

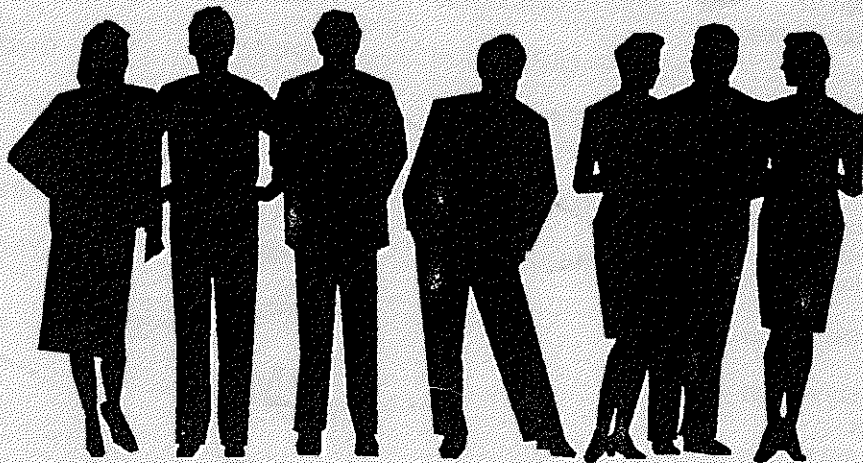
NEWS DIGEST

Focusing on the TI99/4A Home Computer

Volume 14, Number 9

October, 1995

PRINT POST Approved - PP244099/00016



Sydney, New South Wales, Australia

\$3

TiSHUG (Australia) Ltd.
A.C.N. 003 374 383

All correspondence to:
C/o 3 Storey St.
Ryde 2112 Australia

TiSHUG News Digest

ISSN 0819-1984

I N D E X

The Board

Co-ordinator

Dick Warburton (02) 918 8132

Secretary

Thomas Marshall (02) 671 7535

Treasurer

Cyril Bohlsen (02) 639 5847

Directors

Percy Harrison (02) 808 3181

Loren West (047) 21 3739

Sub-committees

News Digest Editor

Loren West (047) 21 3739

BBS Sysop

Ross Mudie (02) 456 2122

BBS telephone number (02) 456 4806

TI Merchandising

Dick Warburton (02) 918 8132

IBM Merchandising

Cyril Bohlsen (02) 639 5847

Software Library

Larry Saunders (02) 644 7377

Technical Co-ordinator

Geoff Trott (042) 29 6629

Regional Group Contacts

Central Coast

Russell Welham (043) 92 4000

Glebe

Mike Slattery (02) 692 8162

Hunter Valley

Geoff Phillips (049) 42 8617

Illawarra

Geoff Trott (042) 29 6629

Liverpool

Larry Saunders (02) 644 7377

Sutherland

Peter Young (02) 528 8775

Membership and Subscriptions

Annual Family Dues \$35.00
Associate membership \$10.00
Overseas Airmail Dues A\$65.00
Overseas Surface Dues A\$50.00

TiSHUG Sydney Meeting

The October Meeting will start at 2.0 pm on the 7th October 1995 at Meadowbank Primary School, Thistle Street, Meadowbank.

Printed by

Kwik Kopy West Ryde

| Title | Description | Author | Page No |
|----------------------------------|------------------|-----------------|---------|
| BABBAGE COMPUTER PIONEER | GENERAL INTEREST | GENE SMITH | 18 |
| DID YOU KNOW | GENERAL INTEREST | LOREN WEST | 4 |
| EDITOR'S COMMENTS | GENERAL INTEREST | LOREN WEST | 13 |
| FILE TYPE ON THE TI 99/4A | HINTS | IRWIN HOLT | 5 |
| LEARN TO KNOW YOUR TI, LESSON 31 | TUTORIAL | PERCY HARRISON | 2 |
| MY COMPUTER AND I | GENERAL INTEREST | LOREN WEST | 8 |
| NEW COMPANY TAKES OVER TEXCOMP | GENERAL INTEREST | LAURA BURNS | 9 |
| PUZZLE | PUZZLE | LOREN WEST | 6 |
| PROGRAM OF THE MONTH | PROGRAM | BOB AUGUST | 10 |
| REGIONAL GROUP REPORT | GENERAL INTEREST | VARIOUS | 23 |
| SNAP CALCULATOR | PROGRAM | TI99ER | 14 |
| SURGE PROTECTION | TECHNICAL | GARY COX | 7 |
| SYDNEY MARITIME MUSEUM | ART UNION | SYDNEY MARITIME | 11 |
| TISHUG SOFTWARE | CLUB NEWS | LARRY SAUNDERS | 13 |
| TREASURERS REPORT | REPORT | CYRIL BOHLSSEN | 10 |
| <u>IBM</u> | | | |
| CPU SPEED | TECHNICAL | CYRIL BOHLSSEN | 19 |
| IBM SHOP | CLUB NEWS | CYRIL BOHLSSEN | 19 |
| SNAP CALCULATOR | PROGRAM | TI99ER | 20 |

LEARN TO KNOW YOUR TI

LESSON 31

with Percy Harrison

Well folks, how did you make out with last month's assignment. Unfortunately, I do not get any feedback from our members so I can only assume that either no one bothers to read and try these lessons which means that my efforts have been a complete waste of time or else those that do follow the lessons find that they work without any problems which means that I have not made any mistakes and this I just can't believe.

The lesson this month treats the numeric values for TRUE and FALSE and should encourage you to use homemade DO WHILE and DO UNTIL constructions.

There are two abstract ideas in this lesson which you may find difficult.

One is that TRUE and FALSE have numeric values of -1 and 0. Any expression which is of the form of an assertion (a "phrase A"), has a number value of 0 or -1. This number can be treated just as any other number. It can be stored in a numeric variable, printed, or used in an expression. Most often it is used in an IF statement.

The other abstract idea compounds the confusion. The IF command doesn't really look to see if "phrase A" is present. Rather, it looks for a numeric value between IF and THEN. Any number which is non-zero is treated as TRUE.

(The value -1 for TRUE may seem artificial. It comes from the idea that FALSE is zero, which is a natural idea. Then NOT FALSE should be TRUE. But the Boolean operation NOT on a number whose digits are all zero gives a number whose digits are all ones. This number has the value -1 in the usual two's complement notation used on the binary level of the computer's operation).

It is easy to build a program using GOTO and IF branching that ends up being a bowl of spaghetti. The programmer must discipline himself or herself to avoid this. A universal scheme limits the programs to single-entry, single-exit blocks. Then those which make conditional branches (IFs) are limited to just two:

DO UNTIL and DO WHILE .

The difference is whether the test for exit is made at

the beginning or at the end of the block.

Enough of the confusing theory, let's get on with the lesson and hopefully learn something worthwhile.

LESSON 31 LOGIC: TRUE AND FALSE

TRUE AND FALSE ARE NUMBERS

The computer says true and false are numbers.

Rule: TRUE is the number -1
FALSE is the number 0

(It is easy to remember that 0 is FALSE because zero is the grade you get if your homework is false (wrong).)

To see these numbers, enter these commands:

```
PRINT 3=7 prints 0 because 3=7 is FALSE
PRINT 3=3 prints -1 because 3=3 is TRUE
```

PUTTING TRUE AND FALSE IN BOXES

The numbers for TRUE and FALSE can be put into boxes just like other variables:

```
Run: 10 N = (3=22)
      20 PRINT N
```

The number 0 is stored in the box N because 3=22 is FALSE.

```
Run: 10 N = "B"="B"
      20 PRINT N
```

The number -1 is stored in the box N because the two letters in the quotes are the same, so the statement "B"="B" is TRUE.

THE IF COMMAND TELLS LITTLE WHITE LIES

The IF command looks like this:

```
10 IF phrase A THEN line number N
```

Try this:

```
10 IF 0 THEN 30
20 PRINT "FALSE"
21 END
30 PRINT "TRUE"
```

It should print FALSE.

Try it again with this line:

```
10 IF -1 THEN 30
```

It should print TRUE.

Now with this:

```
10 IF 22 THEN 30
```

What does it print?.....

Rule: In an IF, the computer looks at "phrase A":

If it is zero, the computer says "phrase A is FALSE", and skips what is after THEN.

If it is not zero, the computer says "phrase A is TRUE", and goes to the line whose number is after THEN.

So the IF command tells little white lies:

TRUE is supposed to be the number "-1".

But the "IF" stretches the truth to say "TRUE is anything that is not FALSE".

That is, any number that is not zero is TRUE.

THE LOGIC SIGN

You can use 6 symbols in the "phrase A":

- = equal
- <> not equal
- < less than
- > greater than
- <= less than or equal to
- >= greater than or equal to

You have to press the SHIFT key and two other keys to make the <> sign and the <= and >= signs.

The last two are new so look at this example to see the difference between < and <=:

```

2<=3 is TRUE      2<3 is FALSE
3<=3 is TRUE      3<3 is FALSE
4<=3 is FALSE     4<3 is FALSE

```

Assignment:

Answer the following questions then type each line into your computer to see if you are correct.

1. Tell what will be found in the box X if:

```

N = 4=4
N = "G"< >"S"
N = 5>7
N = 5>=4

```

2. Tell if the word "JELLYBEAN" will be printed:

```

IF 0 THEN PRINT "JELLYBEAN"
IF 1 THEN PRINT "JELLYBEAN"
IF 9 THEN PRINT "JELLYBEAN"
IF 3<>0 THEN PRINT "JELLYBEAN"
IF "A"="Z" THEN PRINT "JELLYBEAN"
IF 4<=5 THEN PRINT "JELLYBEAN"

```

ONE DOOR TO GO IN AND ONE DOOR TO GO OUT

Use your IF and your GOTO statements carefully to avoid "spaghetti" programs.

Each IF should be used in a block of code which has one entrance and one exit.

There are only two kinds of blocks you need for programming any IF idea:

```

DO UNTIL..... 58 -----
60 REM DOOR IN
62 -----
64 -----
66 -----
68 IF A THEN 60
70 REM DOOR OUT
72 -----

```

The lines in the DO UNTIL block keep repeating until phase A is false.

```

DO WHILE..... 58 -----
60 REM DOOR IN
62 IF A THEN 70
64 -----
66 -----
68 GOTO 62
70 REM DOOR OUT
72 -----

```

The lines in the DO WHILE block keep repeating while phase A is true.

Assignment 3i

1. Write a program to detect a double negative in a sentence. Look for negative words like not, no, don't, won't, can't, nothing, and count them. If there are 2 such words there is a double negative. Test the program

on the sentence "COMPUTERS AIN'T GOT NO BRAINS".

ANSWERS TO LESSON 30

Assignment Question 30-1

```
10 REM ARRAYS
12 DIM D(12)
15 CALL CLEAR
20 FOR I=1 TO 12
22 READ D(M)
24 D(I)=D(M)
25 NEXT I
26 PRINT
27 PRINT
28 PRINT
29 PRINT
30 PRINT "MONTH NUMBER? <1-12>"
31 PRINT
32 INPUT M
33 PRINT
35 PRINT "MONTH NUMBER ";M;" HAS ";D(M);" DAYS."
36 GOTO 26
90 DATA 31,28,31,30,31,30,31,31,30,31,30,31
```

DID YOU KNOW

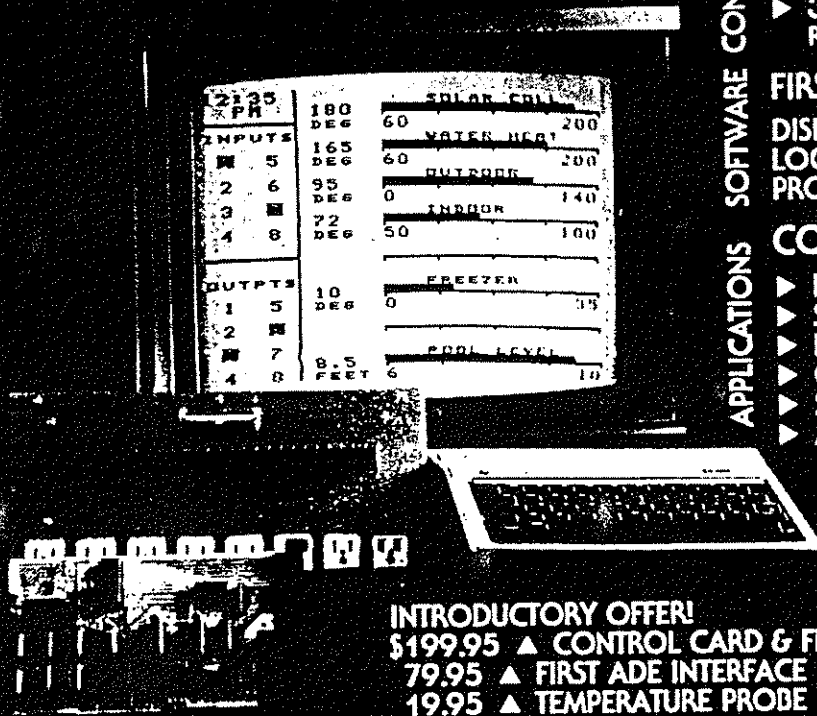
Below is an add I found in an old magazine, 1983 to be exact that's 12 years ago, have a read of the add and see what our TI could do then.

If anyone has one of these units or knows of one in use please drop us a few lines and tell us all about it, and if anyone has other old information about our mighty computer copy it and bring it along to one of the club meetings. (ED)

END OF ARTICLE

FIRST ADE™ FOR YOUR 99/4A!

COMPLETE 8 BIT DATA ACQUISITION AND CONTROL



APPLICATIONS SOFTWARE CONTROL CARD

FIRST ADE™ FEATURES —

- ▶ 8 ANALOG INPUT CHANNELS
- ▶ 8 DIGITAL INPUT CHANNELS
- ▶ 8 DIGITAL OUTPUT CHANNELS
- ▶ REAL TIME CLOCK W/BATTERY
- ▶ 3 MONTH WARRANTY

REQUIRES 32K & EXT. BASIC

FIRST ADE™ VERSION 1.1

DISK BASED/MENU DRIVEN
LOGIC FUNCTIONS
PROGRAMMABLE SETPOINTS

COMPUTER CONTROL:

- ▶ ENVIRONMENT
- ▶ SPRINKLER SYSTEMS
- ▶ POOL/SPA
- ▶ GREENHOUSE
- ▶ SOLAR SYSTEMS
- ▶ ANYTHING!!

