



MAY 1984 Vol. 2 No. 5

The May meeting will be held on Thursday, May 17 at Cuyahoga Falls High School on Fourth and Stow Street in Room 413 - Physics Lab. The June meeting will be held on June 21. Please remember to sign in and get a name tag.

This month's program will be TI Writer barring unforeseen circumstances. Pat has wanted to demonstrate TI Writer for the last few months but other circumstances got in the way.

Remember the survey sent to you in the newsletter last month? Tom has received TWO back. Please fill out the survey and send it in. We would like to know what kind of programs that you are interested in. If you have any suggestions for a good program or you would like to demonstrate computer hardware or software, please write it on the survey sheet and mail it back to Tom.
P.S. He feels like a Maytag repairman.

P I C N I C

Due to the lack of response the First Annual Picnic which was supposed to take place May 20 has been canceled. We would like to thank all that signed up but we feel at this time, the shelter we rented for 120 people could be enjoyed by another group due to the few people who signed up for our picnic.

We are in desperate need for a Vice President, Library Chairperson and Equipment Chairperson. The Vice President stands in for the President; Library Chairperson runs the library and the Equipment Chairperson makes sure the equipment stays put. If you would like any of these positions, we will be accepting volunteers at the May meeting. Only your participation in the group keeps our users group going.

Deadline for articles printing in the June newsletter is June 9. If you have a program, a review or something you think is valuable to the group, write it down and we will publish it in the newsletter.

Did you receive your copy of the Texas Instruments Home Computer Newsletter? If not or you have friends who own a TI Home Computer but

who may not be on their mailing list, please let them know, they would be happy to add their names. (NOTE: If the Home Computer was purchased since October, 1983, chances are they do not have the owner's address). New owners can send in owner information cards or a postcard with name, address and TI-99/4A serial number to:

TEXAS INSTRUMENTS

P.O. Box 53
Lubbock, TX 79408

TI SELECTS SOFTWARE VENDOR

Texas Instruments mailed to the users groups a letter, dated 3/28/84, effective April 2, 1984 TI will no longer sell products for the TI-99/4A Home Computer. TI had previously announced they were committed to providing product sales support until arrangements could be made. The letter stated "that arrangements now have been concluded to provide availability for all existing software through another organization". A firm not associated with TI, was selected to continue the marketing of its software products. The organization to contact for further information is:

TRITON

P.O. BOX 8123
SAN FRANCISCO, CA. 94128
U.S. (800) 227-6900 CA. (800) 632-4777

EDITORIAL: "Poor Ole TI"

Ever wonder whatever happened to all that manufacturing space and capacity TI had dedicated to the TI 99/4A that just suddenly went idle. And consider the employees . . . massive layoffs?

Have no fear my dear. In fact maybe we have a chicken and egg situation here. A news announcement in the April 3, 1984 PC WEEK, a weekly newspaper on IBM micro/computers said in part. . .

"A Texas Instruments manufacturing plant in Johnson City, TN, heavily contributes to the manufacture of IBM PCs. A TI employee who said he works at the plant claims that TI has been under contract to produce central processing units (CPUs) for IBM PCs since December 1983 under "Operation Moses". "The source said that the TI plant operates 24 hours a day, six days a week to produce PC CPUs." "TI laid off 800 workers when they discontinued the 99/4A, the source said, "but they hired them right back to work on the IBM project." "The TI Johnson City plant, with its advanced robotics and other assembly-line machinery already installed, was the simplest answer to IBM's PC production problems, the source said. However, the plant was used to make subassemblies for the defunct TI 99/4A home computer. TI also said it makes process control units at the huge facility. "The labor costs may be lower overseas, but TI's assembly lines just sitting there in Tennessee with nothing to do, IBM couldn't pass that up," he said. "The [TI] warehouse has stacks and stacks of IBM boxes," the TI employee said. "The CPUs are dropped into the IBM boxes and shipped to Boca Raton. Everything is included except the keyboard and CRT," he added."

The question is "Do you think in October when TI stopped they knew by December they would be producing the popular and profitable IBM?" Profit wise—"How many low priced TI's equal one IBM?" Now don't get me wrong, TI has the right to make a profit. That's why they are in business. But looking back over the grief, trama and indecision we users suffered, couldn't it have been handled better? Paul Hayden

GOOD PROGRAMMING

BY: MIKE NOBLE

Programming is a very personal thing. Two people programming the same problem do not necessarily arrive at the same program although their programs may yield the same results. We are all individuals, and we often approach a problem in different ways. Organizational processes can differ as a result of different educational or career backgrounds. An engineer with a great deal of mathematical training would probably choose an approach using complex mathematical equations, however, a liberal arts major may solve his problem using basic arithmetic functions. One person may be satisfied to use a set of instructions taking a great deal of program memory space, while another person may prefer to look for ways to condense their program to use the minimum amount.

As you get into the process of programming your style should grow. You should find this learning period adventurous and best of all fun. Don't be afraid to make mistakes. Explore - your computer won't mind. Your computer charges you essentially nothing for its time - so take advantage of this opportunity and experiment with alternate routes, functions, patterns, and anything else you can think of!

