8612 T.I. SHUG. (042)

TISHUG NEWS DIGEST

It's Christmas

Time Again

DECEMBER 1986

SAM MUDIE

TISHUG NEWS DIGEST

TI.99/4A Owners Home
Computer User Group.
TIsHUG NEWS DIGEST
November 1986
Correspondence to:
P.O. Box 302
CARLINGFORD
NSW.
Australia.
Texpac BBS Tel.(02)3191009

COMMITTEE MEMBERS

Group Co-Ordinator:
Fred Morris.....Tel.(02)8713873
Secretary:
VACANT-VOLUNTEER PLEASE! Treasurer
Mel Copeland.....Tel.(047)351340
Communications Co-Ordinator:
Shane Anderson....Tel.(02)5690926
Merchandise Co-Ordinator:
Chris Buttner.....Tel.(02)8717753
Technical Co-Ordinator:
Robert Peverill...Tel.(02)6024168
Software Co-Ordinator:
Terry Phillips....Tel.(02)7976313

PUBLICATIONS LIBRARIAN

Brian Graham.....Tel.(02)7743223

TEXPAC BBS SYSOP

Ross Mudie.....Tel.(02)4562122

REGIONAL HOME GROUPS

Country Club:
Brian Graham.....Tel.(02)7743223
Glebe:
Mike Slattery.....Tel.(02)6920559
Penrith:
Mel Copeland.....Tel.(047)351340
Central Coast:
Russell Welham....Tel.(043)924000
Liverpool:
Stan Puckle.....Tel.(046)256157
Bankstown:
Peter Pedersen....Tel.(02)7722396
Illawarra:
Geoff Trott.....Tel.(042)296629
Carlingford:
Chris Buttner.....Tel.(02)8717753
Sutherland:
Peter Young.....Tel.(02)5288775
Manly/Warringah
Dennis Norman.....Tel.(02)4523920
Coffs Harbour:
Keir Wells.....Tel.(066)551487

MEMBERSHIP AND SUBSCRIPTIONS

Membership fee.....$8-00
TIsHUG News Digest....$25-00p.a.
Overseas Air Mail...US$30-00p.a.
Publication Library...$5-00p.a.
Texpac BBS.....$5-00p.a.
Members of other
TI.99/4A User Groups..$10-00p.a.
Public access.....$25.00p.a.

MEETINGS - TIME AND PLACE

COMMITTEE MEETING:
Second Wednesday of each month at
Woodstock Community Hall
Church Street, Burwood.
Starts - 6.30pm.

GROUP GENERAL MEETING
First Saturday of each month at
Shirley House
Ethel Street, Burwood.
Starts - 2.00pm.

UP FRONT

IN THIS ISSUE

In the centre of this issue is a continuation of the four page life out section with the following programs for you to type in and enjoy:

Frequencies, Steinway, TI Writer Bit Image Printer Graphics, Tug-a-War, Cool Water, Add & Carry, a 1 liner and Print Music Demo.

As you can see, once again a massive Christmas bumper issue of our TND, and featuring TWO lift-outs, the continuation of our 4 page Program Lift-out section, and a very special 1987 CALENDAR with artwork by Arto. By the way, another 100 copies of this Calendar have also been prepared on HIGH GLOSS POSTER PAPER, which is available for sale from the TIsHUG CLUB SHOP at both the Christmas Meeting/Bar-B-Q & February'87 Annual General Meeting. They'll make great Christmas presents to give & share.

With the inovation of both new Software & Hardware for your TI-99/4(A), we hope that you will stay with us, and keep informed with the latest developments. Each year at this time, I say that Next year holds some great surprises for you, and each year I'm right. Next year, once again, some great things are planned for your computer.

I want to take this time, to challenge you...Our membership has dropped off, and yet our support for you hasn't. You can help to bring our numbers back up, by doing your own promotional work for this, your club. How many friends do you have, who own a TI-99/4(A) and who either havent got around to renewing their membership, or perhaps have'nt joined us yet? There is a very sexist saying which goes like this...THERE ARE 3 FORMS OF ADVERTISING, TELEPHONE, TELEVISION & TELLA-WOMAN, the latter being the most affective! Like I said, sexist, but what it is really saying is that WORD-OF-MOUTH ADVERTISING is the best way to share the news. How about you TELL-A-FRIEND about TIsHUG, and follow it through, by bringing them along to club Regional Meetings and Monthly Get-to-gethers. Show, or even better, GIVE them a copy of your TND (or obtain a few back issues from your Co-Ordinator (he's got stacks put aside for promotional use). Invite them around to your home and share some of your Club Software AFTER THEY HAVE JOINED!!! Those of you who, for whatever reason, are selling your computer, promote the club to those who purchase it!

See you at the Christmas Party...remember, you won't need to bring a thing, except yourself. All the food and softdrinks are on us! See you there, 12 noon.

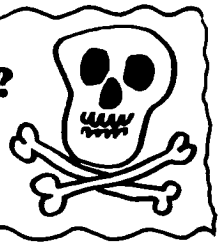
Merry Christmas Shane

Season's Greetings

WE WISH
YOU A
MERRY XMAS
and
HAPPY 1987

from
THE BUSY
FEW

WHO AM I?



A Software Pirate??? Yes Sir, that's me all right. If there is any piece of software for the TI-99/4(A) worth having, I've probably got it. And I guess you could say I STOLE about 80% of it. Why, I've got stuff most people don't even know is on the market yet, and besides that....well, wait a minute; I might as well start this story at the beginning.

I bought my TI-99/4(A) way back when they were around $300+ just for the console. I bought the console, and a cassette recorder, and two, count 'em ... two, cartridges (if you must know, they were Household Budget Management and TI Invaders). I think I really bought the stuff just so I could say "Oh sure, I've got a computer at home, doesn't everybody?" So I took the whole deal home, read the instructions, hooked it to my TV, and fired it up. And in about half an hour, I was the one who was hooked. I mean, really HOOKED!!! I never realized a machine and some electrical impulses could do so much, so fast. Right away, I was off in search of more cartridges. I found 'em too. Lots of them ... $29, $39, $49 apiece. I bought --- my wife grumbled. Then I discovered cassette based software. Now this was great. Software at a fraction of the cost of cartridges. I spent entire weekends combing stores, flling out mail orders, and (so I'm told) cackling loudly over each new piece of software. Each day's mail brought a new cassette or two, each week-end shopping trip brought (and bought) many new cassettes. By now, my wife was speaking in a somewhat more authortative (and disapproving) voice. I paid little attention; if there was a cassette for the TI-99/4(A) I didn't own, I had to have it, and I got it. Then, lightning really struck. I discovered disk drives. What a miraculous device. Well, of course, I HAD to have one of those. Pretty soon, it just didn't make sense to have that wondrful disk drive to load great programs without having more memory; and after that.....well, I'm sure you know how the story goes. My wife screamed, the kids were muttering something about being hungry, and the bank was sending stupid notices in the mail about a process called foreclosure. But in the end, I had my fully (I'm talking SIX CARDS IN THE BOX, PRINTER, 1200 BAUD MODEM, 2 DSDD drives) configured system.

Now it was time to buy some really good software to use on my beautiful set-up. Unfortunately, now was also the time my wife decided to call a halt to this affliction she refers to as "computermania". She quietly informed me that I could be married to her, or I could be married to the computer, but I could not do both. She was willing to share; I could still "play" (as she calls it), but only as long as it occupied a good deal less of my time, and a GREAT deal less of OUR money. I certainly wasn't going to junk my wife for a computer, but I had to have more super software. What to do, what to do???

This was about the time when I met a couple of other people who also owned TI's (up to that point, I guess I thought I was the only person in the universe who owned a TI-99/4(A). At first, we just got together and talked about "computer stuff". After that, we sort of swapped a few simple programs, etc., etc., and before you could say Captain Kidd, we were just trading programs back and forth like there was no tomarrow. I think in the back of my mind I knew we shouldn't be doing it; but after all, I rationalized, I could certainly not afford to buy most of this software on my own. Besides that, at least one of us was paying for the original program (maybe), so that probably made it OK! We weren't swapping anything of major importance (I kept telling myself), and most of it was old stuff anyway. Then the came day when I was copying a disk with Disk Manager 2 and up popped the message "PROPRIETERY DISK ERROR". Well, I darn sure knew what that meant. I was not supposed to have this disk of copyrighted material unles I paid for it. But, I wanted the program, and thus I fell into the next phase of my pirating career; namely, unlocking programs and disks.

Now, TI's disk protection system is pathetically weak; and so, armed with Navarone's old Disk Fixer version 1.1 (which I pirated), it took me about 7 minutes to solve the problem. PROPIETRY DISK indeed!! Not any more! From there, it was an easy slide into working on programs. Mind you, this was long before anyone had ever heard of "CALL LOAD(-326__,0)". back when the only way I could find to unprotect an extended basic program was to change bytes 0 and 1 in the first sector of a program to their hexadecimal twos complement values. As programmers and manufacturers found more sophisticated ways to protect programs and disks, I found better and quicker ways to unprotect them. I've seen a barrel full of protected schemes: destroyed file directories; partially erased headers on Sector 0; special assembly language routines to load the program sector by sector; scratched disks; 30 track disks; partially uninitialized disks; dead sectored disks; disks with uninitialized tracks. None of them slow me down anymore.

I can honestly say I have yet to run into the protection plan I could not crack. Modules??? Let me tell you something about them. I was dumping modules to disk back when Craig Miller of Miller's Graphics thought a "Gram Kracker" was something made by Keebler that you served to kids after school with a glass of milk! It was a long and laborious process then, but I managed. So, because I have no compunction about pirating software, and am capable (I think) of breaking any sort of protection, I probably have the MOST complete library of TI-99/4(A) software in the whole world. And to be honest, I'm pretty proud of it.

All of this brings me to what I suppose is the main point of all this rambling. In recently reviewing my wonderful and most complete library of software, I have been rather amazed to come to the realization that the very best software for our computer has come into existence SINCE Texas Instruments left the home comuter market. Although I think I understand why, no explanation is necessary here. And in the last nine months or so, about 95% of the really super good software has been brought to the TI community under the FREEware or FAIRware concept. Something totally new. And EXPERIMENTAL!!! You get the program from someone who has it (without having to pirate it, since it always somes unprotected), try it out, and if you like it, send the author what they consider to be their just dues (which is almost always $10.00 or less). To mention a few: William Warren's PRBASE (possibly the best Data Base program I have ever seen); Marty Kroll Jr's CATALOGING LIBRARY (absolutely the best disk cataloger available); Jim Swedlow's very unique (and much needed) SIDE*PRINT. I have them all. What's more, I PAID FOR THEM ALL! Why? In the last year or so, softwae production by third party manufacturers has been slow or nonexistant. Had you noticed? Probably in part due to people like me who pirate software the minute it hits the market, and in part because their programs can't hold a candle to the stuff that's coming out as FREEware and FAIRware. These software authors are part of our TI community; they work with the TI-99/4(A) every day, know what it can and can't do, and have written their programs to take advantage of every last byte of programming power out little orphan has. These people have put in many hours (or hundreds of hours) of their time to develop practical, beneficial, and much needed programs. But, everyone's patience has a limit. No one is going to continue making good efforts if those efforts are not appropriately rewarded. So....if you get a piece of FREEware or FAIRware - try it and like it - intend to use it - for Blackbeards sake, PAY FOR IT! I think the

→

entire concept, properly treated, will provide us with mny extra years of useful life for the TI-99/4(A). FREE(FAIR)ware is the wave of the future, but that wave will surely slow to a trickle if not fed by a small stream of your (and my) dollars.

Now then, before you start to applaud my apparent ideological reversal, stow it! I'm still a software pirate and probably always will be. My friends and I still get together; still swap programs; and probably always will. If some hot new highly protected piece of software comes over the horizon -- I'll snatch it in a second, break it in a minute, and pass it on within the hour. Believe It!!! But if it's FAIRware or FREEware, and I think that piece of software will be valuable to me, I'll do my part. Will YOU do YOURS?

Well, that's all I've got to say, so I'll take my Polly Parrot and peg-leg my way over to your newsletter editor's house to deliver this treasure of a story in time for the Christmas issue of the TIsHUG NEWS DIGEST (perhaps I'll be rewarded with a bottle of rum!). So until my next "YO HO HO" ..... but avast ..... I almost forgot. I suppose some of you are wondering who I am. You're asking yourself, who IS this unscrupulous, nefarious pirate that speaks so glibly of "borrowing" software and seems proud of him(her)self for having done it? Well maybe --- just maybe --- I'm you. THINK ABOUT IT !!!!!!!!!


```
100 !XB SCREEN DUMP*
110 ! MINI UTILITY *
120 !BY ARTO HEINO *
130 !   30/8/86    *
140 !***************
150 PRINT "RS232.BA=4800.DA=8.CR";:: ACCEPT AT(24,1)SIZE(-26)BEEP:DV$ :: OPEN #1
:DV$ :: PRINT #1:CHR$(27);"A";CHR$(8);
160 FOR X=1 TO 32 :: PRINT #1:CHR$(27);"K";CHR$(192);CHR$(0);:: FOR Y=24 TO 1 ST
EP -1 :: CALL GCHAR(Y,X,C):: C=MAX(C,32):: IF C=CA THEN 190 ELSE CALL CHARPAT(C,
CH$)
170 CA=C :: DP$="" :: FOR U=16 TO 1 STEP -2 :: C1=ASC(SEG$(CH$,U,1)):: C2=ASC(SE
G$(CH$,U-1,1)):: C1=C1+(C1>57)*7 :: C2=C2+(C2>57)*7 :: V=0
180 FOR I=0 TO 3 :: V=V+(C1 AND 2^I)+(C2 AND 2^I)*16 :: NEXT I :: DP$=DP$&CHR$(V
):: NEXT U
190 PRINT #1:DP$;:: NEXT Y :: PRINT #1:CHR$(10):: NEXT X :: CLOSE #1
```

Warning! The Screen dump is slow. It should be adaquate for poor people without a full system. If you want to use it as routine just make DV$ your print device name and resequence to higher numbers.

SUPER SPECIALS FOR CHRISTMAS

Don't miss this one:

A DISK STORAGE BOX (100 CAPACITY) AND 20 DOUBLE SIDED DOUBLE DENSITY DISKS. THE LOT JUST $49.00.

Watch this section for forthcoming specials:- MODEMS (with software)

MULTIPLAN AND LOGO SETS

I have received word these are now on the way from the U.S.. If you ordered a set - don't despair. Price should be no more than $55.00.

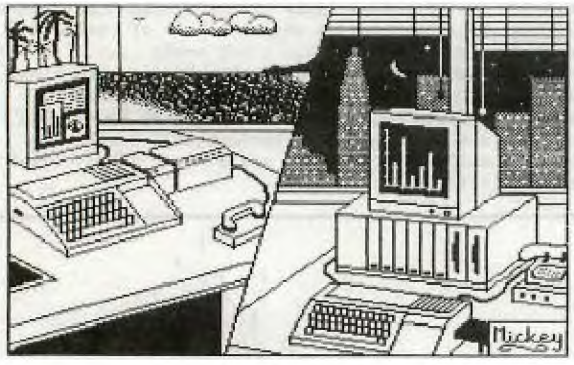
AUTO SPELL CHECKER (by Dragonslayer)

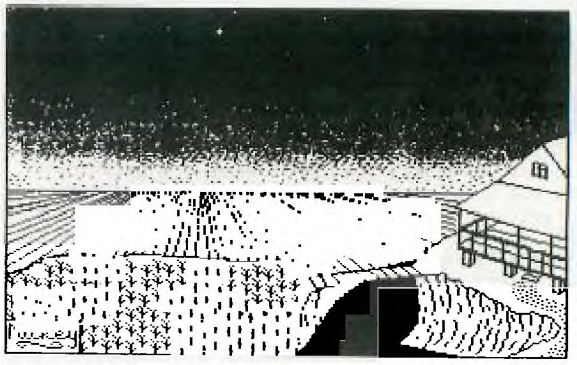
For those who do word processing on the TI I have one copy of the latest version of the Auto Spell Checker. This latest disk is MUCH faster than the old version and really speeds up your work. One of the nice features of this programme (not found on some large systems e.g. Wang) is the ability to view "unrecognized words in context. Price $60.00.

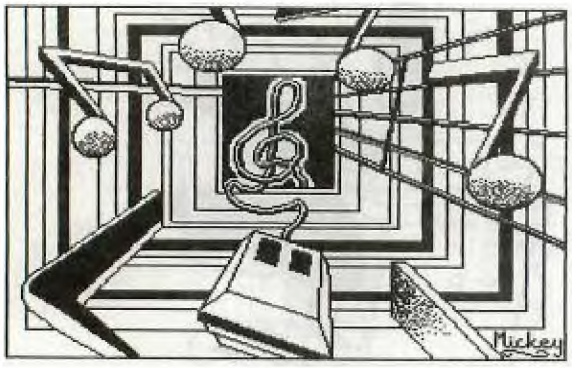
Continued on Page 9

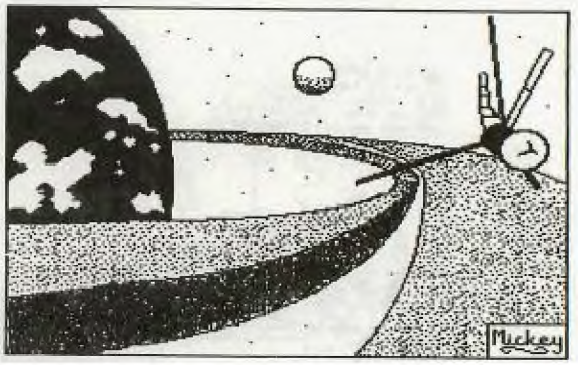
"GRAPHX GALLERY"
by Michael Donovan

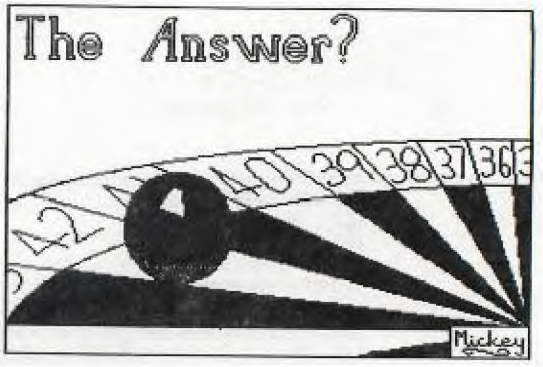
Merry Christmas and
a Happy New Year!

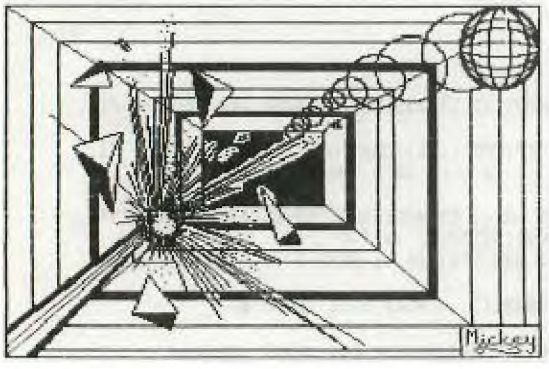


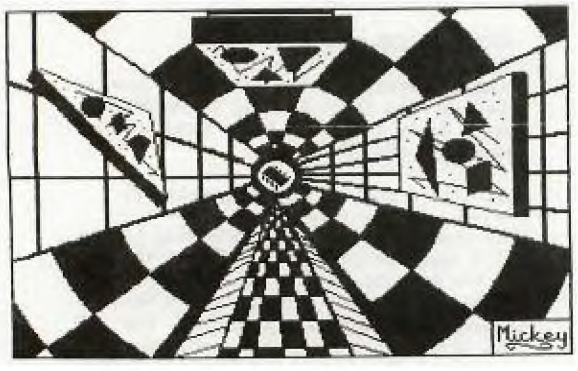


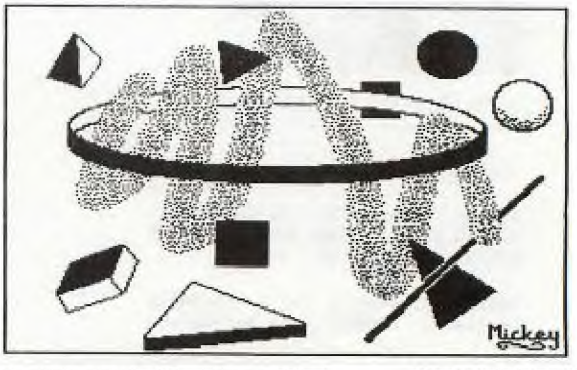








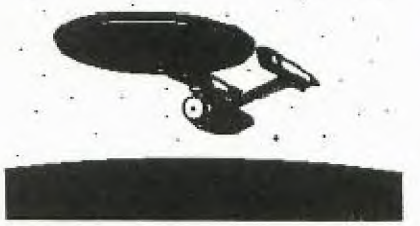




all of these pictures
have been drawn on a
TI-99/4(A) home computer.

Picture Gallery
DIGITALIZED PHOTOGRAPHS
from our TI-99/4(A)

"WHERE NO MAN HAS GONE BEFORE"



Sally Ride

Christa McAuliffe

THESE, AND MANY MORE
DIGITIZED PICTURES
ARE AVAILABLE ON DISK
FROM OUR TISHUG CLUB
SHOP...INCLUDING SOME
<R>PICTURES WE ARE
UNABLE TO SHOW HERE!

BASIC NUMBERS

by
Geoff Trott
TIsHUG Illawarra Regional Group

If we want to write programmes in BASIC that use numbers then some of the properties of those numbers can be important. To find out about how BASIC stores numbers, their range of values or the precision of the numbers, we could read the manual and find out the hard way perhaps. However it should give some clues and we could use these to write programmes to get BASIC to tell us what we want to know. For example, in the "User's Reference Guide" on page II-9, there is a section on Numeric Constants. In this it talks about Scientific Notation, Mantissa, Exponent and states that numeric constants are defined in the range of :

```
-9.9999999999999E127 to -1E-128
0
1E-128 to 9.9999999999999E127
```

If a number is between -1E-128 and 1E128 then the value is set to 0 without a warning or error. If the number is outside the range of the largest positive or smallest negative number a warning is given 'NUMBER TOO BIG' and the largest or smallest number is used as appropriate. The 'E' in the number is a convenient way to say that the number after the 'E', called the exponent, is the power of 10 which the number is multiplied by. To investigate the range of numbers would seem to be quite easy. Consider the following programme.

```
100 X=1
110 FOR I=1 TO 132
120 X=X/10
130 PRINT I;X
140 NEXT I
150 STOP
```

If you run this programme you will see two columns of numbers, one increasing by 1 and the other decreasing by a factor of 10. From I equals 100 to 128 the second number will show *** as the exponent. This is because the normal printout of numbers can only cope with exponents up to 99. We appear to have to trust that the number actually is there after that! There are some ways to get around this problem. The first way is to scale the number so that the scaled value is printed. Replace line 130 with :

```
130 PRINT I;X * 1E50
```

Now the values will start from 1E49 and go to 1E-78 before going to 0. If we use Extended BASIC we can go one better with a PRINT USING statement. Try the following line for 130 :

```
130 PRINT USING " #### ##.#^^^^^":I,X
```

If you want more information about this see the Extended BASIC manual on PRINT USING and IMAGE. Try different starting values for X, particularly that the negative values behave in a similar way to the positive ones.

