PLUS! v.2.0

This dynamic package of utilities and word processing aids creates one of the most exciting environments in the world of TI/99 and Geneve.

Few things in good ol' **USA** are going to make you as happy as this new (and FINAL!) version of the popular PLUS! series.

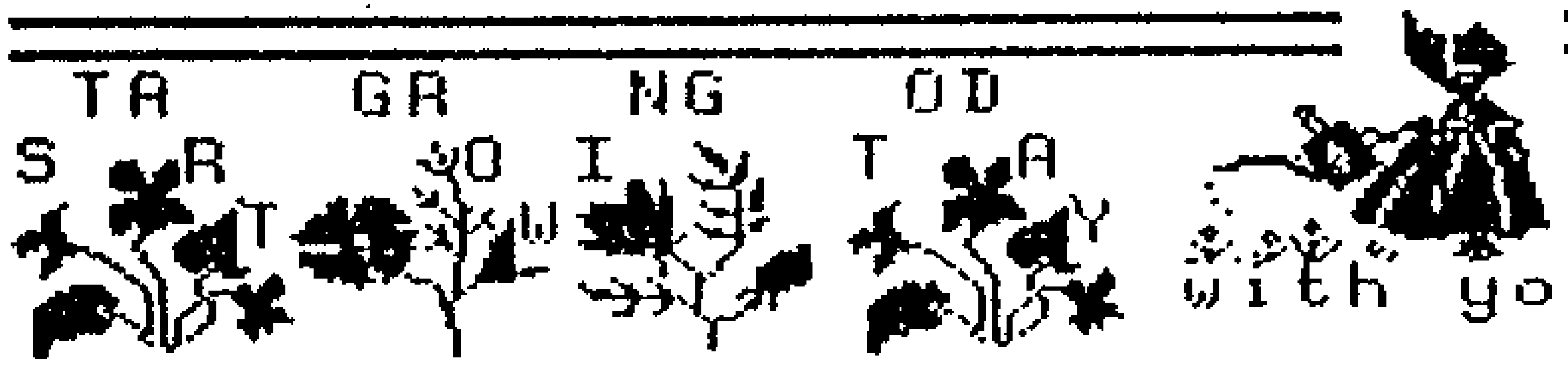
Almost 1400 sectors of pure dynamite is sure to keep you awake many fun-filled nights.

PLUS! 2.0 still has the tiny programs and templates that give you the feeling you've a RAMdisk for many applications even if you have the minimum TI WRITER (or FUNNELWEB) and/or EXTENDED BSIC configurations.

ALL of the original PLUS! files (programs, templates, tutorials, manual) have been updated, improved, or entirely replaced. In addition, numerous new files have been added to make this one of the most complete, most powerful environments of its kind available anywhere at any price.

THE GOOD NEWS

IS THAT PLUS! IS NOW ALMOST TWICE THE SIZE AT STILL THE LOW FAIRWARE PRICE OF JUST $10. Comes complete with additional documentary sheets on DSSD or SSSD. Please specify.



with your TI Computer and PLUS!

Order PLUS! 2.0 now and bring a little fun (and lots of ingenious programs and templates and more excellent tutorials) into your life.

Only $10 from: Jack Sughrue
Box 459
East Douglas, MA 01516

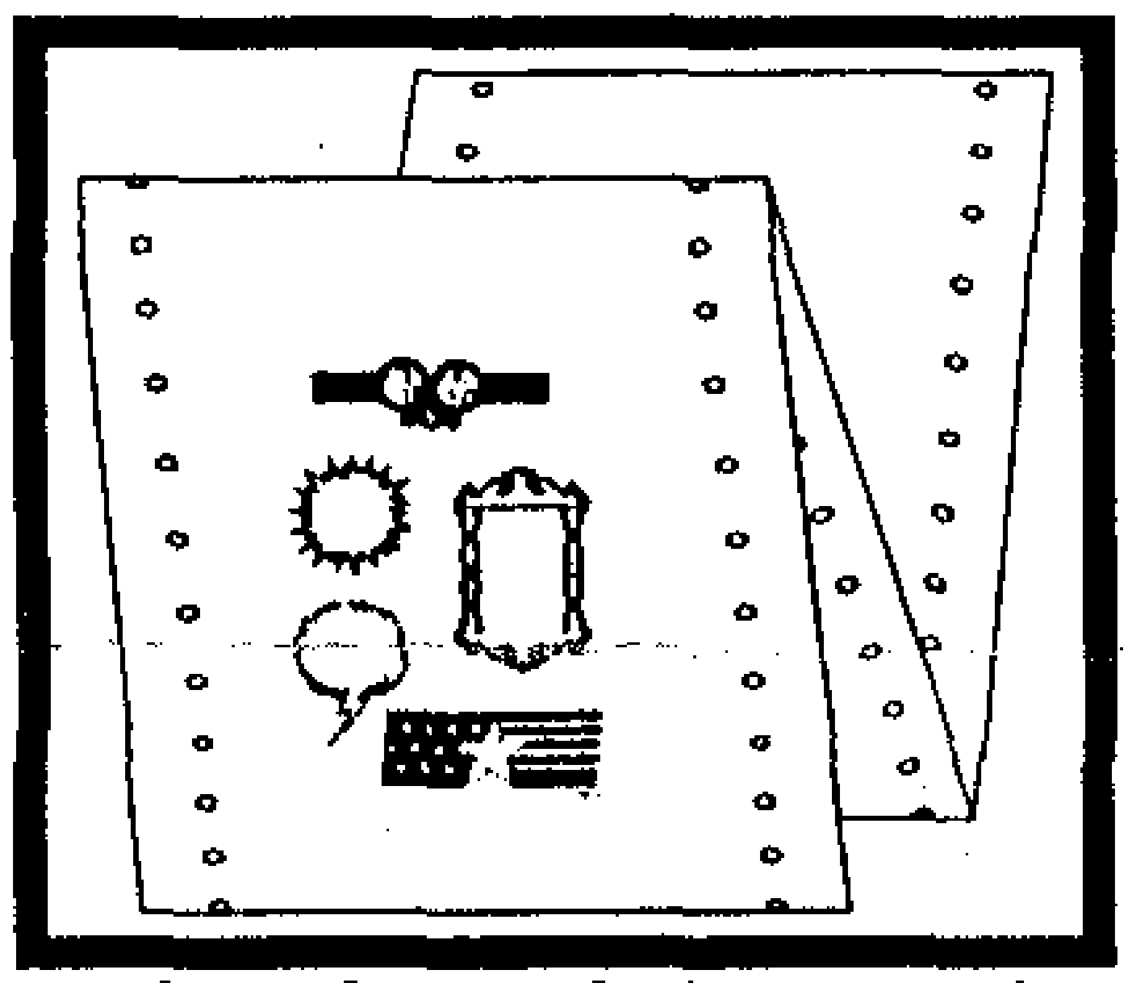
IMPORTANT INFORMATION

SPECIAL REPORT

BY: PAUL A. BROCK

This is a column for those that don't get the Computer Monthly. There is an article written by Barry A. Traver that I haven't seen anywhere else, there is some good information in this article. I thought that I might pass some of that information on to the readers of the WEST PENN 99'ers. the following information comes from Barry's column in the June 1991 issue, called TI-99ers in the '90s. I am coping it and hope that I don't make any typing errors.