TI BASIC AND TI Extended BASIC permits both simple and complex programming. Simple programs may be entered, checked and run with little effort or difficulty. Even though these languages are designed to be as straightforward as possible, a complex program requires forethought and planning.

1. *Define the problem clearly and carefully:* Identify the formulas, variables and desired results. What is known? What is to be determined? How are the known and unknown related.
2. *Develop a method of solution (sometimes called an algorithm):* Define the sequence of the approach you want to use, keeping in mind the programming capabilities of your computer. (Remember strictly speaking, computers do not solve problems, you do. Your computer carries out solutions precisely the way you tell it to!)
3. *Develop a flow diagram:* It is often useful to develop drawings that help you visualize the flow of the program. Here, you can picture interactions between various parts of the solution. It may even be possible to simplify the program structure after it is flow charted.
4. *Begin making variable assignments:* Assign variables to the numerous thing you will be operating on. You'll continue this task throughout the programming process. It is a good idea to never assign a variable without making a written note that the variable in question contains that quantity.
5. *Translate the flow diagram into programming statements.*
6. *Enter the program.*
7. *Test the program:* check out the program using test problems representing as many cases as possible.
8. *Correct any errors:* Even programs written by professionals have mistakes.
9. *Retest the program:* Repeat steps 7 and 8 as needed.
10. *Document the user instructions:* It's always a good idea to carefully write down step-by-step instructions describing how to use your program. Even the most powerful programs are useless if you don't know how to use them.

If you have done little programming, you will find the above ideas useful. If you are an experienced programmer, the ideas will serve as a review and orient you toward better programming practices. You should interpret the following only as a list of suggestions since you will undoubtedly develop your own programming style.

BIGGIES

BITS



```

50 !ORIGINALLY WRITTEN FOR
    MAINFRAME AT CBS FOR
    "WIZKIDS"
60 ! TRANSLATED TO TI FOR
    YOUR ENTERTAINMENT
70 ! PROGRAM DEMONSTRATES
    HOW THE COMPUTER IS
    PROGRAMMED TO LOOK AS IF
    SOMEONE IS ENTERING DATA
    AND IT IS ANSWERING
80 RUTTEN=100
90 OK=20
100 GOOD=7
110 FAST=1
120 SLOW=4
130 CALL CLEAR
140 GOSUB 540
150 LINES=3
160 SPEED=SLOW
170 TYPIST=0
180 PROFICIENCY=OK
190 DATA "SYS-134 LOGON 06:3
8:10   HRD DSK B43-0010"
200 DATA ""
210 DATA "PLEASE ENTER ACCOU
NT NAME:  "
220 GOSUB 400 :: GOSUB 540
230 TYPIST=1

```

```

240 LINES=1
250 DATA "KILROY"
260 GOSUB 400
270 PRINT
280 GOSUB 540 :: GOSUB 540 :
: GOSUB 540
290 PRINT
300 TYPIST=0
310 LINES=4
320 SPEED=SLOW
330 DATA ""
340 DATA "Hello Richie."
350 DATA "This is ralf."
360 DATA "Why didn't you cal
l earlier?"
370 GOSUB 400 :: GOSUB 540 :
: GOSUB 540
380 PRINT : :
390 END
400 FOR MAIN=1 TO LINES
410 READ PHRASE$
420 PL=LEN(PHRASE$)
430 FOR CHARACTER=1 TO PL
440 IF TYPIST=1 THEN SPEED=R
ND*CHARACTER*PROFICIENCY
450 FOR DELAY=1 TO SPEED
460 NEXT DELAY
470 PRINT SEG$(PHRASE$,CHARA
CTER,1);
480 NEXT CHARACTER
490 IF MAIN<LINES THEN PRINT
500 FOR DELAY=1 TO 200
510 NEXT DELAY
520 NEXT MAIN
530 RETURN
540 FOR DELAY=1 TO 1000
550 NEXT DELAY
560 RETURN

