

VAST NEWS

for the VAST Computer User Group

Volume 8 Number 12

December 12, 1992



The True Meaning of Christmas

Happy Holidays from VAST

SECRETARY'S SLATE

MINUTES

VAST Computer User Group
November 14, 1992

President Jim Ely officially called the November 14, 1992 meeting of the Valley of the Sun TI Users Group to order at 10:03 a.m. There were 18 people in attendance. All officers were present for the meeting.

The first call of business was the reading of the minutes of the last meeting. The minutes were accepted as published in the newsletter. The treasury report was read and accepted next. Jim then asked for the committee reports.

LIBRARY: Hazel reported that the entire library was present at the meeting, including all of the audio and video tapes. She also stated that the contents of the library have finally been cataloged. The lists are available at the meetings.

PROGRAM: The program for the December meeting will consist of Christmas programs for the TI. It was suggested that the next president should appoint a new program committee. It was mentioned that the new committee chairperson should be a TI user so that the scope of the programs would be realized. No program has been listed for the January meeting.

PUBLICITY: Ray Frantz had nothing new to report at this time.

RAFFLE: Ron Vanover was in attendance and so there was a raffle at the November meeting. Jim said that he had forgotten to print the results of the October raffle in the

newsletter, but would try to include them in the next edition.

BBS: Jim had nothing new to report. He stated that the BBS is still under used, but he noted that there have been a number of out of state callers lately. He was not sure why this was occurring, but he guessed that it might have something to do with the group being mentioned in MICROpendium magazine.

Speaking of MICROpendium, a discussion was started about the possibility of taking out some ads for the Page Pro artwork that Virginia Davis has donated to the club. It was also suggested that maybe the club should submit the artwork for a review by MICROpendium.

NEWSLETTER: Jim stated that the newsletter would not be mailed next month in keeping with our usual manner of distribution. The last edition was only mailed to get the ballots out to voters.

Jim also mentioned that Chris Taylor is still dropping articles off for publication even though he has not been attending the monthly meetings. Having mentioned this fact, Jim wondered why Chris has not been coming to the meetings. After a lengthy discussion, the group was unanimous in encouraging Chris to come to the meetings and Ron Vanover promised to contact Chris.

Next, Jim raised the topic of **OLD BUSINESS**. Bud Barnes stated that 4 members attended the tour of the ASM Litho facility in Tempe. He described the equipment that was

— See *MINUTES* on page 3

— MINUTES, from page 2

demonstrated and how it is used. Bud also stated that he would try to work on another field trip in the future.

Also under the topic of **OLD BUSINESS**, Bud had the results of the election for 1993 VAST officers. The new president will be Ray Frantz. Ralph Rees was re-elected as vice-president and Tom Pfeffer remains the secretary/treasurer. Congratulations were expressed to all of the candidates and thank yous were extended to all those that took the time to vote. The new officers will take office at the first meeting in 1993.

There being no more old business, the topic was changed to **NEW BUSINESS**. Hazel asked if the club planned to donate anything to the groups that are staging the Fest-West '93. During the lengthy discussion that followed, it was mentioned that VAST has not been contacted by anyone regarding the upcoming fair. Why? It was surmised that maybe the Southwest '99ers were getting all of the correspondence even though they did nothing as far as running the Fest-West '92. Ralph Rees volunteered to contact the people who are going to run the next fest and volunteer our help. It was decided to delay any decision until after Ralph got a response.

This being the last topic to be discussed, the meeting was recessed at 10:46 a.m. The raffle and a demo of *CHECKSUM* by Ralph Rees followed.

Respectfully Submitted,
Tom Pfeffer
VAST Secretary/Treasurer

Raffle Corner

Ron Vanover is now the chairman of the Raffle Committee and runs the Door prize raffle at the meetings. *YOU MUST BE PRESENT TO WIN!*

Here are the results of the raffles held at the October and November User Group meetings...

In October, the number range of tickets sold was number 343 to number 384 (41 tickets). There were five tickets drawn: 378 and 381 were not claimed; Ed Morse won a disk storage case with ticket number 348; Ray Frantz also chose a disk storage case with ticket number 356; Ron Vanover had the last winning number, 343, and chose a disk based version of a Screen Dump Program.

In November, the number range of tickets sold was number 385 to number 447 (62 tickets). Three tickets were drawn: 416 was held by Hazel Knight and she chose a 5 1/4" disk storage case; Ed "Corky" Geissel chose a Gorilla Banana Dot Matrix Printer with winning number 410; Tom Pfeffer was the third winner with number 423 and chose 3 software cartridges: ESPIAL (Starfighter), Teach Yourself Extended BASIC, and Adventure.

Here are just some of the prizes that are available for future drawings...

Blank Floppy disks, a box of printer paper, a disk caddy which holds 12 floppies, and numerous software titles including System III Checkbook program, and TI Software Cartirdges including ALPINER, PHYSICAL FITNESS and TI INVADERS.