A/D ELECTRONICS
Box 26357
Sacramento, California
95826 (916) 363-8331

INTRODUCTORY OFFER!
\$199.95 ▲ CONTROL CARD & FIRST ADE 1.1
79.95 ▲ FIRST ADE INTERFACE
19.95 ▲ TEMPERATURE PROBE
CHECK / M.O. / C.O.D. 3% SHIPPING/CA + 6%



FILE TYPES ON THE TI99/4A

By Irwin Holt.

This article will be concerned with the types of disk files available on the TI. They are: DISPLAY VARIABLE, DISPLAY FIXED, INTERNAL VARIABLE, INTERNAL FIXED and PROGRAM.

We'll start looking at programs. When you think of a program you probably think of a BASIC or EXTENDED BASIC one. There are several other types of programs. There are a few indicators you can use to help you to decide whether the programs on your disk is BASIC, E/A, GK etc. Basic or extended programs may be anywhere from 2 to 52 or so sectors. After a point which is determined by how much memory is needed, EXTENDED BASIC programs are saved as INTERNAL VARIABLE 254. Those programs cannot be run in BASIC.

You may be able to save an INTERNAL VARIABLE 254 program from disk to cassette if it is close to the point where it would be saved as a PROGRAM.

You may also run into EXTENDED BASIC programs that are, say 25 sectors long with just a few program lines.

Those most likely contain embedded assembly files saved with ALSAVE or SYSTEX. Those will not run in BASIC. If you save those in MERGE format, you may lose assembly files.

33 sector programs may be E/A option 5. If you try to run an E/A5 in BASIC or EXTENDED BASIC you will get an error 50. Another indication of E/A5 is that the last character of the file name will be incremented by one ascii value as in TERM TERN TERO or MGR1 MGR2. In most cases E/A5 files will be 33 sectors in length except for the last one. There are some exceptions with E/A5 loaders however.

Files for the GRAMCRACKER follow a somewhat similar format except that they are 34 sectors in length. For the GK you may see file names such as: XB XB1 XB2 XB3 XB4 XB5 XB6 or FAST/61 FAST/62 or GRAM0 GRAM1 GRAM2. Some RLE pictures may also be in programs I have seen 25 and 54 sector format pictures. Just a couple of other notes on programs. There will soon be a program in the C.O.N.M.I. library which is supposed to make it possible to transfer long programs from cassette to disk. ALSO FROM FUMHELWEB FARMS CASSETTE TRANSFER make it possible for you to store/recover your E/A5 programs on cassette. This is great for people who have 32K and

a cassette system but have not added disk.

Now let's take a look at RELATIVE files. They may be used on disk only. The two types of file access are RELATIVE and SEQUENTIAL.

Records in a RELATIVE (random) file may be accessed in any order. An example of a RELATIVE file is the disk directory.

Try the following:

```
10 OPEN #1:"DSK1.",INPUT,INTERNAL, RELATIVE
20 INPUT A :: INPUT #1,REC A:A$
30 PRINT A$ :: GOTO 20
```

Just put in a number in A from 0 to 127, 0 (zero) will give you the disk name. 1-127 will give you the filename on that record. If you get a null string you are beyond the end of the directory. I use a RELATIVE file on the SPIRIT OF '99 TIBBS which contains the alphabetical user list. This makes it possible to search the LIST for a name. RELATIVE files must have FIXED length records. This means that any extra space in the record is filled in by the computer. If you have a 70 character line in a FIXED 80 file the last 10 characters will not be available for use. RELATIVE files may be DISPLAY or INTERNAL.

Variable files (INTERNAL or DISPLAY) may only be accessed sequentially. DISPLAY VARIABLE 80 files are most likely text files. This article is being written in EXTENDED BASIC and stored as D/V 80 on disk. You may print D/V 80 files from TI-WRITER (or equivalent) DM1000 or the E/A editor. The editor will strip out all control characters. You may need to use the TI-WRITER FORMATTER for some files. If you OPEN #1:"DSK1.TEST" The file will default to D/V 80 in UPDATE mode. This means you may read or write to it. If you open it in INPUT mode you may read from it. Interestingly, if you open the file in INPUT mode the default is D/V with no record length. I find this useful if I need to read a variety of files with speech. One seeming hitch to this is with the Horizon Ram-disk.

There the INPUT default is D/V 80. The OUTPUT (WRITE) default is D/V 80. You may also find D/V 80 files as RLE pictures or pics which are intended to be printed directly on your printer. D/V 80 files may also be source code for assembly programs. Those may be assembled with the E/A cartridge.

One of the most interesting files is DISPLAY VARIABLE 163. This is normally used by the computer for ME format. If you try to read one of these files you will get some numbers and strings but mostly unprintable characters. The computer uses TOKENS to represent words such as CALL COLOR, CLEAR etc.. Tokens are used to

represent line numbers, quoted and unquoted strings and much more. If you see a D/V 163 file go into EXTENDED BASIC, type NEW, type MERGE DSKx.(filenane)and the file should be in memory after reading it from disk. If you wish to save it as a program, do so as if you had typed it in. DISPLAY FIXED 80 files may be assembly files. They may be loaded from E/A3 or from EXTENDED BASIC with CALL LOAD. There is no way that I know of to tell the difference without trying to load the file. Note that some early versions of TEII store files in D/F 80 in terminal mode. DISPLAY FIXED 128 files may be RLE pictures or ARCHIVED (packed) files. If you cannot tell by the name you may have to try to unpack the file with ARCHIVER. The usual indicator for an ARCHIVER file is /PAC or /ARC as the last part of the file name. This is not always followed however. With the advent of archiving programs that squeeze or compress the file this may become more confusing. Be sure to read any instructions descriptions etc., that come with the file. This is just an overview of some of the types of files you might run into. You will find numerous exceptions to these guide lines.

If you have any questions, be sure to ask at the meeting.

Retyped for TEXPAC BBS by John Ryan of TISHUG.

END OF ARTICLE



PUZZLE

This months list of words is based around the subject of "PORT'S"

So all men on deck and GOOD LUCK!

```

E N C E I F V Q Z Y A T V N O N G C O I
S I O N H P F S S L J H A D I L L W Q N
F Y P S V O H G I H C T Y L G R H K I M
D Q H V K V R F G M X P A G B Y E P G P
F E X A O C A E W N P U L E B O X Q F H
O M I P R E I V S U E S Z O P P T K P J
Q S F L S C L D U B G J O Q G R I V Y D
G G F N C A O I C M Y U U N S A I R I X
A L J R P S L U Z M F B N P B W N N I G
P M P Y Q R P G R A A W E K Y M O I C E
N A H H W E G J U T B E N Y C T T H J E
E R R H E M X G V O H E N E X Y M O Y J
F X P N Q L B U E K D Q T Y L T H G P G
P B R S A A S R W H B U E H T L O X L B
Y J R A K H K O N O T V X O A A E J N Q
R R G U D C O O N B L I W T G I N H W X
O Q H H H I Q W A F O X L E Z B F N J Z
P K X W P T U A H W C E V H L H E S A Y
P Q L P U X R M P P C L D X S D W P Q B
K Y R I A F F A L W Y Y E E R X A D A C
  
```

Find these hidden words

HOW TO PLAY

In this puzzle there are (20) words somewhere horizontal, vertically, diagonally even backwards

| | | |
|-----------|----------|------------|
| ARTHUR | AUPRINCE | BAINMATYNE |
| CHALMERS | DICKSON | DOUGLAS |
| ELIZABETH | ELLEN | ERIN |
| FAIRY | GENTIL | HARCOURT |
| LOGAN | MORESBY | NELSON |
| PIRIE | RADIUM | SIMPSON |
| TALBOT | WELD | |

This puzzle was compiled using Ashley Lynn's program "Word Puzzle" which can be ordered through TISHUG.

END OF ARTICLE

SURGE PROTECTION

by Gary K Cox

WHAT IS A SURGE PROTECTOR/SUPPRESSOR? A surge protector/suppressor is simply a device which attempts to prevent unwanted voltage surges from reaching your electronic equipment. The surge protector operates by providing an alternative pathway for the additional electric current by either absorbing the energy or diverting the additional energy to ground thus away from your equipment.

WHAT IS A VOLTAGE SURGE? A voltage surge is any voltage above the 110/115 volts in a normal electrical outlet. The power supplies in most electrical equipment can filter out momentary power surges under about 300 to 500 volts but anything above that can damage your equipment.

WHAT CAUSES VOLTAGE SURGES? The most obvious cause of a voltage surge is lightning/electrical storms which can cause serious damage to electrical equipment, especially computers. Other causes are not so obvious such as your air-conditioner kicking on causing a drop and then a surge in electricity which sometimes can be seen by your lights blinking...Motors hooked into the electrical line can cause surges or work being done by the power company can cause a surge...However, power surges are not to be confused by loss of power as a surge protector ONLY protects against overvoltage not undervoltage. However, losing power usually doesn't damage equipment whereas too much voltage will damage equipment.

HOW CAN I PROTECT AGAINST VOLTAGE SURGES? That's what surge protectors are for. Most surge protectors plug into your wall outlet and then the equipment that you wish to protect plugs into the surge protector.

All computer equipment should have some sort of surge protector on it but the quality of the surge protector greatly varies as does the price.

HOW DO I TELL A GOOD SURGE PROTECTOR FROM A NOT SO GOOD ONE? Underwriters laboratories has developed a minimum standard for surge protectors called UL listing 1449. Any good surge protector should carry this UL 1449 listing for absorbing surges between any pair of the three wires in an electrical circuit which is the ground wire, the "hot" or line wire and the neutral wire. Some surge protectors only protect as little as one of these lines, when all three should be protected.

A good surge protector should have an indicator light indicating that the device is working as it is important to know if the surge protection is still operating. Some protectors only have a light to indicate the status of the protection while other protectors will have an additional circuit which shuts down the entire unit until when protection is compromised to prevent any unprotected power from reaching your equipment.

Many protectors have a site wiring fault indicator to indicate whether the outlet that you are plugged into is wired ok or if the outlet is improperly grounded or if

the polarity is reversed which can prevent the surge protector from working properly.

Read and understand the documentation on the surge protector. Even the cheap surge protectors have fancy packages so don't be fooled by a pretty package.

Some surge protectors only have MOV's (Metal Oxide Varistors) to protect against surges. MOV's will absorb a surge but MOV's only provide the minimum protection necessary. The surge protector should have more protection than just MOV's. The more that is between the power line and your equipment the better protection that you have against surges.

Some sophisticated surge protectors have what is called sine wave tracking which tracks the AC sine wave and cuts off a surge off the top of the wave. In other words, this feature can provide for better protection.

Most surge protectors have EMI/FRI filtering which filters out any noise in the line and it is a nice feature to have in a surge protector.

In looking at the specifications on the surge protector look at how much voltage it will let through on a surge. I have seen some surge protectors let through as little as less than 40 volts while others let through 500 volts or more! The UL 1449 standard states that a surge protector should not let through more than 330 volts to comply with the standard. Most power supplies will filter out a 330 volt surge but around 500 volts or more seems to be a little high. Also look at how high of a surge the protector is capable of handling. Some surge protectors fail at 3,000 volts while others will go much higher and of course the higher the better.

Many surge protectors offer a guarantee whereby if your equipment is damaged by a surge they will pay for the repair up to a certain amount. This amount varies from anywhere around \$2,000 to \$25,000.

Another feature to look for in a surge protector is modem/fax protection as a voltage surge can come through your phone line just as easy as your electrical line. Many surge protectors have a phone line protector built in or a phone line protector can be purchased separately.

Most of the good surge protectors tend to sell for \$50 or more so just remember you get what you pay for. If you buy a \$6 surge protector you probably do not have much protection.

Surge protectors typically come in various forms. The most common is in the form of an outlet strip with multiple outlets. Another form is in a complete unit that plugs directly into electrical outlet covering up both outlets yet providing for several plugs. Another type unit is in the form of a single cube where your existing outlet strip or any electrical device plugs into it and plugs into the wall. Surge protectors can be purchased in the form of outlets themselves replacing the standard in the wall electrical outlet or they can be purchased in the form where they go into your electrical feedbox. Phone line protectors can come in the form of a single unit which plugs into your electrical outlet (for grounding) or many of the electrical surge protector units also provide a phone jack for protecting the phone

line.

Additionally if protection against loss of power is needed in addition to surge protector then a UPS (Uninterruptable Power Supply) is the answer. UPS's will provide power to equipment when loss of power from the power company occurs without disturbing the operation of that equipment. However, UPS's are only for temporary power so that a user will have time to finish what he or she is doing and shut down the system properly without data loss. Most UPS's also provide surge protector and can react to losses of power of a duration of less than a second. Your computer equipment must have a continuous power as an interruption of less than a second can reboot your system and possibly cause some data loss. The UPS also must be matched with the amount of equipment that you wish to protect as too much equipment attached to a UPS can overload it unless the UPS is large enough for the amount of equipment connected to it. Furthermore the larger the UPS the longer it will continue to provide power in a power failure. Depending on the size of the UPS it can supply power for a duration of typically 15 minutes to several hours. Of course, the larger the UPS the more it will cost and many manufactures also provide a guarantee that if your equipment is damaged they will pay for the repair up to a certain amount. When shopping for a surge protector be sure to read the package for the specifications and do not be fooled by the fancy talk. Just pay attention to the actual specifications.

I hope that this helps a little while you are out shopping for a surge protector.

END OF ARTICLE 

"MY COMPUTER AND I"

By Loren West

This is really a story on how I've been using my computer.