In addition to commercial programs published by Asgard, Page Pro 99 has received fairware support as well. Let me mention a few products and addresses. Ed Johnson (399 S. Lexington Ave.,St Paul, MN. 55105) has released TIPS2PP (converts tips pictures to Page Pro 99 format) and Page Pro Font Editor(creates new Page Pro Fonts. Paul Scheidemantle (2762 Lovington,Troy,MI 48083) has made available a Page Pro Picture Cataloger (creates a catalog of Page Pro Pictures). Bill Gaskill has come out with a Page Pro Editor/formatter (helpful, according to Chris Bobbitt, in generating text for newsletters) And Chris Bobbitt himself has released Medical Clipart."an extensive collection...useful for physicians and students. " (incidentally, this information on fairware products comes from a helpful "Page Pro Productivity Chart produced by Chris.)


Paul is running out of printer paper fast with his big interest in graphics!

I haven't seen the actual program yet, but I understand that Joe Delekto's Screen Preview Program is now available (including disk and 12 page manual) from Asgard for $12.95 plus $2.50 shipping. from the description it sounds a bit like Harry Wihelm's Paper Saver program (in that it allows you to see on the screen in miniature what your formatted printout will look like), but Asgard's Screen Preview sounds more powerful (in that Asgard's program actually allows you to make changes in the text while running the program).

Screen Preview is a replacement for the TI-Writer formatter (it supports most but not all TI-Writer formatter commands) and runs from the TI-Writer Utility option (or any equivalent). Here's how Asgard describes the program: "This program... will format your text file, with embedded TI-Writer formatter commands, to the screen in a miniature format,... You can view an entire page at a glance, checking margins, page breaks and other formatting. if you see an error, you can point to the line that needs correction and change the text on the fly. Finally, when the page is done to your satisfaction,press a key and it's printed on your printer, and you can move on to the next."

According to Asgard, Screen Preview not only works with standard floppy disks, but "is also compatible with hard drive systems and most RAM disks." If (like many people) you use your computer mostly for word processing, you could find this product to be very helpful. I plan on giving it a try myself. <END>

The following are my comments...

Most of this information was explained at the meeting, but I missed it somewhere! Maybe someone else missed the information, then this will be of some help. Since the beginning I have been using Asgard's products, After I got my printer I went crazy. You might say that I am a graphic freak. I got a pile of pictures. I am having a lot of problems with my equipment, but thanks to the WEST PENN U.G. I am getting them worked out.

R2D2&TI2 MADE IN USA

-WP♦

AMORTIZATION---TI TO THE RESCUE
by
Wayne Garrison

I had an interesting situation come up last week that I thought would be of interest to my fellow TI99ers. One of my co-workers relocated to the St.Louis area about 2 years ago from the Detroit area due to the declining American auto sales and the ensuing lay-offs. Upon leaving Detroit he sold his house up there rather hastily and financed the deal himself. The new owner was making regular monthly payments and all was fine. Well, all of a sudden, one day the seller got a phone call from the buyer saying that he had come into a large sum of money and was prepared to make a final payment in full for the balance of the loan. Ray, my co-worker, knew I am a computer enthusiast. He has an IBM compatible Epson computer. He said he needed an amortization schedule printed up for the purpose of determining exactly how much the final payment on his house would be. He didn't have any software for his computer that would perform this task and he needed this schedule in a hurry. He gave me all the particulars about the loan and so that night I sat down at my 99/4a and plugged in the numbers and in short order my printer was spittin' out this desperately needed print-out. I ran off two of them so each party would have their own copy. The next day I showed up at work with what he needed. He was impressed to say the least. "Say, can you give me a copy of the software you used to run this off with?", he said. "Sure!", I told him, "But I don't think it will do you any good. You see I ran this on my TI-99/4a." "You're kiddin'." he remarked. Right about then my chest puffed up with pride and I had a grin from ear to ear.

This is not to say that he couldn't have done it with his machine, but he didn't have the right software and the purchaser was going to get back with him the next day. Dat-Da-Da-Da, TI TO THE RESCUE. This sort of thing just makes my day. Most people who use IBM compatible computers, sort of look down on our machines as if they were a toy computer with little or no capability. I have heard them remark, "Why don't you step up to the IBM world?" Well, although I realize the differences between the two machines, for many, this may not necessarily be a step up. For the average home computer user, the TI Home Computer will do about anything he or she needs it to do. Not only that, who can afford to switch these days. I'm sure by now all of our TIs are paid for.

Texas Instruments sold a solid state module for the 99/4a called Personal Real Estate. This is an excellent module for anyone in the market for a home. It allows the user to punch in numbers to determine monthly payments and gives the bottom line on how much you end up paying for the home you wish to buy. It will help you decide what sort of price range you can afford and lets you print out an amortization schedule. For those who do not know what an amortization schedule is, it is a print-out of a loan on a house which breaks down the monthly payment to show how much money goes toward the principle of the loan, how much goes toward the interest for each month because these two amounts change with each payment. As the loan progresses, the amount in the principle column goes up and the amount in the interest column goes down.

This schedule calculates the total amount paid toward the principal each year and also gives the same information about the interest paid. This module was not as popular as the Parsec and Invaders modules and at the time it was also more expensive, so there are many who don't have this module in their own personal library. I didn't have it for quite a long time.

My employer had a Radio Shack Model One computer in the shop for us to use which had an amortization program. I dumped the program to a printer and started, line for line, translating this program to TI Basic. When I was finished I had a program which I could use at home on my TI. The original program would run the schedule on the computer screen. I went a step further, I wrote it to make a hard copy on the printer. Later, I found one written for the TI listed in a computer magazine. It also did not provide for a printed copy of an amortized schedule, so I sat down and revised it so that it would. The one I found in the magazine was from the old 99er Home Computer Magazine and the program name was Loan Calculator. It is a good program to use for those who don't have the Personal Real Estate module. I'm not sure, but it is quite possible that you can find it in our club software library. If not I will make it available. Though the program and magazine is copyrighted, the Emerald Valley Publishing Company no longer exists, at least I wasn't able to find them in the Thomas Register which is a national register for all legitimate businesses. If you are fortunate to have this program in your library, I have listed the changes and additions necessary to print out an amortization schedule. I make no legal claim to these revisions, so if you would like to change it in some way, feel free to play with it.

For those of you who have the Personal Real Estate module, you may notice a similarity of the printed amortization schedule to that of the one generated with my listed program revision. Well, the one I originally wrote, which was a translation from the Radio Shack program, was not as neat and did not offer the yearly totals, so, upon my acquiring the TI module I changed my print-out routine to somewhat mimic the module. If you should decide to get this program (Loan Calculator) from our librarian or a friend, I recommend that you type these lines into the program so you can see what changes are necessary to achieve the schedule print-out. It shouldn't take you long compared to how long it took me to write it. Do it! This is a great tool if you are in the market to buy a home now or in the future.