Chances are still 3 for \$1 and are available only at the meetings, after the General meeting and before the program/demo. rv ♦



From the Editor's Desk HAPPY HOLIDAYS

GOOD NEWS AND BAD NEWS

This issue turned out to be a lot bigger than I had anticipated due to the length of the article Chris Taylor submitted for *Chris' Corner* (page 6) and the *Build Your Own... Disk Of The Year* program, this month featuring the CHECKSUM programs demoed by Ralph Rees at last month's meeting. Those programs start on page 9. *4/A Foray* is on page 5 with Ralph discussing several ideas including Virginia Davis' ART that is in the VAST Library EXCLUSIVELY! And on page 3 are the results of the last 2 month's Raffles. That's all the GOOD news.

Now for the BAD... As most of you know, I have been doing VAST News for almost 8 years. Due to some changes taking place in my personal life I must now turn the reigns over to someone else in the group. This December issue is the last I will edit. I'm sure there must be someone else in the group that can carry on with the newsletter and there have got to be some other ideas as to how the newsletter should be done. Your new board of directors will have to make that decision. I have enjoyed doing the newsletter these many years, I just find I no longer will have the time to spend doing it.

JANUARY 1993							FEBRUARY 1993						
SUN	MON	TUE	WED	THR	FRI	SAT	SUN	MON	TUE	WED	THR	FRI	SAT
3	4	5	6	7	8	9	7	8	9	10	11	12	13
10	11	12	13	14	15	16	14	15	16	17	18	19	20
17	18	19	20	21	22	23	21	22	23	24	25	26	27
24	25	26	27	28	29	30	28						
31													

UPCOMING MEETINGS

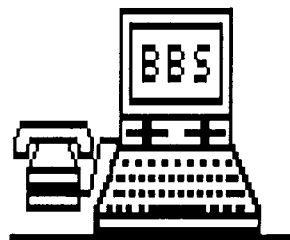
Above is the 2 month calendar with our meeting dates indicated. That's it for this issue. I'll see you all at the December meeting.



Just a reminder...

The **VAST BBS** can be reached 24 hours a day, 7 days a week at **(602) 233-0790**.

This is a local call for most area residents. Give it a call for the latest information about the VAST Computer User Group! And now it's **FASTER** than ever!



4/A Foray

by Ralph Rees

It's beginning to look a lot like Christmas, everywhere I go!

Boy, isn't that the truth! I have been wondering the last couple of years why merchants don't just have "Christmas Year-Round" sales and be done with it!

Don't get me wrong, I have the Spirit, but that's just it — "The Spirit of the Baby Jesus" NOT commercialism!

Oh Well, Everyone have an Excellent Christmas and a Marvelous New Year!

* Concerning Chris Department

As I had stated at the November meeting, We as a group should do all we can to "re-vitalize" Chris Taylors excitement in the TI. **WE BLEW IT!** Here we have in our OWN group a programmer of such fantastic talent that I think he could make the TI do just about anything he wanted, and showed most of us this **FACT!** I believe that we need to BEG Chris to forgive us and give us another chance to do a "Group-Project" or what ever with Chris leading the venture. Remember folks, Superior TI programmers are a scarce commodity and we need to keep them. Let's do our best to keep Chris!

* Virginia Davis DepARTment

Here is another area we need to work on, Virginia Davis' HUGE amount of DONATED artwork. This little lady has given us a literal TON of PagePro Pics and stuff for our group to sell as we see fit. If you made our FestWest then you know that Virginias products sold **EXTREMELY** well! We need to promote this to other groups, as I am sure there is a large percentage of TI'ers that would love to have some of this artwork. In fact, Have any fellow VAST members purchased any of

these disks? Look for them in our library that Wallace & Hazel take care of, You'll want at least a dozen diskfuls!

* Another VAST Fest? Department

A few folks have mentioned that they would like to have VAST do another TI event. This idea has some merit BUT I would have to enter these ideas into the mix.

1) Make it a VAST event, NOT Fest West. I have read a few other UG newsletters and the emphasis was on FESTWEST not that we as a group held this thing and did it well. It would have been nice to read some recognition of our efforts.

2) How about holding it Memorial Day in May? That would make it another 3 day weekend for most folks and the weather should be bearable to most cold country folk.

3) Try and hold it at Days Inn Camelback again. The Hotel folks seemed greatly suprised that we were able to get that many people and I think they would welcome the business at that time of year.

There are other variables I have in mind, but we'll discuss them if any interest is shown at all.

That's about all my cluttered mind has this month. Remember to write an article, use the library and BBS, feed the Kitty at meetings and to buy me a nice expensive TI related Christmas Gift!

FCTN =

—Ralph...