By changing the / to a * in line 120 the large numbers can be investigated with the same programme. Using the scaling method to see the numbers up to the largest will require the multiplier to be 1E-50 of course. It is interesting to note how much faster Extended BASIC is at this simple programme.

To investigate the precision of the numbers requires a simple trick as only 10 digits are printed normally (see page II-66 URG). If we calculate X+D-X the answer should be the value of D. This will only be so if the value of D is larger than the smallest resolution of X+D. If adding D to X results in the loss of precision, then the result of our calculation will not be equal to D. The programme could look like this :

```
100 X=1
110 D=1
120 FOR I=1 TO 20
130 X=X * 10
140 PRINT I;X+D-X;X/D
150 NEXT I
160 STOP
```

This gives 3 columns of numbers with the third one giving the theoretical precision. The middle one gives the value of D, which should be constant, but which will go to 0 when the maximum precision is reached. The maximum precision changes as the initial value of X changes. Try some initial values of X between .9999 and 99 to see if you can recognise a pattern. If we change line 130 to :

```
130 D=D/10
```

then the value of D changes but the over effect should be the same. Other computers may run the first version quite well but will not be as good with this version in that the value of the result will become incorrect before it finally goes to 0. With our BASIC numbers the result is exactly correct until it finally goes to 0.

If you have a play with these programmes and ones like them you can have a bit of fun and learn about your computer in the process. The best way to learn about your computer is to get it to teach you itself!

If you have followed this so far you may want to find out all about the numbers. I shall try to explain how they are stored, but will not be upset if you still have trouble. It is probably necessary to go over several different explanations several times before it all comes clear. Any way here goes! Any number is first scaled to get its magnitude to be greater than or equal to 1.0 and less than 100.0. This means that the integer part of the number is between 1 and 99 (unless it is 0 - a special case). The scaling factor is a power of 100. The easiest way to see this may be with examples.

Number	Scaled Number
1.11	1.11 * 100^0
11.1	11.1 * 100^0
111.1	1.111 * 100^1
0.111	11.1 * 100^-1
0.0111	1.11 * 100^-1

And so on. As you can see any number can be scaled in this way except 0.

Then the integer part and each pair of digits after the decimal point are converted into Hexadecimal (or Binary) in the range 0 to 99 (>00 to >63). Each of these pairs of digits are stored in a byte with the integer byte first followed by the values after the decimal point. There are 7 bytes used in all, which allow 12 digits after the decimal point of the scaled number and either 1 or 2 digits in front of the decimal point giving 13 or 14 digits all together. This is how the magnitude of the scaled number is stored in the computer, but to store the original number the scaling factor needs to be stored.

Since all the scaling is done in powers of 100, the 100 itself does not need to be stored, only the power of the hundred. This is just an integer, but can be positive or negative and so must be stored as a signed integer. Seven bits in one byte are set aside for this signed integer, the exponent, which gives a range of 128 numbers. These are divided up into 64 negative numbers (-64 to -1) and 64 positive numbers (0 to 63). For several reasons which can at first seem obscure, the codes used for these are as follows :

Exponent	Hex code
-64	>00
-63	>01
-48	>10
-32	>20
-16	>30
-1	>3F
0	>40
1	>41
32	>60
48	>70
63	>7F

This byte, containing the exponent, is the first byte in the number, preceeding the byte containing the integer part of the number. The 8 bytes for the numbers considered first would be as follows :

```
1.11    >40,>01,>0B,>00,>00,>00,>00,>00
11.1    >40,>0B,>0A,>00,>00,>00,>00,>00
111.1   >41,>01,>0B,>0A,>00,>00,>00,>00
0.111   >3F,>0B,>0A,>00,>00,>00,>00,>00
0.0111  >3F,>01,>0B,>00,>00,>00,>00,>00
```

Notice the effect of taking the digits in pairs to the right of the decimal point (.1 gives value of 10 not 1). That was not too bad (I hope) but what about negative numbers?

Negative numbers are normally shown by the first bit in the number being a 1. The first bit in our numbers is the most significant bit in the exponent byte, which is why only 7 of the 8 bits were used for the exponent. So far that bit has always been 0, which is correct for positive numbers (distinct from the sign of the exponent which determines the magnitude of the number). The simple way would have been to simply set that bit for negative numbers, and reset it for positive numbers. That is too straight forward in one sense, and too difficult in another. It is easier to set that bit by negating the byte or word, and to reverse the process by negating again. This is what TI has done. To convert a negative number, ignore the negative and convert the magnitude as above, then negate the first two bytes as a single 16 bit word. This is easy for the processor to do, but not quite as easy for us to do. Try these examples.

```
-1.11     >BF,>FF,>0B,>00,>00,>00,>00,>00
-0.111    >C0,>F5,>0A,>00,>00,>00,>00,>00
-.99999999999999E128
          >80,>9D,>63,>63,>63,>63,>63,>63
-1.0E-128
          >FF,>FF,>00,>00,>00,>00,>00,>00
0         >00,>00,>00,>00,>00,>00,>00,>00
1.0E128   >00,>01,>00,>00,>00,>00,>00,>00
0.99999999999999E128
          >7F,>63,>63,>63,>63,>63,>63,>63
```

One of the rules of numbers of this type is that if the first digit of the number, in this case the second byte, is zero the value of the number is zero. This is why zero is different, and why there is the gap between 1E-128 and 0. If you examine the first two bytes of the numbers shown here, you will find that the relative sizes of the numbers follow the same ordering as used for 2's complement signed integers, which is one of the reasons for some of the decisions taken about the coding. Well if you struggled through to the end you deserve a medal, but all I can offer is perhaps a better understanding of BASIC numbers. There are plenty of ways to get the computer to give you information about itself to check all this. Go to it!

Continued from Page 5

MICROPENDIUM MAGAZINES

September issues now in stock.

Articles include:

(a) a programme to print your own disk sleeve and catalogue.
(b) creating Wycove Forth menus
(c) a further article on GPL by Mack McCormick.
(d) Super 4th. (An upgrade)
(e) a comparison between GRAPHX and TI-ARTIST.
(f) a review of the Mechatronic 128K card.

plus User Notes and other articles.

I expect price to remain stable for the foreseeable future at $2.90 (+ postage if applicable).

CENTRAL COAST REGIONAL NEWS - our next meeting will be a big family fun day get together around the pool - weather permitting. It will be held on Saturday 13 December, commencing 2pm at 48 Monoa Road, Budgewoi. Club software will be available and it promises to be a great day so all you Central Coasters get there. For more details you can contact Russell Welham on (043)924000.

COFFS HARBOUR REGIONAL NEWS - A new group is in the process of being formed in the Coffs Harbour area. The contact for this Regional Group is Keir Wells, telephone (066)55.1487. So come on all you members who live in the area, give Keir a call and get a strong group going. Regional Groups are the best way to learn what your TI can do for you and conversely what you can do with it.

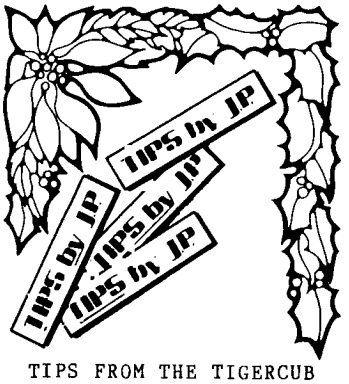
SST EXPANDED BASIC COMPILER

Have you ever wanted to speed up your own basic programmes? Perhaps you should be writing with the SST Compiler. There is only one copy of the expanded system available at $80.00. To run it you will require the Extended Basic Module and a TI Disk Controller Card.

NAVARONE INDUSTRIES PRODUCTS

Cartridge Expander ($43); Console Writer ($45) Paint & Print ($90) are available ex-stock.

Continued on Page 13



TIPS FROM THE TIGERCUB

#27 & #28

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Now, here is the new, and final, version of the Tigercub Menu Loader.

```
100 !by A. Kludge/M. Gordon/
T. Boisseau/J. Peterson/etc.
Version #5, 9/85
110 CALL PEEK(8198,A):: IF A
<>170 THEN CALL INIT
120 OPTION BASE 1 :: DIM PG$
(127),V(127,3):: CALL LOAD(-
31806,16):: ON ERROR 130 ::
GOTO 160
130 DISPLAY AT(12,9)ERASE AL
L:"I/O ERROR" :: RUN 100
140 @,@@,A,A$,B,C,D$,FLAG,I,
J,K,KD,KK,M,M$,N$,NN,P,P$,PG
$(),PP,PP$,Q$,S,ST,T$(),TT,V
T,V(,),W$,X,X$,Y,K2,S2
150 CALL LINK :: CALL PEEK :
: CALL KEY :: CALL SCREEN ::
 CALL COLOR :: CALL CLEAR ::
 CALL VCHAR :: CALL SOUND ::
 !@P-
160 CALL CLEAR :: CALL LOAD(
8196,63,248):: CALL LOAD(163
76,67,85,82,83,79,82,48,8)
170 CALL LOAD(12288,129,195,
126,165,129,153,102,60)
180 CALL LOAD(12296,2,0,3,24
0,2,1,48,0,2,2,0,8,4,32,32,3
6,4,91):: CALL LINK("CURSOR"
)
190 CALL CLEAR :: FOR S=1 TO
 14 :: CALL COLOR(S,7,16)::
NEXT S :: CALL COLOR(0,2,16)
200 T$(1)="d/f" :: T$(2)="d/
v" :: T$(3)="i/f" :: T$(4)="
i/v" :: T$(5)="pro" :: ON WA
RNING NEXT
210 IMAGE ###
220 IMAGE ###  Quit
230 IMAGE ###  Delete
240 IMAGE ###  Print
250 IMAGE ###  Rescan
260 CALL SCREEN(5):: CALL VC
HAR(1,31,1,96):: DISPLAY AT(
1,4):"TIGERCUB MENU LOADER"
270 ! IF YOU HAVE MORE THAN
ONE DISK DRIVE, DELETE THE !
 IN LINE 280 AND THE FIRST S
TATEMENT IN 290
280 ! DISPLAY AT(12,6):"DISK
? (1-3):" :: ACCEPT AT(12,19
)SIZE(-1)VALIDATE("123"):D$
:: D$="DSK"&D$&"."
290 D$="DSK1." :: OPEN #1:D$
,INPUT ,RELATIVE,INTERNAL ::
 INPUT #1:N$,A,J,K :: DISPLA
Y AT(1,2)SIZE(27):SEG$(D$,1,
4)&" - Diskname= "&N$;
300 DISPLAY AT(2,2):"Availab
le=";K;"Used=";J-K:" Prog Fi
lename Size  Type":"---- ---
------- ---- -------" :: I,V
T=0 :: TT=J-K
310 FOR X=1 TO 127 :: IF X/2
0<>INT(X/20)THEN 340
320 DISPLAY AT(24,1):"Choice
? Enter for more 0" :: ACCEP
T AT(24,24)VALIDATE(DIGIT)SI
ZE(-3):K :: IF K=0 THEN 330
:: IF K>0 AND K<NN+1 THEN 60
0 ELSE 320
330 X=1
340 I=I+1 :: IF I>127 THEN K
=X :: GOTO 430
350 INPUT #1:P$,A,J,B :: NN=
NN+1
360 IF LEN(P$)=0 THEN 430
370 DISPLAY AT(X+4,1):USING
210:NN :: DISPLAY AT(X+4,5):
P$ :: PG$(NN)=P$ :: DISPLAY
AT(X+4,16):USING 210:J :: DI
SPLAY AT(X+4,20):T$(ABS(A))
380 V(NN,1)=A :: V(NN,2)=ABS
(B):: V(NN,3)=J
390 X$="  "&STR$(B):: DISPLA
Y AT(X+4,24):SEG$(X$,LEN(X$)
-2,3):: VT=VT+J
400 IF A>0 THEN 410 :: DISPL
AY AT(X+4,28):"Y"
410 CALL KEY(0,KK,ST):: IF S
T=0 THEN 420 :: FLAG=1 :: GO
TO 430
420 NEXT X
430 DISPLAY AT(X+4,1):USING
220:NN :: DISPLAY AT(X+5,1):
USING 230:NN+1
440 IF VT=TT OR FLAG=1 THEN
460 :: DISPLAY AT(2,25)SIZE(
4):VT
450 FOR @=1 TO 10 :: DISPLAY
 AT(2,25)SIZE(1):CHR$(30)::
DISPLAY AT(2,25)SIZE(1):" "
:: CALL SOUND(-99,110,0,-4,0
):: NEXT @
460 IF FLAG=1 THEN 470 :: DI
SPLAY AT(X+4,13):USING 240:N
N+2 :: DISPLAY AT(X+5,13):US
ING 250:NN+3
470 DISPLAY AT(X+6,1):"    C
hoice?" :: ACCEPT AT(X+6,16)
SIZE(-3)VALIDATE(DIGIT):K
480 IF FLAG=1 THEN 500
490 IF K=NN+2 THEN 840 ELSE
IF K=NN+3 THEN CLOSE #1 :: N
N=0 :: GOTO 190
500 IF K<>NN AND K<>NN+1 THE
N 590
510 IF K=NN THEN CALL CLEAR
:: CLOSE #1 :: END
520 DISPLAY AT(X+5,12)SIZE(1
2):" #?" :: ACCEPT AT(X+5,15
)SIZE(3)VALIDATE(DIGIT):KD :
: IF KD<1 OR KD>NN THEN 520
530 IF V(KD,1)>0 THEN 550
540 FOR J=1 TO 10 :: DISPLAY
 AT(11,1):" ":"  PROTECTED -
 CANNOT DELETE":" " :: DISPL
AY AT(12,1):" " :: NEXT J ::
 GOTO 570
550 DISPLAY AT(X+6,1)SIZE(27
)BEEP:" Verify - Delete ";PG
$(KD);"?" :: DISPLAY AT(X+6,
28)SIZE(1):"Y" :: ACCEPT AT(
X+6,28)SIZE(-1)VALIDATE("YN"
):Q$ :: IF Q$<>"Y" THEN 570
560 DELETE D$&PG$(KD)
570 CLOSE #1
580 CALL VCHAR(1,3,32,672)::
 NN=0 :: X=0 :: FLAG=0 :: GO
TO 260
590 IF K<1 OR K>127 OR LEN(P
G$(K))=0 THEN 430
600 IF ABS(V(K,1))=5 OR ABS(
V(K,1))=4 AND V(K,2)=254 THE
N 640
610 DISPLAY AT(12,1)ERASE AL
L:"Print to ? S": :"(P)rinte
r?":"(S)creen?" :: ACCEPT AT
(12,12)SIZE(-1)VALIDATE("PS"
):Q$ :: IF Q$="S" THEN PP=0
:: GOTO 630
620 DISPLAY AT(12,1)ERASE AL
L:"PRINTER? PIO" :: ACCEPT A
T(12,10)SIZE(-18):P$ :: OPEN
 #3:P$ :: PP=3
630 CALL CLEAR :: CALL SCREE
N(16):: ON ABS(V(K,1))GOTO 6
80,690,750,760
640 CLOSE #1 :: IF SEG$(PG$(
K),LEN(PG$(K)),1)="*" THEN D
ISPLAY AT(12,1)ERASE ALL:"RE
TURN TO BASIC AND LOAD BY":"
TYPING OLD ";D$&PG$(K):: STO
P
650 CALL PEEK(-31952,A,B)::
CALL PEEK(A*256+B-65534,A,B)
:: C=A*256+B-65534 :: A$=D$&
PG$(K):: CALL LOAD(C,LEN(A$)
)
660 FOR I=1 TO LEN(A$):: CAL
L LOAD(C+I,ASC(SEG$(A$,I,1))
):: NEXT I :: CALL LOAD(C+I,
0)
670 CALL VCHAR(1,3,32,672)::
 CALL SCREEN(8):: FOR S=0 TO
 14 :: CALL COLOR(S,2,1):: N
EXT S :: DISPLAY AT(12,2):"L
OADING ";A$ :: GOTO 900
680 OPEN #2:D$&PG$(K),INPUT
,FIXED :: GOTO 700
690 OPEN #2:D$&PG$(K),INPUT
700 LINPUT #2:W$ :: PRINT #P
P:W$ :: IF EOF(2)THEN 730
710 CALL KEY(0,K,S):: IF S=0
 THEN 700
720 CALL KEY(0,K2,S2):: IF S
2<1 THEN 720 ELSE 700
730 CLOSE #1 :: CLOSE #2 ::
PRINT "   >>>press any key<<
<" :: IF Q$="P" THEN CLOSE #
3
740 CALL KEY(0,K,ST):: IF ST
<1 THEN 740 ELSE 580
750 OPEN #2:D$&PG$(K),INPUT
,INTERNAL,FIXED :: J=0 :: GO
TO 770
760 OPEN #2:D$&PG$(K),INPUT
,INTERNAL :: J=0
770 IF EOF(2)=1 THEN 730 ::
J=J+1 :: INPUT #2:M$ :: IF L
EN(M$)=8 THEN 790
780 PRINT #PP:M$ :: GOTO 820
790 FOR Y=1 TO 8 :: @@=ASC(S
EG$(M$,Y,1)):: IF @@<32 OR @
@>127 THEN 810
800 NEXT Y :: GOTO 780
810 RESTORE #2 :: FOR X=1 TO
 J-1 :: INPUT #2:M$ :: NEXT
X :: INPUT #2:M :: PRINT #PP
:M
820 CALL KEY(0,K,S):: IF S=0
 THEN 770
```

```
830 CALL KEY(0,K2,S2):: IF S
2<1 THEN 830 ELSE 770
840 DISPLAY AT(24,1):"PRINTE
R NAME? PIO" :: ACCEPT AT(24
,15)SIZE(-14):PP$ :: OPEN #2
:PP$ :: PRINT #2:SEG$(D$,1,4
)&" - Diskname= "&N$
850 PRINT #2:RPT$("*",28):"A
vailable=";358-VT;"Used=";VT
:RPT$("-",28)
860 PRINT #2:"FILENAME  SIZE
   TYPE":RPT$("-",28)
870 FOR P=1 TO NN-1 :: PRINT
 #2:PG$(P);TAB(15);V(P,3);TA
B(20);T$(ABS(V(P,1)));TAB(25
);V(P,2):: NEXT P :: CLOSE #
2
880 DISPLAY AT(12,3)ERASE AL
L:"(P) to print again":"  (R
) to rescan":"  (Q) to quit"
890 ACCEPT AT(15,4)VALIDATE(
"PQR")SIZE(-1)BEEP:Q$ :: IF
Q$="P" THEN 840 :: CLOSE #1
:: NN=0 :: IF Q$="R" THEN 19
0 ELSE END
900 RUN "DSKX.1234567890"
```

This version turns off the Quit key, restarts itself rather than crashing on an I/0 error, and has pre-scan for faster start-up. It displays disk name, sectors available and sectors presumably used - it also totals up actual sectors used and sounds a warning if any sectors are not accounted for.

It lists up to 127 programs and files by number, filename, number of sectors, program or file type, file record length, and write-protection. It will stop for menu selection on any keypress or at the end of each screen, continuing on Enter. It will load and run any program that can run from Extended Basic, displaying its filename while loading. If the filename ends in an asterisk, it will warn you to return to Basic. It will delete any unprotected program or file, after first requiring verification by filename, or will inform you if the file is protected. It will read any readable file, including internal numeric, and list it to screen or printer. It will dump a catalog of the disk to your printer, and it will offer the option of quitting or rescanning the disk or another disk. And it's free, I don't even want a freeware donation - but I would appreciate if you would take a look at my catalog and see if, somewhere among those 140 programs, there might be something you would be willing to pay $3 for? The Menu Loader is included as a bonus on every disk I sell!

```
100 CALL CLEAR :: RANDOMIZE
:: DISPLAY AT(3,4):"TIGERCUB
 MATH PUZZLE"
110 DISPLAY AT(6,1):"Insert
+, -, * (multiply) OR / (div
ide) between the  digits
to equal the total": :"Type
Q to give up"
120 DISPLAY AT(12,1):"Level
1 or 2?" :: ACCEPT AT(12,15)
VALIDATE("12"):L$
130 T,X=INT(9*RND+1):: M$=ST
R$(X):: Z$=M$&" "
140 FOR J=1 TO 4 :: Y(J)=INT
(9*RND+1):: Z=INT(4*RND+1)::
 ON Z GOSUB 240,250,260,270
:: Z$=Z$&STR$(Y(J))&" " :: N
EXT J
150 IF L$="1" AND T<>INT(T)T
HEN 130 :: Z$=Z$&"="&STR$(T)
160 DISPLAY AT(12,1):Z$ :: D
ISPLAY AT(18,1):" " :: DISPL
AY AT(20,1):" " :: DISPLAY A
T(22,1):" "
170 P=2 :: FOR J=1 TO 4 :: A
CCEPT AT(12,P)VALIDATE("Q+-*
/")SIZE(1):S$
180 IF S$="Q" THEN 200 ELSE
IF S$="+" THEN X=X+Y(J)ELSE
IF S$="-" THEN X=X-Y(J)ELSE
IF S$="*" THEN X=X*Y(J)ELSE
X=X/Y(J)
190 P=P+2 :: NEXT J :: IF X=
T THEN 230 :: DISPLAY AT(18,
1):"WRONG!"
200 DISPLAY AT(20,1):"ANSWER
 IS ";M$
210 DISPLAY AT(22,1):"PRESS
ANY KEY"
220 CALL KEY(0,K,ST):: IF ST
<1 THEN 220 :: GOTO 130
230 DISPLAY AT(18,1):"RIGHT!
" :: GOTO 210
240 M$=M$&"+"&STR$(Y(J)):: T
=T+Y(J):: RETURN
250 M$=M$&"-"&STR$(Y(J)):: T
=T-Y(J):: RETURN
260 M$=M$&"*"&STR$(Y(J)):: T
=T*Y(J):: RETURN
270 M$=M$&"/"&STR$(Y(J)):: T
=T/Y(J):: RETURN
```

NUTS & BOLTS DISK No. 2 is now ready, and I think it's better than the first one. It contains 108 utility subprograms in merge format, including many new character fonts and screen display routines as well as 2-dimensional array sorts, variable line numbers in GOSUB, GOTO and RESTORE, on-screen editing and much, much more. The price is $19.95 postpaid, or you can order both Nuts & Bolts disks for $37 ppd (PLEASE NOTE: U.S.prices).