Chances are, you may not run into the same situation as I did with my co-worker and his little emergency, but like I said, it is a useful tool if you don't have the TI module and who knows, you may be able to impress someone who has what they refer to as a "real" computer. I know you will. When my wife and I were in the market for a home, we got a lot of "OOO"s and "AHH"s from people at the real estate office and especially the bank when I produced an amortization schedule upon our approval for a loan. They told us we would have to wait a couple of weeks to get a print-out from them at a greatly inflated price. I had one the next day thanks to my Home Computer.

Buying a home? Financing a home? Wouldn't it be nice to have an amortization schedule? Have no fear, TI is here.

Happy Computing!

--- Changes And Additions To Loan Calculator ---

```
Change  170 REM REVISIONS BY:Wayne Garrison - ST.Louis 99ers rev.050991
       1660 DISPLAY AT(12,1):"SHOW SCHEDULE FROM": :"PAYMENT #" :: DISPLAY AT(24,3):"ENTER -0- FOR A PRINTOUT" :: ACCEPT
            AT(14,10)BEEP:STRT
       1670 IF STRT=0 THEN 2000 :: DISPLAY AT(16,1):"TO PAYMENT #" :: ACCEPT AT(16,13)BEEP:STP
       1830 DISPLAY AT(23,2):"PRESS: P - FOR A PRINT OUT" :: DISPLAY AT(24,1):"       R - TO RETURN"

ADD     172 DIM M$(13)
       1834 CALL KEY(0,K,S) :: IF S=0 THEN 1830 :: IF K=82 THEN 210 :: IF K=80 THEN 2000
       1836 GOTO 1834
       2000 CALL CLEAR :: DISPLAY AT(10,2):"ENTER MONTH LOAN STARTS" :: DISPLAY AT(16,4):"EXAMPLE: 1 -- FOR JAN. :: DISPLAY
            AT(17,13):"12 -FOR DEC."
       2010 M$(1)="JAN" :: M$(2)="FEB" :: M$(3)="MAR" :: M$(4)= "APR" :: M$(5)="MAY" :: M$(6)="JUN" :: M$(7)="JUL" ::
            M$(8)="AUG"
       2020 M$(9)="SEP" :: M$(10)="OCT" :: M$(11)="NOV :: M$(12)= "DEC"
       2030 ACCEPT AT(10,26)BEEP VALIDATE("123456789101112"):B
       2040 CALL CLEAR :: DISPLAY AT(10,2):"ENTER THE YEAR LOAN STARTS" :: ACCEPT AT(12,12)BEEP VALIDATE("0123456789"):C
       2050 CALL CLEAR :: DISPLAY AT(10,2):"MAKE SURE PRINTER IS READY" :: DISPLAY AT(12,3):"PRESS ANY KEY TO START"
       2060 CALL KEY(0,K,S) :: IF S=0 THEN 2060
       2100 OPEN #1:"PIO" :: PRINT #1 :: PRINT #1 :: PRINT #1 :: PRINT #1:TAB(29):"AMORTIZATION SCHEDULE"
       2110 PRINT #1 :: PRINT #1 :: PRINT #1 :: PRINT #1 :: PRINT #1:"LOAN DESCRIPTION:  MORTGAGE"
       2120 PRINT #1 :: PRINT #1 :: PRINT #1:"LOAN AMOUNT = $"; LOAN :: PRINT #1:"MONTHS FINANCED = ";N :: PRINT #1: "INTEREST RATE
            = ";IN*1200
       2130 PRINT #1:"MONTH LOAN BEGINS = ";M$(B) :: PRINT #1: "YEAR LOAN BEGINS = ";C :: PRINT #1:"MONTHLY PAYMENT = $";PAY
       2140 C=C-1
       2160 GOSUB 2500 :: REM SUB TO PRINT BLANK LINE, YEAR, AND COLUMN HEADINGS
       2170 Z=1 :: K=(1+IN)^(-(Z-1)) :: L=1/K*(PMT*(K-1)/IN+LOAN)
       2200 FOR Z=1 TO N
       2210 K=(1+IN)^(-Z) :: BAL=1/K*(PMT*(K-1)/IN+LOAN)
       2220 I=BAL-L+PAY :: L=BAL ::PR=PAY-I
       2230 REM MI=MONTHLY INTEREST, MP=MONTHLY PRINCIPLE, MB=MONTHLY BALANCE
       2240 MI=INT(100*I+.5)/100 :: MP=INT(100*PR+.5)/100
       2250 MB=INT(100*BAL+.5)/100 :: YTP=YTP+MP :: YTI=YTI+MI
       2260 PRINT #1:M$(B);TAB(7):MP;TAB(21):MI;TAB(32):MB :: B=B+1
       2270 IF B=13 THEN GOSUB 2400 :: REM SUB 2400 PRINTS YEARLY TOTALS
       2272 IF Z=N THEN 2280 :: IF B=1 THEN GOSUB 2500 :: REM SUB 2500 PRINTS YEARLY HEADER
       2280 NEXT Z
       2290 GOTO 2600 :: REM 2600 IS ROUTINE TO PRINT FINAL TABULATIONS AND END OF REPORT
       2400 PRINT #1 :: PRINT #1:"YEAR";TAB(7):YTP;TAB(21):YTI; TAB(32):MB
       2410 PRINT #1:"TOTAL PAYMENTS = ";TAB(18);YTP+YTI
       2412 LTP=LTP+(YTP+YTI) :: LTI=LTI+YTI :: REM KEEPS TAB ON TOTALS FOR DURATION OF THE LOAN
       2420 YTP=0 :: YTI=0 :: B=1 :: REM RESET TOTALS TO ZERO PLUS RESET M$(B) TO JAN
       2430 RETURN
       2500 PRINT #1 :: C=C+1 :: PRINT #1:"------------ YEAR  ";C; "------------"
       2510 PRINT #1:"MONTH";"  PRINCIPAL";"     INTEREST";" BALANCE" :: RETURN
       2600 PRINT #1 :: PRINT #1:"------------ TOTALS ------------" :: PRINT #1:"TOTAL PAYMENTS ON LOAN"
       2610 PRINT #1:TAB(13):LTP :: PRINT #1:"TOTAL INTEREST PAID" :: PRINT #1:TAB(13):LTI
       2620 PRINT #1:"TOTAL PRINCIPAL PAID" :: PRINT #1:TAB (13); LOAN :: PRINT #1:"--------------------------------" :: PRINT
            #1:"END OF REPORT"
       2630 CLOSE #1 :: LTP=0 :: LTI=0 :: GOTO 210
```