Chris' Corner

Copyright © October 1992
by Chris Taylor
Reprinted by permission
of Chris Taylor

Why the TI, part 4

For the past few months I have shared with you reasons why I still use the 99/4A. I feel that just because a computer is not state of the art, unless you need state of the art features, use what you have. Unfortunately, in the TI community very few powerful applications were ever produced that exploited its features. I realize that the decline of its popularity (coupled with the withdrawal of Texas Instruments support in 1983) is responsible for the lack of power programs. I, also, realize that if I devoted all my efforts to developing the most powerful programs possible for the 99/4A, those efforts would pale in comparison to having programs developed by a team of programmers. My objective, therefore, is to provide simple tools and concepts to aid you in expanding the use of your computer by becoming a member of the TI power users team.

Most of you know that I selected forth as my language of choice. The particular version of forth I use is Wycove forth for which I now have product rights. I chose forth because it has no competition as far as programming productivity is concerned, especially on the 99/4A. It offers access to the all the computer's resources: the video display processor, including the bit-map mode, the sound processor, the speech processor, ROM and GROM routines, and simple access to the peripherals. All of this at speeds which approached or equal assembly language speed. Furthermore, accessing the machine's hidden capabilities can be done as easily in forth as in BASIC.

When I initially purchased forth, I ordered two

versions of forth: Wycove and TI. While forth allows the programmer the ability to re configure the language to assume any form, the amount of effort required to make TI forth as powerful as Wycove forth was not worth the effort nor could I have done so without extensive knowledge of assembly language programming and the inner workings of forth. Had I had the knowledge to make such changes at that time I would have, because if I had marketed any programs using Wycove forth, I would have had to pay a royalty fee. Now, that is not the case provided certain guidelines are met.

So, why am I trying to sell you on forth. Simple. I believe that over the next few months I will be able to provide you with enough easily digestible information, which will allow you to take control of your computer. In preparation for this information, the following TI BASIC program is offered for you to study. If you can master this program, you are on the way to power computing. I hope that you realize that this is a very simple database program.

```
100 CALL CLEAR
```

```
110 INPUT " Enter Disk Drive  
Number: " :DRIVE
```

```
120 PRINT
```

```
130 INPUT " Enter data base  
Filename: " :FILENAME$
```

```
140 INPUT " Enter number of  
names: " :MAXNAMES
```

```
150 PRINT
```

```
160 PRINT " TO QUIT type 'q  
uit' or 'QUIT'"
```

```
170 PRINT " for a data name  
entry."
```

— See CHRIS' CORNER on page 7

— CHRIS' CORNER, from page 6

```

180 REM

190 REM

200 OPEN #1:"DSK"&STR$(DRIVE
)&". "&FILENAME$

210 FOR ENTRY = 1 TO MAXNAME
S

220 INPUT "Enter name: ":DB$

230 IF DB$="QUIT" THEN 270

240 IF DB$="quit" THEN 270

250 PRINT #1:DB$

260 NEXT ENTRY

270 CLOSE #1

280 END

```

This program uses quite a few concepts, the most important of which is the concept of variables. Each time you are asked to "INPUT" something, you are putting that something in to a variable. A variable is a place (computer memory location) where values vary. Variables are alphabet and number combinations (alphanumeric) used to show that place. The reason we use variables is to enable us to change values within a program. The changes (INPUTs) may be made by a person or machine.

There are two types of variables "string" and "number" variables. The difference between the two is that string variables hold alphanumeric entries which cannot be used arithmetically and number variables only hold numbers which can. Since you can store 12345 in either a string or number variable, you need a way to tell one from the other. If the variable is a string then the value will be enclosed in quotes and the variable will end

in the dollar sign ("\$"). On the other hand if the variable is a number then its value has no quotes nor does it end in any special symbol.

```

e.g. string variable: 100 ADDR$="12345"
number variable: 110 UNITS=12345

```

The second concept is ease of use or the man machine interface. In the absence of the tools necessary to make a fancier display, we use what is available. For example the first thing we do is to clear the screen. We would, also, make any changes to the screen color or load any special character sets or symbols at this point. Then we must decide how to make the program invisible to the user as well as providing simple instructions. This is accomplished by deciding just what we want to say and using combinations of the "INPUT" and "PRINT" commands to format how information appears on the screen. The look and feel of the program (did I say Mac?) will determine if the user wants to spend time with the program. As an important side note, the use of REM (remark) statements helps to make our program easy for us to debug (correct) because it is easy for us to read.

The main program begins in line 200 and illustrates three additional concepts. First, it shows how to open a device. Second, it shows how to combine strings to create the proper syntax – the exact language structure that the computer understands – to open that device. And finally, it shows the concept of defaults, which are values that the command uses, invisibly (they ARE there), when you don't tell it otherwise. (See your TI BASIC reference manual for the full "OPEN" command.)

The sixth concept is looping which is in line 210. This is the core of all computing. By placing an instruction outside a selected group of instructions we can repeat that instruction(s) as many times as we desire. In this case we will repeat lines 220, 230, 240, and 250. As long as the "MAX-NAMES" variable (we entered in line 140 to show the number of maximum number of times that we wish to enter names) is less than what we input,

— See CHRIS' CORNER on page 8

— CHRIS' CORNER, from page 7

every time the program reaches line 260 it will kick back to line 220 and repeat the instructions in those lines: 220, 230, 240, 250. When we reach the value in "MAXNAMES" the program will continue with line 270. Note that the "1" used in line 210, also, could have been a numeric variable. Also, note that looping uses a default. (See your BASIC reference manual.)