And I have put together 18 different collection disks each containing 5 or 6 of my catalog programs for just $12 postpaid. The programs on each disk are all of the same category, and I have filled up the rest of the disk with public domain programs of the same category, as a bonus.
I want to make it very plain that I am NOT - repeat, NOT - selling public domain programs!

My own programs on these disks are offered at a great discount and the public domain programs are just thrown in for free! Together with this issue of the Tips I have mailed to your user's group a copy of my catalog #6 with an added page describing these new offerings, and a rebate offer to user's groups.

My catalog will be sent to individuals for $1, which is deductable from your first order. If you already have my catalog #6, the added page will be sent to you free on request.

And so, on to old business. Yes, I know that RESequencing a program does not resequence references to line numbers in REMs. I just forgot! In line 270 of the Menu Loader in Tips #27, the reference should be to lines 280 and 290, of course.
While programming the file reader in that menu loader, I ran into a peculiarity of the TI-99/4A that surprised most of the expert programmers whom I called for help. When you "read blind" you must read everything as a string, because attempting to read a string as numeric will crash the program. This is no problem with DISPLAY files - but when I tried it with INTERNAL files, I got the strangest garbage! My solution (not quite fool-proof) was to identify a record as numeric if it was 8 bytes long and contained an ASCII out of printable range, and then RESTORE the file, read back to that point and re-read it as numeric. Not very efficient!
The following routine will save a numeric input in an internal file, read it back out as a string, show you the way it was saved, and then attempt to translate it back to numeric. It works for positive and negative integers or non-integers of not less than -99, but not for less than that.

```
100 INPUT X :: OPEN #1:"DSK1
.TEST",INTERNAL,OUTPUT :: PR
INT #1:X :: CLOSE #1
110 OPEN #1:"DSK1.TEST",INTE
RNAL,INPUT :: INPUT #1:A$ ::
 PRINT A$ :: CLOSE #1
120 FOR J=1 TO 8 :: PRINT AS
C(SEG$(A$,J,1));:: NEXT J
130 FOR J=1 TO 8 :: A(J)=ASC
(SEG$(A$,J,1)):: NEXT J
140 X=A(1)-63 :: IF X<73 THE
N 150
142 X=192-A(1):: N$="-" :: F
OR J=2 TO X+1 :: N$=N$&STR$(
256-A(J)):: NEXT J :: GOTO 1
60
```

```
150 FOR J=2 TO X+1 :: N$=N$&
STR$(A(J)):: NEXT J
160 IF A(J)<>0 THEN N$=N$&".
"&STR$(A(J))
170 J=J+1 :: IF A(J)<>0 THEN
 N$=N$&STR$(A(J)):: GOTO 170
180 N=VAL(N$):: N$="" :: PRI
NT N :: GOTO 100
```

So, here is another Tigercub Challenge! Can you fix it? Let's HEAR from you this time!

Another problem that I ran into was in recovering from an I/O error. When ON ERROR is used to prevent crashing on such an error, the file is "ajar" - you can't close it and you can't open it. My solution was to simply RUN the program again - and this will show you how the pre-scan speeds that up. Since then, I have learned of three other ways. The method described in the TIsHUG (Australia) NEWS-DIGEST is a bit complicated, but Irwin Hott gave me a simple solution - just increment the file number! Works fine if you don't increment it into the number of another open file on the disk. Chuck Grimes gave me an even better way - open and close anything else, even "PIO"! Example -

```
100 ON ERROR 110 :: OPEN #1:
"DSK1.TEST",OUTPUT :: ON ERR
OR STOP :: PRINT "CONTINUE P
ROGRAM" :: END
110 OPEN #1:"PIO" :: CLOSE #
1 :: PRINT "I/O ERROR":"CHEC
K DISK AND DRIVE":"THEN PRES
S ANY KEY"
120 CALL KEY(0,K,S):: IF S=0
 THEN 120 ELSE 100
```

There is a reason for that ON ERROR STOP, and it's why I don't use ON ERROR if I can avoid it. When an error occurs, the program goes to the line number specified by the last open ON ERROR statement, takes whatever action is directed by that line, and RETURNs as directed. If the error was not one that you expected to happpen, the results can be very confusing!

For that reason, when you set out to modify a program, the first thing you should do is delete, temporarily, all the ON ERROR statements. The next thing you should do, if the program has a routine to turn off the pre-scan, is to disable that. Otherwise, you will be driven crazy by invalid SYNTAX ERROR messages and other strange happenings.

The third thing you should do is to make a list of all the lines that a GOTO or GOSUB goes to, so you don't delete or change them. And here is a program to do just that for you -

```
100 !GO-SEARCH by Jim Peters
on searches a MERGE format f
ile, finds all line numbers
containing a jump, sorts int
o "to" line number sequence,
110 !prints "to" line number
, statement (GO, GOTO or GOS
UB) and "from" line number
120 DIM C(200):: A=1 :: GO$(
1)="GO" :: GO$(2)="GOTO" ::
GO$(3)="GOSUB"
130 INPUT "FILENAME? DSK1.":
F$
140 OPEN #1:"DSK1."&F$,INPUT
 ,VARIABLE 163 :: OPEN #2:"P
IO"
150 LINPUT #1:A$
160 IF POS(A$,CHR$(133),1)=0
 AND POS(A$,CHR$(134),1)=0 A
ND POS(A$,CHR$(135),1)=0 THE
N 210
170 LN=ASC(SEG$(A$,1,1))*256
+ASC(SEG$(A$,2,1)):: T=133 :
: P=1
180 G$=CHR$(T):: X=POS(A$,G$
,P):: IF X=0 THEN 200 :: LRE
F=ASC(SEG$(A$,X+2,1))*256+AS
C(SEG$(A$,X+3,1))!:: PRINT #
2:LN;GO$(T-132);LREF :: P=X+
1 :: GOTO 180
190 C$=STR$(LREF)&"."&STR$(L
N)&STR$(T-132):: C(A)=VAL(C$
):: A=A+1 :: P=X+1 :: GOTO 1
80
200 IF G$=CHR$(135)THEN 210
:: T=T+1 :: P=1 :: GOTO 180
210 IF EOF(1)THEN CLOSE #1 :
: GOTO 220 :: ELSE 150
220 A=A-1 :: CALL LONGSHELLN
(A,C())
230 FOR J=1 TO A :: A$=STR$(
C(J)):: X=POS(A$,".",1):: Y=
VAL(SEG$(A$,LEN(A$),1)):: A$
=SEG$(A$,1,LEN(A$)-1)
240 PRINT #2:SEG$(A$,1,X-1);
TAB(7);GO$(Y);" FROM ";TAB(2
1);SEG$(A$,X+1,LEN(A$)):: NE
XT J
250 SUB LONGSHELLN(N,NN())
260 D=N
270 D=INT(D/3)+1 :: FOR I=1
TO N-D :: IF NN(I)<=NN(I+D)T
HEN 300 :: T=NN(I+D):: J=I
280 NN(J+D)=NN(J):: J=J-D ::
 IF J<1 THEN 290 :: IF T<NN(
J)THEN 280
290 NN(J+D)=T
300 NEXT I
310 IF D>1 THEN 270
320 SUBEND
```

According to the User's Reference Guide that came with your computer, if you open a file without specifying INPUT, OUTPUT, UPDATE or APPEND, the computer will assume the UPDATE mode as the default and "UPDATE files may be both read and written. The usual processing is to read a record, change it in some way, and then write the altered record back out on the file." This is a very dangerous bit of misinformation! It is true only if you are using RELATIVE files with the REC clause. In any other case, the first record you write to the file will become the record FOLLOWING the last record you read, and it will also become the LAST record in the file - any records beyond that point will be lost!

The moral of the story - get in the habit of NEVER opening a file without specifying the mode. The only way to update a sequential file is to read it ALL into an array, update it, and then write it back to the file.

I reviewed hundreds of programs, in my PD library of about 2600, in order to select some of the best to fill up the collection disks. Often they needed only a few minor changes to greatly improve them.

One frequent flaw was in interpreting the status of CALL KEY. The User's Reference Guide says that a status variable of -1 means that "the same key was pressed during the performance of CALL KEY as was pressed during the previous performance." This is misleading. It actually means that the same key is STILL BEING pressed. Try this -

```
100 DISPLAY AT(12,1)ERASE AL
L:"TYPE YOUR NAME" :: R=14 :
: C=3
110 CALL KEY(0,K,S):: IF S=0
 THEN 110 :: DISPLAY AT(R,C)
:CHR$(K):: C=C+1 :: GOTO 110
```

Difficult to type without unwanted repetition of letters? Now try changing the S=0 to S<1 !

IF S<1 (if S is less than 1) means that if no key is pressed (S=0) or if the same key is still being held down (S=-1) then CALL KEY again.

Another frequent flaw is

```
INPUT "WANT TO PLAY AGAIN? "
:Q$ :: IF Q$<>"Y" THEN END
```

- or, more professionally programmed, IF SEG$(Q$,1,1)<>"Y" THEN....,

which will accept either "Y" or "YES" as a reply. The problem is still that this question is often asked at the end of a joystick game, for which the Alpha Lock will be unlocked - and a response of a lower case "y" then terminates the program! One solution is to precede the INPUT with a dummy CALL KEY(3,K,S), which will cause any subsequent upper case CALL KEY, INPUT, LINPUT or ACCEPT AT response to be read as lower case until you turn it off with CALL KEY(5,K,S).

Here's one that does nothing except look pretty.

```
100 DISPLAY AT(3,8)ERASE ALL
:"COLORSQUARES" :: DISPLAY A
T(8,1):"Select option 1, 2 o
r 3" ! by Jim Peterson, Tige
rcub Software
```

```
110 CALL KEY(0,K,ST):: IF ST
=0 OR K<49 OR K>51 THEN 110
:: ON K-48 GOTO 150,120,130
120 FOR CH=38 TO 142 STEP 8
:: CALL CHAR(CH,RPT$("A55A",
4)):: NEXT CH :: GOTO 150
130 FOR CH=38 TO 142 STEP 8
:: FOR L=1 TO 4 :: RANDOMIZE
 :: X$=SEG$("0018243C425A667
E8199A5BDC3DBE7FF",INT(16*RN
D+1)*2-1,2)
140 B$=B$&X$ :: C$=X$&C$ ::
NEXT L :: CALL CHAR(CH,B$&C$
):: B$,C$=NUL$ :: NEXT CH
150 CALL CLEAR :: RANDOMIZE
:: FOR SET=0-(K>49)TO 14 ::
CALL COLOR(SET,SET+2+(K>49),
SET+2):: NEXT SET
160 Y=INT(4*RND+3):: R=INT(1
2*RND+1):: R2=25-R-Y :: C=IN
T(7*RND+7):: C2=32-C-Y :: IF
 K=49 THEN X=INT(14*RND+1)*8
+22 ELSE X=INT(13*RND+1)*8+3
0
170 FOR T=R TO R+Y :: CALL H
CHAR(T,C,X,Y):: CALL HCHAR(T
,C2,X,Y):: NEXT T
180 FOR T=R2 TO R2+Y :: CALL
 HCHAR(T,C,X,Y):: CALL HCHAR
(T,C2,X,Y):: NEXT T :: GOTO
160
```

The asterisk on the Gemini printer looks rather like a bug squashed side- ways, and it was confusing some folks in the condensed print of my newsletter, so I improved it with this -

```
150 PRINT #2:CHR$(27);CHR$(4
2);CHR$(1);CHR$(42);CHR$(0);
CHR$(8);CHR$(34);CHR$(8);CHR
$(0);CHR$(62);CHR$(0);CHR$(8
);CHR$(34);CHR$(8);
```

And at the same time I improved the slashed zero -

```
140 PRINT #2:CHR$(27);CHR$(4
2);CHR$(1);CHR$(48);CHR$(0);
CHR$(64);CHR$(30);CHR$(96);C
HR$(17);CHR$(72);CHR$(5);CHR
$(66);CHR$(61);CHR$(0);
```

```
90 !THIS WON'T WORK, WILL IT
?
100 DISPLAY AT(9999,9999)ERA
SE ALL:SEG$("CAN'T DO THAT!"
,1,3)&SEG$("CAN'T DO THAT!",
6,8)
```

If the Tigercub Math Puzzle was a bit too tough, these changes will add a couple of easier levels...

```
105 DISPLAY AT(6,1):"Level 1
, 2, 3 or 4?" :: ACCEPT AT(6
,21)VALIDATE("1234"):L$ :: L
=VAL(L$)
106 IF L<3 THEN M$="Insert +
, -, or * (multiply)" ELSE M
$="Insert +, -, * (multiply)
 or / (divide)"
110 DISPLAY AT(5,1):M$;" bet
ween the digits":" to equal
the total": :"Type Q to give
 up"
120 ! **DELETED LINE **
130 DISPLAY AT(12,1):" " ::
T,X=INT(9*RND+1):: M$=STR$(X
):: Z$=M$&" "
140 FOR J=1 TO 4 :: Y(J)=INT
(9*RND+1):: @=3+ABS(L>2):: Z
=INT(@*RND+1):: ON Z GOSUB 2
40,250,260,270 :: Z$=Z$&STR$
(Y(J))&" " :: NEXT J
150 IF L/2<>INT(L/2)AND T<>I
NT(T)THEN 130 :: Z$=Z$&"="&S
TR$(T)
```

Have a Happy & Safe Christmas
See you in the February issue of the TIsHUG NEWS DIGEST.

Jim Peterson

Merry Christmas and a Happy New Year from the Committee of TIsHUG

Continued from Page 9

PRINTERS

With Christmas just around the corner now is a good time to purchase that printer. We are offering the EPSON LX-86 printer, complete with tractor feed at the unbeatable price of $526.00. (For the curious, suggested retail for the printer alone is $559 + sales tax.) These units come with a Roman Type NLQ mode - 12 month warranty - and are compatible with the TI RS232 interface.

If you need a Daisy Wheel printer I suggest the EPSON RV-90. This is a 20 CPS machine complete with centronics, RS232 and IEEE interface again available at an unbeatable price.

PAPER SUPPLIES

By arrangement with our supplier, we are able to offer one of the most extensive ranges of paper products - weight; form length; form width; colour; even custom designed continuous stationery if you need it; labels etc., - and our prices INCLUDE DELIVERY in capital city metro. areas. (Prices are now around $5.00 / box dearer than when last published.)

REMINDER

The shop is there to serve your needs so let me know what you are interested in: Hardware or Software.

TECHO TIME

Due to the enthusiastic response to the first 'Poormans Disk System' article in the September TND, all of the TI Disk Controller Cards have sold ,and some Disk systems have already been constructed and are running. Some are still under construction and for these in particular I will supply further information on the improvements I have made to the design of the System which will now be referred to as the Mini-PE system.

If you do not have a disk controller card then there is no point starting this project until you can get one. Also the TISHUG Shop is still negotiating on the purchase of TI PE Boxes from the U.S.of A, and if that does'nt work perhaps I will eventually find enough time to complete design on my own controller card.

First here is the circuit of the Power supply used on the Mini-PE Board. It provides power to the 60 way sockets for the controller card and also a negative supply for cards that require it. The main board also uses +5 volt regulated for the pull-up resistors on some pins. These resistors are all 56 or 47 Ohm except one of 10K. There are 16 resistors all-up. This board has the 2 bridge rectifiers mounted flat on the PCB and even though these are rated at 6 amp they require heatsinking if used to supply more than one disk drive and this is not possible on this PCB unless the rectifiers are bolted to a heatsink or your box, with leads run from the PCB. Then you can run 2 drives. This is the limit of the regulators and transformer used. If you are making this system for a single disk drive (or 2 genuine half power drives) then construction is fairly simple with all parts mounting on the main PCB.

If you wish to run more than one drive then I recomend leaving out the power supply components altogether and constructing a separate power board. I have designed a Switch-mode Power Board which will supply up to 4 Amp with adequate cooling and this PCB is called the Power board MPE 3A, available from me. It uses a specially wound air-core inductor which I can supply, or you can make one yourself. The cost of parts for this is is about $20 , however some parts are saved from the main PCB power components. If separate power is used then +5V must be connected to the board for the pull-up resistors, and the unregulated voltages to supply the 60 way sockets.

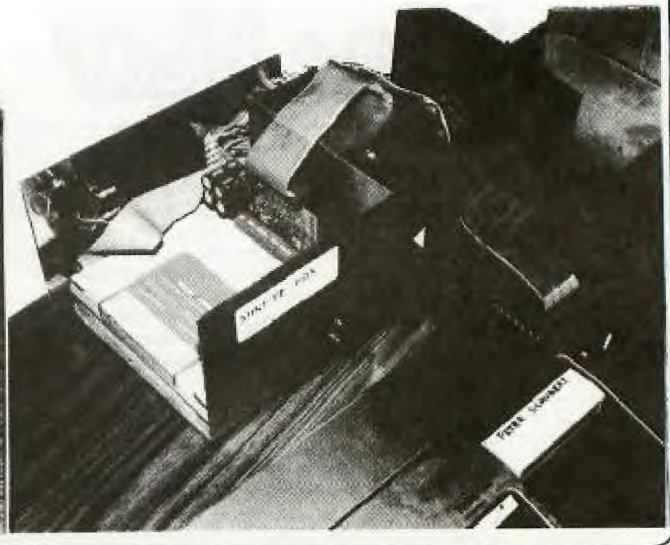
The Mini-PE board mounts vertically so that the ribbon connector is at the top and the regulators at the bottom can bolt to the metal base or heatsink bracket. I used 12mm long brass spacers to mount the board to an aluminium plate which is bent at right angle to sit on the bottom of your box, and I mounted the regulators to this. The disk controller card sits flat almost on the bottom of the box when plugged into the bottom 60 way socket. I had to space mine up about 6mm so I used the screws that mount the rubber feet as they were just the right height to keep the card level. If a card is plugged into the second socket it will lay flat on top of the disk card.

A 40 way flex cable is used to connect to the console. On the PE box end it is plugged straight onto the edge connector. The console end can be connected to a 44 way connector by not using the end 4 pins(2 on each end) however pin 44 must be connected to the ribbon wire that would normally go to 39.

If you require a connection to other peripherals such as the speech or RS232/Modem box, etc, then you will need the special PC board I made titled MINI-PE BOARD/2. It has a 44 way to 44 way straight thru, and also a 40 way for the ribbon cable. If, however, any of you have a Parallax printer interface this too can be used to plug onto (see me for details).

The ribbon cable can be soldered to the boards at both ends if you wish, or you can use crimp-on 40 way conectors as I did, and plug onto the PC boards. Regarding the ribbon cable itself, it must be shielded in some way otherwise it will not work over a small length. I shielded it by wrapping in alfoil and then taping it with PVC tape to seal the foil on and protect it from damage. A short length of bare wire was inserted under the foil before taping it, and the end was connected to a ground pin (21,23,25 or 27 on console end. remember the 40 way starts from 3 and 4 and finishes on 41,42. Also the 40 way on the 3 way console PC board is inverted with odds on top and evens on bottom. This allows the same connectors on each end of ribbon both same way up.