```

50 !THIS IS A REPRINT OF THE
FAMOUS "LOVE PRINT" ART WOR
K BY ROBERT L. INDIANNA PASS

```

ED ON BY MICHAEL NOBEL OF SU
MMIT 99ERS.
60 !THE PRINTER CALL IS FOR
A TI99 PRINTER IN LINE 100 S
IMPLY CHANGE THIS TO YOUR PR
INTER.
100 INPUT A$ :: OPEN #3:"RS2
32.BA=9600.DA=B",OUTPUT,VARI
ABLE 255 :: PRINT #3:CHR$(15
)
110 INPUT "YOUR MESSAGE (60
CHARACTERS MAXIMUM? ":A$ ::
L=LEN(A$)
120 DIM T$(120):: FOR X=1 TO
10 :: PRINT #3: :: NEXT X
130 FOR J=0 TO INT(60/L)
140 FOR I=1 TO L
150 T$(J*L+I)=SEG$(A$,I,1)
160 NEXT I :: NEXT J
170 C=0
180 A1,P=1 :: C=C+1 :: IF C=
37 THEN 370
190 PRINT #3:
200 READ A :: A1=A1+A :: IF
P=1 THEN 220
210 FOR I=1 TO A :: PRINT #3
:" ";; NEXT I :: P=1 :: GOT
O 230
220 FOR I=A1-A TO A1-1 :: PR
INT #3:T$(I);:: NEXT I :: P=
0
230 IF A1>60 THEN 180
240 GOTO 200
250 DATA 60,1,12,26,9,12,3,8
,24,17,8,4,6,23,21,6,4,6,22,
12,5,6,5
260 DATA 4,6,21,11,8,6,4,4,6
,21,10,10,5,4,4,6,21,9,11,5,
4
270 DATA 4,6,21,8,11,6,4,4,6
,21,7,11,7,4,4,6,21,6,11,8,4
280 DATA 4,6,19,1,1,5,11,9,4

```

```

,4,6,19,1,1,5,10,10,4,4,6,18
,2,1,6,8,11,4
290 DATA 4,6,17,3,1,7,5,13,4
,4,6,15,5,2,23,5,1,29,5,17,8
300 DATA 1,29,9,9,12,1,13,5,
40,1,1,13,5,40,1,4,6,13,3,10
,6,12,5,1
310 DATA 5,6,11,3,11,6,14,3,
1,5,6,11,3,11,6,15,2,1
320 DATA 6,6,9,3,12,6,16,1,1
,6,6,9,3,12,6,7,1,10
330 DATA 7,6,7,3,13,6,6,2,10
,7,6,7,3,13,14,10,8,6,5,3,14
,6,6,2,10
340 DATA 8,6,5,3,14,6,7,1,10
,9,6,3,3,15,6,16,1,1
350 DATA 9,6,3,3,15,6,15,2,1
,10,6,1,3,16,6,14,3,1,10,10,
16,6,12,5,1
360 DATA 11,8,13,27,1,11,8,1
3,27,1,60
370 PRINT #3:CHR$(12):: REST
ORE :: GOTO 110 B

```

```

50 ! UNCLE BIGGIES ROOM
PAINTING CALCULATOR
60 CALL CLEAR :: FOR SET=1 T
O 12 :: CALL COLOR(SET,16,1)
:: NEXT SET :: CALL SCREEN(5
):: DISPLAY AT(12,6):"PAINT
CALCULATER": :TAB(6);" PRESS
ANY KEY"
70 CALL KEY(O,K,S):: IF S=0
THEN 60 ELSE CALL CLEAR
80 DIM WALLS(6)
90 HEIGHT,GALLONS$,SQFT,S1,S2
,S3,S4,S5,S6,TCST,CST=0
100 ! MAIN WALL LENGTH LOOP
110 FOR C=1 TO 6 :: WALL(C)=
0 :: CALL CLEAR :: PRINT "LE

```

```

NGTH OF WALL";C;"(feet)";
120 INPUT WALL(C)
130 INPUT "IS LENGTH CORRECT
? (Y/N) ":A$
140 IF A$<>"Y" THEN 100
150 NEXT C
160 ! WALL HEIGHT CALCULATIO
N INPUT
170 CALL CLEAR :: PRINT "HEI
GHT OF WALLS (feet)";
180 INPUT HEIGHT
190 INPUT "IS HEIGHT CORRECT
? (Y/N) ":A$
200 IF A$<>"Y" THEN 160
210 CALL CLEAR :: INPUT "NO.
OF SQ.FT./GALLON
(from label) ":SQFT
220 INPUT "IS SQ.FT. FIGURE
CORRECT ? (
Y/N) ":A$
230 IF A$<>"Y" THEN 210
240 ! CALCULATE AND PRINT VA
LUES
250 CALL CLEAR :: INPUT "COS
T PER GALLON ? (
from store) $":CST
260 INPUT "IS COST CORRECT ?
(Y/N)":A$
270 IF A$<>"Y" THEN 250
280 CALL CLEAR :: S1=HEIGHT*
WALL(1):: S2=HEIGHT*WALL(2):
: S3=HEIGHT*WALL(3):: S4=HEI
GHT*WALL(4):: S5=HEIGHT*WALL
(5):: S6=HEIGHT*WALL(6)
290 S7=S1+S2+S3+S4+S5+S6 ::
GALLONS=S7/SQFT+.5 :: TCST=I
NT(GALLONS)*CST
300 DISPLAY AT(4,1)ERASE ALL
:"LOOKS LIKE A JOB FOR SOME
FRIENDS AND A COUPLE OF
SIX PACS!"
310 DISPLAY AT(8,1):"GALLONS

```

```

REQUIRED";INT(GALLONS): ;"
OST $";TCST: ;"TOTAL SQUARE
FOOTAGE": : "=";S7;"SQ.FT. "
: TAB(5);"PRESS ANY KEY TO I
ND"
320 CALL KEY(O,K,S):: IF S=0
THEN 310