FROGRAMMING MUSIC THE EASY WAY

?ART 2

ɔy Jim Peterson

In Part 1 I showed you how to set up a musical scale to create notes, and ɿow to merge in various little routines to create a variety of musical effects, ɔut I didn't tell you how to figure out what numbers to put in between those 5OSUBs. So, here is the little program that makes it all easy.

```
100 CALL CHAR(127,"000F080F0
868F870000F08080868F87000080
8080868F870000808080868987O"
):: CALL CHAR(131,"000000000
0609070")
110 CALL CHAR(132,"0000120C4
83020400000221C0810200000201
0201030200000003CFF"):: CALL
 CHAR(136,"000000FF3C")
120 CALL CLEAR :: S$="GFEDCB
A" :: CALL CHAR(45,"0000000
FF"):: A$=RPT$(S$,3):: FOR R
=2 TO 22 STEP 2 :: IF R=12 T
HEN 130 :: DISPLAY AT(R,1):R
PT$("-",28)
130 NEXT R :: CALL CHAR(98,"
0020202B34242830")
140 FOR R=1 TO 21 :: DISPLAY
 AT(R,1):SEG$(A$,R,1);:: NEX
T R
150 DATA 127,127,128,128,129
,129,130,130,131,131
160 DATA 1/16,1/8,1/4,1/2,1/
1
170 FOR R=1 TO 20 STEP 2 ::
READ N :: DISPLAY AT(R,15):C
HR$(N);:: NEXT R :: FOR R=3
TO 19 STEP 4 :: DISPLAY AT(R
,16):".";:: NEXT R
180 C=132 :: FOR R=1 TO 17 S
TEP 4 :: DISPLAY AT(R,17):CH
R$(C);:: C=C+1 :: NEXT R
190 FOR R=1 TO 17 STEP 4 ::
READ M$ :: DISPLAY AT(R,20):
M$;:: NEXT R
200 DATA 35,33,32,30,28,27,2
5,23,21,20,18,16,15,13,11,9,
8,6,4,3,1
210 FOR R=1 TO 21 :: READ N
:: N$=N$&CHR$(N):: DISPLAY A
T(R,6):STR$(N);:: NEXT R
220 G$="b" :: Z=-1 :: GOSUB
320 :: IF F=0 THEN 230 ELSE
GOSUB 330 :: GOTO 240
230 G$="#" :: Z=1 :: GOSUB 3
20 :: IF F<>0 THEN GOSUB 330
240 DISPLAY AT(24,1):"Shorte
st note? 1/" :: ACCEPT AT(24
,18)VALIDATE("12468")SIZE(2)
BEEP:L :: T$="1/"&STR$(L)::
RESTORE 160 :: FOR J=1 TO 5
:: READ L$ :: IF L$=T$ THEN
260
250 NEXT J :: GOTO 240
260 DISPLAY AT(24,1):"Is it
dotted? Y/N" :: ACCEPT AT(24
,19)VALIDATE("YN")SIZE(1):D$
 :: D=1-(D$="Y")
270 T=-3+J*4
280 FOR R=T TO 19 STEP 4 ::
DISPLAY AT(R,11):STR$(D);::
DISPLAY AT(R+2,11):STR$(D*1.
5);:: D=D*2 :: NEXT R
290 GOTO 360
300 FOR R=1 TO 20 STEP 2 ::
READ N :: DISPLAY AT(R,15):C
HR$(N);:: NEXT N
310 GOTO 310
320 DISPLAY AT(24,1):"How ma
ny "&G$&" on upper scale?" :
: ACCEPT AT(24,28)VALIDATE("
01234567")SIZE(1)BEEP:F :: R
ETURN
330 Y$="" :: FOR J=1 TO F ::
 DISPLAY AT(24,1):"On which
letter?"
340 ACCEPT AT(24,18)VALIDATE
(S$)SIZE(1)BEEP:L$ :: IF POS
(Y$,L$,1)<>0 THEN 340 ELSE Y
$=Y$&L$
350 S=1 :: FOR K=1 TO 3 :: P
=POS(A$,L$,S):: DISPLAY AT(P
,2):G$;:: DISPLAY AT(P,6):ST
R$(ASC(SEG$(N$,P,1))+Z);:: S
=P+1 :: NEXT K :: NEXT J ::
RETURN
360 OPEN #1:"PIO" :: FOR R=1
 TO 22 :: FOR C=3 TO 30 :: C
ALL GCHAR(R,C,G):: CALL HCHA
R(R,C,30):: R$=R$&CHR$(G)::
NEXT C :: PRINT #1:R$ :: R$=
"" :: NEXT R :: STOP
```

Get yourself a piece of sheet music and compare it to the screen display from that program. You will see that music is written on two sets of 5 lines. The upper set is marked at the left end with something like a fancy script capital S; it is used to write the higher notes, including the melody, which a pianist plays with the right hand. The lower set, marked with a sort of a backward C, contains the low notes played with the left hand. Your sheet music probably has a wide space between the sets, to make room for the lyrics, but there are really only three notes between them.

The screen display shows letters at the left, which are not on the sheet music. Those are the names of the notes, which we will have to refer to a couple of times to get started; observe that the notes are named A through G and then repeated.

The numbers along the left side are the numbers you will key in to play those notes. However, the screen display is set up in the key of C, which is played entirely on the piano white keys. The sheet music you want to program from may be in a different key, so -

The computer is asking you how many there are of something that looks like a squashed lower case b - I guess that's why they call it a flat? It means that the note will be played a bit lower, on the black key just left of the white key - and we will program it one number lower. So, look next to that capital S and see how many flats there are. If none, type 0. Otherwise, the computer will ask which letters they are next to. Type them in, one at a time, and presto - the computer will put them on the staff and adjust the numbers accordingly.

If there were no flats, the computer will want to know if there are any sharps - those are what you get by typing a shift 3 on the keyboard, and they mean that the note is played on the black key above the white key, and is programmed one number higher.

Now, the computer needs some information in order to help you set u the length of your notes - how long they are sounded. The various notes ar depicted at the right. A 1/16 note is little black egg with a stem (it may g up or down, makes no difference) and two flags on the stem. A 1/8 has only one flag and a 1/4 note has none. A 1, note is a hollow egg with a stem and a whole note has no stem.

Those little doodads to the right o the notes are rests, used to indicate silent pause of the same length as th note - more on that later.

Look through your sheet music and find the shortest note. Tell the computer. It will want to know if any of those shortest notes are dotted -

have a little dot to their right, as the screen display shows. A dotted note is played half again as long as normal.

Presto again, the computer will show you the duration number to key in for each note. Then, if you have a printer attached, it will print out an XBasic screen dump of that screen - you will have to squash your own b's and sketch in the notes and rests.

If your software library contains an assembly screen dump, delete that last program line and put in a CALL INIT, CALL LOAD and CALL LINK to get a better printout - or ask me for it. If you don't have a printer, why not copy those numbers right onto the corresponding lines and spaces on your sheet music, and number some of the notes.

Now we're ready to make music! Let's keep it simple at first, just a single note melody - and I hope you picked a simple piece of music. Clear the TI's brain with NEW, then merge in that line 100 scale from part 1 by MERGE DSK1.SCALE . In the same way, merge in one of those line 1000 CALL SOUND routines. Put in a temporary stopper line 999 STOP, and a line 110 D=200 to set the duration.

The melody is almost always on the upper set of 5 lines. If a note has 2 or 3 eggs on its stem, as they usually do, the upper one is the melody note - we will get into harmony later.

Start with line 110. Check your chart to see what number denotes the length of the first note - maybe 2, if so key in T=2 :: Then check to see what number applies to the position of the upper egg of that note. Maybe 22, so key in A=22 :: GOSUB 1000 Enter RUN, and if you've done everything correctly, you will hear the note. You might decide already that you want to change that 200 in line 110.