→




```
100 REM *************
110 REM *FREQUENCIES*
120 REM *************
130 REM BY P AXEL-LUTE
140 REM IN TI BASIC
150 PREV=-32
160 DIM FDIG(26,26),CH$(26),
DF(26),IND(26),FLAG(26),FREQ
(26),FR$(12),DI$(12)
170 CALL CLEAR
180 PRINT "      FREQUENCY C
OUNT":"BY P. AXEL-LUTE, S.OR
ANGE NJ": : :
190 PRINT "GIVES FREQUENCIES
 OF ALL":"LETTERS AND 12 MOS
T FREQUENTDIGRAMS. LETTERS S
EPARATED  BY SPACES ARE NOT
COUNTED ASA DIGRAM. CHARACTE
RS OTHER"
200 PRINT "THAN LETTERS ARE
IGNORED.   ENTER TEXT-UPPER
CASE ONLY  UP TO 4 SCREEN LI
NES ATE A  TIME. MAXIMUM OF
2000 CHAR- ACTERS. ENTER A '
*' AT INPUT"
210 PRINT "PROMPT TO SIGNAL
END OF TEXT": : : : :
220 SN=1
230 S$(SN)=""
240 INPUT A$
250 IF A$="*" THEN 340
260 IF TL+LEN(A$)>2000 THEN
960
270 IF LEN(S$(SN))+LEN(A$)>2
55 THEN 300
280 S$(SN)=S$(SN)&A$
290 GOTO 240
300 TL=TL+LEN(S$(SN))
310 SN=SN+1
320 S$(SN)=A$
330 GOTO 240
340 FOR NS=1 TO SN
350 FOR N=1 TO LEN(S$(NS))
360 L$=SEG$(S$(NS),N,1)
370 L=ASC(L$)-64
380 IF L<1 THEN 450
390 IF L>26 THEN 450
400 FREQ(L)=FREQ(L)+1
410 TOT=TOT+1
420 IF PREV<1 THEN 450
430 IF PREV>26 THEN 450
440 FDIG(PREV,L)=FDIG(PREV,L
)+1
450 PREV=L
460 NEXT N
470 NEXT NS
480 FOR J=1 TO 26
490 MAX=-1
500 FOR I=1 TO 26
510 IF FLAG(I)=1 THEN 550
520 IF FREQ(I)<=MAX THEN 550
530 MAX=FREQ(I)
540 IND(J)=I
550 NEXT I
560 FLAG(IND(J))=1
570 NEXT J
580 FOR I=1 TO 26
590 CH$(I)=CHR$(IND(I)+64)
600 DF(I)=FREQ(IND(I))
610 NEXT I
620 FOR I=1 TO 8
630 PRINT TAB(4);CH$(I);DF(I
);TAB(12);CH$(I+9);DF(I+9);T
AB(20);CH$(I+18);DF(I+18)
640 NEXT I
650 PRINT TAB(4);CH$(9);DF(9
);TAB(12);CH$(18);DF(18);TAB
(18);"TOT";TOT
660 K=1
670 MAX=-1
680 FOR I=1 TO 26
690 IF FREQ(I)<=1 THEN 770
700 FOR J=1 TO 26
710 IF FREQ(J)<=1 THEN 760
720 IF FDIG(I,J)<=MAX THEN 7
60
730 MAX=FDIG(I,J)
740 MI=I
750 MJ=J
760 NEXT J
770 NEXT I
780 IF MAX<=1 THEN 910
790 FR$(K)=STR$(MAX)
800 B$=CHR$(MI+64)
810 E$=CHR$(MJ+64)
820 DI$(K)=B$&E$
830 FDIG(MI,MJ)=-2
840 IF K=12 THEN 910
850 K=K+1
860 GOTO 670
870 FOR M=K TO 9
880 DI$(M)=" "
890 FR(M)=1
900 NEXT M
910 PRINT
920 FOR K=1 TO 4
930 PRINT TAB(4);DI$(K);" ";
FR$(K);TAB(12);DI$(K+4);" ";
FR$(K+4);TAB(20);DI$(K+8);"
";FR$(K+8)
940 NEXT K
950 END
960 PRINT "TEXT OVER 2000 CH
ARACTERS   RE-ENTER NO MORE
THAN";2000-TL:"CHARACTERS OF
 LAST PORTION"
970 GOTO 240
```

```
100 REM **********
110 REM *STEINWAY*
120 REM **********
130 REM M CHRISTIANSON
140 REM AND J BOWMAN
150 REM IN TI BASIC
160 CALL CLEAR
170 CALL SCREEN(15)
180 CALL CHAR(65,"FF81818181
818181")
190 CALL CHAR(66,"8181818181
818181")
200 CALL CHAR(67,"8181818181
8181FF")
210 CALL CHAR(80,"FF")
220 CALL CHAR(81,"FF81818181
818181")
230 CALL CHAR(133,"040605061
D3C3C18")
240 CALL COLOR(5,2,16)
250 CALL COLOR(8,2,2)
260 PRINT "   QPPPPPPPPPPPPP
PPPPPPPQ"
270 PRINT "   Q&  steinway 9
9/4a  &Q"
280 PRINT "   7777777777777
ZZZZZZZZ"
290 PRINT "   ZAAAAAAAAAAAAA
AAAAAAAZ"
300 PRINT "   ZBBBBBBBBBBBBB
BBBBBBBZ"
310 PRINT "   ZCCCCCCCCCCCCC
CCCCCCCZ"
320 PRINT "   Zabcdefgabcdef
gabcdefZ"
330 PRINT "   ZAAAAAAAAAAAAA
AAAAAAAZ"
340 PRINT "   ZBBBBBBBBBBBBB
BBBBBBBZ"
350 PRINT "   ZCCCCCCCCCCCCC
CCCCCCCZ"
360 CALL COLOR(5,2,16)
370 OPTION BASE 0
380 DIM NO(20),N(20)
390 FOR A=0 TO 20
400 READ NO(A),N(A)
410 NEXT A
420 DATA 40000,30,247,2,349,
6,392,7,440,8,659,12,698,13,
784,14,1047,17,1175,18,1319,
19,1397,20,880,15,494,9
430 DATA 294,4,262,3,220,1,3
30,5,523,10,587,11,988,16
440 CALL KEY(1,K1,S)
450 CALL KEY(2,K2,S)
460 K1=K1+1
470 K2=K2+1
480 CALL HCHAR(19,7,67,20)
490 CALL HCHAR(23,7,67,20)
500 CALL COLOR(5,2,16)
510 CALL SOUND(-4250,NO(K1)*
.5,0,NO(K2),0,N(K1)+200,30)
520 GOSUB 540
530 GOTO 440
540 IF K1=0 THEN 560
550 CALL HCHAR(23,N(K1)+6,13
3)
560 IF K2=0 THEN 630
570 CALL HCHAR(19,N(K2)+6,13
3)
580 RANDOMIZE
590 A=INT(RND*12)+1
600 B=INT(RND*32)+1
610 CALL HCHAR(A,B,133)
620 CALL COLOR(13,7,1)
630 RETURN
```

```
24900 !TI-WRITER BIT-IMAGE
 PRINTER GRAPHICS
  by Richard Mitchell in
  Super 99 Monthly V.1 #9
24902 !with corrections and
enhancements from V.1 #10
24910 !save in MERGE format,
 merge into any program  -
 will dump any screen of tex
t and/or graphics into a D/V
 80 disk file which can be
24920 !printed thru the TI-W
RITER Formatter mode on any
Epson/Gemini type printer.
24930 !at the point in the p
rogram where screen is to be
 dumped, insert a line in th
e form CALL DUMP_TIW(BR,ER,F
$,DE,I$,T)
24940 !BR is the beginning s
creen row(1 to 24) to be dum
ped, ER (1-24) is the ending
 screen row, F$ is the filen
ame to be used, such as
24950 !"DSK1.FLOWERDUMP", DE
 is the density (1 or 2) to
use, I$ is the inverse optio
n ("I" for inverse, any othe
r valid string for normal)
24960 !and T is the tab valu
e (18 will center the printo
ut on most printers)
24970 !when printing the fil
e from the Formatter, you
MUST MUST use a CR designati
on, such as "PIO.CR"
24971 !if you have an Epson
or TI printer, change line 2
5070 to a REM and remove the
 ! from line 25072
```

```
24975 !for a compressed (hal
f-width) dump, use .5 as the
 DE parameter, 29 as the tab
 to center. Change 25070 to
a REM, delete the ! from
24976 !line 25073 (or change
 the end of 25072 for Epson)
, remove ! from 25032, 25035
 and 25075, make 25030 a REM
25000 SUB DUMP_TIW(BR,ER,F$,
DE,I$,T)
25010 ON ERROR 25220
25020 IF (T<0)+(T>40)+(BR>ER
)+(BR<1)+(BR>24)+(ER<1)+(ER>
24)THEN GOSUB 25220
25030 IF DE<>2 THEN DE=1
25032 !IF (DE<>2)*(DE<>.5)TH
EN DE=1
25035 !R=DE :: IF DE=.5 THEN
 DE=2
25040 OPEN #1:F$,DISPLAY ,VA
RIABLE 80,OUTPUT :: B$="0123
456789ABCDEF" :: C$=" *.^&@"
&CHR$(27)&CHR$(10)&CHR$(13):
: D$=""
25050 FOR I=2 TO 122 :: IF P
OS(C$,CHR$(I),1)=0 THEN D$=D
$&CHR$(I)
25060 NEXT I
25070 PRINT #1:".TL 1:27,65,
8,10,13,27,108,"&STR$(T)&",2
7,"&SEG$("7576",DE*2-1,2)&",
0,"&STR$(DE)
25072 !PRINT #1:".TL 1:27,65
,8,10,13,27,68,"&STR$(T)&",0
,9,27,"&SEG$("7576",DE*2-1,2
)&",0,"&STR$(DE)
25073 !PRINT #1:".TL 1:27,65
,8,10,13,27,108,"&STR$(T)&",
27,"&SEG$("7576",DE*2-1,2)&"
,0,"&STR$(MIN(DE,R*2))
25075 !IF R=.5 THEN DE=1
25080 FOR I=32 TO 143 :: CAL
L CHARPAT(I,H$)
25090 C1,C2,C3,C4,C5,C6,C7,C
8=0 :: FOR P=1 TO 15 STEP 2
:: X=POS(B$,SEG$(H$,P,1),1)-
1 :: Y=POS(B$,SEG$(H$,P+1,1)
,1)-1 :: Z=2^((15-P)/2)
25100 C1=C1+Z*SGN(X AND 8)::
 C2=C2+Z*SGN(X AND 4):: C3=C
3+Z*SGN(X AND 2):: C4=C4+Z*S
GN(X AND 1)
25110 C5=C5+Z*SGN(Y AND 8)::
 C6=C6+Z*SGN(Y AND 4):: C7=C
7+Z*SGN(Y AND 2):: C8=C8+Z*S
GN(Y AND 1):: NEXT P
25120 IF I$="I" THEN C1=255-
C1 :: C2=255-C2 :: C3=255-C3
 :: C4=255-C4 :: C5=255-C5 :
: C6=255-C6 :: C7=255-C7 ::
C8=255-C8
25130 A$=".TL "&STR$(ASC(SEG
$(D$,I-31,1)))&":"&RPT$(STR$
(C1)&",",DE)&RPT$(STR$(C2)&"
,",DE)&RPT$(STR$(C3)&",",DE)
&RPT$(STR$(C4)&",",DE)
25140 A$=A$&RPT$(STR$(C5)&",
",DE)&RPT$(STR$(C6)&",",DE)&
RPT$(STR$(C7)&",",DE)&RPT$(S
TR$(C8)&",",DE):: A$=SEG$(A$
,1,LEN(A$)-1):: PRINT #1:A$
25150 NEXT I
25160 FOR I=BR TO ER :: A$=C
HR$(1):: FOR J=1 TO 32
25170 CALL GCHAR(I,J,C):: C=
MIN(MAX(C,32),143):: A$=A$&S
EG$(D$,C-31,1):: NEXT J :: P
RINT #1:A$ :: NEXT I
25180 PRINT #1:".TL 1:1"
25190 FOR I=2 TO 122 :: IF P
OS(C$,CHR$(I),1)=0 THEN PRIN
T #1:".TL "&STR$(I)&":"&STR$
(I)
25200 NEXT I
25210 PRINT #1:CHR$(27)&CHR$
(64):".PL 1" :: CLOSE #1 ::
SUBEXIT
25220 PRINT "BAD PARAMETER"
:: END :: RETURN
25230 SUBEND
```

```
100 REM ***********
110 REM *TUG-A-WAR*
120 REM ***********
130 REM IN TI BASIC
140 GOTO 190
150 FOR I=1 TO LEN(A$)
160 CALL HCHAR(R,C+I,ASC(SEG
$(A$,I,1)))
170 NEXT I
180 RETURN
190 RANDOMIZE
200 CALL COLOR(14,1,7)
210 CALL SCREEN(2)
220 PC(0)=5
230 PC(1)=7
240 P$(0)="BLUE"
250 P$(1)="RED"
260 Y$(0)="<- 1  2  3  4  5
 6  7  8  9   "
270 Y$(1)="   9  8  7  6  5
 4  3  2  1 ->"
280 KHAR(0)=0
290 KHAR(10)=5
300 FOR I=96 TO 136 STEP 8
310 CALL CHAR(I,"00000000000
00000")
320 CALL CHAR(I+1,"0F0F0F0F0
F0F0F0F")
330 CALL CHAR(I+2,"3078FCFCF
C7830")
340 CALL CHAR(I+3,"001030101
0101038")
350 CALL CHAR(I+4,"003844040
810207C")
360 NEXT I
370 PS=5
380 PL=0
390 BP=17
400 CALL CLEAR
410 GOSUB 1040
420 PRINT TAB(11);"TUG-A-WAR
"
430 PRINT : :
440 B$=CHR$(128)&CHR$(128)&C
HR$(129)
450 PRINT TAB(9);"hhippq xxy
";B$
460 PRINT TAB(9);"hkiptq x{y
";CHR$(128);CHR$(132);CHR$(1
29)
470 PRINT TAB(9);"hhippq xxy
";B$
480 PRINT
490 PRINT TAB(11);"<-";TAB(1
8);"->"
500 FOR I=1 TO 15
510 PRINT
520 NEXT I
530 FOR I=1 TO 9
540 RANDOMIZE
550 KHAR(I)=INT(4*RND)+1
560 NEXT I
570 FOR R=13 TO 15
580 CALL HCHAR(R,2,96,2)
590 FOR I=1 TO 9
600 KH=96+KHAR(I)*8
610 CALL HCHAR(R,I*3+1,KH)
620 CALL HCHAR(R,I*3+2,KH)
630 CALL HCHAR(R,I*3+3,KH+1)
640 NEXT I
650 CALL HCHAR(R,31,136,2)
660 NEXT R
670 CALL HCHAR(14,BP,96+KHAR
(PS)*8+2)
680 IF (PS=0)+(PS=10)THEN 12
20
690 A$=Y$(PL)
700 R=17
710 C=1
720 GOSUB 150
730 CALL HCHAR(24,17,32)
740 A$=P$(PL)&"'S TURN "
750 R=20
760 C=11
770 GOSUB 150
780 R=22
790 C=14
800 A$="(1-9)"
810 GOSUB 150
820 GOSUB 1060
830 CALL KEY(0,K,H)
840 IF H=0 THEN 830
850 IF (K<49)+(K>57)THEN 830
860 AN=K-48
870 CALL HCHAR(24,17,K)
880 IF PL=0 THEN 930
890 AN=10-AN
900 S=AN
910 E=9
920 GOTO 950
930 S=1
940 E=AN
950 GOSUB 1140
960 FOR Q=S TO E
970 IF KHAR(Q)<>4 THEN 1000
980 KHAR(Q)=1
990 GOTO 1010
1000 KHAR(Q)=KHAR(Q)+1
1010 NEXT Q
1020 PL=-(PL=0)
1030 GOTO 570
1040 CALL COLOR(9,1,5)
1050 CALL COLOR(14,1,7)
1060 FOR I=1 TO 8
1070 CALL COLOR(I,PC(PL),2)
1080 NEXT I
1090 CALL COLOR(10,PC(PL),16
)
1100 CALL COLOR(11,PC(PL),11
)
1110 CALL COLOR(12,PC(PL),8)
1120 CALL COLOR(13,PC(PL),14
)
1130 RETURN
1140 IF ((AN<PS)*(PL=0))+(AN
>PS)*(PL=1)THEN 1210
1150 A=(KHAR(PS)=1)+(KHAR(PS
)=2)*2-(KHAR(PS)=3)-(KHAR(PS
)=4)*2
1160 BP=BP+A*3
1170 PS=PS+A
1180 IF (PS>0)*(PS<10)THEN 1
210
1190 PS=-(PS=-1)+(PS=11)+PS
1200 BP=-(BP<5)*3-(BP>29)*31
1210 RETURN
1220 R=14
1230 C=7
1240 A$=P$(-(PS=10))&" IS TH
E WINNER!"
1250 GOSUB 150
1260 A$="LIKE TO PLAY AGAIN
(Y/N)?"
1270 R=24
1280 C=4
1290 GOSUB 150
1300 CALL KEY(0,K,H)
1310 IF H=0 THEN 1300
1320 IF (K<>78)*(K<>89)THEN
1300
1330 IF K=89 THEN 370
1340 REM TUG-A-WAR BY PATRIC
K PARRISH IN COMPUTE! 4/86
1350 REM THE LOWER SERIES OF
 BOXES IS THE PLAYING FIELD.
 THE OUTERMOST BOXES REPRESE
NT EACH PLAYER'S HOME POSITI
ON.
```

TISHUG

T I's Homecomputer Us
P.O. Box 302, Carling

january

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

february

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

march

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

april

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

may

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

june

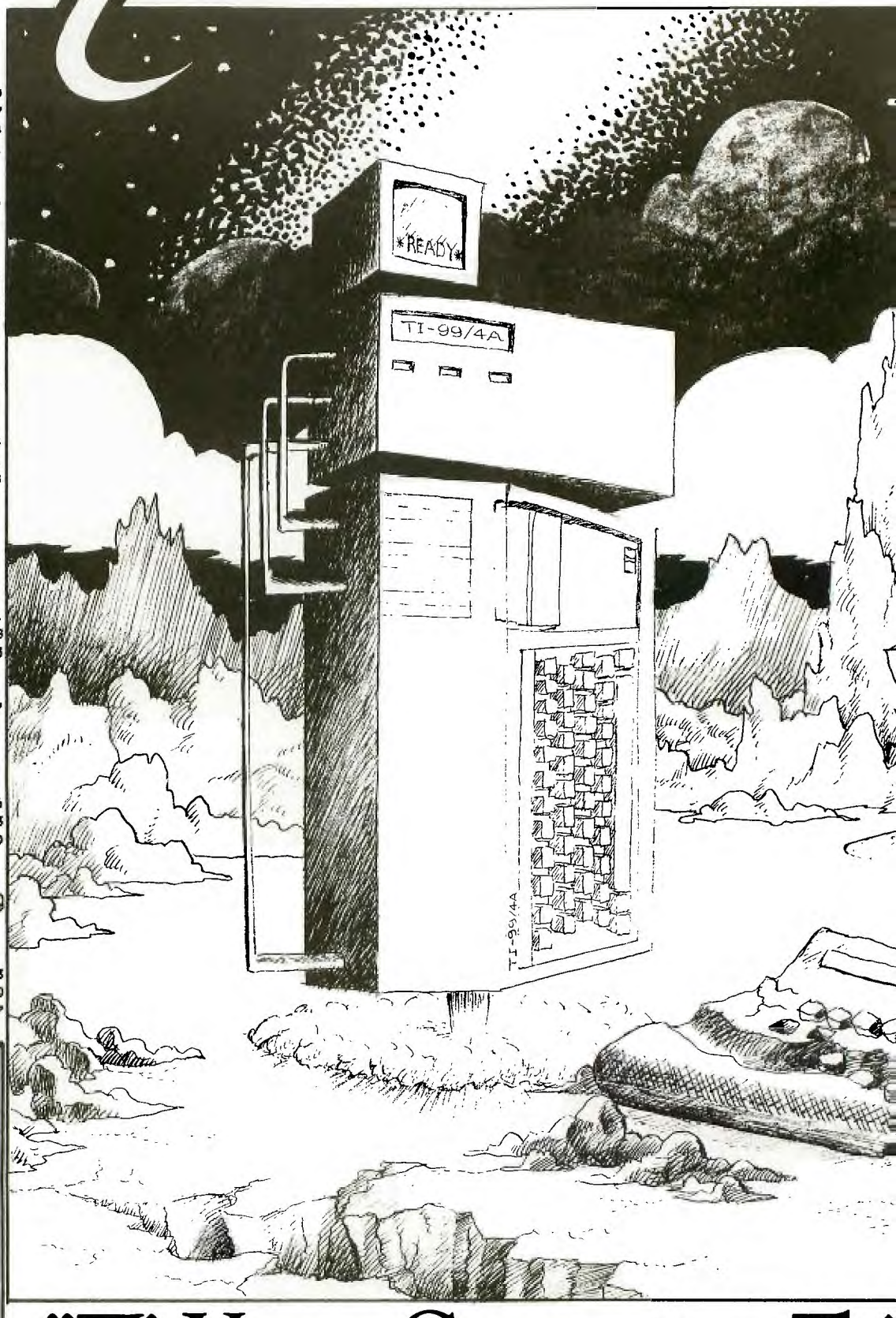
S	M	T	W	T	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30				