```

```

10 ! OUTPUT SUBROUTINE/
K-POWER MARCH 84 WITH
MINOR IMPROVEMENTS BY
BIGGIE. XBASIC REQUIRED
20 CALL CLEAR :: VT=1 :: A$=
"THIS IS A DEMONSTRATION OF
HOW EASY IT IS TO USE THIS
"
30 A$=A$&"SIMPLE SUBROUTINE
IN YOUR OWN PROGRAMS. LIST
THE PROGRAM AND LOOK AT ..."
40 GOSUB 1030
50 VT=10 :: B$=" LINES 10-15
0 TO SEE " :: GDSUB 2010
80 B$=" HOW TO PRINT LINES,
OR " :: GOSUB 2010
110 B$=" LINES 1000 TO 2080
TO " :: GOSUB 2010
140 B$=" SEE THE ROUTINES."
:: GOSUB 2010
150 B$=" PRESS ANY KEY TO CO
NTINUE" :: GOSUB 2010 :: CAL
L KEY(O,K,S):: IF S=0 THEN
50 ELSE 20 :: END
1000 ! OPTIONAL LINE BREAK
ROUTINE. USE THIS IF YOU
WANT THE OUTPUT SUBROUTINE
TO BE ABLE TO HANDLE LINES
THAT ARE LONGER THAN YOUR
1001 ! COMPUTER SCREEN IS
WIDE.
1030 IF LEN(A$)>27 THEN 1080
1040 B$=A$ :: A$="" :: GOSUB
2010 :: RETURN B

```

```

1080 Y=28
1090 FOR X=2 TO 29 :: IF SEG
$(A$,X,1)<>" " THEN 1120 ::
Y=X-1
1120 NEXT X
1130 B$=SEG$(A$,1,Y):: A$=SE
G$(A$,Y+2,LEN(A$)):: GOSUB 2
010 :: GOTO 1030
2000 ! HERE IS THE OUTPUT
SUBROUTINE
2010 M=LEN(B$)
2020 IF M/2=INT(M/2)THEN 205
0
2030 B$=B$&" "
2040 M=M+1
2050 DISPLAY AT(12,1):" " ::
FOR N=1 TO M/2
2070 DISPLAY AT(12,14-N):SEG
$(B$,1,N);SEG$(B$,M-N+1,M)::
: NEXT N :: CALL WAIT :: RET
URN
2080 SUB WAIT :: FOR X=1 TO
200 :: NEXT X :: SUBEND

```

```

50 CALL CLEAR :: ! HERE IS T
HE SAME THING WITHOUT ALL
THOSE LINES
60 DISPLAY AT(12,8)ERASE ALL
:"SHOW A MESSAGE" :: DISPLAY
AT(14,7):"BY NIRAJ N. SHAH"
70 INPUT M$
71 CALL CLEAR
80 L=LEN(M$):: H=INT(L/2)::
H=H+((L/2)-H>0)*(-1):: IF L=
0 OR L>28 THEN RETURN
90 FOR I=1 TO H :: DISPLAY A
T(12,14-I+1)SIZE(1):SEG$(M$,
H-I+1,1):: DISPLAY AT(12,14+
I)SIZE(1):SEG$(M$,I+H,1):: N
EXT I
100 GOTO 70

```



An exciting new way to learn about computers

A · C · A · D · E · M · Y
ON
COMPUTERS

Beginning Wednesday
April 18 at 6 p.m.
Repeated Thursdays at 11 p.m.
and Saturdays at 8 a.m.

LIST OF BOARD MEMBERS AND THEIR HOME PHONE NUMBERS

President, Pat Bowen	920-1884
V.P. Hard and Software, Librarian, Michael Noble	929-7453
V.P. Program, Tom Sanders	923-4131
Secretary, Paul Hayden	923-1958
Treasurer, Betty Duncan	633-5217
Educational Director, John Curry	929-8824
Editor, Kathi Anderson	923-7530



* TIGERCUB SOFTWARE *

156 COLLINGWOOD AVE.
COLUMBUS, OHIO 43213

TIPS FROM THE TIGERCUB #8

(614) 235-3545

I NOW HAVE 127 PROGRAMS READY, AND SOME MORE IN THE FUMIGATOR BEING DEBUGGED, FOR SALE AT JUST \$3 EACH, PLUS \$1.50 PER ORDER FOR ACTUAL COST OF CASSETTE, PACKAGE AND MAILING. NOW ALSO AVAILABLE ON DISKETTE, WITH A FEW EXCEPTIONS, AT THE SAME PRICE PLUS \$3 PER ORDER FOR THE DISK AND P&P.

MY CATALOG IS \$1, WHICH WILL BE REFUNDED ON YOUR FIRST ORDER.

THESE TIPS MAY BE REPRINTED BY ANYONE, WITH CREDIT TO TIGERCUB SOFTWARE. IF YOU PUBLISH A NEWSLETTER, PLEASE DO PUT ME ON YOUR MAILING LIST. MANY THANKS TO THE EDITORS WHO HAVE MENTIONED MY KITCHEN TABLE ENTERPRISE.

