THE NATIONAL NINTY-NINER

VOL I - NO. 10 - SEPTEMBER, 1984

MONTHLY NATIONAL NEWSLETTER OF THE

99ERS USERS GROUP ASSOCIATION
3535 SO.H ST., #93
BAKERSFIELD, CALIF. 93304
(805) 397-4361
DON VEITH - EDITOR/PRESIDENT

CREATED FOR TI 99/4A HOME COMPUTER OWNERS AND USERS GROUPS

CORCOMP NEEDS OUR SUPPORT

The plea for owner and User Group assistance came directly from Jackie Sagouspe, Director of Marketing for CorComp. We had a telephone conversation about CorComp, its present problems, and redirection out of their present dilemma. I cannot emphasize enough how important it is for each of us to contact our distributors of CorComp products. Jackie and I both personally thank each of you in advance for your assistance.

CorComp has filed for Chapter 11 Bankruptcy. NOW, before you panic and get all excited, this form of bankruptcy is where a corporation admits it has some problems, then works with the court AND its creditors to find a solution. The firm is presently reorganizing itself to streamline operations and reduce costs.

Product development on the 9900 Micro-Expansion System is proceeding on schedule. Products previously developed by the firm include the RS-232 card, 32K Memory Expansion and the Double-Sided, Double-Density Disk Controller Card. These products are still being manufactured. In fact, solutions to all of the previously discover problems with the new Double-Sided, Double Density Disk Controller Card have been corrected. Watch this newsletter for additional details.

Now, we have a firm here, that despite its own admitted internal problems, has developed four (4) new products in just one short year of existence. I feel the survival of this firm is very important to owners of the TI-99/4A. Some of CorComp's distributors feel the market CorComp states exists is really not out here among 4A owners. Our project is to inform the distributors this is an incorrect assumption on their part.

What can I as an owner of a 99/4A do about this problem? If you believe CorComp's products are required to extend or enhance the utilization of YOUR COMPUTER, then call or write a major distributor and make your opinion known. NOW IS THE TIME TO MAKE THAT CALL YOU ALWAYS PUT OFF ON AN ISSUE YOU FEEL STRONGLY ABOUT! Let the distributors know we believe CorComp has products required by TI-99/4A owners and that you **WILL** purchase their products. For once, stand up and be counted among the people who got involved and make their opinion known.

DataBioTics Announces Products Available 4th Quarter 1984

This firm will have three (3) new products available in the Fourth Quarter of this year. Each product under development is detailed below. Further releases of information will appear in this newsletter as they become available.

The first product discussed was a word processing module. WordMaster I is a word processor in a cartridge for a price of $39.95. WordMaster II will contain the same software and a Centronics Parallel Interface for a printer for less than $100.00.

A new cartridge named Rembrandt will be available. The software in this cartridge will be based on FORTH and will be contained in a 48K chip. A 32K memory expansion card will be needed to use this cartridge. The software will also include a graphics generator and presentation capabilities with storage to cassette or disk.

SuperDisk is an Expansion System Card which can be addressed as any disk. The card will contain 128K of memory capacity. It will look like another disk controller and drive to the 99/4A. A possible option for virtual power is being discussed. The final development details on this product are still under review. A set of utiities to enhance operation will be included with the SuperDisk card.

WordMaster I, WordMaster II, Rembrandt, and SuperDisk are Registered Trademarks of DataBioTics.

NEW UGA STAFF WRITERS

I am pleased to announce the addition of three new staff writers for this newsletter. These additional writers will allow us to increase our coverage of programming languages to include Pascal. An additional two staff writers will also cover Assembly Language assisting our present outstanding articles by John Phillips. Additional articles on Basic and Extended Basic will be added by the third new staff contributor.

Edgar Dohmann appears for the first time this month with his introductory article on Pascal. He will also be writing additional articles on various aspects of Assembly Language. Edgar is a member of the Johnson Space Center Users Group (JUG). He possesses an outstanding background in both A/L and Pascal and contributed the Sprite Editor programs distributed as Public Domain to each Coordinator last month.

John Clulow is known to many of you in the Midwestern area of the United States. He has donated many fine programs to the Users Groups in his area of the country programmed in Assembly Language. John will be contributing articles when he is able due to an extremely busy personal schedule. He recently developed software for a Bulletin Board System.

Our third new staff writer this month is profiled later in this issue. This writer had her start with one computer magazine and presently is writing for another national computer publication. This individual writes articles, teaches school, assists schools systems in establishing computer programs, and is mom to five children. Cheryl Whitelaw is a name few readers will recognize. The pen name of REGENA is known to almost every owner of a TI-99/4A. We feel priviledged to add this individual to our staff of writers. She will begin contributing articles to the newsletter on a regular basis with our December, 1984 issue.

I know we all heartily appreciate the outstanding articles provided by our Staff Writers. Their articles create the newsletter our Coordinators and other Staff distribute each month. Every person associated with the 99'ERS UGA is important to our organization and the effort to provide you this newletter. Support and assist these people in any way possible. Each person pays their own expenses and donates their time to bring the articles you read here each month. Thank you!

PROFILE-REGENA

Cheryl Whitelaw got a TI-99/4A for Christmas of 1980 (that was when the price was about $650) and entered the exciting world of home computers. She started writing programs and articles for the 99'er MAGAZINE then became the Program Editor for that magazine for the second through sixth issues. She wanted to use an author name that might be easier to remember, so she chose REGENA, which happens to be her real middle name. All her computer work is done under the name REGENA.

Regena heard about the International 99/4 Users Group and called Charles LaFara only to find out there were about five other IUG members in Utah. Now, of course, there are hundreds of TI users in Utah. One of her first programs won the IUG contest to draw a state flag, the Utah flag. She currently is a regular columnist for the IUG publication, ENTHUSIAST '99.

In November of 1981 she was able to get the new TI-99/4A with the improved keyboard. In 1982 she got a TRS-80 Color Computer and has had articles and programs published in various TRS-80 magazines. Later that year home computer marketing trends were changing and a company hired her to write the sales programs for a VIC-20, so she entered the Commodore area. She also has worked a little with Scott, Foresman and was happy to visit their company and meet the people developing TI educational courseware. A lot of her early programming was in exchange for hardware. She has both the "old-style" and the peripheral box style expansion system.

In the Fall of 1982, her family moved to Cedar City, Utah, so her husband, Chandler Whitelaw, could become the Computer Center Director at Southern Utah State College. She submitted an article to COMPUTE! Magazine but didn't put a phone number because she was living in a motel. COMPUTE!, however, was able to trace her and invited her to visit their offices. She has been writing a monthly column for programming the TI since the January 1983 issue. At that time she also started writing books which are published by COMPUTE! Books.

Regena's first book, PROGRAMMER'S REFERENCE GUIDE TO THE TI-99/4A, is a compilation of her programs with explanations of how they were programmed. There were 40 non trivial programs in the book covering a variety of topics and illustrating the versatility of the TI-99/4A. This book hit the computer best seller lists in August 1983 and continued through January. It is still available from COMPUTE! or in major bookstores.

Next Regena edited COMPUTE!'s FIRST BOOK OF TI GAMES, which contains listings for 29 games for the TI. She translated several of the popular games that had previously been published in COMPUTE! for other computers.