HELP

who ya gonna call

PROGRAMMERS
CRISIS
LINE

Ph : 992229

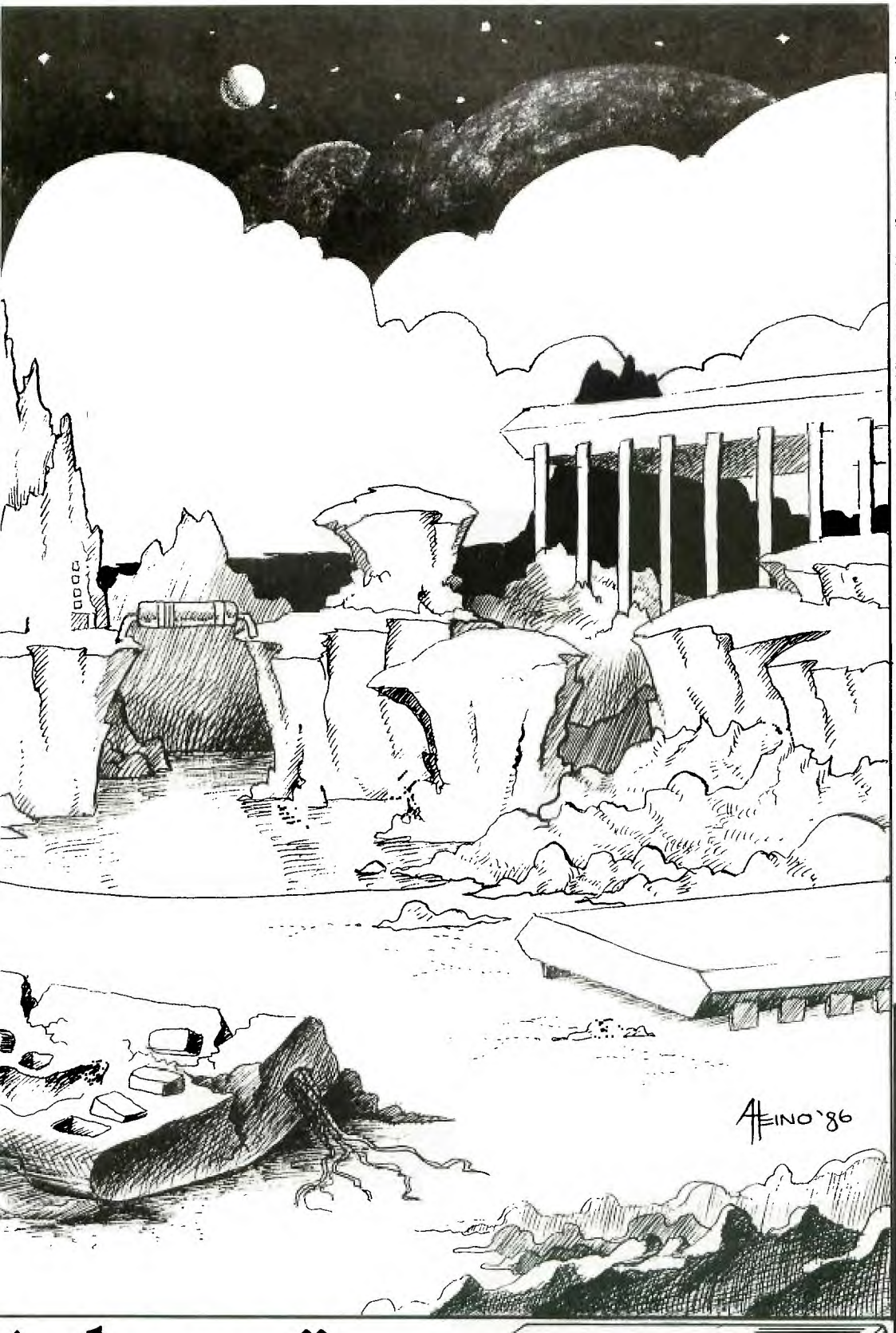


"TI's Home Computer. Thi

It makes you the one with the

s Group(Aust)Ltd
rd, N.S.W. 2118

CALENDAR 1987



july

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

august

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

september

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

october

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

november

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

december

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

```
1360 REM THE PLAYERS ALTERNA
TE TURNS, EACH TRYING TO MOV
E THE BALL IN THEIR OWN DIRE
CTION UNTIL IT REACHES A HOM
E SQUARE.
1370 REM THE COLOR OF THE SQ
UARE THE BALL IS ON DETERMIN
ES WHICH DIRECTION IT MOVES
AND HOW FAR.
1380 REM ON ANY TURN, THE BA
LL CAN MOVE EITHER 1 OR 2 SQ
UARES IN EITHER DIRECTION, O
R NONE.
1390 REM THE 4 BOXES AT THE
TOP OF THE SCREEN SHOW WHICH
 COLORS ARE LINKED TO EACH D
IRECTION. THE LEFTMOST BOX S
HOWS THE COLOR THAT MOVES
1400 ' THE BALL ONE SQUARE
TO I ' ', ETC.
1410 REM BELOW EACH BOX IS A
 NUMBER REPRESENTING ITS DIS
TANCE FROM THE HOME POSITION
 OF THE PLAYER WHOSE TURN IT
 IS.
1420 REM ON YOUR TURN YOU MU
ST SELECT A NUMBER (1-9) WHI
CH DETERMINES HOW MANY BOXES
 WILL CHANGE COLOR.
1430 REM FOR INSTANCE, IF YO
U PRESS 1 ONLY THE SQUARE NE
AREST YOUR HOME POSITION WIL
L CHANGE COLOR.
1440 REM EVERY SQUARE CYCLES
 THROUGH THE SAME SERIES OF
FOUR COLORS WHICH ARE SHOWN
IN THE UPPER 4 BOXES - WHITE
, YELLOW, BLUE, PURPLE.
1450 REM A WHITE SQUARE ALWA
YS CHANGES TO YELLOW, A YELL
OW ALWAYS CHANGES TO BLUE, B
LUE TO PURPLE, AND PURPLE BA
CK TO WHITE.
1460 REM THE BALL ONLY MOVES
 WHEN YOU CHANGE THE COLOR O
F ALL THE BOXES BETWEEN YOUR
 HOME POSITION AND THE CURRE
NT POSITION OF THE BALL.
1470 REM THE DIRECTION THE B
ALL MOVES DEPENDS ON THE COL
OR OF ITS SQUARE BEFORE YOU
TAKE YOUR TURN.
1480 REM IN MANY CASES YOU W
ILL WANT TO MOVE THE BALL ON
LY IF IT'S ON A COLOR THAT M
OVES IT TOWARD YOUR GOAL.
1490 REM BUT SOMETIMES IT IS
 WISE TO MAKE A SMALL TEMPOR
ARY SACRIFICE IN ORDER TO BE
NEFIT LATER IN THE GAME.
```

```
100 ! ************
110 ! *COOL-WATER*
120 ! ************
130 ! EXTENDED BASIC
140 ! BY BILL KNECHT
150 CALL CLEAR :: CALL SCREE
N(16):: CALL COLOR(9,8,16)::
 CALL CHAR(97,RPT$("F",16)):
: CALL CHAR(98,"FF7F3F1F0F07
0301"):: CALL CHAR(99,"0080C
0E0F0F8FCFE")
160 DISPLAY AT(2,8):"aaa aaa
 aaa a":"       a   a a a a
a":"       a   a a a a a":"
      a   a a a a a":"
 aaa aaa aaa aaa"
170 DISPLAY AT(8,5):"a a a a
aa aaa aaa aaa":"    a a a a
 a a   a a":"    a a a a
aa  a  aaa aaa" :: DISPLAY A
T(11,5):"a a a a a  a a  a
bc":"    aaaaa a a  a  aaa a
 b"
180 D=15 :: C$="A MUSIC PROG
RAM BY BILL KNECHT
 " :: C$=RPT$(" ",28)&C$ ::
FOR I=1 TO LEN(C$):: DISPLAY
 AT(14,1):SEG$(C$,I,28):: FO
R J=0 TO D :: NEXT J :: NEXT
 I
190 CALL CHAR(42,RPT$("F",16
)):: CALL CHAR(65,RPT$("F",1
6)):: CALL CHAR(72,RPT$("F",
16)):: FOR I=1 TO 5 :: READ
CH,CH$ :: CALL CHAR(CH,CH$):
: NEXT I
200 DATA 48,005AD1DEFFEEFFFF
,66,FFFF,67,0103071FFFFFFFFF
,68,00000000FFFFFFFF,73,FFD7
D5D5C1F7F7F7
210 CALL COLOR(2,5,1,3,13,8,
5,12,6,6,11,13):: DISPLAY AT
(16,1):RPT$("*",28):"*"&RPT$
("a",26)&"*":"*"&RPT$("a",26
)&"*":"*"&RPT$("a",26)&"*"
220 DISPLAY AT(20,1):"*"&RPT
$("a",21)&"AAAaa*":"0000"&R
PT$("a",16)&"AAAAAa*":"*BBBA
AAHHHIHHIHHHIIHAAAAAAA*":"*D
DC"&RPT$("A",23)&"*" :: DISP
LAY AT(24,1):RPT$("*",28)::
DU=425
230 CALL CHAR(128,"000E1F7F7
F7F3F1F3E0000000000000000078F
CFEFEFEFEFC7800000000000000"
)
240 CALL CHAR(132,"FFFFFFFFF
FFFFFFFFFFFFFFFFF00000
00000000000000000000000000"
)
250 CALL MAGNIFY(3):: CALL S
PRITE(#1,132,5,140,17,0,0,#2
,132,5,140,233,0,0)
260 CALL SPRITE(#3,128,16,14
0,64,0,0,#4,128,16,140,192,0
,0)
270 LLB=116 :: LC=131 :: LD=
147 :: LE=156 :: LF=175 :: L
G=196 :: LA=208 :: LB=233 ::
 C=262 :: D=294 :: E=311 ::
F=349 :: G=392 :: A=415 :: B
=466 :: HC=523 :: HD=587 ::
HE=622
280 HF=698 :: HG=784 :: W=40
000 :: T=1 :: X=F :: Y=W ::
Z=LLB :: GOSUB 470 :: Y=D ::
 GOSUB 470 :: Z=LC :: GOSUB
470 :: Z=LLB :: GOSUB 470 ::
 Y=W :: GOSUB 470 :: Y=D ::
GOSUB 470 :: Z=LC :: GOSUB 4
70 :: Z=LLB :: GOSUB 470 ::
Y=W :: Z=LE :: G
OSUB 470 :: X=E :: Y=LB :: Z
=LLB :: GOSUB 470
290 Z=LC :: GOSUB 470 :: Z=L
LB :: GOSUB 470 :: Y=HE :: Z
=LG :: GOSUB 490 :: Z=LE ::
GOSUB 470
300 CALL MOTION(#3,0,1,#4,0,
1)
310 X=E :: Y=W :: IF T=2 THE
N Y=B
320 Z=W :: GOSUB 470 :: X=G
:: IF T=2 THEN Y=HC
330 Z=LB :: GOSUB 450 :: GOS
UB 450 :: IF T=2 THEN Y=B
340 GOSUB 450 :: GOSUB 450 :
: GOSUB 480 :: Y=W :: GOSUB
450 :: GOSUB 450 :: X=F :: Z
=LA :: GOSUB 470 :: GOSUB 45
0 :: GOSUB 480 :: X=G :: Z=L
B :: GOSUB 450 :: X=F :: Z=L
G :: GOSUB 450 :: X=E :: GOS
UB 470 :: GOSUB 450 :: GOSUB
 480 :: X=C :: G
OSUB 450 :: X=D :: Z=LA :: G
OSUB 450 :: X=F :: GOSUB 450
 :: GOSUB 500 :: Y=D :: GOSU
B 510 :: Y=LB :: Z=LG
350 GOSUB 470 :: X=E :: GOSU
B 500 :: Y=HC :: GOSUB 470 :
: Y=B :: GOSUB 500 :: X=B ::
 Y=W :: Z=LB :: GOSUB 470 ::
 X=HC :: Z=LA :: GOSUB 470 :
: Z=LE :: GOSUB 470 :: Z=W :
: GOSUB 480 :: Z=LE :: GOSUB
 450 :: X=HD :: Z=LB :: GOSU
B 470 :: GOSUB 4
70 :: Z=LF :: GOSUB 480 :: G
OSUB 450 :: X=HE :: Z=LE ::
GOSUB 500 :: Z=LB :: GOSUB 4
70 :: X=HC
360 Z=LA :: GOSUB 490 :: Z=L
E :: GOSUB 470 :: X=A :: GOS
UB 470 :: X=E :: GOSUB 470 :
: X=G :: GOSUB 500 :: Y=HC :
: GOSUB 470 :: Y=B :: GOSUB
500 :: X=F :: Y=D :: Z=LB ::
 GOSUB 490 :: Y=HC :: GOSUB
450 :: Y=B :: GOSUB 480 :: Y
=D :: GOSUB 490
:: Y=HC :: GOSUB 450 :: Y=B
:: GOSUB 480 :: Y=LB :: Z=LE
 :: GOSUB 470 :: X=E :: GOSU
B 500 :: Y=HC
370 GOSUB 470 :: Y=B :: GOSU
B 500 :: Y=HC :: GOSUB 460 :
: GOSUB 440 :: Y=B :: GOSUB
460 :: GOSUB 440 :: GOSUB 47
0 :: Y=W :: GOSUB 460 :: GOS
UB 440 :: X=D :: Y=HC :: Z=L
LB :: GOSUB 460 :: GOSUB 440
 :: Y=B :: GOSUB 460 :: GOSU
B 440 :: GOSUB 4
70 :: Y=W :: GOSUB 460 :: GO
SUB 440 :: X=E :: Y=HC :: Z=
LE :: GOSUB 460 :: GOSUB 440
 :: Y=B :: GOSUB 460 :: GOSU
B 440
380 GOSUB 470 :: Y=W :: GOSU
B 460 :: GOSUB 440 :: X=F ::
 Y=HC :: Z=LLB :: GOSUB 460
:: GOSUB 440 :: Y=B :: GOSUB
 460 :: GOSUB 440 :: GOSUB 4
70 :: Y=W :: GOSUB 470 :: Y=
HC :: Z=LE :: GOSUB 450 :: X
=E :: GOSUB 450 :: Y=B :: GO
SUB 500 :: Y=HC
:: GOSUB 470 :: Y=B :: GOSUB
 500 :: X=HC :: Y=HF :: Z=LA
 :: GOSUB 470 :: Y=HE :: GOS
UB 460 :: GOSUB 440
390 GOSUB 470 :: GOSUB 470 :
: X=B :: Y=HF :: Z=LG :: GOS
UB 470 :: Y=HE :: GOSUB 470
:: GOSUB 470 :: GOSUB 460 ::
 GOSUB 440 :: X=HC :: Y=HF :
: Z=LA :: GOSUB 460 :: GOSUB
 440 :: Y=HE :: GOSUB 460 ::
 GOSUB 440 :: GOSUB 470 :: G
OSUB 460 :: GOSU
B 440 :: X=HD :: Y=HG :: Z=L
B :: GOSUB 460 :: GOSUB 440
:: Y=HF :: GOSUB 460 :: GOSU
B 440 :: GOSUB 470 :: GOSUB
470
400 X=HE :: Y=HC :: Z=LE ::
GOSUB 470 :: Y=B :: GOSUB 50
0 :: IF T=1 THEN 430 ELSE 41
0
410 X=G :: Y=HC :: GOSUB 470
 :: Y=B :: GOSUB 470 :: Y=HC
 :: GOSUB 470 :: Y=B :: GOSU
B 470 :: Y=HE :: GOSUB 490 :
: Y=HC :: GOSUB 450 :: X=B :
: Y=HE :: GOSUB 450 :: X=HC
:: Y=HF :: GOSUB 450 :: X=HE
 :: Y=HG :: GOSUB 450 :: X=H
F :: Y=932 :: GO
SUB 490 :: X=HE :: GOSUB 490
```

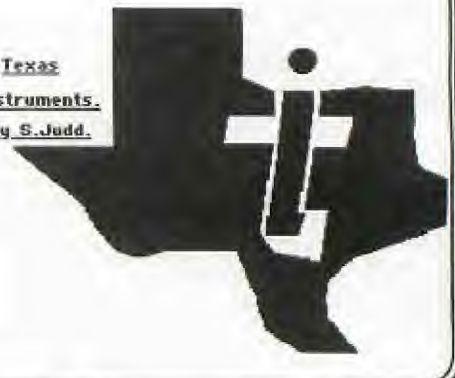
```
420 FOR I=1 TO 200 :: NEXT I
 :: CALL DELSPRITE(ALL):: RU
N "DSK1.LOAD"
430 Y=HC :: Z=LF :: GOSUB 47
0 :: Y=B :: GOSUB 470 :: X=F
 :: Y=HC :: Z=LB :: GOSUB 47
0 :: T=2 :: GOTO 310
440 CALL SOUND(DU/4,X,2,Y,7,
Z,10):: RETURN
450 CALL SOUND(DU/2,X,2,Y,7,
Z,10):: RETURN
460 CALL SOUND(DU/1.5,X,2,Y,
7,Z,10):: RETURN
470 CALL SOUND(DU,X,2,Y,7,Z,
10):: RETURN
480 CALL SOUND(1.5*DU,X,2,Y,
7,Z,10):: RETURN
490 CALL SOUND(2*DU,X,2,Y,7,
Z,10):: RETURN
500 CALL SOUND(3*DU,X,2,Y,7,
Z,10):: RETURN
510 CALL SOUND(4*DU,X,2,Y,7,
Z,10):: RETURN
```

```
100 CALL CLEAR :: CALL TITLE
(5,"ADD & CARRY")!by Jim Pet
erson
110 DISPLAY AT(3,10):"COPYRI
GHT":TAB(10);"TIGERCUB SOFTW
ARE":TAB(10);"FOR FREE":TAB(
10);"DISTRIBUTION":TAB(11);"
SALE PROHIBITED"
120 CALL PEEK(-28672,A@):: I
F A@=0 THEN 160
130 DATA FINE,NO,GOOD,UHOH,R
IGHT,TRY AGAIN,YES,THAT IS N
OT RIGHT
140 FOR J=1 TO 4 :: READ RIG
HT$(J),WRONG$(J):: NEXT J
150 FOR D=1 TO 1000 :: NEXT
D :: CALL DELSPRITE(ALL)
160 CALL CLEAR :: CALL CHAR(
95,"FFFF"):: CALL MAGNIFY(2)
:: RANDOMIZE :: CALL SCREEN(
14):: FOR SET=5 TO 8 :: CALL
 COLOR(SET,16,1):: NEXT SET
170 CALL CHAR(120,"E70042001
8007E0000E700420099423CE7004
20099423C00E7004218003C4200"
)
180 CALL CHAR(124,"0E0004010
00708007000208000E01000")
190 DISPLAY AT(3,8):"ADD AND
 CARRY" :: CALL CHAMELEON
200 CALL COLOR(14,2,2):: CAL
L HCHAR(4,4,143,2):: CALL HC
HAR(5,4,143,2):: CALL SPRITE
(#25,120,11,25,25)
210 T=T+1 :: IF T=6 THEN T=0
 :: GOTO 250
220 Z=INT(8*RND+2):: IF Z=Z2
 THEN 220 ELSE Z2=Z
230 Y=INT(Z*RND):: IF Y=Y2 T
HEN 230 ELSE Y2=Y :: X=Z-Y
240 N=1 :: GOSUB 470 :: GOTO
 210
250 T=T+1 :: IF T=11 THEN T=
0 :: GOTO 290
260 X=INT(10*RND):: IF X=X2
THEN 260 ELSE X2=X
270 Y=INT(10*RND):: IF Y=Y2
OR X+Y<10 THEN 260 ELSE Y2=Y
 :: Z=X+Y
280 N=1 :: GOSUB 470 :: GOTO
 250
290 T=T+1 :: IF T=11 THEN T=
0 :: GOTO 330
300 X=INT(90*RND+10):: IF X=
X2 THEN 300 ELSE X2=X
310 Y=INT(90*RND+10):: IF Y=
Y2 THEN 310 ELSE Y2=Y :: Z=X
+Y
320 N=2 :: GOSUB 470 :: GOTO
 290
330 X=INT(900*RND+100):: IF
X=X2 THEN 330 ELSE X2=X
340 Y=INT(900*RND+100):: IF
Y=Y2 THEN 340 ELSE Y2=Y :: Z
=X+Y
350 N=3 :: GOSUB 470 :: GOTO
 330
360 R=96 :: CC=96 :: FOR J=1
 TO N :: CALL SPRITE(#J,48+A
(J),11,R,CC):: CC=CC+16 :: N
EXT J
370 R=116 :: CC=96 :: FOR J=
1 TO N :: CALL SPRITE(#4+J,4
8+B(J),11,R,CC):: CC=CC+16 :
: NEXT J
380 CALL HCHAR(18,12,95,N*3)
:: CC=CC-16 :: CALL SPRITE(#
22,43,16,R,80):: RETURN
390 R=140 :: FOR J=LEN(STR$(
Z))TO 1 STEP -1 :: CALL SPRI
TE(#20,63,11,R,CC)
400 CALL KEY(3,K,ST):: IF ST
<1 OR K<48 OR K>57 THEN CALL
 PATTERN(#20,32):: CALL PATT
ERN(#20,63):: GOTO 400
410 CALL DELSPRITE(#20):: CA
LL SPRITE(#12+J,K,11,R,CC)
420 IF K-48<>C(J)THEN GOSUB
480 :: CALL DELSPRITE(#12+J)
:: CALL SPRITE(#20,63,11,R,C
C):: GOTO 400
430 IF A(J-W)+B(J-W)>9 THEN
CALL SPRITE(#28,49,16,80,CC-
16)
440 CC=CC-16 :: NEXT J :: GO
SUB 510 :: RETURN
450 FOR J=1 TO LEN(STR$(X)):
: :: A(J)=VAL(SEG$(STR$(X),J
,1)):: NEXT J :: FOR J=1 TO
LEN(STR$(Y)):: B(J)=VAL(SEG$
(STR$(Y),J,1)):: NEXT J
460 FOR J=1 TO LEN(STR$(Z)):
: C(J)=VAL(SEG$(STR$(Z),J,1)
):: NEXT J :: W=LEN(STR$(Z))
-LEN(STR$(X)):: RETURN
470 GOSUB 450 :: GOSUB 360 :
: GOSUB 390 :: FOR D=1 TO 20
0 :: NEXT D :: CALL DELSPRIT
E(ALL):: DISPLAY AT(18,1)::
CALL CHAMELEON :: CALL SPRIT
E(#25,120,11,25,25):: RETURN
480 DATA 123,124,125,123,124
,125,123,120
490 IF A@=0 THEN 500 :: CALL
 SAY(WRONG$(INT(4*RND+1)))
500 RESTORE 480 :: FOR JJ=1
TO 8 :: READ P :: CALL PATTE
RN(#25,P):: XX=2^250 :: NEXT
 JJ :: RETURN
510 DATA 121,122,121,122,121
,122
520 IF A@=0 THEN 530 :: CALL
 SAY(RIGHT$(INT(4*RND+1)))
530 RESTORE 510 :: FOR JJ=1
TO 6 :: READ P :: CALL PATTE
RN(#25,P):: XX=2^250 :: NEXT
 JJ :: RETURN
540 SUB CHAMELEON
550 M$="1800665AC342DB667E18
8100995AC3A5E78142BD24DB6600
81429924007E5AC3A53C241800FF
DB5AFF7EFF0099188100660018"
560 RANDOMIZE :: CALL CHAR(1
28,SEG$(M$,INT(43*RND+1)*2-1
,16)):: X=INT(14*RND+3)
570 Y=INT(14*RND+3):: IF Y=X
 THEN 570 :: CALL COLOR(13,X
,Y)
580 CALL HCHAR(1,2,128,30)::
 CALL HCHAR(24,2,128,30):: C
ALL VCHAR(1,31,128,96):: SUB
END
590 SUB CHAMWIPE
600 T=T+1+(T=2)*2 :: ON T GO
TO 610,620
610 CALL VCHAR(1,3,128,768):
: GOTO 630
620 CALL HCHAR(1,1,128,768)
630 CALL CLEAR :: SUBEND
640 SUB TITLE(S,T$)
650 CALL SCREEN(S):: L=LEN(T
$):: CALL MAGNIFY(2)
660 FOR J=1 TO L :: CALL SPR
ITE(#J,ASC(SEG$(T$,J,1)),J+1
-(J+1=S)+(J+1=S+13)+(J>14)*1
3,J*(170/L),10+J*(200/L))::
NEXT J
670 SUBEND
```

```
1 !This routine will write w
hat is probably the longest
one-liner that actually does
 anything. RUN it, then type
 NEW, then MERGE DSK1.LONG
2 !Then RUN. It will crash w
ith BREAKPOINT, but RUN agai
n, it will print 53 zeros.
LIST - DISPLAY X repeated 53
 times - over 24 lines!
3 !An even longer line can b
e written by repeating RANDO
MIZE 79 times (over 34 lines
!), but it does not "do" any
thing.
100 OPEN #1:"DSK1.LONG",VARI
ABLE 163,OUTPUT
110 FOR J=1 TO 52 :: M$=M$&C
HR$(162)&"X"&CHR$(130):: NEX
T J :: M$=CHR$(254)&CHR$(254
)&M$&CHR$(162)&"X"&CHR$(0)::
 PRINT #1:M$ :: PRINT #1:CHR
$(255)&CHR$(255):: CLOSE #1
```

```
100 !PRINT USING DEMO by Jim
 Peterson, based on a discov
ery by Chick De Marti
110 CALL CLEAR :: RANDOMIZE
:: CALL SCREEN(5):: FOR S=2
TO 14 :: CALL COLOR(S,S,S)::
 NEXT S
120 N=INT(13*RND+1):: C$=CHR
$(8*N+32-(N=4)*11)
130 FOR J=N TO 12 :: A$=RPT$
(" ",J)&"#"&RPT$(" ",26-J*2)
&"#" :: PRINT USING A$:C$,C$
 :: NEXT J
140 FOR J=12 TO N STEP -1 ::
 A$=RPT$(" ",J)&"#"&RPT$(" "
,26-J*2)&"#" :: PRINT USING
A$:C$,C$ :: NEXT J :: GOTO 1
20
```

Texas Instruments.
By S.Judd.



The shielded ribbon will work to over 800 mm length although I advise to keep it as short as necessary for your layout.(I tested my cable sucessfully to just under one metre!!!)

PARTS SUPPLIES
Power transformer 8V + 15V 2Amp
MB1, MB2 bridge rectifier 6 Amps
2200 uF 25V x4 Electro. capacitor
470 uF 25V x2 Electro. cap.
1 uf 25V Tag x4 Tantalum cap.
7805 Regulator
7812 Regulator
40 way x0.1" connector x2
34 way x0.1" connector for disk drive
34 way IDC IDC connector for controller
300mm of 34 way ribbon cable
600mm of 40 way ribbon cable

These are all available from:
GEOFF WOOD ELECTRONICS
229 Burns Bay Rd
Lane Cove
(02) 427 1676

44 way x0.1" solder connector for Console I/O Port and the power plug for the disk drive (Utilux part 1-428424-1) is available from:
Connect Electronics
45 Dickson Ave
Artarmon.
437 6224

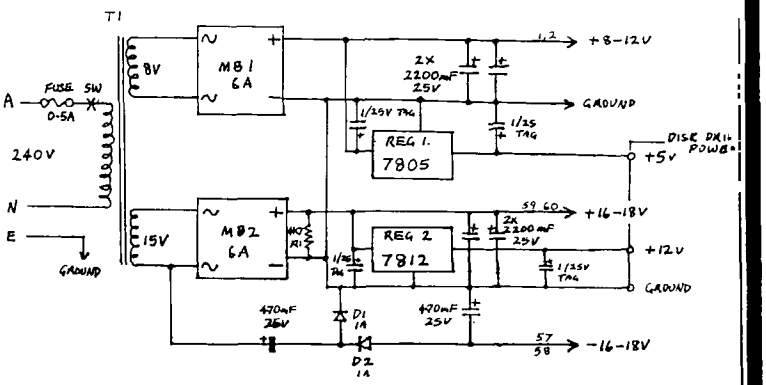
60 way connectors
These were made to size from readily available 100 way S100 type sockets by cutting them to fit, and cutting them the right length so that they just fit into the opening on the TI controller card so you won't have to worry about them sliding sideways. This can also be done with the 44 way if necessary.