LAST MONTH'S CHALLENGE WAS TO WRITE THE EXTENDED BASIC STATEMENT IF X=1 THEN Y=7 ELSE IF X=2 THEN Y=33 ELSE IF X=3 THEN Y=19 ELSE IF X=4 THEN Y=21, IN JUST ONE LINE OF BASIC. THAT DIDN'T SEEM TO STIR UP MUCH INTEREST BUT ANYWAY, HERE'S MY SOLUTION -
Y=VAL(SEQ\$("07331921", X*2-1, 2))
YOU CAN EVEN PUT THAT IN A DEF AND THEN USE Y REPEATEDLY WITHOUT FURTHER DEFINING IT. DON'T BE PUTTING TI BASIC DOWN - IT'S POWERFUL STUFF!

SO HERE'S THE CHALLENGE FOR THIS MONTH. CAN YOU WRITE A PROGRAM IN TWO LINES OF BASIC, OR ONE LINE OF EXTENDED BASIC, TO COMPOSE AND PLAY RANDOM MUSIC IN 2-PART HARMONY IN THE KEY OF C? I'M NOT TALKING ABOUT JUST ANY OLD RANDOM FREQUENCIES, I WANT RANDOM MUSIC COMPOSED OF PAIRS OF NOTES THAT BELONG TOGETHER IN CHORDS OF THE KEY OF C! IF YOU FIGURE IT OUT, WRITE ME OR GIVE ME A CALL.

LAST MONTH WE PASSED ON DUANE FISCHER'S TIP THAT T1-99/4A CONSOLES WHICH DISPLAY V2.2 INSTEAD OF 1981 ON THE TITLE SCREEN WILL NOT RUN 3RD PARTY COMMAND MODULES.

IT IS NOW REPORTED THAT THE WHITE MODULES WITH 1983 ON THE TITLE SCREEN WILL GIVE THE SAME PROBLEM.

IF YOU HAVE SET UP YOUR EQUIPMENT, TRIED TO LOAD A PROGRAM, AND THE SCREEN PRINTS OUT "2.D CSI" OR SOMETHING ELSE GOOFY, DON'T PANIC - JUST UNPLUG THE CASSETTE INTERFACE CABLE FROM THE JOYSTICK PORT AND PLUG IT IN WHERE IT BELONGS!

DID YOU EVER ABSENTMINDEDLY TYPE SAVE CSI INSTEAD OF OLD CSI, PUSH RECORD, AND NOT REALIZE IT UNTIL YOU HAD ERASED A PROGRAM FROM YOUR CASSETTE? DID YOU KNOW THAT THE CASSETTE HAS TWO TABS ON THE BACK EDGE THAT CAN BE REMOVED TO KEEP THAT FROM HAPPENING? JUST SLIP THE TIP OF A KNIFE BLADE UNDER THEM, AND PRY UP TO SNAP THEM OFF. EACH SIDE OF THE CASSETTE IS PROTECTED BY THE TAB ON ITS BACK LEFT EDGE; WHEN THE TAB IS REMOVED, THE RECORDER'S RECORD BUTTON CAN'T BE PUSHED DOWN. LATER ON, IF YOU DO WANT TO RECORD OVER THAT SIDE, JUST PUT A BIT OF SCOTCH TAPE OVER THE HOLE.

ACCORDING TO MICHAEL A. COVINGTON IN COMPUTE! OF DECEMBER '83, IF YOU ARE USING A BLACK&WHITE TV FOR A MONITOR YOU CAN GET A SHARPER SCREEN BY STARTING YOUR PROGRAM WITH A LINE 1 CALL SCREEN(15).

100 REM DOLLARS & CENTS PRINTER DEVELOPED FROM A MICROSOFT ONE-LINER BY CHUCK EMERSON-HEJRY IN COMPUTE! 2/84
110 REM TO GENERATE PRICES TO DEMONSTRATE
120 A=INT(10*RN)+.1*INT(5*RN)
130 REM ONE-LINER TO PRINT
140 PRINT "\$"&SEQ\$(STR\$(A+.001),1,LEN(STR\$(A+.001))-1)
150 GOTO 120

100 !WHERE DID THIS COME FROM?
110 FOR CH=24 TO 30 :: CALL HCHAR(1,1,CH,768):: CALL COLOR(0,5,11):: NEXT CH :: GOTO 110

ASCII CODES 1 THROUGH 31 CAN BE PLACED ON THE SCREEN BY CALL HCHAR OR CALL VCHAR. ASCII 30 IS THE CURSOR, WHICH PRINTS OUT AS THAT BLACK SQUARE. THE OTHERS DO NOT REPRESENT A CHARACTER, CANNOT BE REDEFINED, AND ARE MOSTLY BLANK, BUT ASCII CODES 24 THROUGH 27 OFTEN CONTAIN ODD BITS OF GRAPHICS. ASCII CODES 1 THROUGH 23 ARE ALWAYS TRANSPARENT. CODES 24 THROUGH 31 ARE IN CHARACTER SET 0 WHICH IS NORMALLY TRANSPARENT, BUT IF YOU ARE IN EXTENDED BASIC IT CAN BE COLORED.