BASIC PROGRAMS FOR SMALL COMPUTERS is another book of programs and includes listings for the TI (over 35 programs). While writing this book, Regena acquired a few more home computers and also added a Commodore 64 and an IBM PCjr so she could write the BASIC programming columns for COMPUTE!'s Gazette and COMPUTE!'s PC & PCjr.

Regena has written three other books, PROGRAMMER'S REFERENCE GUIDE FOR THE COLOR COMPUTER, COMPUTE!'s COMMODORE 64 TEACHER, and COMPUTE!'s IBM PC TEACHER (over 50 educational programs). She also has presented seminars and workshops about using microcomputers and assists in the school districts in implementing computers.

Regena has five children ranging in age from 4 to 15, and they give her ideas for programs to write. They are also anxious to try out any new games for the computers.

When the price dropped, Regena purchased several more TI computers. She also still uses her original TI-99/4A. Even though she has written for many brands of computers, the TI is still her "first love", and she will probably continue to write TI programs in the future years--it's a great computer!

Double-Density FORTH
by J. W. Vincent

This article is intended for all TI FORTH users who have (or plan on having) double density and/or double sided disk capabilities. While the techniques described should work with any disk controller capable of double density, the author's CorComp 9900 Disk Controller card is the only one that has been tested. The purpose of this article is to illustrate both how to access the additional screen capacity and how to modify the FORTH words and disc to be compatable with the new format and Disk Manager. Throughout this article lowercase letters used in a FORTH definition will indicate a variable value to be entered. The following terms will be used to refer to the various formats a FORTH disc may have.

```
 90 SCRN or SSSD - the original 90 screen single sided single density format
180 SCRN         - either a SSDD or DSSD disc when comment applies to both
360 SCRN or DSDD - a double sided double density disc
           SSDD - a single sided double density disc
           DSSD - a double sided single density disc
```

The first step is to use Disk Manager to format (initialize) a 180 or 360 SCRN disc. Next, you must copy FORTH from the 90 SCRN disc to the new 180 or 360 SCRN disc. The disc copy feature of CorComp's Disk Manager will do this properly for you. If you have two drives, the FORTH-COPY word in the -COPY screens will also do it properly (do 0 DISK_LO ! first). However, if you are using TI's Disk Manager II, after copying the three files you must use FORTH to copy screens 1 to 9 because Disk Manager II puts them in the wrong place! To do this, enter the following for each of the nine screens.

```
n BLOCK UPDATE ( where n is the screen number to be read from old disc)
FLUSH          ( after inserting the new disk - note: up to five screens may be entered at a time)
```

Now edit screen 3 of your new disc and add the following commands:

```
x DISK_SIZE !  ( where x = 180 or 360 as approprate)
y DISK_HI   !  ( where y = x times 1, 2, 3, or 4 depending on the number of drives you have)
```

Unfortunately, TI FORTH does not provide a method for configuring each drive individually. Therefore, the user must be cognizant of which screens are available on each drive when there are differences between them. At this point, FORTH can be booted and it will recognize the full capacity of your 180 or 360 SCRN disc. You can create, edit, list, and load from screens greater than 89. However, neither Disk Manager nor FORTH-COPY will recognize this disk as having more than 90 screens. To fix this problem you must modify the -COPY screens (39 and 40), the disc header (sector 0) and, the SYS-SCRNS file header (sector 4). First edit screen 39. Change the value 90, which appears once in DTEST and twice in FORTH-COPY to 180 or 360 as appropriate. Next, edit screen 40 as follows:

```
Line  3 - change 168 to 2D0 for 180 SCRN or 5A0 for 360 SCRN
Line  4 - change 944 to 1244 for SSDD or DSDD (no change for DSSD)
Line  5 - replace entire line with:
          DUP 10 + 2028 SWAP ! DUP 12 + a SWAP ! DUP 14 + 24 0 FILL
          where a = 0201 for DSSD, 0102 for SSDD, or 0202 for DSDD
Line 10 - change 165 to 2CD for 180 SCRN, or 59D for 360 SCRN
Line 13 - change 4016 to C02C for 180 SCRN, or C059 for 360 SCRN
```

Next edit screen 33 to modify the FORMAT-DISK word to:

```
: FORMAT-DISK 1+ a 33616 ! 18 SYSTEM ;   ( where a = 258 for DSSD, 513 for SSDD, 514 for DSDD)
```

Finally, you need to create a word that will modify the header sectors on your new disc. This word only needs to be executed once since copies of this disk, once it's modified, will not require modification. Here is the way to do it:

```
HEX 0 DISK_LO !              ( removes disc fence)
: DD-FORTH 0 BLOCK UPDATE    ( read screen 0 and mark as updated)
     DUP  A + a SWAP !       ( a = 2D0 for 180 SCRN, 5A0 for 360 SCRN)
     DUP  C + b SWAP !       ( b = 944 for DSSD, 1244 for SSDD or DSDD)
     DUP 10 + c SWAP !       ( c = 2028 for all versions)
     DUP 12 + d SWAP !       ( d = 201 on DSSD, 102 on SSDD, 202 on DSDD)
         38 + C8 FF FILL     ( flag all sectors as in use)
         1 BLOCK UPDATE      ( read screen 1 and mark as updated)
     DUP  E + f SWAP !       ( f = 2A0 for 180 SCRN, 570 for 360 SCRN)
     DUP 1C + g SWAP !       ( g = 4D20 for 180 or 360 SCRN versions)
     DUP 1E + h SWAP !       ( h = 2805 for 180 SCRN, 5205 for 360 SCRN)
         20 + i SWAP !       ( i = F029 for 180 SCRN, F059 for 360 SCRN)
     FLUSH ;                 ( write modified screens to disc)
DECIMAL DD-FORTH             ( execute it)
```

Now your new high capacity copy of FORTH is fully compatable with Disk Manager, the FORTH format, copy, test, and header words and your double density and/or double sided disk drives and controller. Enjoy!

PASCAL NOTES

by Edgar Dohmann - JSC Users Group (JUG)

I will be writing a regular monthly article on Pascal for the TI-99/4A. My goal is to present information of interest to all /4A owners who want to use Pascal. To accomplish this goal I will need your assistance and support. I will be glad to pass along any tips, experiences, and comments about Pascal on the /4A which you want to share with other users. I would like to include a question and answer column in my articles and I will appreciate hearing from some of you experienced Pascal users out there who would be willing to help solve some of the problems other users might have.

One project I would like to tackle is to prepare a library of public domain Pascal programs for the /4A that we can distribute to Users Groups through the UGA. If you would like to contribute to such a library, please let me know. As this project develops, I will publish guidelines for submitting programs.

I am in the process of upgrading my /4A system by adding a P-Code Card and replacing my lone SS/SD disk drive with two DS/DD drives. I regularly use Pascal at work and I am looking forward to being able to write Pascal programs on my home computer. At work I use two other Texas Instruments Pascal compilers. We use TI Pascal (TIP) on the TI minicomputers (DS990 machines) and Micro-Processor Pascal (MPP) on the TI microprocessor (TM990) products. We use both of these compilers to generate machine code for the target machine rather than for a P-System interpreter because our systems are used in real time applications and we have to optimize our programs for execution speed.