Recently my wife and I purchased a WET and DRY thermometer from a garage sale. Well what is a wet and dry thermometer? It has two thermometers mounted together on the one frame, one thermometer has a dish to hold water, with a wick around the bulb and the other end in the dish of water, which keeps the bulb "WET".

Due to the evaporative effect on the WET thermometer there is usually a difference in temperature compared to that of the "DRY" one.

With this in mind there is a calculation that can be used to work out the humidity in %. After finding the calculations in our encyclopedia out comes the computer, with it's powerful calculating skills, it's quick reflexes and ability to control a printer.

After a very short time a simple program was put together to display the humidity in % on the screen but to turn the computer on each time you wondered what the humidity is, wasn't feasible, so back to the program, enter a few printer commands and now I have a graph of percentages. By looking down the WET temperature side and going across the DRY temperature side, meeting in the centre, and you have the answer.

This sheet is fixed inside one of our cupboards for quick, walking past reference.

Below is the program and a sample page.

```
100 !SAVE DSK1.TEMPCHECK
110 !NEXT 3 LINES ARE USED TO PRINT
STRAIGHT TO PRINTER :Condensed,
Superscript
115 !OPEN #1:"PIO"
120 !PRINT #1:CHR$(15)
130 !PRINT #1:CHR$(27);CHR$(83);CHR$(0
)
135 !NEXT 2 LINES ARE USED FOR
INDIVIDUAL SCREEN INPUTS TURN OFF LINES
160, 170, 230, 240, 250.
140 INPUT "Tc DRY..":D
150 INPUT "Tc WET..":W
160 !FOR W=18 TO 45
170 !FOR D=18 TO 45
175 !CALCULATION = CENTIGRADE TO
FAHRENHEIT THEN TO PERCENTAGES.
180 TFW=W*9/5+32
190 TFD=D*9/5+32
200 WD=TFW+TFD
210 H=40*WD/100+15
220 PRINT :"HUMIDITY % ";H
221 !THE NEXT 3 LINES WERE TO CONTROL
THE OUTPUT TO THE PRINTER
230 !PRINT #1,USING "###.# ":H;
240 !NEXT D
250 !NEXT W
```

By turning lines 140, 150 off and turning lines 160, 170, 240, 250. you could display to the screen a list of different percentages as lines 160 and 170 go through their paces. Try changing the values in 160 and 170. Try changing line 230 to (PRINT USING "###.# ":H;)

HUMIDITY †
TEMPERATURE DRY Centigrade

| | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 |
|----|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| 9 | 53.6 | 54.3 | 55.0 | 55.7 | 56.4 | 57.2 | 57.9 | 58.6 | 59.3 | 60.0 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 |
| 10 | 54.3 | 55.0 | 55.7 | 56.4 | 57.2 | 57.9 | 58.6 | 59.3 | 60.0 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 |
| 11 | 55.0 | 55.7 | 56.4 | 57.2 | 57.9 | 58.6 | 59.3 | 60.0 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 |
| 12 | 55.7 | 56.4 | 57.2 | 57.9 | 58.6 | 59.3 | 60.0 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 |
| 13 | 56.4 | 57.2 | 57.9 | 58.6 | 59.3 | 60.0 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 |
| 14 | 57.2 | 57.9 | 58.6 | 59.3 | 60.0 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 |
| 15 | 57.9 | 58.6 | 59.3 | 60.0 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 |
| 16 | 58.6 | 59.3 | 60.0 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 |
| 17 | 59.3 | 60.0 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 |
| 18 | 60.0 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 |
| 19 | 60.8 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 |
| 20 | 61.5 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 |
| 21 | 62.2 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 |
| 22 | 62.9 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 |
| 23 | 63.6 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 |
| 24 | 64.4 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 |
| 25 | 65.1 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 |
| 26 | 65.8 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 | 84.5 |
| 27 | 66.5 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 | 84.5 | 85.2 |
| 28 | 67.2 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 | 84.5 | 85.2 | 86.0 |
| 29 | 68.0 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 | 84.5 | 85.2 | 86.0 | 86.7 |
| 30 | 68.7 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 | 84.5 | 85.2 | 86.0 | 86.7 | 87.4 |
| 31 | 69.4 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 | 84.5 | 85.2 | 86.0 | 86.7 | 87.4 | 88.1 |
| 32 | 70.1 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 | 84.5 | 85.2 | 86.0 | 86.7 | 87.4 | 88.1 | 88.8 |
| 33 | 70.8 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 | 84.5 | 85.2 | 86.0 | 86.7 | 87.4 | 88.1 | 88.8 | 89.6 |
| 34 | 71.6 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 | 84.5 | 85.2 | 86.0 | 86.7 | 87.4 | 88.1 | 88.8 | 89.6 | 90.3 |
| 35 | 72.3 | 73.0 | 73.7 | 74.4 | 75.2 | 75.9 | 76.6 | 77.3 | 78.0 | 78.8 | 79.5 | 80.2 | 80.9 | 81.6 | 82.4 | 83.1 | 83.8 | 84.5 | 85.2 | 86.0 | 86.7 | 87.4 | 88.1 | 88.8 | 89.6 | 90.3 | 91.0 |

TEMPERATURE WET Centigrade

COMFORTABLE HUMIDITY IS AROUND 70%

NEW COMPANY TAKES OVER TEX-COMP TI994A DIVISION

By LAURA BURNS

Tex-Comp's TI99/4A division has new owners as of April 1, according to Jerry Price, Tex-Comp vice president.

The new company, Tex-Comp Ltd., is operated by Carey Hoffman. The 24-year-old Hoffman has been working with the TI for 17 years, he says.

"I like the TI," he says. "It comes up a lot quicker than an IBM, that's for sure."

He will be aided in the venture by his father, Larry Hoffman of the West Covina 99ers. The elder Hoffman programmed TI's Speak and Spell to work with Extended BASIC in a project sponsored by Tex-Comp and approved by Texas Instruments, according to Price.

Address of the new company is Tex-Comp Ltd., 425 East Arrow Highway, Suite 732, Glendora, CA 91740-5684. Voice phone is (818) 339-8924 and Fax is (818) 858-2785. In addition, callers can contact the company under the user name TEX-COMP on the bulletin board of the West Covina 99ers, (818) 339-1134.

Carey Hoffman says one project for the new company is "bringing the Databiotics line back to life." The new company

has bought out the remaining DataBioTics stock and will be selling its products, he says.

DataBioTics produced numerous cartridges for the TI99/4A including word processors, multi-screen games, spreadsheets and printer interfaces.

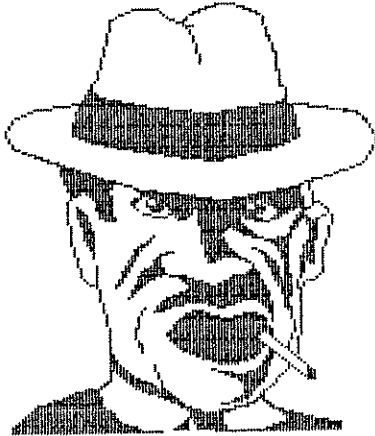
Price says he began negotiating the transfer at Fest-West, held in San Diego, California in February. During the transition months, he will consult with the new firm, he says.

"I feel very comfortable with these people taking over," Price says. "I would not have sold it to just anybody."

Price began his mail order company for the TI99/4A in 1981 when he noticed a need for a source of software other than the popular titles then carried in department stores.

At the time he entered the mail order business, "a couple of distributors and stores had mail order divisions," but he was one of the first TI dealers to make it his major division. Tex-Comp's TI business was in the millions-a-year category at one time.

A tornado that took the roof off a company storage building in December 1991. Later the company sustained minor damage in the Los Angeles-area January 1994 earthquake.



TREASURER'S REPORT

by Cyril Bohlsen

Income for previous month \$ 3095.80
 Expenditure for previous month \$ 3674.38
 Loss for previous month \$ 578.58
 Membership accounted for \$ 70.00 of income
 Shop sales \$ 3025.90 of income

The expenditure was made up of the following :-

Administration \$ 0.00
 Printing and posting of TND \$ 250.02
 Shop purchases \$ 3043.36
 Purchase of new printer for Editor..... \$ 381.00

END OF ARTICLE 

PROGRAM OF THE MONTH

by Bob August

The program this month is for the 4th of July. It's a flag program that is documented to help you understand what the program is doing

the program is written in Basic, but will run in both Basic or Extended Basic.

Hope you enjoy.

```

100 REM < FLAG PROGRAM >
110 REM IN TI-BASIC OR
120 REM EXTENDED BASIC
130 REM BY R.W. AUGUST
140 CALL CLEAR
150 REM ++ WHITE SCREEN ++
160 CALL SCREEN(16)
170 REM ++ GRAY BORDER ++
180 CALL CHAR(112,"FFFFFFFF
FFFFFFFF")
190 CALL COLOR(11,15,16)
200 CALL VCHAR(4,31,112,24*4
)
210 CALL HCHAR(20,1,112,32*8
)
220 REM ++ RED STRIPS ++
230 CALL COLOR(10,7,16)
240 CALL CHAR(104,"FFFFFFFF
FFFFFFFF")
250 CALL CHAR(105,"FFFFFFFOO
OOOOOO")
260 CALL CHAR(106,"OOOOOOOO
OFFFFFF")
270 CALL CHAR(107,"OOOFFFFF
FFFFFF")
280 CALL CHAR(108,"FFFFFFFO
OOOOOO")
290 CALL CHAR(109,"OOOOOOOO
OFFFFFF")
300 CALL HCHAR(4,14,104,17)
310 CALL HCHAR(5,14,105,17)
320 CALL HCHAR(6,14,106,17)
330 CALL HCHAR(7,14,104,17)
340 CALL HCHAR(9,14,107,17)
350 CALL HCHAR(10,14,108,17)
360 CALL HCHAR(11,14,109,17)
370 CALL HCHAR(12,14,104,17)
380 CALL HCHAR(14,3,107,28)
390 CALL HCHAR(15,3,108,28)
400 CALL HCHAR(16,3,109,28)
410 CALL HCHAR(17,3,104,28)
420 CALL HCHAR(19,3,107,28)
430 CALL COLOR(12,7,15)
440 CALL CHAR(120,"FFFFFFFO
OOOOOO")
450 CALL HCHAR(20,3,120,28)
460 REM ++ WHITE STARS ++
470 R=4
480 CALL COLOR(9,5,16)
490 CALL CHAR(96,"FFE7E781C3
99BDFF")
500 CALL CHAR(97,"FFFFFFFF
FFFFFF")
510 FOR C=3 TO 13 STEP 2
520 CALL HCHAR(R,C,96)
530 IF C=13 THEN 550
540 CALL HCHAR(R,C+1,97)

```

```

550 IF R=12 THEN 590
560 CALL HCHAR(R+1,C,97)
570 IF C=13 THEN 590
580 CALL HCHAR(R+1,C+1,96)
590 NEXT C
600 IF R=12 THEN 640
610 R=R+2
620 GOTO 510
630 REM ++ HOLD ON SCREEN ++
640 CALL KEY(O,K,S)
650 IF S=0 THEN 640
660 CALL CLEAR
670 END

```

END OF ARTICLE

SYDNEY MARITIME MUSEUM ART UNION

The objectives of the Sydney Maritime Museum is to restore and operate Australia's Maritime Heritage and keep it afloat for the future generations.

The organisation is a charitable one with minimal Government funding, hence raises funds for continuity of work by having Art Union Raffles.

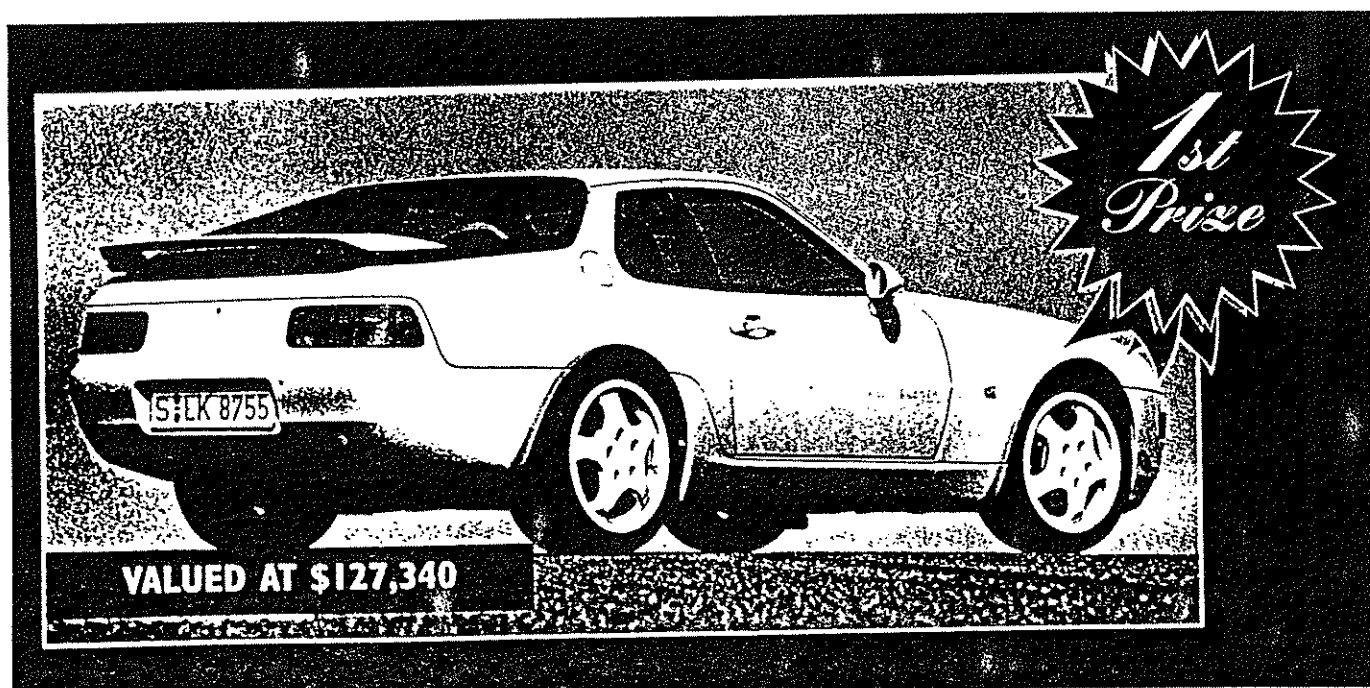
The next Art Union is to be drawn on 28th September 1995 and for the First Prize, it is the ultimate sports car, a red 968 Club Sport PORSCHE valued at \$127,000.00 while the Second Prize is a Gold Coast Holiday for two with \$200.00 spending money and the Third Prize is Two Mongoose Mountain Bikes.