The seventh concept is conditional branching. This is the concept which allows the computer to make decisions. You tell it in an algebraic form which conditions or things you are comparing and then instruct the computer to branch (GOTO) another portion of the program if certain conditions are met.

The final concept of this article is the "PRINT" command. The "PRINT" command is extremely powerful and all it does is just print. But WHERE it prints is the question. The answer is anywhere you tell it: to disk drives, ramdisks, rs232, pio, ram, printers, etc. Using the syntax in line 250, the information you input in line 220 (your name data) is stored in the variable DB\$. This is PRINTed to the device OPENed in line 200 (which in this case is a disk drive). Of course, you OPENed a printer in line 200 then DB\$ would be printed to a printer which if set up properly would print the output. In the case of a disk drive, when it is printed to, the information is saved. It should be noted that the reverse of PRINTing to a device is INPUTing from that device. You should study your reference manual because there are many defaults associated with: OPEN, PRINT, and INPUT. Understanding how to use these commands will allow you to control the output of your computer. If there is enough interest I will explain the defaults in a future installment.

So there you have eight programming concepts: variables (string and numeric), look and feel interface, opening a device, combining strings, defaults, looping, branching and printing. I hope you will spend two to three hours studying the information

presented.

Next month: An introduction to the most awesome language available for the TI 99/4A, Wycove forth with the t_extensions™. (t_dos™ is a subset of the t_system™. t_dos will be gradually introduced as we explore the t_extensions to Wycove forth and the t_system later in this series.)

PS. In case you did not know it Wycove forth runs on the 9640 and with the t_extensions allow access to all available memory and video resources.

Tech Comments:

Although I have not received my TIM back from Bud Mills which I sent it to him in February 1992 for repair, I feel obligated to express my enthusiasm for the new HRD 4000. If it functions as advertised in the latest edition of MICROpendium then it is the one piece of hardware that every active TI user should have if you don't already own a Horizon Ramdisk. Why? Because the Horizon protects your investment in the TI by providing ram and ram disk capabilities. While most of you may not know it, the vast majority of programs written for the IBM class of computers are written for only 64K code blocks with extended memory programs using 16K blocks of program code. A process called bank switching is used to allow larger programs to be written. This scheme is available on the TI using the HRD. In my last article I said that had two Megs of ram. One meg of this is for programs (the t_system) and the other meg is for ramdisk data. To understand how bank switching can expand the potential of the TI 99/4A, examine the following table.

Memory Address	Computer function	Size
>0000	console ROM: two 4K ROM chips used for the system monitor	8K bytes
>2000	Low Memory expansion: 8K of the 32K memory expansion	8K bytes

>4000	Peripheral ROMs: controls devices like disk drives	8K bytes
>6000	Cartridge space: used for TI cartridges like Extended BASIC	8K bytes
>8000	VDP, Sound and Speech input/output area	8K bytes
>A000	High Memory: 24K of the 32K memory expansion	24K bytes

The maximum amount of memory that the 99/4A can directly address is 64K. That memory is subdivided electrically into eight areas of 8k. The 6000 hex memory space reserves 8K for use with different cartridges. Because a lot of software for the 99/4A is cartridge based, programs can be changed by simply changing cartridges. We can combined the contents of several cartridges into one cartridge and add a selector switch to choose which cartridge program we wanted. Each cartridge represents a bank of programs. Every time we manually choose a different cartridge we are in effect selecting a different bank. Of course, the smarter route would be to develop a method which would allow us to accomplish the switching by way of software. Such software could contain pull-down menus with choices being selected possibly by a mouse.

Most of the cartridges for the TI contain ROM or GROM and, therefore, cannot be changed. If we used RAM, then we could have our choice programs available instead. Programs like a calendar, a drawing program, a word processor, etc., would be instantly available. The HRD offers such a capability!

When I first explained that to use the `t_system™` you had to have an Horizon ramdisk many people complained. What they did not understand is that although companies such as Myarc, Foundation, Corcomp, and Rave created memory boards which, also, functioned as ramdisks and RAM, the

most significant difference was that they forced the programmer to bank switch ALL four 8k banks the main 32k used for memory expansion in and out at a time whereas the HRD leaves the 32K area alone. This difference permitted me to use a stable (virtually bug-free) programming environment (using forth) without having to develop custom software for the very small market of users of these memory cards. That is, I could write programs using my `t_system™` kernel for 32K as well as several megabytes. My code will work with the original HRD which only supported 2K banks. I used these 2K banks in a database in my military job. Imagine being able to search a thousand records in a matter of seconds on a TI. Think about it. Till next month. ct ♦

Build Your Own... Disk Of The Year

Last month, Ralph Rees demoed a great utility for persons who enjoy typing in their own programs. It was called CHECKSUM. This month's BYO...DOM is the text versions of those programs. This article appeared in the January 1988 issue of VAST News and also as noted below.