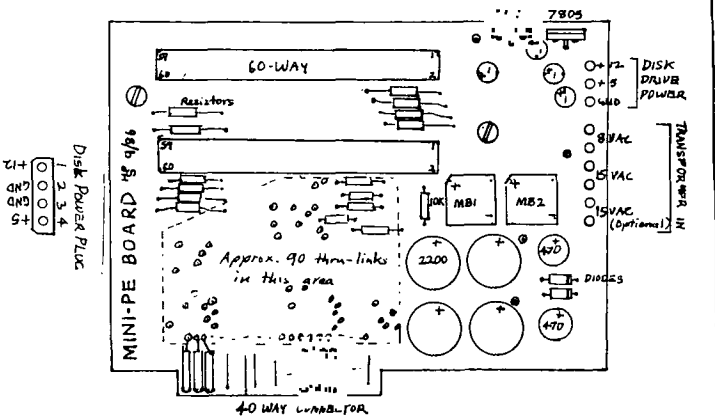
The 3-WAY CONNECTOR BOARD
This board designated Mini PEB/2 is made with very fine spacing between conductors and about 130 thru-links that have to be soldered close to each other. As we don't have facilities for plated-thru holes this is a very difficult board to assemble, so I prefer to do these myself and supply them complete and tested for $16, which includes all parts except the 44 way connector which must be supplied by you, unless you want the cut-to-size type connector for $3. The parts for the Load-Interrupt switch are also included on this, but for easy postage I'll let you solder in the P.B. switch.

GENERAL CONSIDERATIONS
This constructional series of articles is intended for those with some experience in Electronic construction therefore I have omitted some of the basic details which apply to such projects. To include all the details would most likely fill this whole magazine, and that is not the purpose of the TND. If you do not have the experience then perhaps another member can assist you in their spare time, otherwise do not attempt this project. Also as this is 240V Mains powered, relevent safety codes must be followed such as a suitable 240V cord, plug, anchoring of cord, ON/OFF switch and Terminal Block with Earth connection, and most important of all the FUSE. This fuse is all the protection your disk controller card has so it is an important part.
The -16 volt supply will only deliver about 100mA and is intended for the RS232 card. If only disk cont. card is to be used the capacitors and diodes can be omitted. Note that the 4K7 resistor on the +16V output is necessary to protect the card from residual charge when it is plugged in. This was not provided for on the PCB so must be added by soldering across the 1uF TAG.

If you have a VHF modulator and your TV is close to the computer you will get a little interference to your screen unless you do as TI did and make your box in Iron sheet. Another important consideration is cooling and adequate ventilation is necessary otherwise the controller card will overheat. I left the back of my box open across the top about 25mm down from lid AND also drilled many holes in the bottom near the front to allow cross ventilation. Or you could install a small quiet fan in the back. DO NOT EVEN TEST THE UNIT IN A CLOSED BOX OR DAMAGE WILL RESULT. And on that dramatic note I conclude this article.

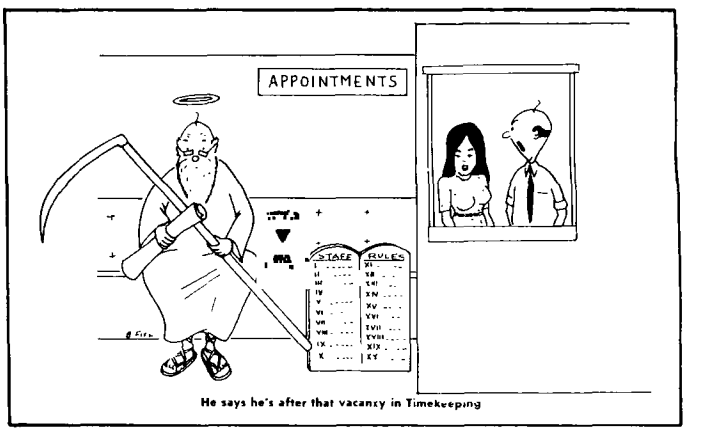
POWER SUPPLY
MINI PE BOX





By Peter Schubert

FOOTNOTE: A late message comes to hand from Bernie Elsner in WA who says he still has 2 or 3 disk controller cards and will send them to me so if you need one give me a call.



He says he's after that vacancy in Timekeeping

ASSEMBLER.

for the beginner with

ARTO

```
*****************************************
*                                       *
* XB CALL TO BLWP ASSEMBLY ROUTINES #1  *
*                                       *
*          by   ARTO HEINO 1986         *
*                                       *
*****************************************
```

I hope these routines will help you make the transition to assembly writing.

These BLWP modules are very useful if you want to speedup your screen handling routines.

If you are a complete novice, write your program in basic first to see if it works O.K. then write it in assembly.

Listed below are all the commands available for you to use. Place your own code where it says 'FILL WITH YOUR OWN CODE'.

When using these routines you can use variables, only if you declare them.If you are a novice just use numbers.eg..

```
     BLWP @HCHAR
     DATA 12,4,65,16
```

Which is the same as:

```
   CALL HCHAR(12,4,65,16)
```

```
*****************************************
BLWP @GCHAR      =CALL GCHAR(Y,X,RTCHR)
DATA Y,X,RTCHR

BLWP @HCHAR      =CALL HCHAR(Y,X,CH,RPT)
DATA Y,X,CH,RPT

BLWP @VCHAR      =CALL VCHAR(Y,X,CH,RPT)
DATA Y,X,CH,RPT

BLWP @CLEAR      =CALL CLEAR

BLWP @SCREEN     =CALL SCREEN(COL)
DATA COL

BLWP @COLOR      =CALL COLOR(CSET,FG,BG)
DATA CSET,FG,BG

BLWP @CHAR
DATA CHR,>0000,>0000,>0000,>0000

=CALL CHAR(CHR,"0000000000000000")
*****************************************
```

When using these BLWP modules try to make sure you check your values before you assemble the code, I put very little error checking.

```
*****************************************
*      CALL LINK("TEST")                *
*****************************************
       DEF  TEST
VSBW   EQU  >2020
VSBR   EQU  >2028
VMBW   EQU  >2024
VMBR   EQU  >202C
VWTR   EQU  >2030
GPLWS  EQU  >83E0
STATUS EQU  >837C
SAVRTN DATA >0000
MYWS   BSS  32
SWSP   BSS  32
CHBUFF BSS  8
*=====================================
TEST   MOV  R11,@SAVRTN
       LWPI MYWS
*
*
********************
*FILL WITH YOUR OWN
*CODE.
********************
*
*
EXIT   LWPI GPLWS
       MOV  @SAVRTN,R11
       CLR  @STATUS
       RT
*=====================================
XYLOC  MOV  *R14+,R1      :CONVERT XY
       DEC  R1            :TO SCREEN
       SLA  R1,5          :POSITION
       MOV  *R14+,R0      :ROUTINE
       DEC  R0
       A    R1,R0
       RT
*-------------------------------------
GCHAR  DATA SWSP,GCH1     :GCHAR
GCH1   BL   @XYLOC        :ROUTINE
       BLWP @VSBR
       SRL  R1,8
       AI   R1,->60
       MOV  R1,*R14+
       RTWP
*-------------------------------------
VCHAR  DATA  SWSP,VCH1    :VCHAR
VCH1   BL   @XYLOC        :ROUTINE
       MOV  *R14+,R1
       AI   R1,>60
       SWPB R1
       MOV  *R14+,R2
VCH2   BLWP @VSBW
       AI   R0,32
       CI   R0,>2FF
       JGT  VOVER
VCH3   DEC  R2
       JNE  VCH2
       RTWP
VOVER  AI   R0,-767
       JMP  VCH3
*-------------------------------------
HCHAR  DATA SWSP,HCH1     :HCHAR
HCH1   BL   @XYLOC        :ROUTINE
       MOV  *R14+,R1
       AI   R1,>60
       SWPB R1
       MOV  *R14+,R2
HCH2   BLWP @VSBW
       INC  R0
       CI   R0,>300
       JEQ  HOVER
HCH3   DEC  R2
       JNE  HCH2
       RTWP
HOVER  CLR  R0
       JMP  HCH3
*-------------------------------------
CLEAR  DATA SWSP,CLR1     :CLEAR
CLR1   CLR  R0            :ROUTINE
       LI   R1,>8000
CLR2   BLWP @VSBW
       INC  R0
       CI   R0,>300
       JLT  CLR2
       RTWP
*-------------------------------------
SCREEN DATA SWSP,SCN1     :SCREEN
SCN1   MOV  *14+,R0       :ROUTINE
       AI   R0,>6FF
       BLWP @VWTR
       RTWP
*-------------------------------------
COLOR  DATA SWSP,COL1     :COLOR
COL1   MOV  *R14+,R0      :ROUTINE
       AI   R0,>80F
       MOV  *R14+,R2
       DEC  R2
       SLA  R2,12
       MOV  *R14+,R1
       DEC  R1
       SWPB R1
       SOC  R2,R1
       BLWP @VSBW
       RTWP
*-------------------------------------
CHAR   DATA SWSP,CH1      :CHAR
CH1    MOV  *R14+,R0      :ROUTINE
       AI   R0,-29
       SLA  R0,3
       AI   R0,>3E8
       LI   R3,CHBUFF
       MOV  R3,R1
       MOV  *14+,*3+
       MOV  *14+,*3+
       MOV  *14+,*3+
       MOV  *14+,*3+
       LI   R2,8
       BLWP @VMBW
       RTWP
       END
```

Adverts Classifieds

*** FOR SALE *** FOR SALE ***
One Thunderer Modem with 2yr warranty.Come on-line to TEXPAC BBS at 1200 baud. Also Viatel modes. Comes with the best TI Comms software on disk (Freeware).$190 P.O.Box 28 Kings Cross.2011, or phone Peter on (02)358 5602.

Stand-alone RS232 with 300+1200/75 Modem. Plugs into side of Console, new with 2yr warranty $270. Optional parallel port (PIO) $30 extra Peter Schubert (02)358 5602.

FOR SALE: Mini Memory Module and Assembly Books - $30. Touch Typing Tutor - $15. Phone Terry on (02) 80.2317 (After Hours)

TI ... "The great deceiver"

by George Meldrum
TiSHUG Illawarra Regional group.

This is a story of much to do about nothing. It started when Mr Ti developed his BASIC, you know, TI BASIC. He decided the user (us) could use the video memory area to store their (our) program. Wanting to give us as much as possible of the 16K (16,384) bytes of memory available, he set upon a deception to ponder the talents of the great Pink Panther.

Now the video memory area (VDP RAM) is needed for storing screen information. The screen displays 24 lines of 32 characters, i.e. 24x32=768 characters in view on the vdu! Mr Ti set the VDP RAM addresses 0-767 for the screen map. We use this area by writing DISPLAY "HI" and the ASCII values for "H" and "I" are loaded into the screen memory. Well that's not the end of it. What about the shape of each character? Alphanumeric characters have their shapes automatically defined by the system and we redefine character shapes with CALL CHAR. Just where are these character shape definitions stored?, you guessed it, VDP RAM.

Now back to Mr Ti's dilemma, he wants us to have as much free area as possible for our program code. However character definitions take a lot of memory, 8 memory locations for each character in fact. But it gets worse. The video processor chip (TMS9929A) allows the character table to start only at certain points in VDP memory. It can start at memory address 0, but that clashes with the screen map. Next it can start at address 2048, but that will take away valuable programming space. Anyway to cut a long story short Mr Ti decided to start the character table at memory address 0. How? you ask. Good question because now when you write something to the screen memory you are also redefining some character shape!

Just a few extra comments so you know where we are at : (1) The video processor allows up to 256 character shapes to be defined ; (2) When you write an ASCII value to the screen memory the video processor looks for the character shape at an address calculated by :- ASCII value x 8 + base address of the definition table (in our case this is 0).

What Mr Ti did next was truely very tricky. For the screen display he compromised and gave us around 130 of the 256 characters using only the high character values i.e. CHR$(128-255). Now remember what was said earlier about when we DISPLAY "HI" on the screen. The ASCII values for "H" and "I" are loaded into the screen memory. This does not happen! What Mr Ti's BASIC does is to add 96 to each ASCII value and load that into the screen memory e.g. ASC("H")+96, ASC("I")+96. You see the character shapes for "H" and "I" have been predefined by the system at memory locations : (ASCII value + 96) x 8.

Have you ever wondered why the graphic characters range to CHR$(159) and don't go as high as CHR$(255)? Try adding 96 to 159 and what do you get? yes 255. The video processor allows characters 0-255 so why can't we use some of the lower value characters for graphics? Answer : because the character shapes would be defined in the screen memory area. And why didn't Mr Ti place the character shape table much higher in memory away from the screen memory? Answer : because he wanted us to have more memory area for programming.

Finally, does any of this make any difference to the BASIC programmer? No!, all of this is transparent to the BASIC user. However some machine language programmers have hopefully just seen the light. Remember, all of this is top secret, so forget it!


```
50 REM **********************
       SCREEN TEST
      FOR THAT OLD TV SET
       SHOWS YOU IF YOUR
      SCREEN IS CENTERED
60 REM     PROPERLY
      **********************
65 REM  by 'Biggie' in
       Central Ohio 99'ers
       Spirit of 99
       June 1984
70 CALL CLEAR
80 CALL CHAR(32,"FF818181818
181FF")
90 PRINT : : : : : : : : : :
 :
100 CALL HCHAR(10,1,88,5)
110 CALL HCHAR(10,28,88,5)
120 GOTO 120
130 REM ********************
       YOU SHOULD SEE 32
       PERFECT VERTICAL
       LINES AND 24 PERFECT
       HORIZNTAL LINES.
140 REM IN ROW 10 YOU SHOULD
       SEE FIVE X'S ON THE
       LEFT SIDE OF THE
       SCREEN AND FIVE ON
       THE RIGHT SIDE...
150 REM THE LAST TWELVE
       BOTTOM ROWS SHOULD
       HAVE COLUMNS 1,2,31
       & 32 MISSING....
160 REM THAT IS, IF YOUR SET
       IS ALIGNED RIGHT.
```

REVIEWS

Russel's Cataloguer

a review by Ben Takach

The Hunter Valley is famous for its excellent wines, however there are many other good things coming out of the New Castle district. We have some very talented TI- programmers in this part of the State.
I received my copy of the Russel's Cataloguer program by curtesy of its creator, Russel Welham. He worked over a year on this project, and it was obvious that indeed it needed a lot of time to give it its professional polish as soon as I tried it. The program is just about crash proof, it is well insulated against the usual operators errors.
The program was not ment for general distribution, hence it contains no documentation files. Anyway, it is very "user friendly", thus documentation is hardly needed.
It was written for systems having two single sided drives. The data files are on a second disk or on the B side of a flippy. This naturally does not impede the use of DSDD drives, one can simply acknowledge the request to change disks by a stab at the space bar.
The main menu displays 7 options:

1 Read Disks
2 Delete Disks
3 Show Lists
4 Search Lists
5 Sort
6 Change
7 End

READ DISKS.
The disk file is searched, then the program asks if it should be added to the list. A "Y" prompt is followed by the execution of the command, and the totalisator is updated: -the number of disks on file,
-the number of program or data files in the list,
-the name of the last disk read.
It will give 3 choices if a disk is read which is already on the file, or has the same name:

U Update,
T give it a Temporary name,
R Return to main menu.

These options cater for all possibilities: one may have changed the disk content, there could be more than 1 disks bearing the same name or one simply could not remember that the disk is already listed.
Updating is a tedious job for the computer, it could take some time, especially when it has to look through the files of some 100 odd disks. The screen display tracks the process to keep you informed.
DELETE DISKS.
This segment will remove all the relevant disk files and the disk name from the catalogue. The operation is automatic, it needs no explanation.
SHOW LIST.
The show list option will display and/or print out disks or files of the catalogue. The listing could be global or sectional. The printer default is PIO, this may be changed in line 8060. The listing is not in alphabetical order if it is executed before the sort option.
SORT.
The sort option will rewrite the entire catalogue in alphabetical order. The user has to select the sort mode, (S)hell or (B)ubble sort. Default is Shell sort. Program and disk lists are sorted separately, the choice is determined by the user (single key push P or D). The program list sort option provides an other time saver; one can opt for a complete sort or any letter of the alphabet. This is a useful feature after an update where only a few files have been added.
The show list option run after the sort will naturally list the catalogue in alphabetical order. One may also choose a partial listing of just one or more consecutive letters of the alphabet. This enables the user to locate a specific program or file without the need to scan through the entire catalogue.
CHANGE.
The catalogue totals may be changed with this option. These are:

- the number of disks on file,
- the total number of files listed,
- the total number of sectors used, and
- sectors free.

The print-out is configured for 120 column format. Printing starts at tab 10 and produces 3 37 column length lists across the page. Page length is 67 lines. The validate parameters in line 9020 may be changed from ("12") to ("123"). This will allow the selection of DSK3. Subroutine calls to 9160 in the main program and 420 in the Shell-sort resp. 410 in the bubble-sort programs may be deleted if the program and data files are on the same disk. The user will not be called then to change disks.
The 37 program and data files fill 213 sectors of the disk. Of these, 28 data files are used to store file names, 1 data file contains the disk names, there are 6 XB program files and 2 assembly object codes. The record lengths are carefully worked out, data packing is very dense. The catalogue of 68 disks containing 1288 files fill an additional 179 sectors.

To sum it up, it is a delight to work with this excellent utility program. It is a very essential program to keep a growing disk inventory in order. The program structure is very rational, thus it runs quite fast, considering that the bulk of the program is in basic. The program makes no reference to fair ware status albeit it deserves a reward more than many others.

```
100 CALL CLEAR
110 PRINT "       DISK SWEEP
ER            ERASE AN ENTIR
E DISK         IN LESS THAN
A MIN."
120 PRINT
130 PRINT "     STEVE PATTER
SON"
140 FOR T=1 TO 5
150 PRINT
160 NEXT T
170 PRINT "PLACE DISK IN DIS
K DRIVE #2   PRESS ANY KEY T
O DELETE"
180 CALL KEY(0,K,S)
190 IF S=0 THEN 180
200 CALL CLEAR
210 R=1
220 OPEN #1:"DSK2.",INPUT ,I
NTERNAL,RELATIVE
230 INPUT #1:A$
240 B$=A$
250 GOSUB 430
260 R=R+2
270 INPUT #1:B$
280 IF B$="" THEN 320
290 GOSUB 430
300 DELETE "DSK2."&B$
310 GOTO 270
320 CLOSE #1
330 CALL CLEAR
340 PASS=PASS+1
350 IF PASS=5 THEN 390
360 PRINT ">>> PASS #"
370 CALL HCHAR(23,13,PASS+49
)
380 GOTO 210
390 PRINT "    COMMAND COMPL
ETED            PRESS ANY K
EY"
400 CALL KEY(0,K,S)
410 IF S=0 THEN 400
420 END
430 IF R=24 THEN 510
440 FOR U=1 TO LEN(B$)
450 C$=SEG$(B$,U,1)
460 A=ASC(C$)
470 CALL HCHAR(R,U+3,A)
480 NEXT U
490 R=R+1
500 RETURN
510 PRINT
520 R=23
530 GOTO 440
```

SEASONS GREETINGS

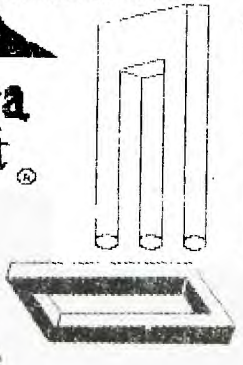
YOUNGER SET

UNDER 18'S PAGE

Welcome to another Younger Set section for members under 18 years of age. I have received a few of letters, one from Richard Exton with some really great Graphx pictures, one from the very dizzy "TICTACMAN" (that's his Username of the new improved Texpac BBS) Stephen Judd who helps us out with FATHOM module, and another one from Vincent Maker who writes... DEAR JENNY, I HOPE YOU LIKE THE PROGRAM I HAVE ENCLOSED WITH THIS LETTER. I'D LIKE TO WISH BRYAN APPLEBY A MERRY CHRISTMAS (HOPEING HE'S READING THIS), AND ALL READERS TO, A VERY MERRY CHRISTMAS TOO. BY THE WAY, THE PROGRAM IS IN BASIC. WHAT SHOULD I DO TO CONTRIBUTE AN ADVENTURE COLUMN WITH HINTS TO THIS MAGAZINE?
WISHING YOU A MERRY CHRISTMAS,
VINCENT MAKER.

Thank you Vincent, and a very happy & fun Christmas & New Year to both you and our other readers. If you want to contribute anything to this magazine, please feel free to do so. Just send any written articles to the Younger Set address (P.O.Box 595, Marrickville, NSW 2204). If you have any artwork (drawings) to go with your article, then please use a BLACK fine tipped felt pen. While I am talking about this, if any one want to send me PRINTED goodies for this page, please ensure that you set your printer to DOUBLE STRIKE, as I need your print/art work nice and black, the quickest way to do this is to go to BASIC (or Ex-Basic) with your printer on, and type the following...
OPEN#1:"PIO",OUTPUT (then press ENTER)
PRINT#1:CHR$(27);CHR$(71) (& ENTER)
you will then see the printer advance the paper a line wich indicates that the internal switches of the printer have now been set, and will stay at that setting 'till you change it or turn off the printer. Now, back to Vincent, I have passed on your program to Terry (Our Software Librarian) for him to check out and add to the Club Library. Remember that when you send in program of your own doing, that you can select any 3 programs from the Club Library of your choice, and you could be in the running for the supurb Cerficate "SPECIAL COMMENDATION FOR EXCELLENCE" as pictured in last months TND. Shane has designed it to look really OK on your wall, on stiff matt white card, and something to both strive for, and be proud of.