The Sydney Maritime Museum has specially designed Harbour Cruising Tour packages including discount rates for business and esteem reputable organisations who may be in mind of celebrating some occasion or are in mind to entertain overseas or local business associates, staff or family members and leave a lifetime impact on their minds and hearts. This is a 2 hour cruising tour over the Sydney waters via under the Harbour Bridge and is obtained at buying 20 tickets or 50 tickets!

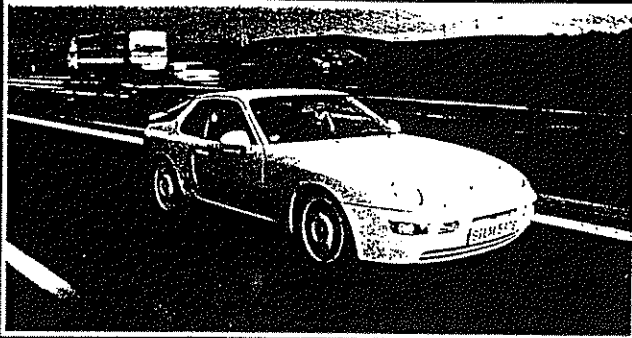
In summary this is a generous contribution towards the preservation of Australian Maritime Heritage for the future generations. Whilst buying Art Union tickets at a discounted rate, increase the winning odds and the opportunity of entertaining the beloved ones with a cruising tour and to leave an impact to be ever fresh in heart and mind. Looking forward to your generous support.

(ED)

The above letter was recieved by our organisation and we felt it deserved being brought to the notice of our members. The tickets come at a cost of \$40.00 per ticket and anyone interested should ring or see Percy at our next meeting. Remember, we need groups of 20 or groups of 50 to qualify for the discount and Harbour Cruise.

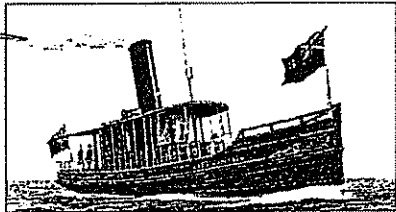


An invitation to excitement

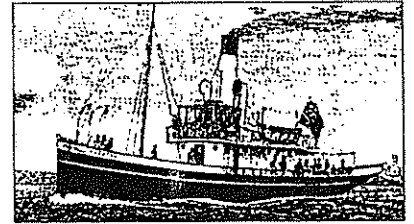


Receive a free 2 hour cruise on the 1902 VIP steam tug "Waratah". This authentically restored vessel with the never to be forgotten smell of steam and coal smoke accommodates 49 passengers.

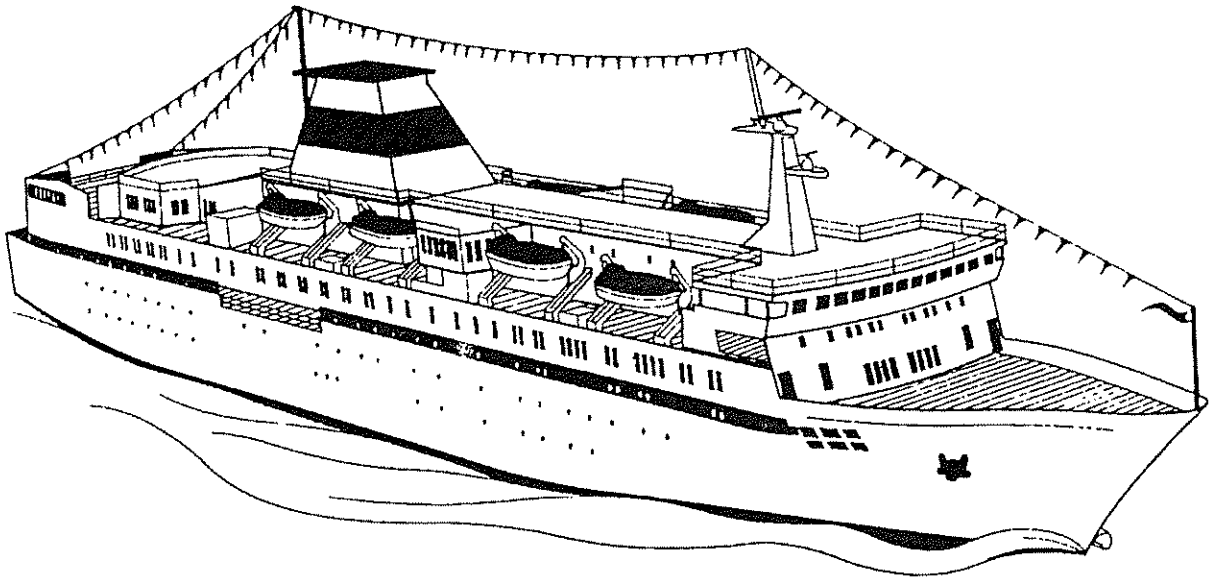
beautiful Edwardian vessel can accommodate 20 passengers.



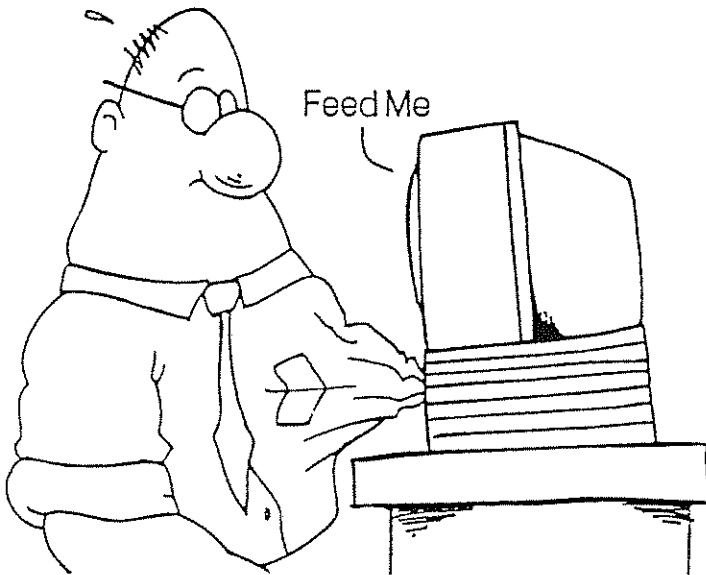
Receive a free 2 hour cruise on the 1902 VIP steam launch "Lady Hopetoun". This beautiful Edwardian vessel can accommodate 20 passengers.



END OF ARTICLE



By Larry saunders



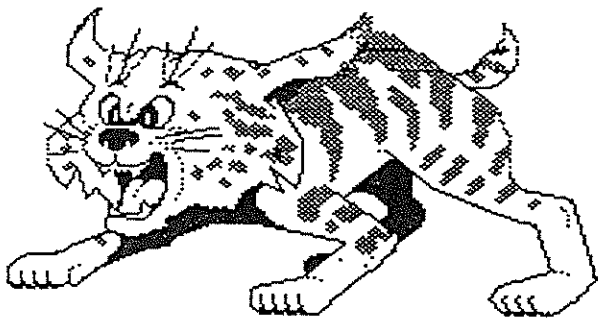
EDITORS COMMENTS

Hello to everyone, I'd like to say what an interesting meeting we had last month, being offered tours around the famous Sydney Harbour to buying cheap items from our TI shop. The deal for the tours should appear else were in this magazine and the cheap and free items from the shop you will have to come and look for your self, I had purchased quite a few of the cassette tapes and I intend to purchase some more at the next meeting, after loading the programs to disk, they are erased and then used for a very good recording tape.

This month in the magazine we are going to try something different, we have found a programe that is written in TI994A and also in IBM language, so have a go and type this little program in and let us know how it turns out. Please look in the TI section for the main part of the text to this programe.

I hope to see you all at the next meeting, bye for now (ED)

END OF ARTICLE



Diskname P144
Used= 358 Free= 0

Page Pro headliner maker. This program makes headlines for page pro. It uses HF Fonts that comes with the program or you can put together a large font your self by one of the programs in this package. If you do make one each part must be save away as a picture first. e.g. A as A . B as B . a as AA etc. on a blank disk. Them run the Font Maker and it will catalog the disk, make a font, and save it away. The program is MENU driven and is easy to use.

| | | | |
|-----------|---------|------------|---------|
| 3D HF | 80 I 13 | ANTIQUE HF | 75 I 13 |
| BKMN1 HF | 76 I 13 | BRDWY HF | 61 I 13 |
| HL/FILE1 | 2 d 80 | HL/PIC-01 | 4 I 13 |
| LOAD | 11*Prog | PEFS-V100 | 16*Prog |
| PPHM-V100 | 33*Prog | | |

Diskname P145
Used= 352 Free= 6

Page Pro Pictures.

| | | | |
|------------|----------|----------|---------|
| SANDCASTLE | 128*I 13 | SHELL1 | 28*I 13 |
| SHELL4 | 13*I 13 | SPOTFISH | 31*I 13 |
| WHEEL | 152*I 13 | | |

Diskname P146
Used= 356 Free= 2

Page Pro Titles. Large fancy Font pictures.

| | | | |
|------------|---------|-----------|---------|
| 10TH | 14*I 13 | 15TH | 14*I 13 |
| 1ST | 10*I 13 | 20TH | 14*I 13 |
| 25TH | 14*I 13 | 2ND | 10*I 13 |
| 3RD | 10*I 13 | 4TH | 12*I 13 |
| 5TH | 11*I 13 | ANNIVRSRY | 39*I 13 |
| ANNOUNCING | 39*I 13 | ANNUAL | 26*I 13 |
| MEETING | 38*I 13 | MONTHLY | 38*I 13 |
| REUNION | 32*I 13 | WSEKLY | 35*I 13 |

Diskname P147
Used= 282 Free= 76

Page Pro Templates. Invitations cards.

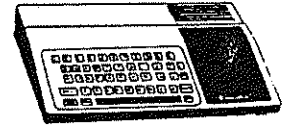
| | | | |
|------------|---------|------------|---------|
| B/FCR01 | 10 I 13 | B/SHOWER01 | 46 I 13 |
| CCMEC4 | 21 I 13 | DATE:1 | 2 I 13 |
| DR01 | 40 I 13 | HAT01 | 44 I 13 |
| HELPO01 | 19 I 13 | INSIDE00 | 18 I 13 |
| INV/CARD04 | 21 Prog | INV/CARD06 | 21 Prog |
| INV/CARD06 | 21 Prog | OVER01 | 6 I 13 |
| PLACE:1 | 2 I 13 | SURPRISE1 | 9 I 13 |
| TIME:1 | 2 I 13 | | |



The spreadsheet program is a handy tool for doing many financial calculations simultaneously. Its inherent versatility turns a computer into a super "What if . . ." machine. Using very simple commands, you can easily customize a spreadsheet for your own applications. These could include tracking your investments, evaluating alternative purchase options, calculating your company payroll, or tabulating accounts receivable. *Snap-Calc™* is such a spreadsheet program written in BASIC.

The *Snap-Calc™* spreadsheet itself is a matrix of numeric values. After you have entered values into the program, it will execute calculations that you have previously specified, and display the results through screen windows.

With *Snap-Calc™* you will be able to create a formula for each row of the matrix. This same formula will then be used for all of the columns. You may also create names for each of the rows, and indicate which rows are to be included in the printed reports. (You may want to use certain rows to calculate other data.) In addition, you can specify the number of columns the spreadsheet will have, and set up a *totals* column which sums all columns used. The photo at the right is an example of the data entry screen.



NEW When this command is used, the memory will clear and you will be prompted to enter one of four suboptions. If you select Data, then all of the data entered on the spreadsheet will be cleared. If you select Logic, your current logic model will be erased from memory. If the third suboption—Both Data And Logic—is selected, both data and the logic model will be cleared from memory. The fourth suboption, Abort New Command, is an escape which allows you to return to the normal logic entry screen without clearing any memory.

| ROW NAME | 1 | 2 |
|--------------|-------|-------|
| 1 COST 1 | 10.00 | 20.30 |
| 2 COST 2 | 25.00 | 15.75 |
| 3 COST 3 | 37.15 | 12.00 |
| 4 TOTAL COST | 72.15 | 68.25 |
| 5 5 | 0.00 | 0.00 |
| 6 6 | 0.00 | 0.00 |
| 7 7 | 0.00 | 0.00 |
| 8 8 | 0.00 | 0.00 |
| 9 9 | 0.00 | 0.00 |
| 10 10 | 0.00 | 0.00 |

This command should not be confused with the BASIC command NEW, which erases the program in memory. Exercise caution when using this command. Always save your data and logic model to disk first.

LIST This command will list the logic model on the screen. The first item listed is the logic name. If no logic name has been assigned, then the message LOGIC NAME IS with no name following it will be printed.

TOTAL COLUMN IS col is always on the second line; here col is the totals column. If col is set to 0, then the totals column has been turned off. The default value for the totals column is 13.

LAST COLUMN IS col is always the next item listed and col is the last column in which data entry can occur. This value must always be less than the total column value. The default for this value is 12.

From this point on, the information listed is determined by the logic commands that you have entered into the model. If you created a row label for a row, the message *r IS nnn* will list, where *r* is the row number and *nnn* is the name of the row. Even if you entered the row name in the data entry section of the program, it will be displayed here. Following the row name will be all equations defined for that row. The message *r = eq* will list, where *r* is the row number and *eq* is the equation for that row.