Now for the second note. If it is of the same length as the first, you don't have to type anything - that's what makes this shorthand method so quick and easy. If the note position is also the same, you don't key that in either - just another GOSUB 1000.

If you have EZ-KEYS or another "hot keys" program, you can program a control key to put in the GOSUB 1000 with just one keypress - wish I had thought of that when I was programming music by the diskfull!

So keep plugging along, keying in durations and notes. After every half dozen notes or so, type RUN to see if everything sounds OK so far - it's easier to catch errors before they are too far back in the music.

You can get up to 5 screen lines on one line number, but you might better stick to 3 lines. You will note that the sets of notes are divided by vertical bars. You might program the notes between bars on a separate line, then add a ! followed by the words of the song that go with those notes - I find that a very good way to track down sour notes.

Regarding those bars - it might help you sometime to know this. At the beginning of the music, right after the big script S and the flats and sharps, you will see something like a 3 over a 4, or a 4 over a 4, or whatever - but often a symbol such as a barred C is used instead. A 3 over a 4, for instance, means that the notes between two of those bars will add up to 3/4 - might be three quarter notes, or two eighth notes and two quarter notes, or whatever, but they will add up to 3/4. Sometimes the very first notes will add up short, but in that case the very last ones will make up the difference.

The notes between those two bars make up a bar of music, and the emphasis is on the first note - for instance, that 3/4 is the 1-2-3, 1-2-3 beat of waltz time.

While you are keying in that music, you might come to one of those rests. You can just key in its T= value and then A=0 for a silent note. However, computer notes stop so abruptly that somehow a rest just doesn't sound right, so I often just use the previous note instead.

You may come across one of those flat or sharp symbols next to a note in the music. Give the note a number 1 lower if a flat, one higher if a sharp, and the same for any subsequent occurrences of that note, until you find next to it a symbol that looks like the sharp sign with half its legs knocked off; that means to go back to normal. You might also come across that symbol to tell you to play a normally flat or sharp note as if it was not.

I think that covers all that you absolutely have to know for now, and I have horrified all serious students of music just about enough. There are all kinds of other squiggles on the sheet music but usually they are not essential in programming music.

There is one other time-saving shortcut that I should tell you about right now. Most music consists at least partly of musical phrases, of a series of notes, which are repeated two or more times within a melody. So, the first thing you should do before you start programming a song is to search through the music for such phrases.

If you find one, of more than a few notes, that is repeated elsewhere - and make sure it is repeated exactly the same - mark it off each place it occurs and label it 500. If you find a second repeating phrase, label it 600, and so on.

Then, when you start programming, start with line 500, key in that series of notes first, and end it with RETURN. If you have another phrase, put it in lines starting with 600, again ending with RETURN.

Now, start programming from the beginning of the song in line 120, but when you come to one of those phrases, just put in GOSUB 500 - the program will jump to that line number, play those notes, and come right back to where it was.

In Part 3, we will get into programming in 3-part harmony, bass notes, auto-chording, and all kinds of things.

TIPS FROM THE TIGERCUB

#55

Tigercub Software
156 Collingwood Ave.
Columbus OH 43213

I am still offering over 120 original and unique entertainment, educational and utility programs at just $1.00 each, or on collection disks at $5.00 per disk.

The contents of the first 52 issues of this newsletter are available as ready-to-run programs on 5 Tips Disks at $10 each.

And my three Nuts & Bolts Disk, $15 each, each contain over 100 subprograms for you to merge into your own programs to do all kinds of wonderful things.

My catalog is available for $1, deductable from your first order (specify TIGERCUB catalog).

################################

TI-PD LIBRARY

I have selected public domain programs, by category, to fill over 200 disks, as full as possible if I had enough programs of the category, with all the Basic-only programs converted to XBasic, with an E/A loader provided for assembly programs if possible, instructions added and any obvious bugs corrected, and with an autoloader by full program name on each disk. These are available as a copying service for just $1.50 postpaid in U.S. and Canada. No fairware will be offered without the author's permission. Send SASE for list or $1, refundable for 9-page catalog listing all titles and authors. Be sure to specify TI-PD catalog.

################################

The Tigercub has dipped a cautious paw into the cold dark mysterious waters of asembly, while still keeping a firm grip on trusty old Extended Basic. The result is an XBasic program that writes an assembly program!

The following subprogram, when merged into any program which has reidentified characters, and called after the characters have been reidentified, will write a source code which can be assembled into object code, loaded from XBasic and linked to instantly access the character set.

The source code is based on 2FONTS/S by Barry Traver, who gives credit to Mac McCormick, David Migicovsky and Karl Schuneman.

```
19000 SUB CHARSUB(HX$())
19001 DISPLAY AT(12,1)ERASE
ALL:"Source code filename?":
"DSK" :: ACCEPT AT(13,4)SIZE
(12)BEEP:F$ :: OPEN #1:"DSK"
&F$,OUTPUT
19002 DISPLAY AT(15,1):"LINK
ABLE program name?" :: ACCEP
T AT(16,1)SIZE(6):P$
19003 DISPLAY AT(18,1):"Rede
fine characters from    ASCI
I     to ASCII"
19004 ACCEPT AT(19,7)VALIDAT
E(DIGIT)SIZE(3):F
19005 ACCEPT AT(19,21)VALIDA
TE(DIGIT)SIZE(3):T
19006 PRINT #1:TAB(8);"DEF";
TAB(13);P$ :: PRINT #1:"VMBW
   EQU  >2024" :: PRINT #1:"
STATUS EQU  >837C"
19007 NB=(T-F+1)*8 :: CALL D
EC_HEX(NB,H$):: A=768+F*8 ::
 CALL DEC_HEX(A,A$)
19008 FOR CH=F TO T :: IF CH
<144 THEN CALL CHARPAT(CH,CH
$)ELSE CH$=HX$(CH)
19009 IF FLAG=0 THEN PRINT #
1:"FONT";:: FLAG=1
19010 FOR J=1 TO 13 STEP 4 :
: M$=M$&">"&SEG$(CH$,J,4)&",
" :: NEXT J :: M$=SEG$(M$,1,
23)&"  *"&CHR$(CH)
19011 PRINT #1:TAB(8);"DATA
"&M$ :: M$="" :: NEXT CH
19012 PRINT #1:P$;TAB(8);"LI
   R1,FONT" :: PRINT #1:TAB(
8);"LI   R0,>"&A$ :: PRINT #
1:TAB(8);"LI   R2,>"&H$
19013 PRINT #1:TAB(8);"BLWP
@VMBW":TAB(8);"CLR  @STATUS"
:TAB(8);"RT":TAB(8);"END" ::
 CLOSE #1
19014 SUBEND
19015 SUB DEC_HEX(D,H$)
19016 X$="0123456789ABCDEF"
:: A=D+65536*(D>32767)
19017 H$=SEG$(X$,(INT(A/4096
)AND 15)+1,1)&SEG$(X$,(INT(A
/256)AND 15)+1,1)&SES$(X$,(I
NT(A/16)AND 15)+1,1)&SEG$(X$
,(A AND 15)+1,1):: SUBEND
```

Now to try it out. You probably know that CALL CHARSET will restore reidentified characters below ASCII 96 to normal form, but not those above, so let's write a routine to restore those. Clear the memory with NEW, merge in the above, which you should have SAVED with -
SAVE DSK1.CHARSUB,MERGE
by MERGE DSK1.CHARSUB. Add a line -
100 CALL CHARSUB(HX$()) and RUN. Answer the filename prompt with DSK1.OLDLOW/S, the next prompt with OLDLOW and select ASCII 97 to 127.