Given a choice of compiled machine code or an interpretive language, I would usually prefer the compiled code because of the faster execution speed. However, the only Pascal compiler TI offered on the /4A was P-System Pascal so that's the best we can do. Next month I hope to be able to write an article on actual /4A Pascal experience. This month though, I will present a little background on the P-System.

The P-System was developed at the University of California at San Diego and thus is often called the UCSD P-System. The P stands for pseudomachine because the machine language for the P-System was designed around a hypothetical processor. This hypothetical processor was given an architecture which lends itself well for execution of Pascal programs. It has a simple, idealized stack structure and all machine opcodes for the processor are one byte in length. One of the main goals in development of the P-System was to implement a method of making Pascal programs portable from one computer to another and allowing Pascal programs to be run in computers with limited memory space.

The two TI Pascal compilers I mentioned above produce very efficient machine code and are ideal high level languages for real time applications. However, the compilers themselves are quite large and the run-time support requirements generally occupy anywhere from 8K to 20K bytes depending on the sophistication of the program. The P-System avoids much of this problem by compiling the Pascal source code into the "idealized" pseudomachine code (P-code) which is then executed by an interpreter in the host computer. Since the P-code instructions are one byte long, the compiled P-code is generally much more compact than a similar program compiled to native machine code or even a program written in Assembly Language. In addition, run-time support for an interpretive language is generally less demanding than for a compiled language.

Once a program is compiled into P-code, it is theoretically portable from machine to machine. This is one way of running object code programs on various computers without having to have a compiler for each machine. Once a single editor and compiler are written for program development to the P-code level, all that is needed for different machines is an interpreter for the P-code. Since interpreters are much easier to design and develop than compilers, this was seen as an advantage to encourage adoption of the P-System.

To summarize things so far, the P-System offers the advantages of compressed object code which allows programs to fit into machines with limited memory space. It is also portable and software development for the host machine is limited. The primary disadvantage is that since the P-code runs in an interpretive mode, it is slower than most programs which are actually run in machine code. This drawback can sometimes be overcome by writing time critical routines in Assembly Language and linking it with less time critical portions which are compiled into P-code. Another technique available with some P-System implementations is a Native Code Generation utility which allows P-Code to be translated into a hybrid mixture of P-code and native machine code. This allows programs to be speeded up (at the expense of more memory storage space) for testing purposes without actually having to perform the Assembly Language conversion by hand. Unfortunately the TI Utility package for the /4A P-System does not include an NCG.

As you are probably aware, TI chose to implement the P-code interpreter for the /4A in ROM. This is why it is necessary to add a P-Code Card to the Peripheral Expansion Box in order to use the P-System on the /4A. The nice thing about this implementation is that we do not have to wait for the interpreter to be loaded off a diskette. Also TI included a console monitor operating system on the P-Code Card to create a complete P-System operating environment for developing, testing, and running programs. However, the price for this convenience is very high. Not only is the P-Code Card expensive, they are becoming very scarce and hard to find now that TI has dropped the /4A product line.

While I have described the P-System in conjunction with Pascal, it certainly is not limited to Pascal. SofTech Microsystems has taken over licensing and support of the P-System from UCSD and they currently have available a Fortran and a Basic compiler for the P-System. Like the Pascal compiler, these compilers also generate P-code which should be portable to any P-code interpreter. SofTech also distributes Native Code Generators for a number of processors including the 9900.

I hope this article has provided some additional insight into the P-System. If you are interested in using the P-System on the /4A, I suggest you start looking for a P-Code Card right away if you do not already have one. As I mentioned above, they are becoming very scarce. I would also like to hear from those of you who are already using the P-System. Let me know what you are doing, any problems you have had, how you solved them, and any unanswered questions you might have. Hopefully through this newsletter we can share knowledge and experience and make our P-Code Cards even more valuable.

TIPS FROM THE TIGERCUB No. 14

Distributed by Tigercub Software to users groups for promotional purposes and in exchange for their newsletters.

Tigercub Software, 156 Collingwood Ave., Columbus Ohio 43213, has over 130 original programs available at only $3.00 each. My catalog is available for $1.00 which may be deducted from your first order.

It has come to my attention that the members of some users groups have never heard of my kitchen-table enterprise, although their group has been receiving my newsletter for several months. It appears that many users groups have no method of making available to their members the information from newsletters which they receive in exchange.

My software business is a failure, and my Tips are a complete failure as a promotional effort. During the month of July I received a total of 31 requests for my catalog, of which only 9 were the result of newsletter publicity. I do not want to discontinue these Tips, because of the many interesting newsletters that I receive in exchange. However, I can no longer afford to distribute them to those groups which never give any indication that my Tips are reaching their members.

You may have observed that the Tigercub now possesses a Gemini 10X printer. The only fault I could find with it was that it wouldn't print Chinese, so I remedied that defect with this little program.

```
100 !THIS ROUTINE INITIALIZE
S THE GEMINI 10X TO PRINT ^C
HINESE' UNTIL IT IS TURNED O
FF - by Jim Peterson
110 OPEN #1:"PIO"
120 PRINT #1:CHR$(27);CHR$(4
2);CHR$(0);:: CALL CLEAR
130 FOR CH=65 TO 90 :: PRINT
 "WORKING..." :: FOR J=1 TO
7 :: FOR L=1 TO 9 :: RANDOMI
ZE :: IF (INT(3*RND+1)<3)+((
L>1)*(D(J,L-1)>0))THEN 150
140 D(J,L),D(J,10-L)=(1+ABS(
J>1)) J
150 NEXT L
160 NEXT J
170 FOR L=1 TO 9 :: FOR J=1
TO 7 :: X(L)=X(L)+D(J,L):: N
EXT J :: NEXT L
180 PRINT #1:CHR$(27);CHR$(4
2);CHR$(1);CHR$(CH);CHR$(1);
CHR$(X(1));CHR$(X(2));CHR$(X
(3));CHR$(X(4));CHR$(X(5));C
HR$(X(6));CHR$(X(7));CHR$(X(
8));CHR$(X9))
190 FOR J=1 TO 7 :: FOR L=1
TO 9 :: D(J,L)=0 :: NEXT L :
: NEXT J
200 FOR L=1 TO 9 :: X(L)=0 :
: NEXT L :: NEXT CH
210 PRINT #1:CHR$(27);CHR$(3
6);CHR$(1);
220 PRINT #1;CHR$(27);CHR$(8
7);CHR$(1)
230 PRINT #1:CHR$(27);CHR$(7
1)
240 STOP
```

Now, without turning off the printer, type LIST "PIO" or run any program that puts out text to the printer. It won't fool a Chinaman but it might impress your friends.