PRINT This command performs the same functions as the LIST command except that the output is directed to the system printer.

MOVE xx TO yy This command allows you to move all of the data from column *xx* into column *yy*. You would then have duplicate entries—the information in column *xx* is not erased after the transfer.

END This command will cause you to exit the Logic Entry mode and return to the Data Entry mode. You may at any time return to Logic Entry mode and append changes to your logic model.

The following logic function commands affect the logic model itself:

LOGIC NAME IS nnn This command lets you create or change the name of the logic model, where *nnn* is the name of your logic. LOGIC NAME IS SAVINGS, for example. The name you assign to your logic model will be used as the file name when you save your model to disk or tape.

TOTAL COLUMN IS col
or

TOTAL COLUMN IS OFF This command lets you create or change the column that you want to use as your totals column. If the Off option is used, then the totals column will be set to zero and will not be calculated when the logic model is calculated.

LAST COLUMN IS col This command creates or changes the last column in which you will be able to enter data.

Continued on next page

DATA ENTRY IS A SNAP

To make an entry in any field, simply move the cursor to the field that you wish to change and type in the new value. A second set of cursor keys lets you move three columns left or right, or five rows up or down at one time. Not only is the screen updated with every key press, but the current value on the screen is stored in the spreadsheet. You can leave that entry field at any time, and the value displayed will be saved automatically.

LOGIC ENTRY MODE

When you select this mode, the screen will clear and a prompt > will appear at the bottom of the screen. At this point, you can enter logic commands. These commands will let you specify the *totals* column and *last* column in the spreadsheet. You can also assign a logic name to your model, as well as row labels. The most powerful aspect to this option is that it lets you define equations for any of the rows in the spreadsheet. (See *r = eq* below.)

Two types of commands are used in this option: immediate action commands, and logic function commands. The immediate action commands are as follows:

Logically, this should be the column just before the totals column (if the totals column is in use), because you may not move your cursor beyond the LAST column when entering data. If the totals column is several columns beyond that point, you may not be able to display it on the screen.

The only restriction in designating the last column is that it must be a lower column number than the totals column, unless of course the totals column is turned off. In that case you can make the last column any value your machine will allow. The computer system you use will determine the maximum number of columns available.

r IS nnn This command is used for naming any of the rows in the spreadsheet. The row number is *r* and *nnn* is the name you assign to the row. An example might be: 3 IS ROW#3, where the name ROW#3 is given to row number 3.

r = eq This is by far the most powerful of the logic model's commands. It lets you assign an equation *eq* to a row *r*. The equation can be made up of other row numbers; real numbers called constants; an operator called LAG, which lets you access a previous column; or arithmetic operators. Rules for constructing the equation are as follows:

(1) A row can be specified by simply using the row number in the equation. For example $5 = 1 + 2$ means that the value from row 1 is to be added to the value in row 2 and the total is to be placed in row 5. This same equation format is then used for all of the columns in the spreadsheet. Notice that there are spaces between each item. This is a requirement of the system so that the computer can decipher it easier, and execute it more quickly.

(2) A real number can be used as a constant in the equation when placed within parentheses. Taking the example from the previous paragraph, let's try: $5 = 1 + 2 * (12.53)$. The equation will now take the sum of row 1 plus row 2 and multiply it times 12.53. There is no priority of multiplication over addition—everything is executed from left to right. Rows 1 and 2 are added together before the sum is multiplied by 12.53.

You can use data from the previous column with the LAG modifier. When this modifier precedes a row number, the information will come from the same row number, but from the previous column. For example: $5 = 1 + \text{LAG } 2$. Here row 1 from the current column will be added to row 2 from the previous column. If the column currently being calculated is 8, then LAG 2 would cause the value from column 7 row 2 to be used. If you set up an equation such as: $5 = \text{LAG } 2$, then the value of the previous column, row 2 would be placed in row 5. In another example, $5 = 4 + \text{LAG } 5$, a running total for row 4 would be created in row 5.

There are four operators that you can use in your equations. They are:

| | |
|---|-----------|
| + | Add. |
| - | Subtract. |
| * | Multiply. |
| / | Divide. |

The length of an equation is limited only by the maximum string length for each system. There are minor differences in how the equation is to be entered on each machine, so consult the *Snap-Calc™* section which covers your machine.

Once you have entered one of the above logic function commands, you can use the LIST command to check whether the command has been received and interpreted properly, or to simply check logic statements that have already been entered.

CALCULATE MODEL

When the key for the Calculate Model function is pressed, the computer will begin calculating your logic model using the data currently entered on the spreadsheet. All of the equations for each row will be executed in ascending order of row number. For example, if an equation in row 3 uses the result of an equation in row 5, you may have a problem. Because you can't enter data into a calculation field, field number 5 will probably be set to zero and will not contain the proper information when row three does its calculation. You should design your logic model with this in mind.

In addition, Calculate Model provides an automatic row total in the totals column. The totals column defaults to column 13 when the program is first run, but you may move it to a different column, or turn it off completely using logic model statements. If the totals column has been turned off with a logic statement, then the row total will not be calculated.

LOAD DATA FILE

When you select the Load Data File option, a menu of three suboptions will prompt you for the type of file to be loaded. The first suboption loads the data file only, which contains the data that has been entered into your spreadsheet. Select the second suboption if you wish to load the logic model. The third suboption lets you load both data and the logic model in one step. Once you have entered your file type, you will be prompted for the file name of the file you requested. The name you enter must reside on the data disk currently in the drive, or on a cassette tape.

SAVE DATA FILE

The primary difference between the Save Data File and the Load Data File options is in creating file names. If you choose to save the logic model in either the Load Logic Only, or Load Data and Logic modes, the file created for the logic model will use the LOGIC NAME as the name of the file. This is important to keep in mind so that when creating a logic name you only use characters that are legal in the file system being used. If the logic name has not been declared, then you will be prompted for the name of the file. That name will then become the logic name for that model when loaded at a later time. If you no longer want to keep a logic name, or wish to create a modified version, a logic name can be changed at any time.

CLEAR FIELD/ERASE

The Clear Field and Erase option can be used to back out of an entry and clear the field, or to simply reset a field back to zero. To use this function, place the cursor over the value that you wish to set to zero and press the key associated with this function. If the field is numeric, it will be redisplayed with a value of zero. If the field is a row label, it will be erased and left blank. Erasing a label from a row does not affect the rest of the row, or any equations set up for the row.

PRINT REPORT

The Print Report option allows you to generate a hard-copy report of your spreadsheet data on your system printer. Before the program starts printing, you will be prompted for the title of the report, the date, and the maximum number of rows to be printed in the report. The report generated here is formatted for an 80-column printer. This limitation restricts the report to printing only six columns of data across the page. Since there are more than six columns in most spreadsheets, the report will print all of the rows for the first six columns, then form feed to the top of the next sheet of paper and print the next six columns. This continues until all of the columns have been printed. The report will include every column up to the column designated as the totals column (or the column designated as the last column, if the totals column is turned off).

An asterisk (*) as the first or only-character in a row name will prevent that row from being printed in the report. This allows you to "pretty up" your report by not printing rows that contain intermediate data for calculations. You may want to print only a summary report, and not a detailed report on every row in your spreadsheet.

EXIT

If you want to exit any of the programs, you can do so by pressing the exit key for your machine, as described in the text for each computer. The Exit option allows you to back out and return to the data entry screen—but you will first be prompted with a message inquiring whether you want to halt the program and lose any data currently in memory.

Text continues for the IBM in the IBM section.

Text continues for the TI in the following pages



REQUIRED EQUIPMENT: TI Extended BASIC

The TI-99/4A is the machine on which *Snap-Calc*™ was originally designed. The TI version differs from the other versions because of the 99/4A's keyboard layout and screen size incompatibilities.

The Keyboard

The TI keyboard does not have separate function keys. On the 99/4A, you select functions by pressing the [FCTN] key in the lower right corner of the keyboard, in conjunction with the numeric keys. The functions perform as follows:

| Press [FCTN] | for FUNCTION |
|--------------|----------------------------------|
| 1 | Load data from disk or cassette. |
| 2 | Save data to disk or cassette. |
| 3 | Cancel entry/erase. |
| 5 | Calculate the logic model. |
| 6 | Print report. |
| 7 | Logic entry mode. |
| 9 | Exit <i>Snap-Calc</i> ™. |

In addition, several other keys work with the [FCTN] key to move the cursor:

| KEYS | FUNCTIONS |
|---------------------|-------------------------------|
| [FCTN] E | Move the cursor up 1 cell. |
| [FCTN] S | Move the cursor left 1 cell. |
| [FCTN] D | Move the cursor right 1 cell. |
| [FCTN] X or [ENTER] | Move the cursor down 1 cell. |

The [CTRL] key is useful for moving the entire window of the spreadsheet.

| Press [CTRL] | for FUNCTION |
|--------------|---------------------|
| E | Move up 5 cells. |
| S | Move left 3 cells. |
| D | Move right 3 cells. |
| X | Move down 5 cells. |

Screen Size

The 99/4A is capable of printing only 28 columns, using the PRINT and DISPLAY AT commands. This limits screen displays of row labels to nine characters wide, and numeric entries to seven characters wide—including a decimal point.

Data Storage

The 99/4A is capable of saving and recalling information from both a cassette recorder and a disk drive. Either device will work with this program. When you select either SAVE or LOAD, you must decide whether you want to work with the spreadsheet data or the logic model, which are kept in two different files.

After making your selection, you are prompted to input the device name. If you are using a cassette, simply type CS1 and press [ENTER]. For a disk drive, enter DSK1 or DSK2. It is not necessary to enter the period separator or the file name at this time, because the computer will just ignore them.

If you selected a device other than a cassette, you will also be prompted for the file name. The only exception to this is when you are saving the logic model. In this case, if you have already assigned a logic name, the logic model will automatically use that name for the file. If no name has been assigned, or you are loading the logic model, you will be prompted for the file name. The file name will then be used as the logic model name. When saving or loading data you are always asked for the file name.

Reports

The 99/4A can work with a maximum of only 13 columns because of its memory limitations, so this program is designed to print all 13 columns on one page. To do this, it is necessary to set the printer to condensed mode, which allows an 80-column printer to print 132 columns. This is done in line 1440. CHR\$(15) sets condensed mode on most printers. If your printer requires a different code to enter condensed mode, however, you may need to change line 1440 to your printer's

specifications. If you are using a letter-quality printer, or one of the less expensive dot matrix printers, you may not be able to print in condensed mode.

Logic Mode

The maximum string length on the 99/4A is 127 characters. You are limited to this size when you enter commands. If the calculation you enter gets too big, it should be broken down into smaller parts.

It's in this mode that you set up your logic model. It can include row labels, a totals column, a last column, a name for your model, and row equations. If you assign a row name longer than nine characters, it will be truncated. All spaces are removed. If you want spaces in your row name, you can enter them directly in Data Entry mode.

The maximum number of rows you can use for data or equations is 20, and the maximum number of columns is 13. But, if you run the program with the 32K memory expansion connected, you could easily modify the program to accept more columns and rows. The modifications are simple using the following procedure:

In line 190 change the value assigned to A to the number of rows you want. Then change the value assigned to B to reflect the number of columns you want. You also need to change the values used to dimension the arrays in this same line. DS() and ES() need to be expanded to the number of rows you wish to use. The J() array should have its first value set to the number of rows you wish to use, and the second value set to the proper number of columns.

| SNAP-CALC (TI-99/4A) | |
|----------------------|---|
| Line Nos. | Explanation of the Program |
| 100-160 | Program header. |
| 170-200 | Initialize program variables and functions. |
| 210-280 | Main control loop for data entry mode. |
| 290-300 | Exit program routine. |
| 310-320 | Subroutine to clear an entry and display the new value. |
| 330-470 | Subroutine to calculate the logic model. |
| 480-490 | Subroutine to return the value to be used in the equation. |
| 500-690 | Control loop for logic entry mode. |
| 700 | Subroutine to return a numeric value from the entry string. |
| 710 | Subroutine to delete the cursor. |
| 720-750 | Subroutine to take entry string apart by placing each word into a cell of the FS() array. |
| 760-910 | Subroutine to rebuild the entry string for listing to the screen or printer. |
| 920-1090 | Subroutine for file handling. |
| 1100-1120 | Builds the row names from the entry string. |
| 1130-1140 | Creates a cell value for the spreadsheet during data entry mode. |
| 1150-1160 | Control loop repaints the data entry screen. |
| 1170 | Routine to display the current mode. |
| 1180-1190 | Input the logic entry string. |
| 1200 | Direct output to either the screen or the printer. |
| 1210 | Key scan routine. |
| 1220 | Relocate the cursor. |
| 1230-1250 | Key scan routine waits for a different key to be pressed before continuing. |
| 1260-1300 | Display the data entry screen. |
| 1310-1390 | Move the cursor up, down, left, and right one position. |
| 1400-1490 | Print the report. |
| 1500-1520 | Image formats for printing single items. |
| 1530-1540 | Clears the logic model from memory. |
| 1550-1570 | Error routine. |
| 1580-1590 | Image formats for a full screen. |
| 1600-1620 | Open and close the printer port. |