CHECKSUM

By Tom Freeman

(This article and the accompanying programs first appeared in the LA TopIcs, newsletter of the Los Angeles 99/4A User Group.)

Have you ever typed in a TI-99/4A version of a BASIC program from a magazine and noticed that the versions of the program for other computers have little numbers at the end of the lines that you don't have? They were for error checking on your typing, to ensure no mistakes.

— See *CHECKSUM* on page 10

— CHECKSUM, from page 9

Have you ever laboriously typed in a long program and run it, only to find that it crashes? Or doesn't work as it is supposed to, all because of a simple typing error that you can't find?

So, why doesn't TI have one? Now you do!

This may be the most useful program that I have published for general use, because almost everyone does BASIC at one time or another. It involves only one extra step for the programmer, and one for the user who is typing the published program in.

It is really a simple method, and depends on the manner in which the TI stores BASIC programs. However, it requires a memory expansion and disk drive and works only in Extended BASIC (although BASIC programs can be entered in XBASIC, SAVED and then RUN in BASIC).

You may remember the format in which "MERGE" type programs are stored on disk. The MERGE format is actually a duplicate of the way in which the actual program is stored in memory, or on disk. The difference is it is a display type file, with each record starting with two bytes for the line number, and then the actual program line. In memory, however, the program lines are stored contiguously, and in seemingly random order. (Actually, the order depends on the order in which they were entered.)

A separate line number table is stored below the program area and keeps track of the line numbers and pointers to where each line begins. Each line consists of one byte "tokens" for all reserved words with all strings, including the names of subprograms such as LOAD, SCREEN, etc., being spelled directly.

When you enter any line in Extended BASIC (either a command or program line with the line number coming first), it is first moved to the so-called "edit buffer" at address >8C0 in VDP. The BASIC bias is preserved. The purpose of this is that if you press FCTN 8 (REDO) then the whole line or lines can be retrieved.

Next, everything is "crunched" by replacing each reserved word with its token, subtracting the BASIC bias from strings, computing their length, etc., and placing the result in the "crunch buffer" at >820 in VDP. Once it is there, it can be transferred to the appropriate place in memory expansion. This is the area that is used when my program computes the "checksum" by merely adding the value of each byte! The number is never allowed to go over hex >FF - the high byte is ignored (thus in decimal, no number over 255). The assumption is that it is extremely unlikely, probability approaching zero, that a small number of mistakes will result in a number that differs by exactly 256 or a multiple thereof. The one exception is that if you transpose two characters, there's nothing I can do about that.

Now, what does the programmer do? First, his program must be completely debugged, as no changes may be made after the checksums are computed, or they will, of course, differ. Next, he SAVES his program in MERGE format. Now the following program must be run on the result:

```
100 ! CREATE CHECKSUMS FOR X
    BASIC PROGRAMS, BY TOM FREEM
    AN,LA 99'ERS.FROM MICROpendi
    um,OCT,1987,PAGE 28 !210
```

```
110 !SHOULD BE USED TOGETHER
    WITH "CHECK" ASSEMBLY FILE
    THAT WILL PRINT CHECKSUMS ON
    SCREEN !099
```

```
120 DISPLAY AT(2,1)ERASE ALL
    : "CREATE CHECKSUMS FOR XBASI
    C ERROR CHECKING": : " by
    Tom Freeman" !085
```

```
130 ! From MICROpendium OCT.
    87, JAN.88, MAR.88. Modified
    to printer by H.Ryan MAR.90
    !075
```

— See CHECKSUM on page 11

— CHECKSUM, from page 10

```
140 P=0 :: DISPLAY AT(8,1):"
PRINT CHECKSUMS? (Y/N)" :: A
CCEPT AT(8,24)BEEP VALIDATE(
"YN")SIZE(1):R$ :: IF R$="Y"
THEN GOSUB 280 !194
```

```
150 DISPLAY AT(10,1):"INPUT
MERGE FILE?": " DSK1" !216
```

```
160 ACCEPT AT(11,6)SIZE(-15)
BEEP:I$ :: OPEN #1:"DSK"&I$,
VARIABLE 163,INPUT !039
```

```
170 DISPLAY AT(13,1):"OUTPUT
MERGE FILE?": " DSK1" !061
```

```
180 ACCEPT AT(14,6)SIZE(-15)
BEEP:O$ :: OPEN #2:"DSK"&O$,
VARIABLE 163,OUTPUT !156
```

```
190 LINPUT #1:A$ :: IF LEN(A
$)=2 THEN CLOSE #1 :: PRINT
#2:CHR$(255)&CHR$(255):: CLO
SE #2 :: IF P>=1 THEN 300 ::
STOP !231
```

```
200 Z=ASC(A$)*256+ASC(SEG$(A
$,2,1))!000
```

```
210 B$=SEG$(A$,3,163):: L=LE
N(B$):: IF L>157 THEN 260 !1
93
```

```
220 N=0 :: FOR X=1 TO L :: Y
=ASC(SEG$(B$,X,1)):: N=N+Y :
: NEXT X :: N=N AND 255 :: N
$=STR$(N):: N$=RPT$("0",3-LE
N(N$))&N$ !088
```

```
230 DISPLAY AT(20,1)BEEP:"AN
ALYZING LINE ";Z:"CHECKSUM T
OTAL ";N$ !093
```

```
240 IF P>=1 THEN GOSUB 290 !
```