When done, insert the Editor/Assembler module and its disk Part A. Select Assembler, Y to load assembler, give the source code DSK1.OLDLOW/S, object code DSK1.OLDLOW/O, just press Enter at next prompt, and R for options. You should get 0000 ERRORS.

Now key in this routine to test your program.

```
100 CALL INIT :: CALL LOAD("
DSK1.OLDLOW/O"):: FOR CH=33
TO 126 :: CALL CHAR(CH,"FF81
818181818181FF"):: PRINT CHR$(
CH);:: NEXT CH
101 CALL KEY(0,K,S):: IF S=0
 THEN 101 ELSE CALL CHARSET
102 CALL KEY(0,K,S):: IF S=0
 THEN 102 ELSE CALL LINK("OL
DLOW")
110 GOTO 110
```

Press any key to restore the upper case characters by CALL CHARSET, any key again to use the CALL LINK.

You are now ready to use the routine to copy all kinds of character sets from the programs in your library. You don't have any such programs? Not to worry. You don't have to reidentify characters one by one with one of those graphics editor programs. You can just manipulate the existing hex codes of the normal characters. I have created nearly 50 different character sets by that method!

The space occupied by a character on the screen is really an 8x8 square of 64 tiny dots. Various dots are turned on (colored) and off (transparent) to create a pattern - just the opposite of light bulbs on a scoreboard.

And those on-and-off dots are really the binary numbers which the computer uses. But fortunately the computer lets us use hexadecimal numbers rather than binary. The following will print out a reference chart of decimal to binary to hexadecimal. You can easily convert it to dump to a printer.

```
10 DISPLAY AT(6,1)ERASE ALL:
"DEC BIN HEX"
100 FOR J=0 TO 15 :: CALL DE
C_BIN(J,B$):: CALL DEC_HEX(J
,H$):: DISPLAY AT(J+8,1):J;T
AB(5);B$;TAB(10);SEG$(H$,4,1
):: NEXT J
21020 SUB DEC_BIN(D@,B$):: D
=D@ :: IF D=0 THEN B$="0000"
 :: SUBEXIT
21021 IF D=1 THEN 21022 :: X
=D/2 :: B@$=STR$(ABS(X<>INT(
X)))&B@$ :: D=INT(X):: IF D>
1 THEN 21021
21022 B@$="1"&B@$ :: B$=RPT$
```

```
("0",4-LEN(B@$))&B@$ :: B@$=
"" :: SUBEND
21039 SUB DEC_HEX(D,H$)
21040 X$="0123456789ABCDEF"
:: A=D+65536*(D>32767)
21041 H$=SEG$(X$,(INT(A/4096
)AND 15)+1,1)&SEG$(X$,(INT(A
/256)AND 15)+1,1)&SEG$(X$,(I
NT(A/16)AND 15)+1,1)&SEG$(X$
,(A AND 15)+1,1):: SUBEND
```

And this routine will show you how each letter is formed, by binary 0's (off) and 1's (on), for each key you press. I put it in merge format so you can MERGE it into any program and CALL it to examine the characters.

```
17000 SUB CHARVIEW
17001 !programmed by Jim Pet
erson Feb 1989
17002 DISPLAY AT(1,1)ERASE A
LL:"CHARACTERS IN BINARY & H
EX":;:"Press any key to see
the   binary representation
 of thescreen character and
its hexcode."
17003 DISPLAY AT(8,1):"Press
 Enter to see the character
."
17004 CALL KEY(0,K,S):: IF K
=13 THEN 17005 ELSE IF S=0 O
R K<32 OR K>143 THEN 17004 E
LSE 17007
17005 CALL CHAR(48,"FF"&RPT$
("81",6)&RPT$("FF",9))
17006 CALL KEY(0,K,S):: IF S
<1 THEN 17006 ELSE CALL CHAR
(48,"00384444444444380010301
010101038"):: GOTO 17004
17007 CALL CHARPAT(K,CH$)
17008 R=12 :: FOR J=1 TO 15
STEP 2
17009 H$=SEG$(CH$,J,1):: CAL
L HEX_BIN(H$,B$)
17010 DISPLAY AT(R,8):B$
17011 H$=SEG$(CH$,J+1,1):: C
ALL HEX_BIN(H$,B$)
17012 DISPLAY AT(R,12):B$ ::
 DISPLAY AT(R,18):SEG$(CH$,J
,2):: R=R+1 :: NEXT J :: DIS
PLAY AT(22,5):CH$ :: GOTO 17
004
17013 SUBEND
17014 SUB HEX_BIN(H$,B$):: H
X$="0123456789ABCDEF" :: BN$
="0000X0001X0010X0011X0100X0
101X0110X0111X1000X1001X1010
X1011X1100X1101X1110X1111"
17015 FOR J=LEN(H$)TO 1 STEP
 -1 :: X$=SEG$(H$,J,1)
17016 X=POS(HX$,X$,1)-1 :: T
$=SEG$(BN$,X*5+1,4)&T$ :: NE
XT J :: B$=T$ :: T$="" :: SU
BEND
```

And to reidentify a character, you just change the numbers and letters in the 16-digit hex code which represents the binary pattern. By writing little routines to switch those digits around, all kinds of things can be done.

For instance, the normal characters always have the top row of dots turned off, to provide spacing between lines of text on the screen. If you want taller characters you will have to double-space the lines, but you can create them by making the numerals and upper case characters consist of the 2nd-7th rows, the 7th row again, and the 8th row - it just happens to work out.

```
18000 SUB HIGHCHAR :: FOR CH
=48 TO 90 :: CALL CHARPAT(CH
,CH$):: CALL CHAR(CH,SEG$(CH
$,3,10)&RPT$(SEG$(CH$,13,2),
2)&SEG$(CH$,15,2)):: NEXT CH
 :: SUBEND
```

I made that a subprogram so you can MERGE it in and use it to modify other character sets.

If we take the hex code apart, 2 digits at a time, and reassemble it backward,

```
100 CALL CLEAR :: FOR CH=33
TO 90 :: CALL CHARPAT(CH,CH$
):: FOR J=1 TO 15 STEP 2 ::
CH2$=SEG$(CH$,J,2)&CH2$ :: N
EXT J :: CALL CHAR(CH,CH2$):
: CH2$="" :: NEXT CH
110 DISPLAY AT(12,1):""NWOD
EDISPU":"VT EHT DENRUT OHW !
YEH" :: GOTO 110
```

That one was in my first Tips newsletter, years ago, but it is much more effective at assembly speed.