SNAP-CALC

| | | |
|-----|-----|------------------------|
| 100 | REM | |
| 110 | REM | SNAP-CALC |
| 120 | REM | |
| 130 | REM | BY GARY STRAUSS |
| 140 | REM | HOME COMPUTER MAGAZINE |
| 150 | REM | VERSION 4.3.1 |
| 160 | REM | TI EXTENDED BASIC |

```

170 DEF RN=RW+R-1 :: DEF CN=CW+Q-1
180 CRS="FF000000000000FF00" :: CALL CH
AR(136 CRS 138 CRS) :: CALL MAGNIFY(
4) :: DISPLAY AT(9,10) ERASE ALL: SNA
P-CALC :: ON ERROR 1550
190 OPTION BASE 1 :: A=20 :: B=13
LC=F-1 :: R,Q,RW,G,H,I=1 :: DIM DS(
20),ES(20),FS(40),GS(21),J(20,13),X
(50)
200 DEF L(M)=ASC(SEGS(ES(N),M,G)) :: FOR
N=1 TO 500 :: NEXT N :: ON WARNING
NEXT :: DVS="DSK1" :: FNS="DATAFI
LE" :: PDVS="RS232 BA=9600 DA=8"
210 O=C :: CALL SCREEN(6) :: HS=KS :: GO
SUB 1150
220 GOSUB 1210 :: IF ((P>47 AND P<58) OR
P=46) AND CN>0 THEN GOSUB 1130 :: G
OTO 220
230 IF P>31 AND P<123 AND CN<1 THEN GOS
UB 1100 :: GOTO 220
240 IF P=133 AND RW<6 THEN RW=1 :: GOSU
B 1260 :: AC=0 :: GOTO 220 ELSE IF
P=133 THEN RW=RW-5 :: GOSUB 1260 ::
AC=0 :: GOTO 220
250 IF P=152 AND RW+5>A-9 THEN RW=A-9
:: GOSUB 1260 :: AC=0 :: GOTO 220 EL
SE IF P=152 THEN RW=RW+5 :: GOSUB 1
260 IF P=147 AND CW<3 THEN CW=0 :: GOSU
B 1260 :: AC=0 :: GOTO 220 ELSE IF
P=147 THEN CW=CW-3 :: GOSUB 1260
AC=0 :: GOTO 220
270 IF P=132 AND CW+3>LC-2 THEN CW=LC-1
:: GOSUB 1260 :: AC=0 :: GOTO 220
ELSE IF P=132 THEN CW=CW+3 :: GOSUB
1260 :: AC=0 :: GOTO 220
280 IF P>15 OR P=0 THEN 220 ELSE ON P G
OSUB 500,220,920,920,220,220,510,13
80,1350,1310,1330,1400,1310,350,290
:: IF O=1 THEN 220 ELSE 210
290 CALL CLEAR :: CALL DELSPRITE(ALL)
:: DISPLAY AT(12,1): "HALT: (Y/N)N"
ACCEPT AT(12,1) VALIDATE ("YN") SIZE(
-1): HS :: IF HS="Y" THEN STOP
300 O=0 :: RETURN
310 AC=0 :: IF CN>0 THEN J(RN,CN)=0
:: DISPLAY AT(R*2+3,(Q-1)*8+6): USING 1
500: J(RN,CN): ELSE DS(RN): " :: DIS
PLAY AT(R*2+3,4) SIZE(9)
320 RETURN
330 HS="CALCULATE MODE" :: GOSUB 1170
340 FOR T=G TO LC :: X=" " :: FOR N=G TO
O S :: DISPLAY AT(3,G)
N: "COLUMN": T :: X=0
350 IF ES(N)= " THEN 440
360 J(N,T)=0 :: FOR M=G TO LEN(ES(N))
U=L(M) :: IF M<LEN(ES(N)) THEN V=L(M
+G) ELSE V=G
370 GOSUB 480 :: W=V :: IF W=205 THEN X
=G :: M=M+G :: IF T>G THEN V=L(M+G)
:: W=J(V,T-G) ELSE W=0
380 IF U=201 THEN J(N,T)=J(N,T)+W :: M=
M+G :: GOTO 430
390 IF U=202 THEN J(N,T)=J(N,T)-W :: M=
M+G :: GOTO 430
400 IF U=203 THEN J(N,T)=J(N,T)*W :: M=
M+G :: GOTO 430
410 IF U=204 AND J(N,T)<>0 AND W<>0 THE
N J(N,T)=J(N,T)/W :: M=M+G :: GOTO
430 :: ELSE IF U=204 THEN J(N,T)=0
:: GOTO 430
420 V=U :: GOSUB 480 :: J(N,T)=V
430 NEXT M
440 X=X&CHR$(X) :: NEXT N :: NEXT T
:: DISPLAY AT(3,G) :: TOTAL COL
UMN
450 FOR N=G TO S :: J(N,F)=0 :: IF AV=G
THEN 470 ELSE IF ASC(SEGS(XS,N,G))
=G THEN J(N,F)=J(N,F-G) :: GOTO 470
460 FOR T=G TO F-G :: J(N,F)=J(N,F)+J(N
,T) :: NEXT T
470 NEXT N :: CALL SCREEN(6) :: HS=C$
:: GOSUB 1170 :: GOSUB 1260 :: RETURN
480 IF V>100 AND V<201 THEN V=K(V-100) E
LSE IF V<101 THEN V=J(V,T)
490 RETURN
500 GOSUB 710 :: O=2 :: RS="LOGIC MODE"
:: CALL SCREEN(4) :: GOSUB 1150
510 JS=">" :: GOSUB 1180
520 IF FS(G)="TOTAL" THEN IF FS(4)="OFF
" THEN AV=G :: F=0 :: GOTO 510 ELSE
F=VAL(FS(4)) :: LC=F-1 :: AV=0 :: G
OTO 510
530 IF FS(G)="LAST" THEN IF VAL(FS(4))<
F OR AV=1 THEN LC=VAL(FS(4)) :: GOTO
510 ELSE IF AV=0 THEN LC=F-1 :: GO
TO 510
540 IF FS(G)="END" THEN RETURN ELSE IF
FS(G)<>"MOVE" THEN 560
550 FOR N=G TO A :: J(N,VAL(FS(4)))=J(N
,VAL(FS(2))) :: NEXT N :: GOTO 510
560 IF FS(G)="LOGIC" THEN KS=FS(4) :: GO
TO 510 ELSE IF FS(3)="NULL" THEN DS
(VAL(FS(G)))=" " :: ES(VAL(FS(G)))="
" :: GOTO 510

```

```

570 IF FS(G)="LIST" THEN GOSUB 760 :: G
OTO 510 ELSE IF FS(G)="PRINT" THEN
AW=G :: GOSUB 1600 :: GOSUB 760
GOSUB 1620 :: AW=0 :: GOTO 510
580 IF FS(G)="NEW" THEN GOSUB 1530 :: G
OTO 510 ELSE IF FS(2)="15" THEN Y=V
AL(FS(G)) :: DS(Y)=SEGS(FS(3),1,9)
S=MAX(S,Y) :: GOTO 510
590 IF FS(2)<>" " THEN PRINT "COMMAND N
OT RECOGNIZED" :: GOTO 510 ELSE AA=
VAL(FS(G)) :: ES(AA)=" " :: S=MAX(S,A
A)
600 FOR N=3 TO 40 :: IF FS(N)=" " THEN 5
10
610 IF FS(N)="LAG" THEN U=205 :: GOTO 6
80
620 IF FS(N)="+" THEN U=201 :: GOTO 680
630 IF FS(N)="-" THEN U=202 :: GOTO 680
640 IF FS(N)="*" THEN U=203 :: GOTO 680
650 IF FS(N)="/" THEN U=204 :: GOTO 680
660 IF SEGS(FS(N),G,G)="" THEN GOSUB 7
00 :: AB=AB+G :: X(AB)=Y :: ES(AA)=
ES(AA)&CHR$(100+AB) :: GOTO 690
670 U=VAL(FS(N))
680 ES(AA)=ES(AA)&CHR$(U)
690 NEXT N :: GOTO 510
700 Y=VAL(SEGS(FS(N),2,LEN(FS(N))-2))
RETURN
710 CALL DELSPRITE(%G) :: RETURN
720 FOR N=1 TO 40 :: FS(N)=" " :: NEXT N
AD=G :: FOR N=1 TO 40 :: AE=POS
(LS AD) :: IF AE=0 THEN AE=LEN(L
S)+G
730 FS(N)=SEGS(LS,AD,AE-AD) :: AD=AE+G
:: IF AE>LEN(LS) THEN RETURN
740 NEXT N
750 IS="TO MANY WORDS" :: GOSUB 1200
RETURN
760 IS="LOGIC NAME IS &KS" :: GOSUB 120
0 :: IS="TOTAL COLUMN IS &STR$(F)
" :: GOSUB 1200 :: IS="LAST COLUMN IS
&STR$(LC)" :: GOSUB 1200
770 FOR N=G TO A :: CALL KEY(0,P, AH)
IF AH<>0 THEN CALL SOUND(100,440,0)
:: GOSUB 1230
780 IF LEN(DS(N))>0 THEN IS=STR$(N)&"
S &DS(N)" :: GOSUB 1200
790 IF ES(N)=" " THEN 910 ELSE IS="
800 IS=STR$(N)&" "
810 FOR M=G TO LEN(ES(N)) :: U=L(M)
820 IF U=201 THEN IS=IS&"+" :: GOTO 8
90
830 IF U=202 THEN IS=IS&"-" :: GOTO 8
90
840 IF U=203 THEN IS=IS&"*" :: GOTO 8
90
850 IF U=204 THEN IS=IS&"/" :: GOTO 8
90
860 IF U=205 THEN IS=IS&"LAG" :: GOTO
890
870 IF U>100 THEN IS=IS&"(" &STR$(K(U-10
0))&")" :: GOTO 890
880 IS=IS&STR$(U)
890 NEXT M
900 GOSUB 1200
910 NEXT N :: RETURN
920 IF P=4 THEN TTLS="SAVE MODE" ELSE T
TLS="LOAD MODE"
930 CALL DELSPRITE(ALL) :: DISPLAY AT(1,
1) ERASE ALL: TTLS: "ENTER YOUR CHOICE
" :: 1) DATA :: 2) LOGIC
940 ACCEPT AT(8,1) VALIDATE ("12") SIZE(1)
: LM :: IF LM=1 THEN 1030
950 DISPLAY AT(10,1): "ENTER DEVICE NAME
" :: DVS :: ACCEPT AT(11,1) SIZE(-28)
: DVS
960 IF KS=" " AND DVS<>"CS1" THEN DISPLA
Y AT(13,1): "ENTER FILE NAME:" :: KS
ACCEPT AT(14,1): KS
970 KS=SEGS(KS,1,10) :: IF DVS<>"CS1" TH
EN DVSS=SEGS(DVS,1,4)&"&KS ELSE D
VSS=DVS
980 IF P=3 THEN OPEN %G:DVSS: INPUT %INT
ERNAL:FIXED 192: ELSE OPEN %G:DVSS:O
UTPUT:INTERNAL:FIXED 192
990 IF P=4 THEN 1010
1000 INPUT %G:KS:S:F:LC:A:B :: FOR N=1 T
O A :: INPUT %G:DS(N):ES(N) :: NEXT
N :: O=0 :: CLOSE %G :: RETURN
1010 PRINT %G:KS:S:F:LC:A:B :: FOR N=1 T
O A :: PRINT %G:DS(N):ES(N) :: NEXT
N :: O=0 :: CLOSE %G :: RETURN
1020 IF P=3 THEN 1030
1030 DISPLAY AT(9,1): "ENTER DEVICE NAME
" :: DVS :: ACCEPT AT(10,1) SIZE(-28)
: DVS :: IF DVS="CS1" THEN DVSS=DVS
:: GOTO 1050
1040 DISPLAY AT(12,1): "FILE NAME:" :: F
NS :: ACCEPT AT(13,1) SIZE(-28) :: F
NS=SEGS(FNS&"",1,9)&" "
DVSS=SEGS(DVS,1,4)&"&FNS
1050 IF P=3 THEN 1080
1060 OPEN %G:DVSS: OUTPUT:INTERNAL:FIXED
128 :: FOR N=1 TO A :: PRINT %G:(N
,1):J(N,2):J(N,3):J(N,4):J(N,5):J(N
,6):J(N,7)

```

```

1070 PRINT #G:J(N,8):J(N,9):J(N,10):J(N,
11)J(N,12):J(N,13):NEXT N::CLO
SE #G:O=0::RETURN
1080 OPEN #G:DVS:INPUT:INTERNAL,FIXED
12B:FOR N=1 TO A:INPUT #G:J(N
1)J(N,2):J(N,3):J(N,4):J(N,5):J(N
6)J(N,7)
1090 INPUT #G:J(N,8):J(N,9):J(N,10):J(N,
11)J(N,12):J(N,13):NEXT N::CLO
SE #G:O=0::RETURN
1100 IF AC=0 THEN AC=1::MS=
1110 IF LEN(MS)<9 THEN MS=MS&CHR$(P) ELSE
CALL SOUND(50,220,0)
1120 DISPLAY AT(R*2+3,4)SIZE(9):MS::D$
(RN)=MS::RETURN
1130 IF AC<>G THEN AC=G::MS="0" ELSE 1
F LEN(MS)=8 OR VAL(MS&CHR$(P))>9999
99 THEN CALL SOUND(100,220,0)::RE
TURN
1140 MS=MS&CHR$(P)::J(RN,CN)=VAL(MS)::
DISPLAY AT(R*2+3,(Q-1)*8+6)SIZE(7):
USING 1500:VAL(MS)::RETURN
1150 DISPLAY AT(G,10)ERASE ALL:"SNAP-CAL
C":HS
1160 IF O=G THEN GOSUB 1260::GOSUB 122
0::RETURN ELSE RETURN
1170 DISPLAY AT(2,G):HS::RETURN
1180 Z=G
1190 INPUT ">":LS::GOSUB 720::IF AW=
G THEN PRINT #5:LS::RETURN ELSE R
ETURN
1200 IF AW=G THEN PRINT #5:HS::RETURN
ELSE PRINT #5::RETURN
1210 CALL KEY(0,P,AH)::IF AH=0 THEN 121
0 ELSE CALL SOUND(10,1000,8)::RETU
RN
1220 CALL SPRITE(#G,136,2,R*16+15,(Q-1)*
64+68)::RETURN
1230 CALL KEY(0,P,AH)::IF AH=-G THEN 12
30
1240 CALL KEY(0,P,AH)::IF AH=0 THEN 124
0

```

```

1550 CALL ERR(AR,AS,AX,AU)::IS="ERROR C
D="&STR$(AR)&" TYP="&STR$(AS)&" SEV
="&STR$(AX)&" AT "&STR$(AU)::IF AR
=109 OR AR=130 THEN AW=0
1560 CALL SOUND(200,110,0)
1570 DISPLAY AT(23,1):IS::FOR TD=1 TO
750::NEXT TD::DISPLAY AT(23,1):
XT::ON ERROR 1550::RETURN NE
1580 IMAGE "#####"
1590 IMAGE "#####"
1600 DISPLAY AT(23,1):ENTER:PRINTER DEV
ICE NAME::PDVS::ACCEPT AT(24,1)S
IZE(-28):PDVS
1610 OPEN #5:PDVS:DISPLAY:OUTPUT:VARIAB
LE 132::PRINT #5:CHR$(15)::RETUR
N
1620 CLOSE #5::RETURN

```

END OF ARTICLE

BABBAGE COMPUTER PIONEER

BABBAGE, COMPUTER PIONEER
Compiled by Gene Smith

Did you ever hear of CHARLES BABBAGE? He was a mathematician and engineer who was born in 1791 and was almost successful in making a mechanical computer.