111

```
250 PRINT #2:SEG$(A$,1,L+1)&
CHR$(131)&N$&CHR$(0):: GOTO
190 !098
```

```
260 DISPLAY AT(22,1)BEEP:"WA
ARNING!": " LINE";Z;"IS TOO LO
NG!": "PRESS ANY KEY TO CONTI
NUE" !123
```

```
270 CALL KEY(0,K,S):: IF S=0
THEN 270 ELSE PRINT #2:A$ :
: GOTO 190 !017
```

```
280 OPEN #3:"PIO" :: PRINT #
3:CHR$(27);"S1" :: PRINT #3:
CHR$(27);CHR$(65);CHR$(6);::
PRINT #3:" Line - Sum" ::
PRINT #3 :: P=1 :: RETURN !0
14
```

```
290 PRINT #3:Z;"- ";N$ :: R
ETURN !204
```

```
300 CLOSE #3 !153
```

Notice the exclamation and three numbers at the end of each line? The program was RUN on itself!

Here is what happens: each record of the merge file is read in, the the first two bytes ignored (we don't need the line number) and the rest are added up.

Next, the identical record is printed to the output file with the addition of the token for "!" (REMark) and the three characters of the checksum. This will work even if the program line already contains a REMark (as in lines 100-110). *The user must not type these characters*, since they are not computed into the checksum. At the end (it may take a while for a long program, but only needs to be run once), the programmer types NEW and MERGEs in the output file, then SAVES it in the normal mode, or LISTs it to a printer, or

— See CHECKSUM on page 12

— CHECKSUM, from page 11

whatever. This is the form to be published.

What the user must do once is type in the CALL LOAD version of the object code for the CHECKSUM program. Then SAVE the program to disk (CHECKSUM would be a good name to SAVE it under!). Then RUN the program.

What the assembly routine "CURSOR" does is some housekeeping, such as moving the numbers 0-9 to character sets 13-14, changing the colors there, redefine the cursor, put up the title screen, etc., and then turn on the user-defined interrupt. Now, at every VDP interrupt (each one-sixtieth second), the routine at CHECK begins. The interrupt can be turned off with CALL LINK("OFF") and back on with CALL LINK("ON") at any time and the shape of the cursor will tell you which mode you are in. Every time you enter a new program line (and for some reason after FCTN 8 even if no changes are made) the checksum will appear at the bottom of the screen and one extra line scrolled up. Here is the key — it should correspond to the one published that you are attempting to copy in. Hence, no errors!

I spent many hours with MG Explorer, by Doug Warren, finding out what is going on when you enter a line in Extended BASIC. The address range in GROM of >6AA0 to >6AD8 should be broad enough to cover the various versions of XBASIC, since they differ by a few bytes here and there (the range needed in my module was >6AAE to >6ACA).

I hope that everyone find this program useful and that it is widely used. I would like to thank Doug Warren for writing Explorer, without which I could not have done this, since I needed to find out where XBASIC does what. And I would like to thank Craig Miller for his invaluable help and advice while I was writing the program.