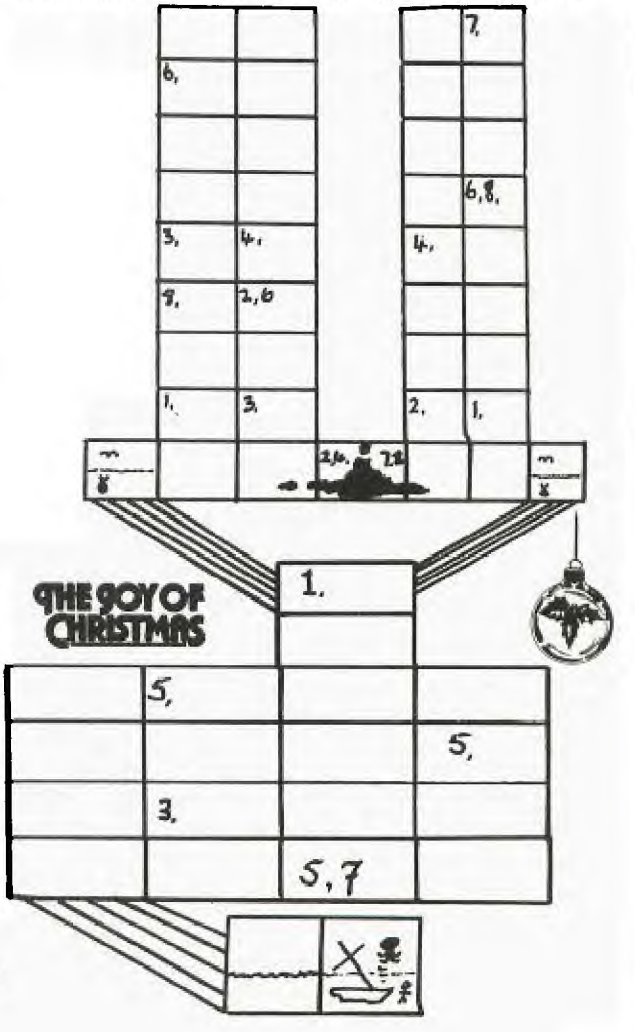
DEAR JENNY,
OVER THE PAST TWO YEARS I HAVE OWNED THE GRAPHX SOFTWARE, AND IN MY SPARE TIME, WHILE NOT STUDYING, I HAVE THESE 'PIECES OF ART'. NOW THAT I HAVE JUST COMPLETED MY LAST HSC EXAM, I THOUGHT I'D SEND YOU A FEW OF MY DIAGRAMS FOR YOU TO PUT IN YOUR COLUMN.
SINCERELY,
RICHARD EXTON.



Hi Richard, nice to hear from you again. I loved the Graphx Picture you've created. Sorry I didn't have room to put your GHOST BUSTERS one in, but hopefully later. Keep up the good work and let's hear more from you with some other creations which you might like to do over the Christmas/New Year Holiday period.
Regards
JENNY.

DEAR JENNY, I RECENTLY BOUGHT THE MODULE 'FATHON'. I FOUND IT HARD TO REMEMBER THE LOCATION OF THREE PIECES OF MAGIC TRIDENT, FOR IF I LOOKED IN EVERY SCREEN, I WOULD QUICKLY RUN OUT OF ENERGY. SO I TOOK OUT A PIECE OF PAPER, DREW UP THE SCREENS, SHOVED IN THE MODULE AND BELOW IS MY WORK. I GOT UP TO THE EIGHT LEVEL BEFORE RUNNING OUT OF ENERGY. THE OTHER LEVELS I'LL LEAVE TO YOU. THE '1's CORRESPOND FOR A PIECE OF TRIDENT ON LEVEL ONE, ETC.
GOOD LUCH! (YOU'LL NEED IT!)

Thanks Stephen, it looks good, and also thanks for your Column Headings which have been apreparing in the TND.



Turn to the next page for more YOUNGER SET information

LINKING BASIC TO ASSEMBLY

#9 LINK IT. #9

BY ROSS MUDIE.

SEASONS GREETINGS

MEMORY MANAGEMENT, WHERE TO PUT THAT TEXT.

The 32K memory expanded TI99/4A home computer has 16K of Video Display Processor (VDP) RAM and 32K of CPU RAM which may be used with basic or extended basic in different ways by your own linked assembly programs.

The memory is broken up into 3 blocks and the usage varies dependant on which basic is in use.

Under TI BASIC with the Editor/Assembler module plugged in the GROM port, the basic program is stored in the VDP RAM, the Basic Support utilities (from the file BSCSUP on the Editor/Assembler disk) in the bottom end of the Low RAM,(from >2000 [8192] to >2695 [9845]). The relocatable assembly programs are stored from >A000 [-24576] to >FFFF [-1].

Under TI Extended BASIC the Basic program is stored in the High expansion RAM (from -26 down), the support utilities (loaded by CALL INIT) are stored in the Low RAM from >2000 [8192] to >24F3 [9459] and the strings are stored in the VDP RAM.

When using BASIC everything must be stored in the VDP RAM which has approximately 14K of user space. If the default of 3 disk files are available, this consumes 2088 bytes, then there are the symbol tables for the variables, the contents of the string and numeric variables, the Peripherial Access Blocks, the line number table & the BASIC program. No wonder that * MEMORY FULL is a familiar message when programing in BASIC.

As soon as the switch is made to Extended Basic, the program, line number table and the numeric value table are moved to the high expansion RAM. Figure 1 shows how the memory resource of the computer is used for extended basic and the location to pointers for the memory assignments.

Once your program is getting big it is likely to contain quite a lot of text messages. Two programs which I have found running out of program space because of the amount of text in the program are TI99-OPOLY (extended basic) and TEXPAC BBS (basic). The answer to the lack of space in the program area is to move the text into the assembly space in the form of a linked assembly program. The example program contained in this article was written for use with TEXPAC which is in c.

This message generator program is loaded as relocatable code for the "program" part, whilst the text storage area is loaded as absolute code in the LOW RAM. The reason for the use of absolute code (AORG) is that in the BBS the high RAM was also full. The basic loader will not load relocatable code (under normal circumstances) in the low RAM thus the easiest method is to use absolute code.

The short basic program allows the loading of the file after it has been assembled. If the program is called a number of times with the same message number and the message is a multi line message then sucessive lines of the message are returned until the end of the message or a new message number is requested. When each message string is returned the optional EOM (End Of Message) flag is a zero until the last string in the message, when the value in the EOM flag becomes a 1. By testing the value of EOM, convenient loop control is available.

The assembly program also contains the necessary changes for use with the extended basic program.

BASIC PROGRAM.

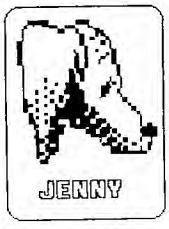
```
100 REM   SAVE DSK1.READ2
110 CALL INIT
120 CALL LOAD("DSK1.BSCSUP","DSK1.TT")
130 INPUT "Message Number? ":N
140 CALL LINK("SYSTXT",N,RS$,EOM)
150 PRINT RS$
160 IF EOM THEN 130 ELSE 140
```

EXTENDED BASIC PROGRAM.

```
100 ! SAVE DSK1.READ3
110 CALL INIT
120 CALL LOAD("DSK1.TT")
130 INPUT "Message Number? ":N
140 CALL LINK("SYSTXT",N,RS$,EOM)
150 PRINT RS$
160 IF EOM THEN 130 ELSE 140
```

More Over →

more YOUNGER SET WITH JENNY



This is one of Stephen Judd's masterpieces?!?! So, Stephen, you think I am a bit of a FOX, or was it a DOG? This brings me to the BIG ANNUAL COMPETITION... Each year, I conduct a CARTOON WITH CAPTION Contest. I know from past experience that we have a lot of very talented Younger Set members out there in TI land, and once again, over the Christmas & New Year period, I want you to create a cartoon like the one pictured here. It must be your own work, not traced or copied from another publication, and there are great CA$H PRIZES to be won. Good luck, and remember that there is no January issue of TND, as we're having a break.
Seasons Greetings.
From JENNY.

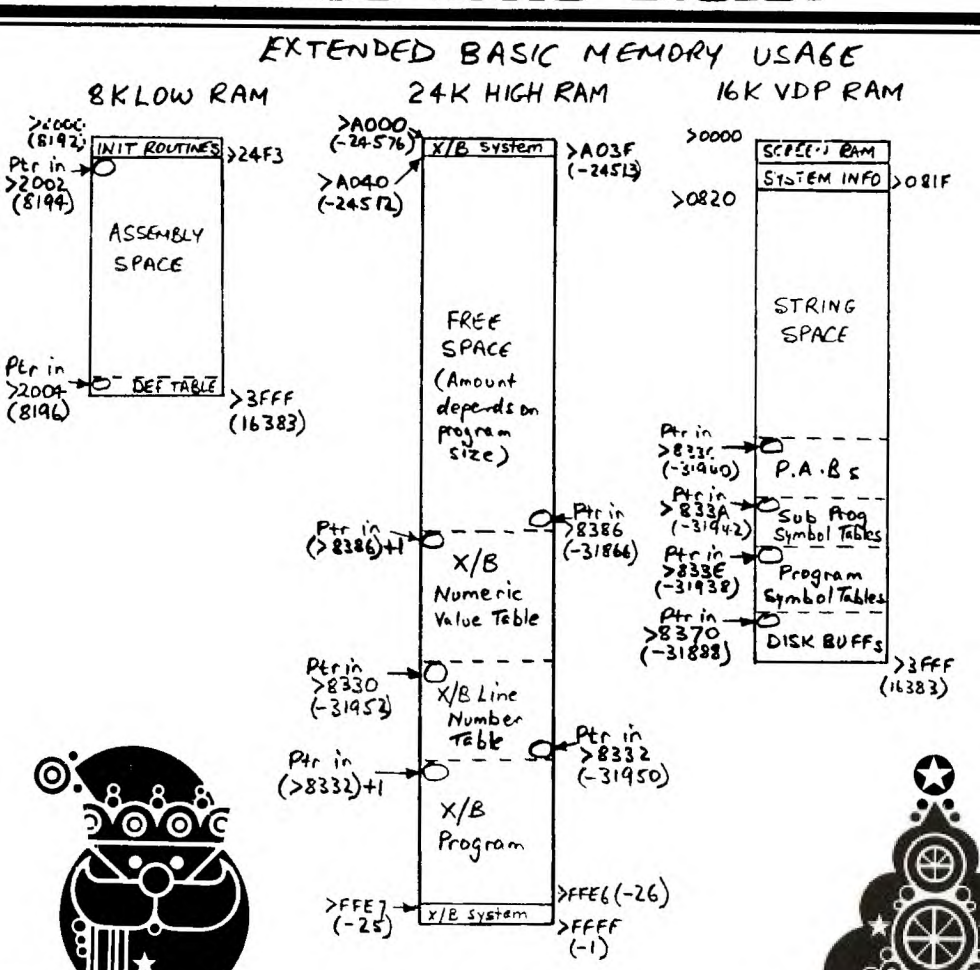


Figure 1

ASSEMBLY PROGRAM.

Only a few of the BBS system messages are shown in the following program example.

```
*  Basic linking statement format:
*
*  CALL LINK("SYSTXT",MSG_No,RS$[,EOM])
*  Where:
*       MSG_No = A numeric variable or  constant  indic-
*                ating the required message number.
*
*       RS$    = A string variable in which  the  string
*                message is returned.
*
*       EOM    = An optional numeric variable in which
*            End Of Message is indicated by a value of 1.
*
        IDT  'SYSTXTmu'    Source file=T  Object file=TT
        DEF  SYSTXT      Ross Mudie, 26th September 1986

        REF  XMLLNK,NUMREF,NUMASG,STRASG      BASIC only
*XMLLNK EQU  >2018           (
*NUMREF EQU  >200C           (  Extended Basic use only
*NUMASG EQU  >2008           (
*STRASG EQU  >2010           (

STATUS EQU  >837C
GPLWS  EQU  >83E0
FAC    EQU  >834A

LASTMG DATA >FFFF              LAST MessaGe number store
ONE    DATA 1                            Word sized one
EOM    DATA 1                            End Of Message
MAX    DATA 5       Maximum number of messages; to trap
*                         out of range message numbers.
BOO    BYTE 0                            Byte size zero
       EVEN

SAVRTN BSS  2                        Return address store
WS     BSS  32                        Register Work Space
BUFFER BSS  82    BUFFER for STRASG max string 81 bytes
```

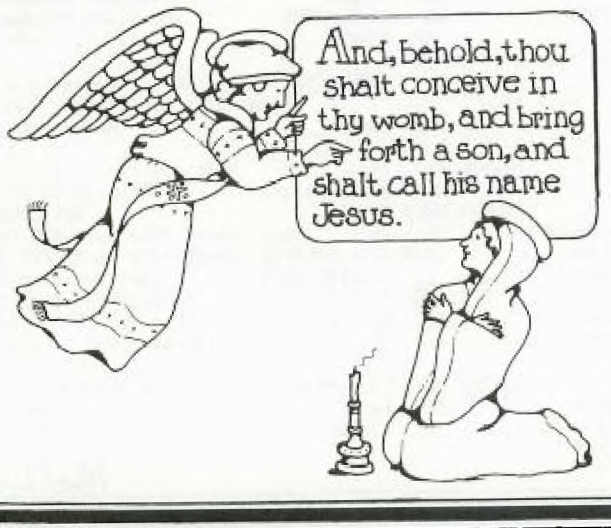
(LINK-IT9) R.Mudie

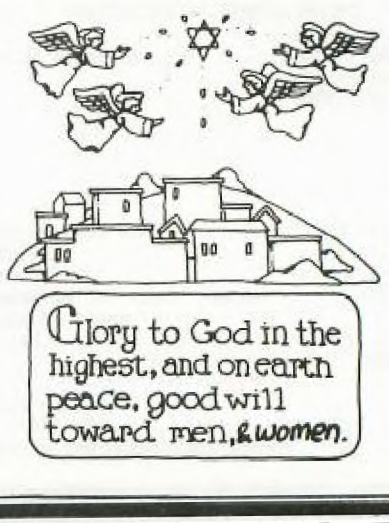
```
SYSTXT MOV  R11,@SAVRTN          Save the return address
       LWPI WS
       MOVB @>8312,R3     Qty of arguments in link list
       SRL  R3,8                 Make it a 2 byte number

       CLR  R0                                Simple variable
       LI   R1,1                          Argument 1 in link list
       BLWP @NUMREF     Get msg number from basic to FAC
       BLWP @XMLLNK     Convert floating point no in FAC
       DATA >1200         to integer in first word of FAC
*      DATA >12B8     Use >12B8 in lieu of >1200 for x/b

       MOV  @FAC,R4            Put message number in FAC
       C    R4,@MAX          Maximum number of messages
       JLE  VALIDN                      Number in range?
       CLR  R4                          If invalid number

VALIDN C    R4,@LASTMG      Is this the same msg number?
       JEQ  SAMEMG               Yes same message number
       MOV  R4,@LASTMG          Store new message number
       JMP  NEWMSG                  It is a NEW MeSsaGe

SAMEMG C    @EOM,@ONE       If EOM is 0 then msg has not
*                      been fully sent; more lines to go.
       JNE  PENDIN                    More message to go

NEWMSG SLA  R4,1           Multiply x2 for look up table
       MOV  @MPTR(R4),R5          Get address of message
PENDIN MOVB *R5,R6                        Get length byte
       SRL  R6,8     Make length byte into a 2 byte word
       MOVB *R5+,@BUFFER Length byte in BUFFER 1st byte
       CB   @BUFFER,@BOO          Test for a null string
       JEQ  TFRSTR             No point loading nothing!
       LI   R2,BUFFER           Put BUFFER address in R2
       INC  R2        R2 points to second byte of BUFFER

FILBUF MOVB *R5+,*R2+          Move message into BUFFER
       DEC  R6                Count down number of bytes
       JNE  FILBUF                        Finished yet?
```

More "LINK-IT" →

MORE "LINK-IT" ↘

```
TFRSTR INC  R1                  For next argument in link
       LI   R2,BUFFER        Tell STRASG where string is
       BLWP @STRASG  Transfer message to basic variable

       MOVB *R5+,R7       Get End Of Message (EOM) byte
       SRL  R7,8        Make EOM byte into a 2 byte word
       MOV  R7,@EOM   Remember that more of msg pending
       CI   R3,3            Is transfer of EOM required?
       JLT  END                          If NO then END
       INC  R1                 Next argument in link list
       MOV  R7,@FAC      Put EOM value in FAC as a word
       BLWP @XMLLNK  Convert Integer in FAC to Floating
       DATA >2300                          point in FAC
*      DATA >20        Use >20 in lieu of >2300 for x/b
       BLWP @NUMASG                 Transfer EOM value

END    CLR  R0                    Return to basic routine
       MOVB R0,@STATUS
       LWPI GPLWS
       MOV  @SAVRTN,R11
       RT

MPTR   DATA M0,M1,M2,M3,M4,M5

       AORG 9846                  Omit if extended basic

* Convention used in messages is as follows:
* 1. Each text string shall be preceeded with a  length
*    descriptor byte.
* 2. Each  text  string  shall  be  followed  with  a
*    terminator of 0 or 1.
* 3. If the string is the last in a  message  then  the
*    terminator shall be 1 else the terminator is 0.
* 4. A null string shall be two bytes of 0 each, or  if
*    the null string is a last  string  then  the  null
*    string is 0,1 .

M0     BYTE 19
       TEXT '* Invalid Request *'
       BYTE 1

M1     BYTE 38
       TEXT '**************************************'
       BYTE 0
       BYTE 38
       TEXT '*     TEXPAC ELECTRONIC MAGAZINE     *'
       BYTE 0
       BYTE 38
       TEXT '**************************************'
       BYTE 0
       BYTE 38
       TEXT '~Another service of TIsHUG Australia.~'
       BYTE 0
       BYTE 0,0
       BYTE 36
       TEXT '>> Please limit calls to 30 minutes.'
       BYTE 0
       BYTE 38
       TEXT '> You may log on again after one hour.'
       BYTE 0
       BYTE 38
       TEXT 'Have you changed your password lately?'
       BYTE 0
       BYTE 0,0
       BYTE 28
       TEXT '    <1> Bulletins/News Menu.'
       BYTE 0
       BYTE 0,0
       BYTE 30
       TEXT '    <2> Program Download Area.'
       BYTE 0
       BYTE 0,0
       BYTE 29
       TEXT '    <3> Send Electronic Mail.'
       BYTE 0
       BYTE 0,0
       BYTE 26
       TEXT '    <4> Read mail for ALL.'
       BYTE 0
       BYTE 0,0
       BYTE 24
       TEXT '    <5> Password Change.'
       BYTE 0
       BYTE 0,0
       BYTE 27
       TEXT '    <6> Terminate the Call.'
       BYTE 0
       BYTE 0,1
       EVEN

M2     BYTE 0,0
       BYTE 38
       TEXT '>>>  Please let others have a go!  <<<'
       BYTE 0
       BYTE 0,0
       BYTE 38
       TEXT '>> Allow 60 minutes between logons. <<'
       BYTE 0
       BYTE 0,1
       EVEN

M3     BYTE 0,0
       BYTE 40
       TEXT '**  Thank you for calling TEXPAC-BBS  **'
       BYTE 0
       BYTE 0,0
       BYTE 40
       TEXT '********** PLEASE HANG UP NOW **********'
       BYTE 1
       EVEN

M4     BYTE 44
       TEXT 'Exit TE2 with CTRL 0  & start receiving now'
       BYTE 1

M5     BYTE 0,0
       BYTE 17
       TEXT 'More to follow...'
       BYTE 1

       END
```





TISHUG NEWS DIGEST

This month we received quite a varied and interesting batch of mail from members each with something to offer. So here here goes:-

ADVENTURE HINTS from Vincent Maker of Winston Hills

1. Can't get into the pyramid? - Go into the pool then dig in the desert somewhere and around the pyramid.
2. Mummy a problem? - Look at the burning leaves
3. Snake a problem? - Look at something around the oyster
4. Bricked doorway? - Try the iron glove
5. Can't find coin? - You can't see but you can do something else
6. Can't find "scarab"? - Move in a certain direction from mirror room
7. Gold teeth? - Look skull before you give to the decapitated skeleton
8. Hole in ceiling? - Throw rope
9. Pharoah statue a problem? - Wash coal and throw it off the ledge
10. Chest a problem? - Wear a glove while unlocking it
11. Window a problem? - Saw bars

Thanks Vincent, I hope that more members will follow your example and share some of their successes!

FOR SALE! Denis Hickey of Westleigh has gone over to that other machine, a big blue XT, and has some software, books and hardware to sell. Give him a ring on (02)634-8526 or (02)84-5461(AH). He writes to advise that his TI has gone to a good home and he hopes that the new owner will enjoy it as much as he did. Also, Denis congratulates all current and past club officers and committee of TIsHUG for providing an interesting and rewarding membership.

Thank you Denis. Sorry to see you go and thank you for your kind comments. I agree that the XT is not as much fun as the TI!

RTTY SPECIAL REQUEST! from B. S. Stevenson. My special request is for the News Digest to publish the "RTTY (Radio Teletype)" program demonstrated at the June workshop. The originator of the program promised to make his program available for publication - but this hasn't happened!

GOOFS! from B. S. Stevenson. The Biorhythm Compatibility Program published Nov. 1984 - page 22 contains errors which cause all Wednesday dates as Tuesdays. This can be corrected by changing lines 490 and 550 to read GOSUB 1070 (ie. not 1080)

Thank you B.S! We will chase up for the RTTY program for you.

CHESS BUGS! from Stephen Marsden a new member of TIsHUG. He writes to ask if any other members have experienced a problem with TI. CHESS - "legal moves" not being allowed?

Welcome Stephen - time will tell. Still TI. CHESS is a terrific game (in strong demand too! All members should be on the look out for this game on tape! (Whispers are about that this will soon be available to users with the 32K expansions!)

SPECIAL SOFTWARE REQUIREMENT! from P. Weaver of Freeman's Reach. What I ask is probably impossible but I will ask it anyway! What I would like to see is a software or hardware add-on that would let me use software written for other computers preferably the Commodore 64 or any other popular brand - Apple, IBM etc. Other software I would like to see includes a "turbo loader" to speed up loading programs, a boxing or wrestling game and the ultimate challenge - A 2 player space game with one acting as pilot and the other as weapons man, or 1 player able to select either position.

This is already underway refer TND November 1986. It seems that Millers Graphics may be marketing a card which will allow the TI to run MS-DOS based software.

WIDER RANGE - CLUB SHOP from P. Weaver of Freeman's Reach. While I am writing, why doesn't the club shop stock more software? There are lots of games overseas that I would probably buy if they were available here. all I need is someone to import them. One item I am particularly interested in is the TI. Adventure editor. Another in the Adventure cartridge itself - which is no longer available here also the Scott Adams adventures themselves also in short supply and the Scott Adams hint book.