SOME OF THE BOOKS OF TI PROGRAMS ARE FULL OF STUFF THAT IS NOT WORTH KEYING IN, BUT THERE IS AT LEAST ONE GOOD ONE, CALLED TERRIFIC GAMES FOR THE TI-99/4A, BY HAL RENKO AND SAM EDWARDS - PROGRAMMED IN THE NETHERLANDS, PUBLISHED IN GREAT BRITAIN AND PRINTED IN FINLAND! YOUR BOOKSTORE PROBABLY WON'T HAVE IT ON THE SHELF, BUT CAN ORDER IT FOR YOU AT HALF THE COST OF MOST BOOKS. THE PUBLISHER IS ADDISON-WESLEY, 1983.

IF YOU ARE MAKING A BACKUP DISK AND YOU ACCIDENTALLY INITIALIZE THE MASTER INSTEAD OF THE COPY DISK, YOU WILL LOSE THE DATA ON THE MASTER. IF YOU KEY IN A PROGRAM, TRY TO SAVE IT TO A DISK, AND FIND THAT NOT ENOUGH SECTORS ARE AVAILABLE, YOU HAD BETTER HAVE ANOTHER INITIALIZED DISK OR A CASSETTE RECORDER AVAILABLE. THE SOLUTION TO BOTH PROBLEMS - WHEN YOU BUY A PACKAGE OF DISKS, INITIALIZE THEM ALL RIGHT AWAY! AND WATCH TO MAKE SURE THAT THEY DON'T HAVE ANY BAD SECTORS.

```
HERE IS A HANDY DEBUGGING ROUTINE.
RIGHT AFTER THE FIRST CALL CLEAR, PUT
IN THESE TEMPORARY LINES:
101 FOR @=1 TO 4
102 CALL COLOR(@,16,1)
103 NEXT @
104 GOTO 104
```

THEN TYPE LIST. AS SOON AS THE FIRST LINES HAVE SCROLLED TO THE TOP OF THE SCREEN, STOP THE LIST WITH FCTN 4. TYPE RUN. ALL THE NUMBERS AND PUNCTUATION WILL TURN WHITE. CHECK FOR 1'S INSTEAD OF I'S AND 0'S INSTEAD OF O'S AND VICE VERSA, EQUAL NUMBERS OF OPENING AND CLOSING PARENTHESES, MISPLACED COMMAS, ETC. THEN BREAK WITH FCTN 4, LIST (THE LAST LINE ON THE SCREEN) AND HYPHEN, ENTER, STOP IT AGAIN, ETC.

IN BASIC, YOU CAN PROGRAM 100 PRINT "HELLO":::::"GOODBYE" TO SCROLL BETWEEN LINES OF PRINT. IN EXTENDED BASIC, THE DOUBLE COLON :: IS USED TO SEPARATE MULTIPLE STATEMENTS. WHEN YOU LOAD A BASIC PROGRAM IN EXTENDED BASIC, YOU WILL USUALLY FIND THAT THE COMPUTER WILL REARRANGE THE COLONS "HELLO" : : : "GOODBYE" AND RUN THE PROGRAM PROPERLY. BUT SOMETIMES, ESPECIALLY IF YOU PUT COLONS IN FRONT, AS PRINT : : : "HELLO", THE COMPUTER MAY BECOME CONFUSED AND GIVE YOU SOME VERY PUZZLING ERROR MESSAGES. ON RARE OCCASIONS IT MAY EVEN REARRANGE THE COLONS INTO STRINGS OF DOUBLE SETS : : : : AND THAT WILL LOCK IT UP COMPLETELY. SO, EVEN WHEN PROGRAMMING IN BASIC IT IS A GOOD IDEA TO SEPARATE YOUR MULTIPLE COLONS : : : : - OR TO BE ABSOLUTELY SAFE, PUT A NUL-STRING :\$:;\$ BETWEEN THEM.

YES, TI BASIC WILL LET YOU USE @ IN A VARIABLE NAME. I NEVER USE IT IN A PROGRAM, BUT I DO TRY TO REMEMBER TO USE IT IN TEMPORARY DEBUGGING LINES, IN UTILITY ROUTINES WHICH I WILL SAVE TO MERGE OR TO BUILD PROGRAMS AROUND, IN MODIFYING OTHER PEOPLE'S PROGRAMS, ETC. THAT WAY, I DON'T BREED NEW BUGS BY DUPLICATING A VARIABLE NAME THAT IS ALREADY IN THE PROGRAM.

STOP THE PRESS! I HAVE JUST RECEIVED A REVIEW COPY OF A NEW BOOK CALLED THE TI-99/4A IN BITS & BYTES

BY ROMO A. LORETO .