Back in his day the term "computer" referred to a person who did math for a living. Usually, it was repetitious sweatshop toil, grinding out tables to be sold for use in engineering, navigation, banking and similar occupations. It was tremendously expensive and tedious to prepare these tables and inevitably contained mistakes with dire consequences.

Around 1820, Babbage decided to change all this. His exasperation with the inherently unreliable nature of man's arithmetical work reached the point where he decided to develop a machine that would do these computations mechanically.

Therefore, he embarked on a lifelong quest to build machines that would take man "out of the loop" and end the "reign of error."

Strangely enough, he almost succeeded.

```

1250 RETURN
1260 IF CW>0 THEN 1290
1270 DISPLAY AT(3,3):"ROW NAME" 1
2::N=R
1280 FOR R=1 TO 10::DISPLAY AT(R*2+3,1
2)::USING 1580:RN,D$(RN),J(RN,1),J(RN
2)::NEXT R::R=N::RETURN
1290 DISPLAY AT(3,7):CW,TAB(16):CW+1:TAB
(24):CW+2::N=R
1300 FOR R=1 TO 10::DISPLAY AT(R*2+3,1
2)::USING 1590:RN,J(RN,CW),J(RN,CW+1)
,J(RN,CW+2)::NEXT R::R=N::RETU
RN
1310 IF RN=A THEN RETURN ELSE R=R+1::A
C=0::IF R>10 THEN R=10::RW=RW+1
::GOSUB 1260
1320 GOSUB 1220::RETURN
1330 IF RN=1 THEN RETURN ELSE R=R-1::A
C=0::IF R<1 THEN R=1::RW=RW-1
::GOSUB 1260
1340 GOSUB 1220::RETURN
1350 IF CN=LC AND Q=2 THEN RETURN
1360 IF CN=LC THEN CW=F-2::Q=2::AC=0
::GOSUB 1260 ELSE Q=Q+1::AC=0
::IF Q>3 THEN Q=3::CW=CW+1::GOS
UB 1260
1370 GOSUB 1220::RETURN
1380 IF CN=0 THEN RETURN ELSE Q=Q-1::A
C=0::IF Q<1 THEN Q=1::CW=CW-1
::GOSUB 1260
1390 GOSUB 1220::RETURN
1400 CALL DELSPRITE(ALL)
1410 O=2::CALL CLEAR::INPUT "REPORT
TITLE:"R:TS::INPUT "DATE:"DT$::
INPUT "NUMBER OF ROWS TO REPORT ON
":NR::IF NR=0 THEN RETURN
1420 CALL CLEAR::DISPLAY AT(20,1):"ENT
ER DEVICE NAME FOR REPORT:PDVS"
ACCEPT AT(2,1,1)SIZE(-28):PDVS::OP
EN #5:PDVS:VARIABLE 132
1430 PG=1::PRINT #5:CHR$(15)
1440 PRINT #5:R:TS:TAB(40):DT$:R:TS(" 1
30)"PAGE":PG:R:TS(" 130)"ROW
":FOR TP=0 TO 13::PRINT #5:TAB
(TP*9+10):(PG-1)*13+TP:RP
1450 NEXT TP::PRINT #5:TAB(5):RP
TS(" 125)"TAB(5):FOR N=G T
O MIN(NR,A)
1460 PRINT #5:N:TAB(5):FOR M=0 TO
13::IF PG=1 AND M=0 THEN PRINT #
5:USING 1520:D$(N):ELSE PRINT #5:US
ING 1510:J(N,(PG-1)*14+M)
1470 IF (PG-1)*14+M>F THEN 1490
1480 NEXT M
1490 PRINT #5::NEXT N::PG=PG+1
IF (PG-1)*14<F THEN PRINT #5:CHR$(
12)::GOTO 1440 ELSE GOSUB 1620
RETURN
1500 IMAGE "#####"
1510 IMAGE "#####"
1520 IMAGE "#####"
1530 CALL DELSPRITE(#G)::FOR N=G TO A
:D$(N):E$(N)::NEXT N::FOR N
=G TO A:FOR M=G TO B:J(N,M)=0
::NEXT M::NEXT N
1540 AV,S=0::E$="":F=13::RETURN

```

The IBM SHOP

with Cyril Bohlsen

Only one remaining CD-ROM kit complete with sound card, SB30 speakers & cables plus 2 CD-disks , the lot \$ 200.00

Grollier 95 CD-rom \$ 42.00
Animals (ZOO) CD-rom \$ 36.00

20mB Conner HDD IDE \$ 25.00

3.5" Disk storage box \$ 10.00
5.25" Disk storage box \$ 10.00
Parallel printer cable \$ 10.00
Serial printer cable \$ 10.00
Keyboard 101 key \$ 30.00
Simm 30 pin 60ns \$ 70.00
Simm 30 pin 70ns \$ 55.00

NOTE : All prices listed are at time of printing, and may change at any time.

For current pricing please contact Cyril Bohlsen at the general meetings or Phone (02) 639 5847

In fact, if things would have gone just a little bit differently, he would have produced a fully mechanical computer more than a century before the advent of similarly capable electronic devices.

He began by developing numerous drawings and called his machine a "Difference Engine". He then turned it's construction over to a toolmaker whom he hired to produce the more than 4000 parts, many of which had complex shapes and had to be machined to close tolerances.

This construction ended in failure when the toolmaker walked off the job after 10 years which caused a great expenditure of money. Some questioned whether or not the Victorian era machine-tool technology was up to the task of producing these parts.

This failure haunted Babbage for the rest of his life (he died in 1871) and he was never able to get any more than a small part of any of his subsequent designs built.

However, recently the National Science Museum in London, England, who had Babbage's plans and drawings in their possession set about to build his Difference Engine No. 2, using only 19th century technology.

Parts were made of steel, bronze and cast iron. It has been completed and stands 7 feet high and weighs 3 tons. It is powered by a hand crank.

It took about a year of fabrication, assembly and testing before it was displayed. It forms the centerpiece of an exhibit commemorating the bicentennial of Babbage's birth.

It is said to bear an uncanny resemblance to computers of the 20th century and is programmed by using punch cards.

Therefore, if political, economic and other factors had been a little more favorable, the dawn of the Computer Age might very well have come over a hundred years sooner than it did.



CPU SPEED

by Cyril Bohlsen

Below is a listing of some of the different CPU's speed as given using NORTON SPEED CHECK VERSION '8', these speeds are related to an XT machine running at one (1).

| Type of CPU | Speed |
|-------------|-------|
| XT | 1 |
| 286 8mH | 4.4 |
| 386DX33 | 35.9 |
| 486DX33 | 71.2 |
| 486DX40 | 79.2 |
| 486DX50 | 99.0 |
| 486DX4/100 | 132.0 |
| Pentium 66 | 211.4 |

If anyone has some further information on any other CPU speeds, would they please pass them on to me so as I can update the listing.

SNAP CALC IBM style



The IBM version of *Snap-Calc*™ will run on the IBM PC, PCXT, and PCjr, with either BASICA, or Cartridge BASIC. The IBM versions contain several enhanced features because of their larger memory capacity and screen size. The program uses TEXT mode only, and will work equally well with monochrome and color monitors.

The best enhancement is the size of the spreadsheet matrix. The program is currently set up for 30 columns and 60 rows, which should be large enough for almost any home application. The matrix size can be altered very easily—simply change the value assigned to variables A and B in line 210. A represents the maximum number of rows set up, and B represents the maximum number of columns. The arrays are then dimensioned with these variables, and all limit checks are done accordingly.

Before haphazardly changing these variables, however, consider how much space you really need. The size of your matrix determines the amount of disk used to store it all. It also affects calculation time if the computer has to scan more rows for equations, and execute equations for more columns.

Because of the complexity of the program, operational differences exist between each of the four program versions. These minor differences, which could not be covered in the main text of this article, are covered here.

The Keys To Success

The IBM keyboard is quite different from other systems. On this program, its function keys and separate arrow keys are used in the following manner:

| FUNCTION KEY | FUNCTION |
|--------------|----------------------------|
| 1 | Load data from disk. |
| 2 | Save data to disk. |
| 3 | Clear entry/erase. |
| 5 | Calculate the logic model. |
| 6 | Print report. |
| 7 | Start logic entry mode. |
| 9 | Exit the program. |

On the PCjr you need to press [Fn] in conjunction with the number keys 1 through 9. On the PC and PCXT you can simply press the function keys on the left side of the keyboard.

The four arrow keys can be used to move the cursor around the screen from one cell to another. When the cursor attempts to go off the edge of the screen, the screen scrolls in the opposite direction to bring the desired cell into view.

On the PCjr you can move by pages by pressing the [Fn] key with an arrow key. This allows you to move up or down five cells at a time, or left or right three cells at a time. On the PC and PCXT you can use the [HOME], [END], [PG UP], and [PG DN] keys to do the same thing:

| PC PCXT | FUNCTION |
|---------|-------------|
| HOME | Page up. |
| BEND | Page down. |
| PG UP | Page right. |
| PG DN | Page left. |

Screen Size

The IBM versions display 40 columns of text. This is enough room to fit three columns of information on the screen at one time. The three columns can be either the row labels and two numeric columns, or three numeric columns. The row labels can be up to ten characters long, and the numeric values can be up to eight characters long, including a fixed decimal point. Only five digits may be entered to the left of the decimal point, and two digits to the right of it. Thus the maximum value that can be entered or displayed is 99999.99.

When entering numeric values in the spreadsheet, simply move the cursor to the selected cell and type in your numbers. The value will be updated on the screen automatically with every key press. After you have entered five digits to the left of the decimal point, the computer will not accept any more input except for a decimal point and digits to the right of it. If you enter only one digit to the right of the decimal, the computer will fill in a zero behind it.

Different Files

When you select either the Save or Load option, you are prompted to indicate whether you wish to work with the spreadsheet data or the logic model. The logic model includes all of the parameters that you set up in the Logic Entry mode. After selecting one of these you will be asked to enter a file name. If you selected the logic model under the Save option, the file name will be taken from the logic model name as specified in the Logic Entry mode. If no logic name has been specified, then you will be prompted for the file name, and the file name will then become the logic model name.

Logic Mode

Commands are entered in Logic mode with the INPUT statement of BASIC. This statement allows inputs of up to 255 characters—the maximum length for any command. The only command which can reach this length and still be a legal command is the one that lets you set up row equations (r = equation). It would take a very long and complicated equation to use up all 255 characters, and you will probably never encounter a case where more characters would be desirable.

The PRINT command will output to the system's default printer device. If the parallel printer interface is installed, it will be connected to the parallel port. If not, then output will be directed to the serial port.

BASIC Precision

If you use the IBM BASICA language, start up BASIC by answering the system prompt (A>) with BASICA/D. (This is to be done any time you RUN *Snap-Calc*™.) This enables the language to operate in double precision mode for increased accuracy. IBM Cartridge BASIC used with the PCjr requires no special start up to ensure high accuracy—just type BASIC.

[Note: The listing states that this program will run with Cassette BASIC. This is incorrect. This version will only run under BASICA or Cartridge BASIC. —Ed.]

| SNAP-CALC (IBM PC & PCjr) | |
|----------------------------|--|
| Explanation of the Program | |
| Line Nos. | |
| 100-200 | Program header. |
| 210-260 | Initialize program variables and key interrupt branching. |
| 270 | Control logic to display the data entry screen. |
| 280-290 | Control loop to accept entry on the data entry screen. |
| 300-320 | Routine to build numeric values and display them on the data entry screen. |
| 330-340 | Routine to build a row name and display it on the data entry screen. |
| 350-420 | Display the spreadsheet on the screen. |
| 430-440 | Two routines to move the cursor. |
| 450-560 | Load data or logic files. |
| 570-650 | Save data or logic files. |
| 660 | Routine to clear an entry or erase a row name. |
| 670-830 | Calculate row equations. |
| 840 | Calculate the totals column. |
| 850-900 | Print spreadsheet report to the printer. |
| 910 | Error routine for the report portion of the program. |
| 920-1100 | Main control loop for the logic entry mode. |
| 1110 | Routine to accept a new logic name. |
| 1120-1130 | Routine to accept a new totals column. |
| 1140-1180 | Routine for the NEW command. |
| 1190-1220 | Routine for the LIST command. |
| 1230-1250 | Routine to put the command string back together for the |

- 1260-1310 LIST and PRINT commands.
- 1320 Routine for the PRINT command.
- 1330 Logic mode error message.
- 1330 Return to data entry screen when END command is used.
- 1340-1350 Move up one cell.
- 1360-1370 Move left one cell.
- 1380-1390 Move right one cell.
- 1400-1410 Move down one cell.
- 1420-1430 Move up five cells.
- 1440-1450 Move left three cells.
- 1460-1470 Move right three cells.
- 1480-1490 Move down five cells.
- 1500-1520 Key scan routine.
- 1530-1550 Routines to turn key interrupts on and off.
- 1560-1590 Check for function arrow key inputs so that paging can be done.
- 1600 Program main error routine.
- 1610 Data containing logic mode commands.
- 1620 Data containing logic mode math operators.
- 1630 Routine to exit the program.