Any stock held by the club represents club member funds tied up in inventory. It is not our policy to "stock" items which members would "probably" purchase. I suggest that you write to the shop with your specific software requirements and enclose the necessary funds. The Shop Co-Ordinator may be able to help you.

RETURN TO PIRATES ISLE HELP! also from P. Weaver. Please print this plea in the TND - "Any advice at all on Return to Pirates Isle greatly appreciated. Come on, someone must have finished it!"

Yes I have - where are you upto in the quest?

THANK YOU JR! from John Kerr of Aldgate S. A. I was sorry to learn of John Robinson's resignation. I have always enjoyed his well written and informative reports in TND but have never told him so. I hope you will pass on my thanks to him. I trust he will continue to contribute to TND as I believe he has a real flair for written communication.

Many members out there echo you senitments. Thank you, I am sure John will be pleased when he reads your kind comments.

EPROM PROGRAMMER AVAILABLE from John Kerr of Aldgate S. A. I don't know if it would be of use to anyone but I do have access to a good Eprom programmer at work and would be happy to program Eproms for anyone interested. To do this I would need a copy of the hex code on disk and of course, the Eprom (2732, 2764 etc.). I could also copy Eproms or PAL devices from a supplied master; but only on the strict understanding that the material was legally copyable. I guess this facility is already available to you but I thought I would mention it.

Thank you John. Members interested should write c/o TIsHUG for onward mailing to John Kerr.

MORE THANKS TO TIsHUG from Tony Madison. At this time (renewal of membership) I really must say a big thank you to all of the people who have done so much, not only to keep TISHUG going, but also to have given of their own precious time,effort and knowledge to others like myself. Were it not for TISHUG many of the things about Computing, Programming, and the way to use computers effectively would not have been imparted to me and hundreds of 99/4A users. Computer literacy is indeed a valuable asset both at work and for enhancing leisure, and has enriched me immensely. There is still much to learn, but then that is what makes it good sense to keep going with the 99/4A. So from me, to TISHUG, a very very sincere THANK YOU!

Praise like that will keep us going for many years to come - more please!

→

THE COMMUNICATORS Special Interest Group.
by Ross Mudie, for December 1986 TND.

The old problem of the BBS program locking up on loss of carrier, due either to the call being cut off or the originator hanging up, has been overcome with changes in the assembly program. The BBS will now reset from all areas except the RS232 program download if carrier is lost. The word END in uppercase will also give a logoff from most prompts.

The Program Download feature, BBS main menu option 2, can only be used when the computer receiving the download is a TI99/4A. With the TI99/4A, to achieve a download you must exit the terminal program in use and go back to either TI Basic or TI Extended Basic. If the Terminal Program is Terminal Emulator 2, don't forget to use <CTRL> 0 (exit) to ensure that any open disk files are closed. The BBS will instruct you when to exit the terminal program, then enter the basic that you want to use and type OLD RS232 or OLD RS232/2 (dependant on where your modem is connected). A number 255 will then appear on the top centre of the screen. As soon as both computers commence communicating the number becomes the size of the program in 256 byte blocks which counts down as the blocks are transferred. Once the cursor returns, save the program to disk or tape. If you have finished with the BBS at this point you should hang up, however if you wish to download another program, just re-enter your terminal program and log on again.

All BBS operations are now being logged on a printer which allows analysis of problems, detection of invalid use & development of an understanding of usage trends. All users are requested to limit session times to 30 minutes between 6am and midnight each day with a 30 minute break between sessions. Between midnight & 6am users are requested to limit sessions to 60 minutes with at least 30 minutes between sessions. If you are performing a multiple program download, each session is from when you log on for the first time, not from the start of each new re-logon. Users who blatently exceed the time limits risk being placed in the "unfinancial" category.

All BBS users are reminded that you should change your password periodically to prevent unauthorised use of your user id. The BBS contains the necesary prompts to make this easy for the authorised users.

Due to the mainly unattended nature of the BBS, a new feature has been provided to allow all users to read the ALL mail file prior to the SYSOP transferring items to the appropiate file. This feature replaces the SYSOP PAGE facility which has been removed from the program. The messages on the ALL mail file are stored in order of receipt since they are stored on disk with the file opened in the APPEND mode. To read the newest items you must read to the end of the file. This feature is entered from the main menu (option 4) but on completion exits to the NEWS menu at present.

There is no formal (written) BBS policy for operations at present. All BBS members are invited to make their own submissions of items for consideration in the draft policy. You may address your ideas to the SYStem OPerator by using the electronic mail facility and the mail address of POLICY .

INFORMATION FOR POTENTIAL USERS OF THE BBS.

For the benefit of readers who are not BBS users, the following will introduce you to the BBS.

The Bulletin Board Service (BBS) is a TI99/4A computer connected to a telephone number via an automatic answer modem. By connecting your own computer via a modem to a telephone line and using a Terminal Emulator 2 module or a terminal program, such as Fast Term or 4A Talk, you can call and communicate with the BBS.
Each TEXPAC BBS user is issued with an individual user number, User Name and password. This restricts access to our members who hopefully use the service in a responsible manner. The BBS membership is $5, in addition to the normal TIsHUG membership. (Persons who are not members of TIsHUG are also eligible to have BBS membership for the following fees: Joining fee=$8, annual fee $10 if a member of another TI user group else $25.) Applications for BBS membership should include your name, address, telephone number and choice of User Name and password. User names and passwords are upper case alpha numeric up to 9 characters. Applications for BBS membership should be sent to:
TIsHUG, PO Box 302, Carlingford. NSW. 2118.

The BBS provides a selection of down loadable basic & extended basic programs, News, Information, Electronic Mail and BBS members adds of a private and non-commercial nature only. Once you are a member on the BBS the only extra cost is the price of your phone calls.

There are 113 registered users on the BBS and in the period 23/9/86 to 31/10/86, a total of 48 users have logged on to the system at least once.

SOFTWARE I WOULD LIKE to own from Ben Von Takach. In response to the Co-Ordinators request for required software suggestions - here is a list of three programs which would help the X/Basic programmer.

Screen dump: X/B screen dump programs run slow. These hold up program execution for long periods while the screen is scanned character by character from to to bottom. An Assembly language sub-routine speeds things up. We exchange the screen refresh addresses with that of the printer for the duration of a complete scan then change it again back to the original

Delete unwanted blocks of XB program to enable a partial program save. Line by line delete is tedious, using one of the utilities involves quite a deal of keyboard gymnastics and Disk fiddle. An Assembly routine loaded with a program accessible in the edit mode by manual command deletes unwanted line number blocks as prompted.

Longer XB lines: Typing text from BASIC or X/BASIC is limited to 4 and 5 lines respectively. (more possible with fiddling). An assembly routine accepts unlimited length of text which is subsequently formatted to user defined line lengths.

VARPTR(x) type sub-routine which returns the memory address of a variable or array so that it may be passed to an assembly language routine. (Refer GW-Basic)

Not a bad ask Ben! I am sure you could work on some of the routines already available from Ross Mudie and create your screen dump - its all there! As far as the others are concerned anybody out there to help Ben!

Well this is it again the final column for another year. I trust that you have enjoyed the software offerings throughout the year, and that you all get the one or two goodies that you want for Christmas. A big thanks to all who contributed throughout the year in the software department.

Seeing that it is the end of the year I have some really big software selections for you this month at the shop and we will start first of with what is available on disk.

1. HAPPY HOLIDAYS - this is a musical and graphic tribute to the holiday season. Written entirely in assembly it is sure to put you into the holiday spirit. All your old favourites are there and young and old should love it. It runs for about 30 minutes so that should give you some idea of how many favourite tunes there are to listen to. For this one you will need a disk system with memory expansion.

2. SNOOPY CALENDAR - if you want a great calendar for 1987 then get hold of this disk. You will get 12 different pictorial calendars, one for each month. Be prepared to wait a while for it to print out - at least an hour. For this one you will need either double sided drives or have a Corcomp or Myarc double density card. There are many files on the disk occupying 681 sectors in total. It will be distributed in both double sided and double density formats. Make sure you get the right one for your system.

3. MAX/RLE PICTURES #2 - a disk containing a selection of these interesting pictures was released in October. More have been obtained and a selection will be available on this disk. Titles to choose from include movie stars, past and present, astronauts, space scenes and a great picture of a tiger which really looks excellent on the screen when you fiddle with the keys to get a black and yellow color scheme.

4.MAX/RLE PICTURES #3 - adults only this one. To see whats on it you will have to buy it. Don't say you haven't been warned.

5. THE MARK BECK CREATIVE FILING SYSTEM - you may recall that I mentioned this last month and that I would do a review on it. Here it is. First up I should tell you that to run CFS you will need at least one disk drive and memory expansion. CFS comes on two disks, one containing the programs and the other documentation and demonstration files. Some features available include the ability to store up to 1440, 8 line records on a DSDD disk, a graph option for graphing files, print mailing labels 1 to 4 across, makes use of Foundations 128K card, converts files to DV80 so they can be used with TI Writer files, can use TI Writers mail merge option and can perform mathematical operations in its report generator. So what is the CFS - basically it's a data base that allows you to create files to your specific needs. According to the author, it was designed to be extremely user friendly and crash proof. I must say here that it never did crash on me despite my rigourous attempts. With most complex pieces of software it pays to read the instructions prior to doing anything and there is no exception to this rule here. While it may be user friendly when you are into it, unless you are familiar with data base operations then without reading the instructions you may not have a clue what to do. There are 24 pages of instructions to print out via the TI Writer/Funlwriter formatter so be prepared for a wait before you can read them. One of the strong points of CFS is the documentation. It is extremely easy to read and follow the authors examples. There is also some useful information on troubleshooting the CFS, plus a gentle reminder from Mark to send him your $10 today. Mark is quite happy for anyone to pass his program on but only those who do send the $10 will receive update notifications. Here's what you get on the program disk:
LOAD - does the obvious
CHARDF - file containing true lower case definitions
CFS - the main program
SORT - sorts files on 1 or 2 keys
GRAPH - graphs file
UTILITY - changes fields, disk and printer configuration, prints files and mailing labels.
REPORT - prints out records in report format.
CATALOG - catalogs disks and merges two files into one.
MENU - stores the main menu screen.
G_SCREEN - stores the menu screen for graph.

Two files are created when you first run CFS - these depend on your system configaration inputs. One is called DISK and the other PRINTER. If by some chance you get a copy of these files on your disk then they should be deleted before you first run CFS.

On the second disk, apart from the documentation are the following files:

#DEMO_FILE - sample CFS file of 25 records.
DEMO_DIS80 - this is the above file converted to DV80 format.
LETTER - a sample form letter.
VALUE_FILE - the #DEMO_FILE converted into a value file.
^#DEMO_FIL - sample default values for the Report generator (Gemini 10X)
_#DEMO_FIL - sample default values for the Utility program. These will print mailing labels, again Gemini 10X

Well there you have a brief description of this very versatile and powerful piece of software. If you are looking for a database then you could do a lot worse than to have a good look at this one. I personally would give it a 10 out of 10.

6. CASINO GAMES - I have only just received this disk of traditional casino games. Included are Blackjack, Craps, Poker and a card game called War, which I think I know what the objective is but I'm not 100% sure. Graphics in these are all very good with colored pictures on the court cards. Great fun to play and you're guaranteed not to lose your shirt.

7. SIDEWAYS PRINT - yes at last I have recived a copy of this useful utility which will allow you to print your Multiplan spreadsheets sideways. No more cutting up and pasting the bits and pieces together. If you've got Multiplan then you need this.

8. DISK 1986/12 - a disk of games, contents as follows:

HOLEYMOLEY - a game of simple fun where you have to jump about hitting moles on their heads before your supply of carrots is gone. Quite nice graphics add to this games pleasure.

ISLAND JUMPER - two programs with this one. The first is the instructions and the second the game. Jump from planes and try to land on the island. Plenty of better games around than this one but it might keep you occupied for a time. Use a CALL FILES(1) before loading from disk or the main game may crash!

ROBIN HOOD - this is a very well done archery game which will take you a fair time to master. The idea is to shoot your arrows into the targets, get 300 points and move onto more difficult levels. I have only got past the first screen once and then was unable to hit any of the moving targets. Hope you have better luck. There are two versions of the game and both will be on this disk - one for unexpanded systems and the other for those with 32K expansion.

DEBROIDS - a very well done game of the asteroids variety. You can use either keyboard or joysticks to shoot the space debris.

9. DECEMBER86 - a disk full of interesting utilities and home use applications - contents are:

99 CALC - a mini spreadsheet with plenty of instructions in program format. Also contains a sample file.

ARCHIVE - a useful utility for packing and unpacking several programs into one and vice versa.

TIKEYS - great macro key routines. This program has actually been packed using Archive and you will need to unpack it before you know what it is about. To do so just load Archive and follow the prompts.

GEMINI 10 PRINTER CONTROL - as mentioned by Shane in last months TND.

Extended Basic and 32K expansion are required for all programs on this disk.

10. SURPRISE - some really good software on this disk and I won't even tell you what it is. Get your copy at the shop. A big thanks to John Paine for the work he has done in this area.

OK that should be enough goodies for disk users to keep them occupied over the holiday period. So now for you tape users and I believe there are still a good number of you around.

TAPE 1986/12 - contents as per disk 1986/12 plus ...

ASTEROID RESCUE - a great little space game where you have to guide you ship to the planets surface avoiding asteroids. Once you pick up the astronaut take him safely back to the top of the screen. Not as easy as it sounds. This one in XB

JIGSAW PUZZLE - a puzzle is scrambled and it's your job to put it back together. You need either Mini Mem or Editor Assembler with basic to run this one.

TAPE 1986/12A - contents .....

BANZAI BUNNY - guide the bunny across the road avoid cars etc. OK you've seen it before with chickens and frogs but not a bunny! - in XB

BILLIARDS - sharpen up on your pool skills. Take on a friend and see if you can win - in XB.

DAY AT SCIOTO - I don't know where Scioto is but they must have a trotting track there, cause that's what this game is all about. Bet on the trotters and watch them go. Animation is very good - in XB

GREEDY GREEN GREENIES - a nice colorful maze escape game written in Basic.

MR KROAKER - this is not another frogger game but a difficult butterfly catching game. You are Mr Kroaker who must catch the insects to score points - in XB.

LEAPER - another good version of a jump obstacles type of game. In this one you have to rescue the princess. Not real easy as there are some meanies in your way - in XB.

HAPPY SPELLER - good educational fun for young and old. Input your own word lists. You'll need the TE2 module.

WORLD CURRENCY & WORLD MYTHOLOGY - 2 similar games where you have to guess the mystery scrambled word. You need to be quick though. Both in XB

TAPE 1986/12B - another surprise tape which you will need to buy to find out what's on it. I will tell you though not to buy unless you have 32K expansion. Note different programs are on this tape and the SURPRISE disk mentioned earlier.

Well that should be enought to keep everybody happy throughout the holidays. There is plenty of great software around for the TI at the moment and a lot of it is just reaching us here. Next year be on the lookout for other great offerings as announced in this column. Time did not permit the issuing of everything received in the past few months, but it will all be available at some time in the future.

Again a special thanks to all those who contributed throughout the year and also thanks to all who patronised the shop for the software available. I hope you got as much enjoyment out of it as I did running and reviewing it. I wish you all a merry Christmas and successful 1987.

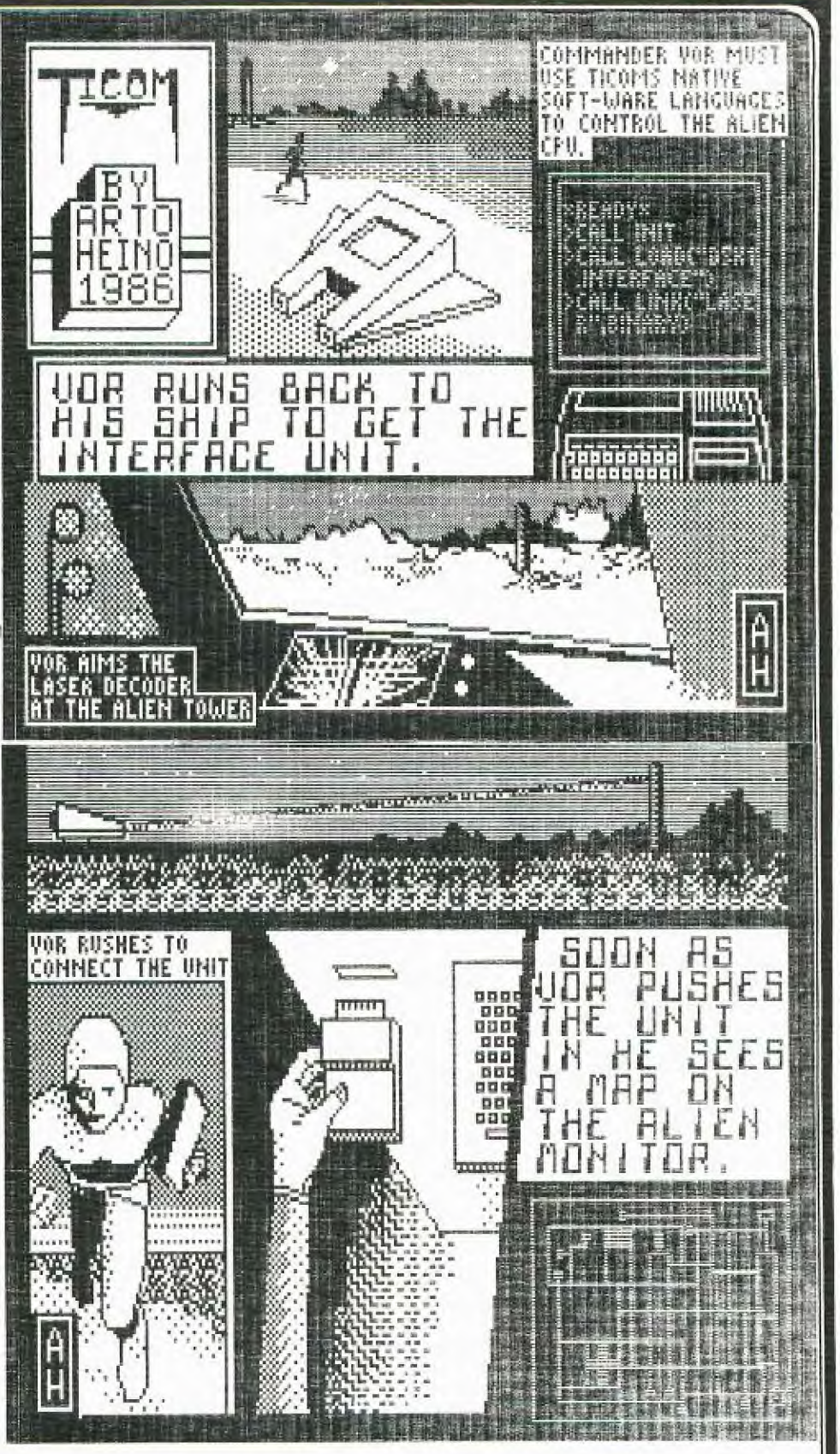
PS - if any country members happen to be in Sydney over the holiday period they are welcome to give me a call on 797.6313 and perhaps pop in and do some software copying. Bring you own disks.

Terry.

Type It

Programs To Try!

```
100 OPEN #1:"SPEECH",OUTPUT
110 CALL CLEAR
120 PRINT #1:"TIGERCUB MIND
READER PROGRAM": :
130 PRINT #1:"I'LL BET YOU A
 DOLLAR I CAN GUESS WHAT YOU
 ARE THINKING.": :
140 GOSUB 450
150 PRINT #1:"AND I'LL BET
ANOTHER DOLLAR I CAN TELL IF
 WHAT YOU ARE   THINKING IS
CORRECT.": :
160 GOSUB 450
170 PRINT #1:"AND I'LL BET A
NOTHER DOLLAR I'M RIGHT BOTH
 TIMES.": :
180 GOSUB 450
190 PRINT #1:"AND I'LL BET O
NE MORE DOLLAR I CAN GUESS W
HAT YOU'LL BE THINKING A MIN
UTE FROM  NOW.": :
200 GOSUB 450
210 PRINT #1:"OK....": :
220 GOSUB 490
230 PRINT #1:"YOU'RE THINKIN
G THAT A COMPU-TER CAN'T POS
SIBLY KNOW    WHAT YOU ARE T
HINKING.............RIGHT?":
:
240 GOSUB 490
250 PRINT #1:"SO I TOLD YOU
WHAT YOU WERE":"THINKING....
....RIGHT?": :
260 GOSUB 490
270 PRINT #1:"YOU OWE ME A B
UCK.": :
280 GOSUB 490
290 PRINT #1:"AND YOU'RE ABS
OLUTELY       RIGHT..I CAN'T
 READ YOUR    MIND.": :
300 GOSUB 490
310 PRINT #1:"SO I TOLD YOU
CORRECTLY THAT":"WHAT YOU WE
RE THINKING WAS":"CORRECT...
...RIGHT?": :
320 GOSUB 490
330 PRINT #1:"YOU OWE ME ANO
THER BUCK.": :
340 GOSUB 490
350 GOSUB 490
360 PRINT #1:"SO I WAS RIGHT
 BOTH TIMES...RIGHT?": :
370 GOSUB 490
380 PRINT #1:"THAT MAKES THR
EE BUCKS YOU  OWE ME."
390 PRINT #1:"AND NOW IT'S A
 MINUTE LATER":"AND YOU'RE T
HINKING YOU'VE":"BEEN PLAYED
 FOR A SUCKER....":"...RIGHT
?": :
400 GOSUB 490
410 PRINT #1:"...SO YOU OWE
ME FOUR BUCKS.": :
420 GOSUB 490
430 PRINT #1:"NEVER NEVER BE
T AGAINST A   COMPUTER!! "
440 END
450 PRINT #1:"WANT TO BET? T
YPE Y(YES)": :
460 CALL KEY(3,K,ST)
470 IF (ST=0)+(K<>89)THEN 46
0
480 RETURN
490 FOR D=1 TO 800
500 NEXT D
510 RETURN
```



I hope you have been following Commander Vors cartoon!! Because in the Feb. issue of TND he will be adventuring into the digital world once more with the aid of his trusty NINA.

Merry Christmas

Each year, we have our Christmas POT-LUCK Dinner, where you bring along a plate of your food, and the club provides the drinks etc. WELL THIS TIME IT'S ALL PROVIDED FREE OF CHARGE...You won't need to bring a thing...we'll provide a great big BAR-B-Q DINNER, Soft drinks, and a visit from Santa. All you have to bring along is yourself (and your membership card). Come and join us nice and early (12:00 NOON) for this fun informal afternoon of sharing, with your fellow TIsHUG members. This will be our final get-to-gether prior to out FEBRUARY '87 ANNUAL GENERAL MEETING & Election time for the club Committee.