THIS ONE IS WORTH THE MONEY! IT IS A THICK BOOK OF 142 BIG PAGES. THE FIRST 20 PAGES ARE FILLED WITH EXPLANATIONS OF PROGRAMMING STATEMENTS AND METHODS, WITH EXAMPLES. THE REST OF THE BOOK CONTAINS 50 PROGRAMS FOR YOU TO KEY IN, AND AT AN AVERAGE OF WELL OVER 2 PAGES PER PROGRAM THESE ARE DEFINITELY NOT MERELY SHORT ROUTINES. THERE IS A WIDE VARIETY OF GAMES, EDUCATION AND UTILITY PROGRAMS, ALL ORIGINAL, WRITTEN BY MR. LORETO'S ASSOCIATES. I HAVE NOT HAD TIME YET TO KEY ANY OF THEM IN, BUT HAVE PREVIOUSLY SEEN A FEW OF THEM RUN, AND THEY WERE GOOD! THE PROGRAMS ARE REPRODUCED FROM LISTINGS, FOR ACCURACY, AND LISTED IN LARGE PRINT IN 28-COLUMN FORMAT FOR EASY COPYING - DON'T YOU WISH EVERYONE DID THAT?

YOU CAN ORDER THIS BOOK THROUGH TIGERCUB SOFTWARE FOR \$14.99, POSTPAID.

OUT OF MEMORY IN LINE 5000, 80

HAPPY HACKIN'

JIM PETERSON

This article comes from A9CUG CALL NEWSLETTER, April, 1984 by Gary Matthews.

FLIPPIES

Everyone has heard about "Flippies" and most people know what they are even if they don't use them. This article will explain to the rest what one is and perhaps give the incentive to make one, to those who have hesitated.

First off - What is a floppy? It is a single sided floppy disk that has been altered so both sides can be used as a single sided initialized disk. Both sides are independent just as if each were a completely separate disk. There are reasons why people use them and reasons why some avoid using them. Some believe that if you turn a disk over and put information on the normally not used side, this will cause future problems. A disk normally rotates in a specific direction; let us say it is clockwise. A Double-sided disk on the other hand uses both sides but still rotates in just one direction. A floppy, when turned over, is now turning in the opposite direction that it normally turns. This is the crux of why some people will not construct and use flippies. They believe that disks were created to rotate in one direction and will become abrasive inside the disksleeve, cause particles of the magnetic oxide to rub off, and destroy the disk's information if the direction of the rotation changed back and forth. You can compare it to a cat's fur that simply doesn't like being rubbed the wrong way. I personally do not know of a single case where this has happened. Problems in disks are usually traced to bad handling and not to the fact that it is a floppy. I mention all this because it will scare the faint/hearted not to try to follow the directions that come past this point. It will also get me of the hook if they do it wrong.

If you look in many leading computer magazines you will find advertisements for these devices that look just like hole punchers, for about \$8.95. They are billed as a wonderful contraption that lets you "Scientiffically" double your disk storage capacity. If you buy one, you have just wasted seven dollars. They are nothing more than hole punchers with something stuck on the side to help locate just where to make the punch. Don't blow your money on those things that you would have to wait five weeks for it to arrive anyway. Instead just follow the directions included here. Buy a regular hole puncher for about two dollars. If you are lucky, you will find one that has a plastic bottom part that will catch the holes as they are punched. If it doesn't you just have to be a little more careful. Now look at the front of one of your floppy disks. You will see a rectangular notch about one inch down on the right side. Now look at the small hole on the same side of the jacket, but further down and closer in to the center hub circle. This smaller hole is where you will see the one soft sector hole when you rotate the disk manually. To turn a disk into a Flippy, you put the same one inch down notch and accompanying soft sector cut-out hole on the left side as a mirror image of the right side. Here is how you go about doing that. The most important part is in placing your alterations at the right place. One way to do this is to take a second disk (not the one you are altering) and after you have rotated it so you can see through the soft-sector hole, lay it backwards on top of the one you are going to alter. This will give the right place to mark the notch along the left edge as well as showing where to punch out the accompanying soft-sector hole. Remember also that the disk to be altered must have the soft-sector hole punched out on the other side too. Just to play it safe let us

say that under no circumstances does the magnetic oxide disk get cut; only the sleeve which houses it.

There are only two things left to point out that could cause you trouble if you aren't aware to watch out for them. Never should the oxide disk actually touch your hole puncher. You could easily scratch the oxide coating. A simple way to ensure this is to lift the envelope away from the disk proper and insert an appropriate width of paper in at the hub area so the puncher will only come in contact with paper instead of the oxide. The other thing to be careful of is when you punch the holes near the hub make sure the cut piece doesn't get loose and fall into the inside of the envelope. That too would cause scratches to the oxide and ruin your disk. That is why I said it would be good if your hole puncher is one that catches the holes that it cuts.