Here's a little something for you who own the Terminal Emulator II and the Speech Synthesizer. Maybe our Congressmen could use it to help them discuss the national debt.

```
100 CALL CLEAR
110 PRINT TAB(7);"NUMBER SPE
AKER": : :"by Jim Peterson":
"      of Tigercub Software"
: : :
120 PRINT " This program wil
l print any":" number of les
s than 67":"digits in number
s and in"
130 PRINT "words, and will s
peak the":"words.": : : :" R
equires Terminal Emulator":"
II and Speech Synthesizer.":
 : :
140 CALL CHAR(39,"0000000000
301020")
150 OPEN #1:"SPEECH",OUTPUT
160 DIM HIGH$(21),NN$(23)
170 DATA ONE,TWO,THREE,FOUR,
FIVE,SIX,SEVEN,EIGHT,NINE
180 DATA TEN,ELEVEN,TWELVE,T
HIRTEEN,FOURTEEN,FIFTEEN,SIX
TEEN,SEVENTEEN,EIGHTEEN,NINE
TEEN

190 DATA TWENTY,THIRTY,FORTY
,FIFTY,SIXTY,SEVENTY,EIGHTY,
NINETY
200 DATA THOUSAND,MILLION,BI
LLION,TRILLION,QUADRILLION,Q
UINTILLION,SEXTILLION,SEPTIL
LION,OCTILLION,NONILLION
210 DATA DECILLION,UNDECILLI
ON,DUODECILLION,TREDECILLION
,QUATTUORDECILLION,QUINDECIL
LION,SEXTEDECILLION
220 DATA SEPTENDECILLION,OCT
ODECILLION,NOVEMDECILLION,VI
GINTILLION
230 FOR J=1 TO 9
240 READ ONE$(J)
250 NEXT J
260 FOR J=1 TO 10
270 READ TEEN$(J)
280 NEXT J
290 FOR J=1 TO 8
300 READ TEN$(J)
310 NEXT J
320 FOR J=1 TO 21
330 READ HIGH$(J)
340 NEXT J
350 PRINT : : :
360 PRINT #1:"NUMBER"
370 INPUT "NUMBER? ":N$
380 L=LEN(N$)
390 FOR J=1 TO L
400 IF POS("0123456789",SEG$
(N$,J,1),1)=0 THEN 360
410 NEXT J
420 IF (VAL(N$)<1)+(VAL(N$)<
>INT(VAL(N$)))THEN 360
430 IF L<67 THEN 470
440 PRINT "HEY! I CAN ONLY C
OUNT TO A":"VIGINTILLION!":
:
450 PRINT #1:"HAY I CAN ONLY
 COUNT TO A VIGINTILLION"
460 GOTO 360
470 IF VAL(N$)>0 THEN 510
480 PRINT : :"ZERO": :
490 PRINT #1:"ZERO"
500 GOTO 360
580 IF J>1 THEN 610
590 P$=STR$(VAL(NN$(JJ)))
600 GOTO 620
610 P$=P$&" "&NN$(JJ)
620 NEXT J
630 PRINT : : :P$: : :
640 FOR J=1 TO X
650 GOSUB 670
660 GOTO 1150
670 IF VAL(NN$(J))<>0 THEN 7
10
680 A$=""
690 FLAG=1
700 GOTO 1140
710 FLAG=0
720 H=VAL(SEG$(NN$(J),1,1))
730 T=VAL(SEG$(NN$(J),2,2))
740 TT=VAL(SEG$(NN$(J),2,1))
-1
750 VV=VAL(SEG$(NN$(J),3,1))
760 IF T=0 THEN 1000
770 IF T>9 THEN 810
780 A$=ONE$(T)
790 SP$=A$
800 GOTO 1000
810 IF T>19 THEN 880
820 A$=TEEN$(T-9)
830 IF T<>19 THEN 860
840 SP$="NINE TEEN"
850 GOTO 1000
860 SP$=A$
870 GOTO 1000
880 IF VV<>0 THEN 950
890 A$=TEN$(TT)
900 IF TT<>8 THEN 930
910 SP$="NINE TEE"
920 GOTO 1000
930 SP$=A$
940 GOTO 1000
950 A$=TEN$(TT$)&"-"&ONE$(VV)
960 IF TT<>8 THEN 990
970 SP$="NINE TEE"&ONE$(VV)
980 GOTO 1000
990 SP$=A$
1000 IF H=0 THEN 1080
1010 IF T=0 THEN 1050
1020 A$=ONE$(H)&" HUNDRED &
"&A$
1030 SP$=ONE$(H)&" HUNDRED &
"&SP$
1040 GOTO 1140
1050 A$=ONE$(H)&" HUNDRED"
1060 SP$=A$
1070 GOTO 1140
1080 IF (J<X)+(T=0)+(VAL(N$)
<100)THEN 1140
1090 A$=" & "&A$
1100 IF (TT<>8)$(TT<>19)THEN
1130
1110 SP$=" & "&SP$
1120 GOTO 1140
1130 SP$=A$
1140 RETURN
1150 PRINT A$
1160 IF FLAG=1 THEN 1200
1170 PRINT #1:SP$
1180 PRINT HIGH$(X-J)
1190 PRINT #1:HIGH$(X-J)
1200 GOSUB 670
1210 NEXT J
1220 PRINT B$
1230 A$=""
```

```
510 IF L/3=INT(L/3)THEN 540
520 N$="0"%N$
530 GOTO 380
540 X=L/3
550 FOR J=1 TO L STEP 3
560 JJ=JJ+1
570 NN$(JJ)=SEG$(N$,J,3)
```

```
1240 JJ=0
1250 B$=""
1260 P$=""
1270 FOR D=1 TO 500
1280 NEXT D
1290 GOTO 350
```

I hope you noticed that all those zeros were neatly slashed so that you wouldn't mistake them for O's. Here's a little routine that will set up your printer to slash the 0's until you turn it off.

```
100 OPEN #1:"PIO"
110 PRINT #1:CHR$(27);CHR$(4
2);CHR$(0);
120 PRINT #1:CHR$(27);CHR$(4
2);CHR$(1);CHR$(48);CHR$(0);
CHR$(92);CHR$(34);CHR$(81);C
HR$(8);CHR$(69);CHR$(2);CHR$
(65);CHR$(34);CHR$(28)
```

```
130 PRINT #1:CHR$(27);CHR$(3
60;CHR$(1)
140 STOP
```

And, somebody might get mad if I don't include a little music -

```
100 REM - BELL MUSIC program
med by Jim Peterson
110 CALL CLEAR :: CALL SCREE
N(5):: RANDOMIZE
120 FOR CH=96 TO 136 STEP 4
:: FOR L=1 TO 4 :: X$=SEG$("
0018243C425A667E8199A5BDC3DB
E7FF",INT(16*RND+1)*2-1,2)::
 B$=B$&X$ :: C$=X$&C$ :: NEX
T L
130 D$=B$&C$ :: Z$=RPT$(D$,4
)
140 CALL CHAR(CH,Z$):: B$,C$
,Z$=NUL$ :: CALL MAGNIFY(4):
: CALL SPRITE(#CH/4-23,CH,IN
T(15*RND+2),255,255):: NEXT
CH
142 FOR J=1 TO 10 STEP 2 ::
X=9*RND-9*RND :: Y=9*RND-9*R
ND :: CALL MOTION(#J,X,Y,#J+
1,X,Y):: NEXT J
150 FOR J=1 TO 20
155 CALL COLOR(#INT(10*RND+1
),INT(15*RND+2))
160 FOR V=0 TO 16 STEP 4
170 ON J GOSUB 250,270,290,3
10,330,350,370,390,410,430,4
10,390,370,350,330,310,290,2
70,250,270,290,310,330,350
180 NEXT V
190 READ X
200 FOR D=1 TO X*5
210 NEXT D
220 NEXT J
230 RESTORE
```

```
240 GOTO 150
250 CALL SOUND(-999,131,V,52
3,V,131/2,30,-4,V)
260 RETURN
270 CALL SOUND(-999,165,V,16
7,V)
280 RETURN
290 CALL SOUND(-999,196,V,19
9,V)
300 RETURN
310 CALL SOUND(-999,262,V,26
5,V)
320 RETURN
330 CALL SOUND(-999,330,V,33
3,V)
340 RETURN
350 CALL SOUND(-999,392,V,39
4,V)
360 RETURN
370 CALL SOUND(-999,523,V,39
2,V,330,V)
380 RETURN
390 CALL SOUND(-999,659,V,66
6,V)
400 RETURN
410 CALL SOUND(-999,784,V,79
2,V)
420 RETURN
430 CALL SOUND(-999,1047,V,1
057,V)
440 RETURN
450 DATA 16,16,2,16,8,16,4,4
,16,2,16,4,16,8,8,16,2,2,16,
4,2,8,16
```