This one shades characters on their left edge by turnon the pixel to the left of the leftmost "on" pixel, if any. Also try it in combination with HIGHCHAR.

```
18001 SUB NEWCHAR3 :: FOR CH
=48 TO 122 :: CALL CHARPAT(C
H,CH$):: FOR J=1 TO 15 STEP
2
18002 CH2$=CH2$&SEG$("0367CD
EF",POS("01234567",SEG$(CH$,
J,1),1),1)&SEG$(CH$,J+1,1)::
 NEXT J :: CALL CHAR(CH,CH2$
):: CH2$="" :: NEXT CH :: SU
BEND
```

This one uses HIGHCHAR to heighten the character and then blanks out three rows. Try following it with NEWCHAR3.

```
18030 SUB NEWCHAR10 :: A$="0
0" :: FOR CH=48 TO 90 :: CAL
L CHARPAT(CH,CH$):: CH$=SEG$
(CH$,3,10)&RPT$(SEG$(CH$,13,
2),2)&SEG$(CH$,15,2)
18031 CH$=SEG$(CH$,1,4)&A$&S
EG$(CH$,7,2)&A$&SEG$(CH$,11,
2)&A$&SEG$(CH$,15,2):: CALL
CHAR(CH,CH$):: NEXT CH :: SU
BEND
```

The next one, which works only on ASCII 97-122, makes tall characters ridiculously elongated above.

```
18050 SUB NEWCHAR20 :: FOR C
H=97 TO 122 :: CALL CHARPAT(
CH,CH$):: CALL CHAR(CH,SEG$(
CH$,7,2)&RPT$(SEG$(CH$,9,2),
4)&SEG$(CH$,11,6)):: NEXT CH
 :: SUBEND
```

This one has the character raised by one line, widened one column at left and two columns at right to make a full 8x8 character which must be double-spaced horizontally and vertically.

```
18090 SUB NEWCHAR27 :: FOR C
H=48 TO 122 :: CALL CHARPAT(
CH,CH$):: CH$=SEG$(CH$,3,10)
&RPT$(SEG$(CH$,13,2),2)&SEG$
(CH$,15,2):: FOR J=1 TO 15 S
TEP 2
18091 CH2$=CH2$&SEG$("014589
CD",POS("01234567",SEG$(CH$,
J,1),1),1)&SEG$("0129",POS("
048C",SEG$(CH$,J+1,1),1),1)
18092 NEXT J :: CALL CHAR(CH
,CH2$):: CH2$="" :: NEXT CH
:: SUBEND
```

Those who have my Nuts & Bolts disks will see how valuable this assembly can be to make instantly available the routines for double height and double width characters, etc., etc. And if you have Todd Kaplan's amazing ALSAVE routine from the Genial Traveler Vol. 1 No. 3, you can imbed them in your XBasic program for fast loading.

And you can merge CHARSUB into any character editor or sprite defining program and, with a bit of modification, use it to convert your creations into fast-loading assembly.

These assembly loads are compatible with my BXB, so you can also load character sets into sets 15 and 16, ASCII 144-159. However, the CHARPAT statement cannot access ASCII above 143, so in this case you must dimension an array in the program you are copying from, as DIM HX$(159), and place the hex codes in the array using the ASCII as the subscript number, such as
CALL CHAR(CH+64,CH$) ::
HX$(CH+64)=CH$, so that they will be passed to the subprogram. And don't CALL INIT after you have called BXB!

So, now you try creating your own screen fonts!

Memory full,

Jim Peterson

NEXT MEETING TUESDAY OCTOBER 8, 1991. FALL IS HERE!!!!!!!!!

MUNCH OFFICERS AND NUMBERS (all in 508 area unless noted)

President	W.C. Wyman	865-9683
Vice President	Bruce Willard	852/3250
Secretary	Jim Cox	
Treasurer	Jim Cox	869-2704
Acting Editor	Jim Cox	
Adv.Prog. Chair	Dan Rogers	248-5502
Library	Al/Lisa Cecchini	
Disk Librarian	Lou Holmes	617 965/3584
Tape Librarian	Walter Nowak	413 436/7675
NEW-AGE/99	Jack Sughrue	476/7630

MUNCH DUES

NEW MEMBERSHIP	$25.00
RENEWAL MEMBERSHIP	$15.00
NEWSLETTER ONLY SUBSCRIPTION	$12.50

SEPTEMBER MEETING. This month's meeting was fun for everyone. We had 18 members and two guests in attendance. Jack did a demo of our Disk of the Month, his Tinygrams demo was very interesting. Corson brought a new twist to the meeting with a meeting newsletter. It was mostly a picture of two "shaady guys" but the future of this new feature is exciteing.

OCTOBER MEETING. This month will feature the new version of Funnelweb Ver. 4.40 with enhancements by Charles Good and Jack Sughrue. Jack will review the configure feature of this program for those interested. I will have blank discs for sale; 100 for $25.00 or 10 for $3.00. I am sure we will have some other surprises also.

RAFFLE. Every month we have a raffle to help defer the rental cost of our meeting hall. A typical raffle will have game and utility programs T-Shirts, books, bumper stickers, blank discs and all sorts of odds and ends for the T.I.

LIBRARY NOTICE. Please return any items borrowed from our library. If you can not come to a meeting or give these items to someone who will be at the meeting.

REPRINTS. Reprints are permitted as long as credit is given to M.U.N.C.H.

ARTICLES. I am always looking for articles for this newsletter, anything which interests you will probably interest other members of the TI community, so please share your ideas and opinions with all of us.