1 !CALL LOAD VERSION OF OBJE
CT CODE FOR CHECKSUM PROGRAM
, BY TOM FREEMAN, LA 99ERS. FR

OM MICROpendium, OCT, 1987, PAG
E 29 !193

100 CALL INIT :: CALL LOAD(9
460,0,0,0,0,0,106,160,106,
216,0,10,11,13,0,0)!180

110 CALL LOAD(9484,0,126,66,
66,66,66,126,0,31,31,32,32,8
8,66,65,83,73,67,32,69)!144

120 CALL LOAD(9504,82,82,79,
82,32,67,72,69,67,75,69,82,3
2,32,32,32,32,32,85,83,73,78
)!107

130 CALL LOAD(9526,71,32,67,
72,69,67,75,83,85,77,83,32,3
2,32,32,32,66,89,32,84,79,77
)!119

140 CALL LOAD(9548,32,70,82,
69,69,77,65,78,44,32,76,65,3
2,57,57,69,82,83,2,132,0,10)
!052

150 CALL LOAD(9570,17,2,2,36
,0,7,2,36,0,48,192,68,2,33,0
,176,6,193,4,32,32,32)!199

160 CALL LOAD(9592,4,91,2,0,
3,240,2,1,37,4,2,2,0,8,4,32,
32,44,2,0,4,128)!121

170 CALL LOAD(9614,2,1,39,22
,2,2,0,80,4,32,32,44,2,0,7,0
,4,32,32,36,4,32)!166

180 CALL LOAD(9636,32,24,0,3
8,2,2,37,22,2,3,96,96,2,4,0,
36,192,66,172,131,6,4)!204

190 CALL LOAD(9658,22,253,2,
0,2,228,2,2,0,24,4,32,32,36,

— See CHECKSUM on page 13

— CHECKSUM, from page 12

4, 32, 32, 24, 0, 38, 2, 0) !067

200 CALL LOAD(9680, 2, 228, 2, 1, 37, 46, 2, 2, 0, 24, 4, 32, 32, 36, 4, 32, 32, 24, 0, 38, 2, 0) !020

210 CALL LOAD(9702, 2, 228, 2, 1, 37, 70, 2, 2, 0, 24, 4, 32, 32, 36, 2, 0, 3, 240, 2, 1, 37, 12) !006

220 CALL LOAD(9724, 2, 2, 0, 8, 4, 32, 32, 36, 2, 0, 38, 36, 200, 0, 13, 1, 196, 4, 91, 2, 0, 3, 240) !119

230 CALL LOAD(9746, 2, 1, 37, 4, 2, 2, 0, 8, 4, 32, 32, 36, 4, 224, 131, 196, 4, 91, 216, 32, 152, 2) !239

240 CALL LOAD(9768, 36, 248, 6, 224, 36, 248, 216, 32, 152, 2, 36, 2, 48, 6, 224, 36, 248, 6, 32, 36, 248, 136, 32) !133

250 CALL LOAD(9790, 36, 248, 36, 250, 26, 8, 136, 32, 36, 248, 36, 2, 52, 27, 4, 4, 224, 36, 244, 4, 224, 1, 31, 4) !013

260 CALL LOAD(9812, 216, 32, 36, 248, 156, 2, 6, 224, 36, 248, 216, 32, 36, 248, 156, 2, 2, 0, 8, 28, 2, 1) !054

270 CALL LOAD(9834, 37, 20, 2, 2, 0, 2, 4, 32, 32, 36, 2, 0, 8, 15, 2, 1, 244, 0, 2, 2, 0, 13) !105

280 CALL LOAD(9856, 4, 32, 32, 3, 2, 5, 128, 6, 2, 22, 251, 2, 0, 7, 4, 4, 32, 32, 48, 7, 96, 36, 244) !204

290 CALL LOAD(9878, 22, 62, 2, 1, 0, 3, 152, 33, 36, 254, 131, 117, 1

9, 3, 6, 1, 22, 250, 4, 91, 200, 32) !180

300 CALL LOAD(9900, 131, 4, 131, 4, 19, 49, 136, 32, 131, 4, 131, 74, 22, 45, 7, 32, 36, 244, 208, 160, 1, 31, 66) !038

310 CALL LOAD(9922, 9, 130, 2, 0, 8, 32, 2, 1, 39, 22, 4, 32, 32, 44, 4, 224, 37, 2, 184, 49, 37, 3) !195

320 CALL LOAD(9944, 6, 2, 22, 25, 2, 200, 11, 36, 246, 4, 32, 32, 24, 0, 38, 2, 0, 2, 226, 193, 96, 37, 2) !138

330 CALL LOAD(9966, 2, 2, 0, 10, 2, 3, 0, 100, 2, 6, 0, 2, 4, 196, 61, 3, 6, 160, 37, 94, 5, 128) !027

340 CALL LOAD(9988, 192, 194, 6, 6, 22, 248, 193, 5, 6, 160, 37, 94, 194, 224, 36, 246, 4, 91) !104

350 CALL LOAD(16376, 79, 78, 32, 32, 32, 32, 37, 244) !042

360 CALL LOAD(16368, 79, 70, 70, 32, 32, 32, 38, 14) !240

370 CALL LOAD(16360, 67, 72, 69, 67, 75, 32, 38, 36) !002

380 CALL LOAD(16352, 67, 85, 82, 83, 79, 82, 37, 122) !053

390 CALL LOAD(8194, 39, 22, 63, 224):: CALL LINK("CURSOR") !143

400 CALL LOAD(-31952, 255, 231, 255, 231) !229

* * * * *

INDEX

Articles appearing in Volume 8 of VAST
News (1992)

ISSUE DESCRIPTION

4A FORAY

(Tid bits from the mind(?) of Ralph Rees)

- 2 Finding new members through the TI supplied mailing labels; Various equipment that will be needed for the FEST; Chris Taylor's "Concept 99."
- 3 Some impressions of FEST-WEST '92; some tips for new members.
- 4 Talk on the Pizza party we had to celebrate the success of the FEST; some of the plans for the group now that the FEST is a success; plea, again, for more people to get involved and contribute.
- 5 Some problems Ralph is having with his TI; the GREAT pizza party we had; the "comedy of errors" in trying to put on last month's demo.
- 6 New motherboard for the group BBS; Ralph is feeling much better now; his "laptop TI project;" for members to please write something for the newsletter.
- 7 CRU addresses of various devices; BBS being under-used; our pre-meeting meetings a Denny's.
- 8 A note in praise of newsletter editors; all should contribute something to the newsletter.
- 9 Some ways of attracting new

ISSUE DESCRIPTION

- members to the group; things not to forget about those new members.
- 10 Examples of "Technospeak"
 - 11 Guest "editorial" by Jim Peterson about what equipment you really need to enjoy computing at home.
 - 12 Comment on Christmas commercialism; Virginia Davis' ART and the Library; Chris Taylor and a group project; mini proposal for another VAST Faire.