Just about MEMORY FULL,

so Happy hackin'

Jim Peterson

I GET QUESTIONS???

and hope to give answers!
by G-S Romano

This time I will begin just where I left off. Last time I ended with a question. Well, as it turns out, the answer came from the person who asked it.
So how can we "customize" a cursor in Extended Basic? Here is the little program that does it:

```
10 CALL CLEAR :: CALL INIT
20 CALL LOAD(8196,63,248)
30 CALL LOAD(16376,67,85,82,83,79,82,48,8)
40 CALL LOAD(12288,0,0,0,0,0,0,0,252)
50 CALL LOAD(12296,2,0,3,240,2,1,48,0,2,2,0,8,4,32,32,36,4,91)
60 CALL LINK("CURSOR)
```

Here's how it works. Line 40 loads the shape description of the new cursor into memory. What you will have to do is translate the more familiar HEX chaacter descriptor string from a CALL CHAR and then change every two segments to decimal numbers. For example, as line 40 now stands it describes a flat line similar to the " " character. If we were using the CALL CHAR of Ex. Basic we would state CALL CHAR(XX,"0000000000000FC"). Compare these two lines:

```
CALL CHAR(XX,"00000000000000FC")
                &/&/&/&/&/&/&/&/
                 & & & & & & & &&
CALL LOAD(12288,0,0,0,0,0,0,0,252)
```

Just break the "00000000000000FC" into 8 pairs and change those "numbers" to decimal - "FC" in decimal is 252. When you run the program the cursor will take on a new shape until you leave Ex. Basic OR use the "NEW" command. If you do use the NEW command, you will have to rerun this program. Here is a version that changes the cursor to the Texas Instruments trademark symbol:

```
40 CALL LOAD(12288,48,48,63,255,254,124,24,12)
```

Now on to new questions!

I have heard a lot about the TI MBX system but it doesn't seem to exist. Did TI every really come out with it?

As far as I can ascertain, the MBX system was introduced for sale through retail outlets only in the Northeast part of the U.S. After a "test run", it would then be made available throughout the country. That never happened because of TI's pullout. The unit and the few program modules that use it are available in limited supply from some mail order sources like Microcomputers Corp. in Armonk, New York or Unisource. If you decide to buy it, remember that what there is is all there will ever be, since Milton Bradley has dropped the unit and will not support it or anything for the 99/4A. This is the second time that Milton Bradley has been stung by TI and they are still smarting from what many perceive to be a "very raw deal".

In Pascal I follow all of the instructions to try to add a UNIT to "SYSTEM.LIBRARY". But when I give the file I want to add to as "SYSTEM.LIBRARY" all I keep getting is "FILE DOESN'T EXIST". What am I doing wrong and how can I overcome this problem?

Unfortunately, you, like many others, are the victim of one of those frustrating "undocumented features" that TI has given us. It really was a smart move on the part of TI to do this, but not telling us is what really drives one up the wall. You see in most UCSD Pascals that people buy for other computers, no SYSTEM.LIBRARY comes with the system. With the TI "hybrid" version of UCSD Pascal (some parts of Pascal that normally would be on disk are on chips on the Pascal card to free up more memory) TI supplied us with a SYSTEM.LIBRARY so that we could use SOUND, GRAPHICS, SPEECH, etc. that exist only on our machine. When installing a new UNIT in a SYSTEM.LIBRARY that file must be open while manipulation of data is done, thus leaving the data already in it quite vulnerable to user error and possible loss of data. Consequently, TI installed that "FILE DOESN'T EXIST" to prevent direct access to it. What you will have to do is call a temporary file like USER.LIB. Then transfer all the UNITS in SYSTEM.LIBRARY to USER.LIB, add any other new UNITS to USER.LIB and then when finished, rename USER.LIB to SYSTEM.LIBRARY. Now you see that it really was quite thoughtful of TI to "protect" the SYSTEM.LIBRARY even though their not telling us causes sleepless nights!!!

In reading in various periodicals, I have seen information that states that one can change certain values starting at address -31888 and increase the memory of my machine and also load programs into memory that would otherwise be too big when the disk drives are on. I tried it and after a while my whole system locked up. Can you tell me more about this procedure.

At address -31888 the pointers are stored for how much memory for file buffers is to be allocated. Changing data there can be great fun when doing a SIZE command to see that you have 65536 bytes of stack free, etc. Unfortunately, it is also about as much fun as chewing on dynamite near a campfire. The address in question is usually accessed through the CALL FILES() command. As you know anytime that a CALL FILES() is done it **MUST** be immediately followed by a NEW command to activate and update all system pointers involved. NOT doing so leaves everything just hanging in mid-air, which leads to those "unpredictable results" we encounter. There is only one instance where I advise people to poke values directly at that address instead of using the CALL FILES() command. And that is for a very specific use to achieve the otherwise impossible "CALL FILES(0)!!!!

Here's an explanation of what I mean. When asked, many people have a fuzzy recollection of having seen something about the CALL FILES() command but they don't know exactly what it is. Let's look at it. If you have any peripherals for you system, this concerns you. When you first turn on your system the very first thing your computer does is to check if any peripherals are connected to it and turned on. If it finds that to be true it automatically sets aside memory buffer space so that 3 different files can be OPENed **simultaneously**. For each of these buffers it takes 512 bytes from stack space in memory. Well, it is really rather unusual for a good program to have more than 2 files open at once. An example might be to have one file to access data on a disk and another file open to send those data to a printer. So, who gave the computer the right to steal MY precious memory from me? Well, I did by not changing the default setting of three files. We can change things to get back more memory if we want by typing in the COMMAND (no line numbers - this is not a program):

```
CALL FILES(2)          (OR ANY NUMBER FROM 1 TO 9)
press ENTER
NEW and press ENTER again.
```

Note that this is a TWO step command. By doing so you tell the computer to reassign the memory to be used for the file buffers. For each number less than three you get back 512 bytes of memory in the stack space (you can verify this with a SIZE command). It is worthwhile to develop the habit of doing a CALL FILES(2) after every powerup to have maximum memory available. By doing so you will find that you are now able to load programs that before only gave you an error message. If you know that you are loading a very large program that will do no file access, you can type in a CALL FILES(1). Why not do a CALL FILES(0) you may ask? You can't, it's not allowed!

