	458-5600
EDITOR/ASSEMBLER..SIG.....	MIKE HOLMES.....	751-7945
TI FORTH.....SIG.....	PETE CROWELL.....	XXX-XXXX
MULTIPLAN.....SIG.....	BEN KRAMER.....	237-1056
THE STAR BOARD....BBS.....	SYSOP-MIKE HOLMES.....	455-3113
ADS UP.....BBS.....	SYSOP-"DR. RON".....	444-6129



* * ROCKY MOUNTAIN 99ers * *
P.O. Box 12605
Denver, CO 80212

<<<NOTE NEW ADDRESS!!>>>

FIRST CLASS

* Please check the expiration *
* date on your label. This is *
* the date your membership exp- *
* ired. Renew your membership *
* today so you don't miss a *
* single issue of TIC-TALK!!! *

Dallas TI H.C.Group
1221 Mosswood
Irving TX 75061