TI WRITER TIPS

(Help to get the most from TI's word processor)

- 3 Form Letters Part 1 by Ray Frantz
- 4 Form Letters Part 2 by Chick De Marti, the LA 99ers.

OUR BBS AND YOU

- 5 30 Commandments of BBS'ing
- 6 Call Waiting and Your Modem by Ray Frantz

SPEAK OUT

(Personal opinion from our members)

- 3 By Ray Frantz. An opinion on dues.
- 4 By Ray Frantz. An opinion on meeting structure and the "swap table" activities.

— See *INDEX* on page 15

— INDEX, from page 14

ISSUE DESCRIPTION

MISCELLANEOUS ARTICLES

- 3 HIND SIGHT. After thoughts on Fest West (or hoof-in-mouth)
- 3 YOU SAY YOU DON'T HAVE A COLD, YOU HAVE A VIRUS? Q & A's regarding Michaelangelo and other viruses by Aryea Goretsky.
- 4 OLD TRICKS FOR NEW (AND NOT SO NEW) USERS. By Joseph Cohen
- 4 PROGRAMMER'S DILEMMA. A Poem by Don Lester
- 5 TI VENDOR'S LIST. Suppliers of goods and services for the TI.
- 6 PROGRAM LISTINGS. How to print a program to your printer. By Phil VanNordstrand.
- 7 BORED. General "chit chat" from Tom. By Tom Pfeffer
- 9 THE HOME COMPUTER by Jim Peterson
- 11 KNOW YOUR CANDIDATES. A look at those running for User Group offices in '92.

CHRIS' CORNER

By Chris Taylor

- 6 Intro to Chris and overview of The Taylor Company
- 7 Why the TI Part 1
- 8 Why the TI Part 2

ISSUE DESCRIPTION

- 9 Why the TI Part 2, continued
- 10 Why the TI Part 3
- 12 Why the TI Part 4, programming

VIEWS AND REVIEWS

A closer look at some programs for the TI by users.

- 8 General look at graphics programs by Ray Frantz

ON THE LIGHTER SIDE

- 6 "These Daze"
From Reader's Digest

BUILD YOUR OWN DISK OF THE YEAR

by Jim Ely

- 9 (Column heading was "Get Your Programs") Day of the Week calculator
- 10 AMIGA, ball demo
- 11 Several programs to demo sprites, motion and sound
- 12 The CHECKSUM Programs

FEST WEST NOTES

(News of Fest-West 92)

- 1 Floor plan and speaker schedule
- 2 Booth assignments, revised speaker schedule

(Index compiled again, as usual, by Mike Marfisi. Thanks a lot for your help, Mike!)

VAST User Group Information

The VAST Computer User Group is a support group for the TI-99/4A Home Computer and compatibles.

The VAST Computer User Group operates a BBS 24 hours a day, 7 days a week. The phone number is (602) 233-0790.

CURRENT OFFICERS

PRESIDENT

Jim Ely Unpublished

VICE PRESIDENT

Ralph Rees 869-8145

SECRETARY/TREASURER

Tom Pfeffer 940-0974

LIBRARIANS

Hazel & Wallace Knight . . . 938-5446

NEWSLETTER EDITOR

Jim Ely Unpublished

BBS SYSOP

Mike Grogan 272-4315

Send any correspondence to the address below.

ADVERTISING RATES

There is **NO CHARGE** to **MEMBERS** for **PERSONAL** advertising. Non-members will be charged \$1.50 per ad.

Please contact the Secretary/Treasurer for **COMMERCIAL** advertising rates.

DISCLAIMER: The VAST Computer User Group can assume **NO** responsibility for the accuracy of the information in this newsletter or for programs or construction projects tried by its members or others. You try them at your **OWN RISK!** Any material herein may be republished by other user groups so long as credit is given to the **ORIGINAL** sources.

VAST NEWS

c/o VAST USER GROUP

P.O. BOX 25576

TEMPE, AZ 85285-5576

INSIDE INFORMATION

ITEM	PAGE
Secretary's Slate	2
Editor's Desk	4
4/A Foray	5
Chris' Corner	6
Disk of the Month	9
1992 Index	14
VAST Information	16

FIRST CLASS MAIL

RALPH REES
18815 N. 13TH AVE
PHOENIX AZ 85027

FIRST CLASS MAIL