OK, so what's this CALL FILES(0) business? If you have Memory Expansion, Extended Basic and/or either Mini-memory or Editor/Assembler you can get even more memory for some otherwise impossible chores. Let's say that you have a program written in Basic and it was saved on cassette long before you got your disk drive. You now want to transfer the program to disk but everytime you try to load it from cassette you only get error messages. It is impossible, because the program takes up more console memory than is available when the system with peripherals is turned on. Even a CALL FILES(1) doesn't help.

The way to transfer this to disk then runs something like this. Have all periphrals turned off at powerup; load the program into memory from tape. Now delete one half of the program, line by line, and save the remaining half to tape. Reload the original, delete the *other* half and save it again. Now turn the system off, turn the peripherals back on and boot up to Extended Basic. Now load the first half of the program from tape and save it to disk in MERGE format. Load the other half from tape and then MERGE the other portion off disk. Finally save the whole reconstituted program to disk. Whew!!! This is a lengthy procedure to be sure but worse, if you have ever tired to edit a large program through console Basic, you know that it takes about 20 seconds to get the cursor back after EVERY line change. A whopping speed of 3 lines per minute - maximum! This is an excrutiatingly frustrating experience at least!

Now you can eliminate the waiting by using the increased speed of Extended Basic. Powerup with all peripherals on. Choose Extended Basic and enter CALL INIT. Then enter the command CALL LOAD(-31888,63,255), press ENTER, enter NEW and press ENTER again. you have just made all of the peripherals invisible to the console and gotten back not only those 512 byte chunks but also the 900 or so extra bytes confiscated for overhead. AND, in addition, you still have the Memory Expansion unit available to you! You can now follow the above procedure for tape to disk transfer with NO WAITING for that cursor. WARNING! When you finish the tape save part, you MUST press FCTN + (Quit) to cancel out that CALL LOAD you did. In this instance you cannot do another CALL LOAD to change the settings because the system will become unstable. ANOTHER WARNING! After the first CALL LOAD do not attempt to enter an OLD or SAVE DSK1.???? or the system will crash. Too, remember to check the program first to see if it can be run through Extented Basic because the final product of your efforts saved on disk will be in Memory Expansion format and you will not be able to load it through console Basic. The usual conflicts are created by the definition of graphics characters with a number greater than 143 or CALL COLOR statements that are defining a character set greater than set 14. Other than that you should have little difficulty in running most Basic programs through Extended Basic.

A closing comment! Although I hear from a great number of people who want help with their 99/4A's from all over, I have yet to be contacted by anyone who has read this column in this newsletter. I DO ask specifically where people learned of me. So come on out there, somebody from this readership must have some questions or want help. Don't be shy! Drop me a line or call Mon-Sat 9AM-3PM Pacific Time (415) 753-5581. I'm here to be of service.

SPECIAL OFFER - DOOM OF MONDULAR

"Doom Of Mondular" is a dungeon fantasy game for the TI-99/4A. You travel through the mystical land of Agnar in search of treasure and power. Beware of traps, hidden passages, and monsters. A 3-D graphic game ideal for fantasy adventurers of all ages.

The previous paragraph was taken from an advertising flyer for the Doom Of Mondular. A review of this excellent game appeared last month in this newsletter. The creator of this product, Randy Romano, has decided to make a special offer to owners of the 99/4A. The normal purchase price for this software package is $24.95. The price has temporarily been reduced to $19.95 plus a shipping and handling charge of $2.50. Equipment requirements for this game include a disk drive, memory expansion, and the Extended Basic module. A three month product warranty is given on this disk based software.

Randy is in the process of developing a new game at the present time named "Tower Of Loom". More details on this game will be made available as we receive the information.

Visa and MasterCard are accepted on orders by Symbiotech. Doom Of Mondular may be obtained by contacting the firm at the location below:

SYMBIOTECH INC
P.O. BOX 320
ROSCOE, ILL. 61073
(815) 623-6751

SPARE PARTS AVAILABLE FOR 99/4A

Spare keyboards and power supplies are available for the TI-99/4A. The cost for each unit is $15.00 which includes shipping and handling. The units are available from Model Masters.

Last month, we reviewed this form's Joyprint Serial Printer Interface and Minied Word Processor Software. The Minied unit's list price is $19.95 plus $2.50 for shipping and handling. A special offer on the Minied is being made to all Users Groups and subscribers of this newsletter. Each Users Group and subscriber may purchase one copy of this software for $17.95 which includes shipping and handling. The software will not be protected in any manner. The author has chosen to forward the software and trust that each person obtaining a copy will prevent its being copied. The Minied will be available in a greatly expanded cartridge version about November 1, 1984. Any person purchasing the Minied software on tape now will receive a discount on the purchase of the cartridge version. All you must do to obtain this discount is mail in your original receipt and instruction manual.

Remember, each Users Group and subscriber may purchase one copy of the Minied at this price. Forward all product purchase requests to the address listed below:

MODEL MASTERS
22411 MTN. LAUREL WAY
DIAMOND BAR, CA 91765

MORNINGSTAR SOFTWARE CPM CARD

I received a pleasant telephone call from Scott Swenson of Morningstar Software. Scott inquired if I was still interested in evaluating the CPM card for our newsletter readers. The card is being forwarded and work with this second computer system mounted within the Expansion System will be undertaken immediately. I will have some initial reactions available for the October issue of the newsletter.

Software for the CPM card, available from Morningstar, will be forwarded for use with the unit. I plan to contact some additional sources for other software compatible with the system. If any of you reading this article possess the CPM card, please write me at the letterhead address about your experiences with the unit.

Scott stated his firm was in the process of investigating the possibility of making their card compatible with the CorComp double-sided, double-density disk controller card. Information updates on this subject will be passed along as we are notified.

We wish to thank Scott Swenson and his firm Morningstar Software for making their product available for testing and evaluation. Many of the third party firms have limited resources and thier efforts to LOAN equipment for evaluation is truly appreciated.

Don Veith, Editor

&BIT MAP OBJECT CODE By John Phillips of Video Magic

The big demand for computer software is **EASE OF USE**. Computer manufacturers are spending big dollars to develop hardware and software that can be operated strictly with a mouse or joystick. You've probably seen the advertisements for Macintosh. They show all these wonderful things that you can do with only a mouse input device. Macpaint and Macwrite (along with Mousepaint) use pull-down windows to select different options. These pull-down menus do make things very easy!

Your TI-99/4A can replicate this same function using **BIT-MAP** mode. If you're not familiar with bit-map mode, it is an enhanced version of graphics available from the TMS9918A VDP chip. Instead of having 256 programmable characters, you have 768. Each character represents a sequential order in the screen image table.

Instead of manipulating characters as you do in pattern mode, you now must manipulate the larger Pattern Descriptor Table. This table now is >1800 bytes long. The color table, as well, is >1800 bytes long. To change patterns, you must calculate the address in the PDT (and CT) and write the information directly to it. For some programmers, this can really "blow their mind", but once you get the hang of it, it becomes very simple to manipulate the bit-map areas.

The programming example listed below is a simulation of a pull-down window. A white-glove will be displayed on a full bit-map screen. You move the glove using the arrow keys. When you press the space bar, a 6-high by 10-wide window is masked. In this example, I am merely changing the color of the background for all the bits in the window. Remember, this is not pattern mode! I have to write 1 byte at a time to the designated color table addresses based upon the position of the sprite (glove).