If you only have single sided disk drives then making your own Flippies is a very good economical way to get more room to store programs without buying another box of disks. As a way to convince people that doing this is NOT harmful, I would like to point out that some of the software that TI has lately put out comes on 'Flippy' type disks. Here is one last note to consider if you are the type to want backup disks of your valuable programs but can't afford to buy an extra disk for every one. Say you have two single-sided disks of programs; call one AAA and the other BBB; make each into a flippy and AAA's programs on the back of BBB, and vice versa. This way if one disk gets ruined, spilled on, stepped on, or whatever, you are still left with a complete copy of everything.

One final note about using a common round hole punch. Just glancing at your disk will let you know whether you are inserting Side 1 (the rectangular notch) or are using the flip Side 2 (the rounded notch). So now. . .go, be fruitful and multiply (your disk capacity). Gary Matthews

PRESIDENTS CORNER

The Summit 99'ers Users Group will lose it's most active member this month. Mike Noble will be going to Mississippi in mid-May for two months and from there wherever his job takes him. Since Mike has been a member of our Users Group he has written many articles for the newsletter, spent many hours teaching others the tricks of the 99/4A, provided usefull information on many topics, and has spent many hours building the library. On behalf of the entire Summit 99'ers Users Group I thank Mike for his contribution and wish him the best of luck in the future.

THANKS MIKE, BEST WISHES.

For those of you interested in your personal copy of the 99'ers U.G.A. newsletter can get on the mailing list by sending your name and address with \$10.00 to:

99'ERS USERS GROUP ASSOCIATION
ATTN: DONALD VEITH
3535 SO. H ST., #93
BAKERSFIELD, CALIF. 93304

I would like to thank Pat and Michael for their articles, and the articles from the other users groups in the USA. I am sorry to see Michael leave this area. I will miss all his articles. I wish him luck wherever he goes and in whatever he pursues. See you all at the meeting. Kathi Anderson, Editor

This review of SXB comes from the Washington DC Area Home Computer Users Group, April, 1984.

REVIEW OF S X B (Super Extended Basic)

In our club newsletter of Nov. 1982 (Vol#1,Noll) Larry Hughes (club Librarian, QUALITY SOFTWARE) wrote an article telling us that the 32K expansion RAM had an 8K sections reserved exclusively for Assembly Language programming and thus could not be used by Basic or Extended Basic programs.

Now comes along club member Jim Hollender (J&KH Software--Video Titles I,II,III) with something he calls SXB--Super Extended Basic. SXB is made up of just over 100 Assembly Language subroutines which are loaded with Extended Basic into the "unavailable" 8K of Expansion RAM. These subroutines just sit there waiting your CALL LINK (subroutinename) found in your program and off they go doing their thing. "Their thing" is found in Data Base, String Arrays, String, Integers and manipulations of the Video Display Processor. Example: you don't like the small letters to be baby caps on the screen, then enter CALL LINK("VMLCL") and all lower case letters will look like a normal typewriter.

SXB is really too big a program to review in one article so let me give you just a few highlights. You receive:

1. A diskette containing SXB program and self testings
2. A 7"x9" 3-ring notebook with 75 pages of information
3. A SXB info registration sheet allowing you to receive all updates, future routines and a subscription to SXBrief for the next 6 months.
4. Almost all SXB routines have an example of how they are invoked and what each one does.

DATA-BASE: allow you to describe, sort, delete, insert, replace, update, count, find, compare, build almost any type of data you want. The heart of SXB.

STRING ARRAY: allow you to count, determine length, translate character to new ones, encode/encrypt, decode, reverse, copy/swap, view string arrays.

STRING: allow you to go hex to binary or binary to hex, swap, restore, reduce length, delete character/spaces, fix length for strings.

INTEGER: allow you to add, multiply, subtract, divide, make 4 integers out of one numeric variable. (example: in ExBasic DIM A(250) allows for 250 integers; in SXB DIM A(250) could allow 1000 integers!)

VIDEO DISPLAY: affect the screen viewing area, you can manipulate the screen to read to it, pick up to 255 character from the screen as an input, change sprite colors, character colors, change the screen window from 32 columns times 24 row to 2xl or whatever. With "VMATYPE" you can type anywhere on the screen or fill it full without ever using the ENTER key.

I suppose the neatest thing about SXB is that it uses an area of RAM really not used by Extended Basic and that all of these subroutines are written in Assembly. This means that SXB is 1.super fast and powerful, and 2.I can use Assembly routines even though I don't know how to program in that language.

However, SXB is intended to be used in your own programs. That means that SXB will not do a thing by itself but sit in your RAM. SXB is not for beginners! It is for those who will be doing their own programs, but really can not or will not write an Assembly Language program to do what these will do.

SXB loads from Extended Basic is less than 30 seconds and will reside in your 32K RAM until you call for them. Jim has put a lot of work into this program and has made a powerful tool available to many who may need what these routines have to offer.

SXB lists for \$99.95, but I have seen it discounted through our club and our local TIBBS shows it at a discounted price.

Rev. Keith G. Koch

SUMMIT'99ers USERS GROUP
Kathi Anderson, Editor
3240 Bailey Road
Cuyahoga Falls, Ohio 44221