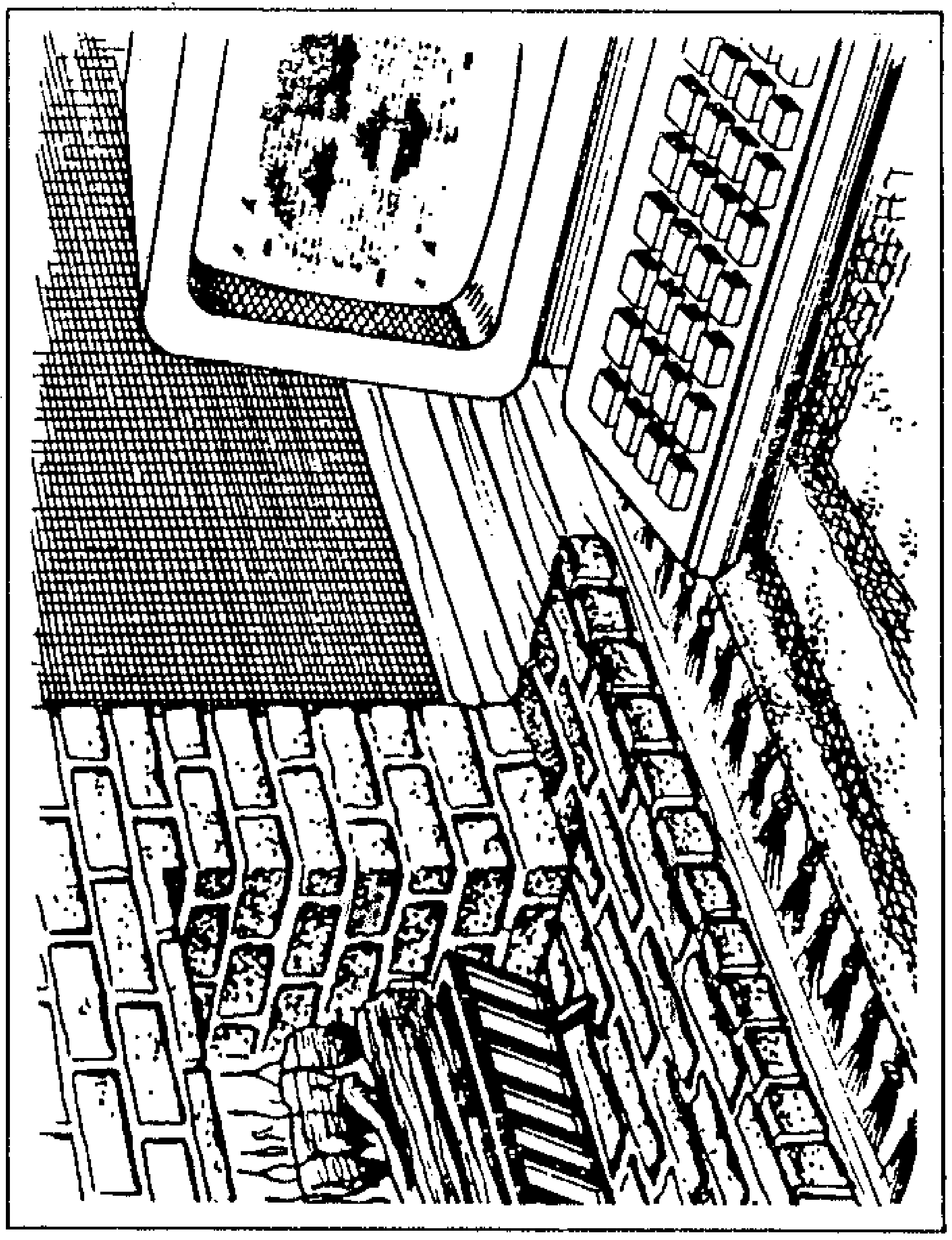
DISK LIBRARY. The disk library will be at the meetings from now on. We have copies of all disks in the library and they are available to members for just $1.50 each for single discs, $2.00 flippies, $3.00 double discs aand $4.00 double flippie.

FOR SALE. The group has a TI Count Business Software package available for sale. If interested contact Jim Cox at the above numer or the club address.

DISK OF THE MONTH. This month's disk is a double flippie of Funnelweb Ver. 4.40 enhanced with additions by Charles Good and Jack Sughrue. This is #49 AB. At the last meeting the group decided to change the price of our discs. A single disc remains $1.50, a flippie is now $2.00, a double disc $3.00 and a double flippie $4.00. We feel these prices are still below those of most groups.

Mass Users of the Ninety-nine and Computer Hobbyists

OCTOBER 1991 Monthly Newsletter Version 10.10



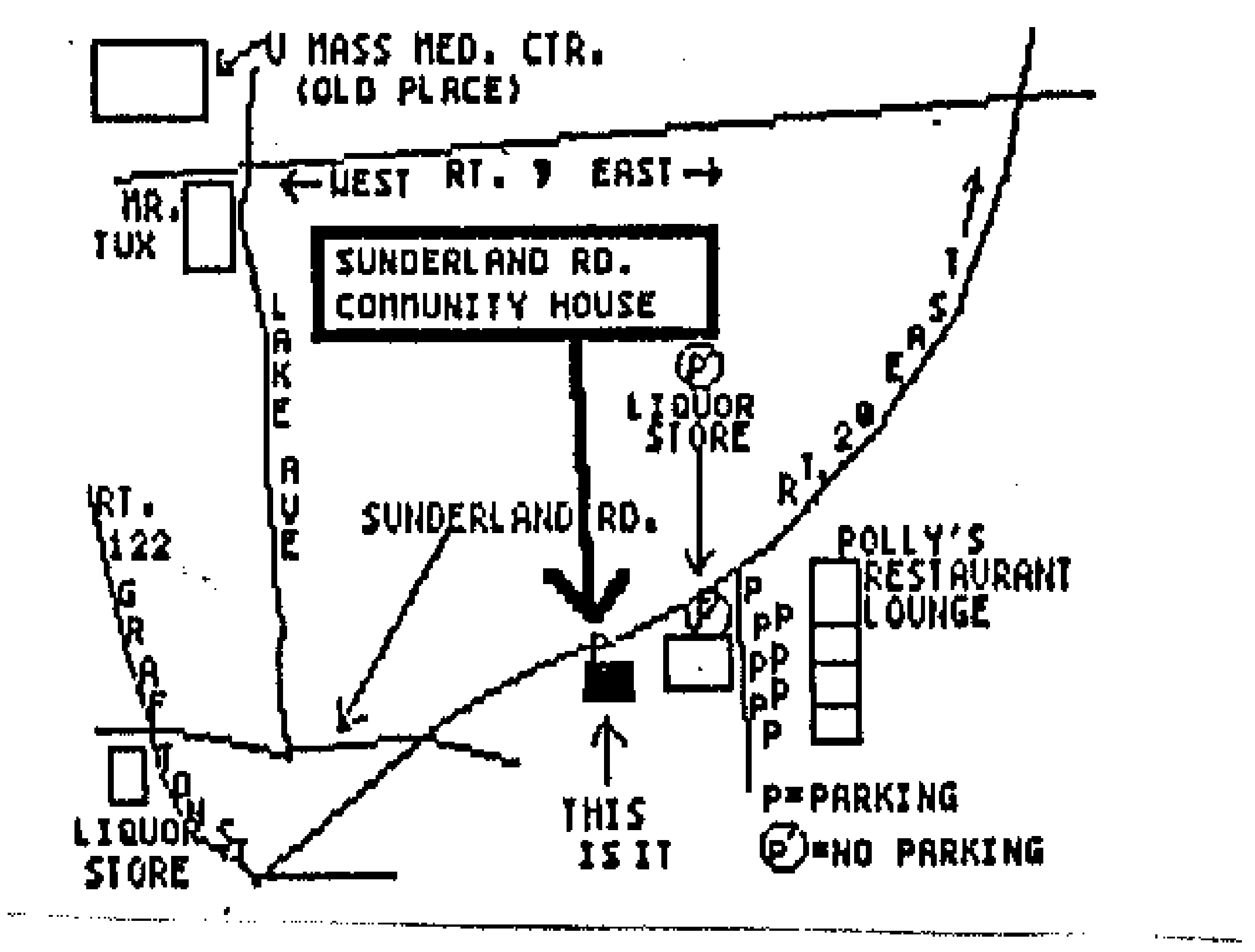
M.U.N.C.H.
560 LINCOLN ST.
P.O. BOX 7193
WORCESTER, MA. 01605-7193

WORCESTER, MA 016
PM
3 OCT
1991

29 USA

FIRST CLASS

Next Meeting OCTOBER 8th



POSTMASTER: Forwarding and Address Correction Requested.