You ambitious programmers might want to modify this program to pull actual pattern data from a table and display it on the screen as well. If anyone wants to take the dare, I'd love to see your results! I have included the EDITOR/ASSEMBLER source and object files for this program on your diskette. Use the LOAD AND RUN option. Enjoy!

John Phillips

```
       DEF BITMAP
       REF VMBW,VSBW,VMBR,VSBR,VWTR,KSCAN
*
SVVDP1 EQU >83D4 SAVE VDP R1
SIT    EQU >1800 SCREEN IMAGE TABLE
SAL    EQU >1B00 SPRITE ATTRIBUTE LIST
CT     EQU >2000 COLOR TABLE
SVT    EQU >1F00 SPRITE VELOCITY TABLE
SDL    EQU >1C00 SPRITE DESCRIPTOR LIST
PDT    EQU >0000 PATTERN DESCRIPTOR TABLE
HIVDP  EQU >3800 FREE VDP AT TOP
*
MYWS   EQU >8300 MY WORKSPACE
TEMP1  EQU >8320 TEMPORARY SAVE
TEMP2  EQU >8322 TEMPORARY SAVE
KEYBRD EQU >8374 KEYBOARD TO SCAN
KEY    EQU >8375 KEY RETURNED
****************************************
* D A T A S T A T E M E N T S *
****************************************
VDPREG DATA >02E2,>06FF,>0336,>0301 VDP REGS BIT MAP MODE
ZEROES DATA 0,0,0,0 8 BYTES OF ZEROES
****************************************
* S P R I T E D E F I N I T I O N S *
****************************************
HAND   DATA >0000,>6070,>381C,>0F0F >80->83
       DATA >0703,>397F,>0F07,>0100
       DATA >0000,>0000,>0000,>B4FE
       DATA >DE6E,>B6DB,>EFFE,>FC38
SALINI BYTE 100,100,>80,>F,>D0 SAL INITS
```

```
*****************************************
* START OF 9900 ASSEMBLY CODE *
*****************************************
BITMAP LWPI MYWS LOAD MY WORKSPACE
       CLR R0 VDP REG VALUE
       LI R1,VDPREG INDEX POINTER
       LI R2,8 8 REGISTERS TO WRITE TO
VDPL   MOVB *R1+,@MYWS+1 MOVE TO ROLB
       BLWP @VWTR WRITE TO VDP
       AI R0,>0100 NEXT VDP REGISTER
       DEC R2 DONE WITH 8 REGISTERS?
       JNE VDPL NOT YET
       MOVB @VDPREG+1,@SVVDP1 KEEP DOUBLE SIZED SPRITES
* AT THIS POINT, VDP IS SET TO BIT-MAP MODE *
*****************************************
* FORMAT THE SCREEN >00->FF 3 TIMES *
*****************************************
       LI R0,SIT ADDRESS TO WRITE TO
       CLR R1 START WITH CHAR 0
       LI R2,>300 LENGTH OF SIT
THIRDS BLWP @VSBW WRITE A BYTE
       INC R0 NEXT SIT SPOT
       AI R1,>0100 WILL WRAP AFTER >FF00
       DEC R2 DONE WITH ALL?
       JNE THIRDS NOT YET
* AT THIS POINT, THE SCREEN IS FORMATTED PROPERLY *
       BL @CLRVDP CLEAR THE PATTERN AREA
       DATA PDT,>1800,0
       BL @CLRVDP NOW, CLEAR THE COLOR AREA
       DATA CT,>1800,0
* SCREEN SHOULD BE BLANK, NOW (AND BLACK)
*****************************************
* CREATE SPRITE ON SCREEN *
*****************************************
       LI R0,SDL SPRITE PATTERNS
       LI R1,HAND HAND PATTERN
       LI R2,32 32 BYTES FOR A DOUBLE SIZE SPRITE
       BLWP @VMBW HAVE 1 SPRITE PATTERN LOADED
       LI R0,SAL NOW SAL DATA
       LI R1,SALINI THERE'S THE DATA
       LI R2,5 INCLUDE THE >D0 TO NULLIFY ALL ELSE
       BLWP @VMBW HAVE A GLOVE ON THE SCREEN!
*****************************************
* M A I N P L A Y I N G L O O P *
*****************************************
LOOP   LIMI 2 ENABLE
       LIMI 0 THEN DISABLE
       CLR @KEYBRD SCAN ENTIRE KEYBOARD
       BLWP @KSCAN SCAN FOR KEY
       MOVB @KEY,R3 COPY KEY PRESS VALUE
       SRL R3,8 MAKE IT A WORD
       CI R3,>00FF KEYS IDLE?
       JEQ LOOP YES, SO KEEP TRYING
       CI R3,32 SPACEBAR?
       JEQ DRAW YES, SO DRAW BOX
       CI R3,69 UP? E
       JEQ UP YES
       CI R3,88 DOWN? X
       JEQ DOWN YES
       CI R3,83 LEFT? S
       JEQ LEFT YES
       CI R3,68 RIGHT?
       JEQ RIGHT YES
       JMP LOOP KEEP GOING!
UP     LI R5,>FF00 -1 FOR Y
       JMP MOVE
DOWN   LI R5,>0100 +1 FOR Y
       JMP MOVE
LEFT   LI R5,>FFFF -1 FOR X
       JMP MOVE
RIGHT  LI R5,>0001 +1 FOR X
MOVE   LI R0,SAL READ Y,X OF SPRITE
       LI R1,MYWS+8 INTO R4
       LI R2,2 2 BYTES
       BLWP @VMBR READ POSITION OF GLOVE
       A R5,R4 ADD DIRECTION OFFSET
       BLWP @VMBW AND MOVE GLOVE
       LI R0,>0800 DELAY
```

```
DLY     DEC R0 DECREMENT COUNTER
        JNE DLY NOT DONE
        B @LOOP NOW SCAN AGAIN
*
DRAW    LI R0,SAL SPRITE Y,X
        LI R1,MYWS+10 READ Y,X INTO R7
        LI R2,2 2 BYTES
        BLWP @VMBR READ IT
        MOV R5,R6 COPY Y,X TO R8
        AI R5,>0100 SPRITE SCREEN STARTS AT >FF,
*         SO ADD 1 FOR CHARACTER SCREEN
        SRL R5,3 DIVIDE Y POSITION BY 8 FOR
*         CHARACTER ROW POSITION (0-23)
        ANDI R5,>FF00 KEEP ONLY ROW IN HIGH BYTE
        ANDI R6,>00FF KEEP ONLY COL IN R6
        SRL R6,3 DIVIDE COL BY 8 FOR CHAR COLUMN
        SWPB R6 NOW PREPARE FOR SUBROUTINE PASS
        MOV R5,@TEMP1 SAVE FOR ERASE
        MOV R6,@TEMP2 SAVE FOR ERASE
*
        BL @FILLER DRAW BOX
        BYTE 6,10 6 HIGH, 10 WIDE
        DATA >0606 RED COLOR
*
DBNCE   BLWP @KSCAN SCAN FOR KEY AGAIN
        MOVB @KEY,R3 COPY KEY PRESS VALUE
        SRL R3,8 MAKE A WORD
        CI R3,>00FF KEY DOWN?
        JNE DBNCE YES, SO WAIT HERE UNTIL HE RELEASES
*
        MOV @TEMP1,R5 RESTORE ROW
        MOV @TEMP2,R6 RESTORE COLUMN
        BL @FILLER ERASE BOX
        BYTE 6,10 SAME DIMENSIONS
        DATA >0000 ONLY TRANSPARENT
        B @LOOP AND LOOP AGAIN
*****************************************
* SUBROUTINE TO FILL ANY GIVEN WINDOW*
* WITH ZERO.  ROW, COL, HEIGTH, WIDTH,*
* COLOR BYTE MUST BE GIVEN.  RETURN IS*
* IN R10.  *
*       *
*       BL @FILL *
*       BYTE 2,4,6,18 *
*       DATA >F0F0 *
*         *
*****************************************
FILL    CLR R5 PREPARE
        MOVB *R11+,R5 ROW * 256
        MOVB *R11+,R6 COL
FILLER  SRL R6,8 ADJUST
        SLA R6,3 MULT BY 8
        A R6,R5 HAVE OFFSET
        AI R5,PDT OFFSET INTO PDT
        MOVB *R11+,R6 HEIGHTH IN ROWS
        SRL R6,8 ADJUST
        MOVB *R11+,R7 WIDTH IN COLUMNS
        SRL R7,8 ADJUST
        SLA R7,3 MULT BY 8
        MOV *R11+,R8 GET COLOR BYTE
        MOV R11,R10 SAVE RETURN
*
FILLL   MOV R5,R0 GET VDP ADDRESS TO WRITE TO
        MOV R7,R3 TELL HOW MANY BYTES TO WRITE
        BL @CLRV CLEAR A SECTION OF THE SCREEN
        DATA 0
*
        MOV R5,R0 START OVER
        AI R0,>2000 POINT TO COLOR TABLE
        MOV R7,R3 TELL HOW MANY BYTES TO WRITE
        MOV R8,R4 TELL THE COLOR BYTES
        BL @CLR PUT BLACK AS BACKGROUND
*
        AI R5,256 NEXT ROW
        DEC R6 DONE?
        JNE FILLL NOT YET
*
        B *R10 RETURN TO CALLER
```

```
*****************************************
* SUBROUTINE TO FILL VDP WITH DATA *
* RETURN IS R12 *
* BL @CLRVDP *
* DATA PDT,>1800,>0000 *
* (destination,length,value to fill *
*****************************************
CLRVDP MOV *R11+,R0 DESTINATION
CLRLV  MOV *R11+,R3 LENGTH
CLRV   MOV *R11+,R4 VALUE(S)
CLR    MOV R11,R12 SAVE RETURN
*
       SRL R3,1 DIVIDE BY 2
       LI R2,2 2 BYTES OF DATA
       LI R1,MYWS+8 FROM R4
CLRVL  BLWP @VMBW WRITE 2 BYTES
       INCT R0 NEXT TWO LOCATIONS
       DEC R3 DONE?
       JNE CLRVL NOT YET
       B *R12 RETURN TO CALLER
*
       END BITMAP LOAD AND GO
```

>>>>>> B U L L E T I N <<<<<<

A one-time offer is being made to all user-groups around the United States. **John Phillips** and **VIDEO MAGIC** are making their software available on disk to all user groups currently listed by the 99'ers U.S.A. The most amazing item about this sale is that each group will have to opportunity to buy just **one** copy of each of the 10 diskettes available in this offer. However, the diskette is **UNPROTECTED!** That's right . . . completely unprotected! You will be allowed to copy this diskette and distribute it freely amongst your group. Let's explain the details.

Listed below are the program packages available. Each user group may purchase one copy of each of the 10 programs for $50.00. You do not have to buy all 10! You may select 1, 2, 4, or 7 for that matter! The thing to remember is that each program will cost $50.00. If you select 3 programs, your total cost will be $150.00. You have total control of the selection!

However, it must be stated that no orders will be shipped until at least **50** orders are received for a particular software item. If fewer than 50 orders are received, it will be determined by VIDEO MAGIC whether or not the partial order will be fulfilled. If the order is not to be filled, your money will be refunded instantly. Here is the process.

1. Fill out the order form and check off the software desired.

2. Calculate the total (# of items x $50.00).

3. Add $2.00 for shipping and handling.

4. Enter the total amount of the order.

5. Mail the order, with your payment to the address listed below. Personal checks will be accepted (make payable to John Phillips). You check **will not** be cashed until your order is ready to be mailed.

6. Enclose a self-addressed, stamped envelope with your order. this will be used to mail your check back in the case of insufficent orders.

That's all there is to it! I will begin processing orders on November 1, so you should have a bundle of Christmas presents. If you have any questions, leave a message on my answering machine at 214-727-8668. Thank you, and have a good time with your new software!

. . . John

DID YOU KNOW ????

MUNCHMAN has, hidden away in its program, a Test Mode. This gives you the ability to choose any round (1-20, 21-40, 41-60), and screen (1-20), and up to nine MunchMen. To enter the Test mode, just press (within three (3) seconds) when the MUNCHMAN title screen appears:

(1) [SHIFT] [8] (2) [SHIFT] [3] (3) [SHIFT] [8]

After this is done, RND(0-2) will appear (what round?). When you have chosen the round you desire, SCN(0-19) will appear (what screen?). Then MM(1-9) or (how many Munch Men). Remember that zero (0) is one (1) in the first responses. Article extracted for the New Hampshire 99'ers newsletter for 9/84.

***** O R D E R F O R M *****

Your name: _____

Address : _____

CT,ST,Zip: _____

Phone : WORK _____ HOME _____

Please check appropriate software desired:

ENTERTAINMENT

____ D-STATION I + D-STATION II (two games for the price of one)

____ BOXER + TIle BREAKER (two games for the price of one)

____ BEYOND SPACE

____ STAR TRAP in 3-D

____ TI-RUNNER (ala LODE RUNNER)

____ FACE CHASE

EDUCATIONAL GAMES

____ THE GREAT WORD RACE

____ STAR GAZER I (astronomy)

____ STAR GAZER II (astronomy)

____ STAR GAZER III (astronomy)

NUMBER OF PROGRAMS CHOSEN _____

SUBTOTAL (# TIMES $50.00) _____

ADD $2.00 SHIP + HANDL. _____

T O T A L O R D E R : $_____

Checklist:

1. Have you marked you intended software?
2. Have you calculated your order correctly?
3. Have you made your check out to John Phillips?
4. Have you included a self-addresses, stamped envelope?
5. Make sure you send it to the address listed below:

VIDEO MAGIC
ATTN: JOHN PHILLIPS
1100 TIMBERBEND TRAIL
ALLEN, TX 75002

SUBSCRIPTION FORM
99'ERS U.G.A.

NAME: Bill Pechnik DATE: _____

ADDRESS: 1467 CARMI DRIVE

CITY: PENTICTON STATE BC ZIP V2A 4R9
CANADA

MAIL CK./M.O. FOR $12.00 TO:

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

ATTN; DONALD VEITH

3/84

THE NATIONAL 99'ER
3535 SOUTH H STREET #93
BAKERSFIELD, CA 93304