SNAP-CALC

IBM PC

```

100 REM *****
110 REM * SNAP-CALC *
120 REM *****
130 REM BY GARY STRAUSS AND
140 REM THE HCM STAFF
150 REM HOME COMPUTER MAGAZINE
160 REM VERSION 4.3.1
170 REM IBM PC & PCjr
180 REM CASSETTE OR CARTRIDGE BASIC
190 REM OR BASICA
200 REM
210 CLS:LOCATE 12,15:PRINT "SNAP-CALC":
OPTION BASE 1:A=60:B=30:TC=13:LC=12
:CL=0:RW=1:USS="":DIM DS(A)
:ES(A),FS(130),J(A,B),K(100):KEY OF
F
220 RPS="":N="":N1="":N2="":N3="":N4="":N5="":
S="+":USS="+":USS="+":USS="+":USS="+":
SS="+":USS="+":USS="+":USS="+":
USS
230 ON KEY(1) GOSUB 450:ON KEY(2) GOSUB
570:ON KEY(3) GOSUB 1660:ON KEY(5)
GOSUB 570:ON KEY(6) GOSUB 850:ON KE
Y(7) GOSUB 920:ON KEY(9) GOSUB 1630
240 ON KEY(11) GOSUB 1340:ON KEY(12) G
OSUB 1360:ON KEY(13) GOSUB 1380:ON K
EY(14) GOSUB 1400:ON ERROR GOTO 160
0
250 LOCATE 24,8:PRINT "PRESS ENTER TO C
ONTINUE":GOSUB 1500:RESTORE 1620:C
EAD NS(1),NS(2),NS(3),NS(4),NS(5):D
EF FNR=RW+R-1:DEF FNC=CL+C-1:R=1:C=
1
260 KEY 15,CHR$(&H40)+CHR$(&HC1):ON KEY
(15) GOSUB 1400
270 KEY(15) ON CLS:GOSUB 350:IF ES(FNR)
>
AND FNR<A THEN GOSUB 430:GOSUB
1400 ELSE IF ES(FNR)>
AND FNR=A T
HEN GOSUB 430:GOSUB 1340 ELSE GOSUB
430
280 DEF SEG=0:POKE 1950,PEEK(1052):GOSU
B 1510:IF FNC>0 THEN IF (KS>="0" AN
D KS<="9") OR KS="." THEN GOSUB 300
ELSE GOTO 280 ELSE GOSUB 330
290 GOTO 280
300 IF CP=1 THEN TIS="":CP=0 ELSE IF VA
L(TIS+KS)>99999.99 THEN RETURN
310 TIS=TIS+KS:J(FNR,FNC)=VAL(TIS)
320 LOCATE R+2+3,(C-1)*12+5:PRINT USING
USS:J(FNR,FNC):RETURN
330 IF CP=1 THEN TIS="":CP=0 ELSE IF LE
N(TIS)=10 THEN RETURN
340 TIS=TIS+KS:LOCATE R+2+3,5:PRINT TIS
-SPACES(10-LEN(TIS)):DS(FNR)=TIS:RE
TURN
350 LOCATE 1,1:PRINT NAMS:LOCATE 3,4:P
RINT CHR$(201):FOR Z=1 TO 35:PRINT
CHR$(205):NEXT:FOR Z=4 TO 23:LOCA
TE 2,4:PRINT CHR$(186):NEXT
360 IF CL=0 THEN LOCATE 2,5:PRINT "ROW
NAME" ELSE LOCATE 2,5:PRINT
CL:
370 LOCATE 2,21:PRINT CL+1:TAB(33):CL+2
:FOR Z=1 TO 10:RS=STR$(INT(RW+Z-1))
:LOCATE Z+2+3,4-LEN(RS):PRINT RS:N
EXT
380 IF CL>0 THEN 420

```

```

390 FOR Z=1 TO 10:LOCATE Z+2+3,5:IF LEN
(DS(Z+RW-1))=0 THEN PRINT Z+RW-1:
ELSE PRINT DS(Z+RW-1)+SPAC
ES(10-LEN(DS(Z+RW-1)))
400 NEXT
410 FOR Z=1 TO 10:LOCATE Z+2+3,17:PRINT
USING USS:J(RW+Z-1,CL+1):LOCATE Z+
2+3,29:PRINT USING USS:J(RW+Z-1,CL+
2):NEXT:RETURN
420 FOR Z=1 TO 10:LOCATE Z+2+3,5:PRINT
USING USS:J(RW+Z-1,CL):PRINT
LOCATE Z+2+3,17:PRINT USING USS:J(R
W+Z-1,CL+1):LOCATE Z+2+3,29:PRINT U
SING USS:J(RW+Z-1,CL+2):NEXT:RETUR
N
430 LOCATE R+2+2,(C-1)*12+5:PRINT STRIN
GS(8,196):LOCATE R+2+4,(C-1)*12+5:
PRINT STRINGS(8,196):RETURN
440 LOCATE R+2+2,(C-1)*12+5:PRINT
:LOCATE R+2+4,(C-1)*12+5:PRIN
T:RETURN
450 CLS:LOCATE 1,1:PRINT "ENTER YOUR CH
OICE":PRINT:PRINT "1) LOAD DATA":P
RINT "2) SAVE LOGIC"
460 KS=INKEY$:IF KS<"1" OR KS>"2" THEN
460 ELSE ON VAL(KS) GOTO 470,500
470 LOCATE 10,1:INPUT "ENTER FILE NAME:
":NS:NS=LEFT$(NS,8)+".HCL"
480 OPEN NS FOR INPUT AS #1
490 FOR Z=1 TO A:FOR Z1=1 TO B:INPUT #1
J(Z,Z1):NEXT:CLOSE #1:RETURN
270
500 IF NAMS="" THEN LOCATE 10,1:INPUT
ENTER LOGIC NAME: NAMS:NS=LEFT$(NA
MS,8)+".HCL"
510 OPEN NS FOR INPUT AS #1
520 INPUT #1,NAMS,S,TC,LC,A,B:IF
A1>A OR B1>B THEN LOCATE 12,1:PRINT
"THE LOGIC IS TOO LARGE. CHANGE LI
NE 210 A = :A1: AND B = :B1:CLOSE #
1:RETURN 270
530 IF A1<A THEN A=A1
540 IF B1<B THEN B=B1
550 FOR Z=1 TO A:INPUT #1,DS(Z),ES(Z):N
EXT
560 FOR Z=1 TO 100:INPUT #1,K(Z):NEXT:C
LOSE #1:RETURN 270
570 CLS:LOCATE 1,1:PRINT "ENTER YOUR CH
OICE":PRINT:PRINT "1) SAVE DATA":P
RINT "2) SAVE LOGIC"
580 KS=INKEY$:IF KS<"1" OR KS>"2" THEN
580 ELSE ON VAL(KS) GOTO 590,620
590 LOCATE 10,1:INPUT "ENTER FILE NAME:
":NS:NS=LEFT$(NS,8)+".HCL"
600 OPEN NS FOR OUTPUT AS #1
610 FOR Z=1 TO A:FOR Z1=1 TO B:WRITE #1
J(Z,Z1):NEXT:NEXT:CLOSE #1:RETURN
270
620 IF NAMS="" THEN LOCATE 10,1:INPUT
ENTER LOGIC NAME: NAMS:NS=LEFT$(NA
MS,8)+".HCL"
630 OPEN NS FOR OUTPUT AS #1
640 FOR Z=1 TO A:WRITE #1,DS(Z),ES(Z):N
EXT
650 FOR Z=1 TO 100:WRITE #1,K(Z):NEXT:W
RITE #1,NAMS,S,TC,LC,A,B:CLOSE #1:R
ETURN 270
660 TIS="":IF FNC>0 THEN J(FNR,FNC)=0:G
OSUB 360:GOSUB 430:RETURN 280 ELSE
DS(FNR)="":GOSUB 360:GOSUB 430:RETU
RN 280
670 LOCATE 25,13:PRINT "CALCULATING...
":FOR Z=1 TO S:IF ES(Z)="." THEN 83
0
680 FOR ZC=1 TO LC:J(Z,ZC)=0:NEXT:FOR Z
1=1 TO LEN(ES(Z)):U=ASC(MID$(ES(Z),
Z1,1)):IF Z1=1 THEN IF U<101 THEN U
1=U:LG=0:GOTO 720 ELSE IF U<201 THE
N U1=U:LG=0:GOTO 730
690 IF U=205 AND Z1=1 THEN Z1=Z1+1:U1=A
SC(MID$(ES(Z),Z1,1)):FOR ZC=2 TO LC
:J(Z,ZC)=J(Z,ZC)+J(U1,ZC-1):NEXT:GO
TO 820
700 LG=0:Z1=Z1+1
710 U1=ASC(MID$(ES(Z),Z1,1)):IF U1<101
THEN ON U-200 GOTO 720,740,760,810
ELSE IF U1<201 THEN ON U GOTO 730,7
50,780,800 ELSE IF U1=205 THEN Z1=Z
1+1:LG=LG+1:GOTO 710
720 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)+J
(U1,ZC-LG):NEXT:GOTO 820
730 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)+A
B(U1-100):NEXT:GOTO 820
740 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)-J
(U1,ZC-LG):NEXT:GOTO 820
750 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)-A
B(U1-100):NEXT:GOTO 820
760 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)+J
(U1,ZC-LG):IF J(Z,ZC)>99999.99 THEN
J(Z,ZC)=99999.99
770 NEXT:GOTO 820
780 FOR ZC=1+LG TO LC:J(Z,ZC)=J(Z,ZC)+A
B(U1-100):IF J(Z,ZC)>99999.99 THEN
J(Z,ZC)=99999.99
790 NEXT:GOTO 820

```

```

800 FOR ZC=1+LG TO LC: J(Z, ZC)=J(Z, ZC)/A
B(U1-100): NEXT: GOTO 820
810 FOR ZC=1+LG TO LC: J(Z, ZC)=J(Z, ZC)/J
(U1, ZC-LG): NEXT: GOTO 820
820 NEXT: Z1
830 NEXT: Z
840 IF TC=0 THEN RETURN 270 ELSE FOR Z=
1 TO A: J(Z, TC)=0: NEXT: FOR Z=1 TO A:
FOR Z1=1 TO LC: J(Z, TC)=J(Z, TC)+J(Z,
Z1): NEXT: NEXT: RETURN 270
850 CLS: LOCATE 1, 1: PRINT "ENTER REPORT
TITLE:"; INPUT TITLE; TS: PRINT PR
INT "ENTER TODAY'S DATE:"; INPUT DAT
E; DS: PRINT PRINT "HOW MANY ROWS:"
; INPUT "ROWS" RWS: PG=1: ON ERROR
GOTO 910
860 PRINT: PRINT "DO YOU WANT A FORM FEE
D WITH EACH PAGE (Y/N)?:"; INPUT FF
S: IF FF#="Y" THEN FF=12 ELSE IF FF#
="N" THEN FF=13 ELSE GOTO 860
870 LPRINT CHR$(15); SNAP-CALC; SPRE
AD SHEET; LPRINT: LPRINT TS: TAB(40):
DS: LPRINT STRINGS(132, 61): LPRINT PG
; LPRINT STRINGS(132, 61): LPRINT:
880 LPRINT "ROW, ROW NAME, TAB(22); PG: T
AB(32); PG+1; TAB(42); PG+2; TAB(52); PG
+3; TAB(62); PG+4; TAB(72); PG+5; TAB(82
); PG+6; TAB(92); PG+7; TAB(102); PG+8; T
AB(112); PG+9
890 FOR Z=1 TO RWS: LPRINT " "; LPRINT
USING RPS: Z, DS(Z), J(Z, PG), J(Z, PG+1
), J(Z, PG+2), J(Z, PG+3), J(Z, PG+4), J(Z
), PG+5), J(Z, PG+6), J(Z, PG+7), J(Z, PG+8
), J(Z, PG+9)
900 NEXT Z: IF PG<=9 THEN PG=PG+10: LPRIN
T CHR$(FF); GOTO 870 ELSE ON ERROR
GOTO 1600: RETURN 270
910 IF ERR<>58 THEN RESUME NEXT ELSE RE
SUME 270
920 GOSUB 1550: CLS: LOCATE 25, 11: PRINT
*** LOGIC MODE ***
930 LOCATE 24, 1: INPUT ">" LS: L=LEN(LS):
P=0
940 RESTORE 1610: FOR Z=1 TO 7: READ LNS:
IF INSTR(LS, LNS)=1 THEN ON Z GOTO 1
110, 1120, 1140, 1190, 1260, 1330, 1300
950 NEXT Z=1 TO 130: FS=INSTR(P+1, LS, " ")
960 IF FS>0 THEN FS(Z)=MID$(LS, P+1, FS-
P-1) ELSE FS(Z)=RIGHT$(LS, LEN(LS)-P
)
970 IF FS=0 THEN F=Z: GOTO 990
980 P=FS: NEXT: PRINT " "; LOGIC SENTENCE
MAY BE TRUNCATED " "; PRINT
990 IF FS(1)="MOVE" THEN 1290 ELSE LN=V
AL(FS(1)): IF LN<1 OR LN>A THEN 1320
ELSE IF FS(2)="" THEN 1000 ELSE IF
FS(3)>15 THEN 1320 ELSE IF FS(
3)="NULL" THEN DS(LN)="": E(LN)=("):
GOTO 930 ELSE DS(LN)=LEFT$(FS(3)+FS
(4)+FS(5)+FS(6)+FS(7)+10): GOTO 930
1000 E(LN)=((LN>S)+LN*(1-1))+(S>LN
)*S*(1-1): IF S=0 THEN S=LN
1010 FOR N=3 TO F
1020 IF FS(N)="LAG" THEN U=205: GOTO 1090
1030 IF FS(N)="+" THEN U=201: GOTO 1090
1040 IF FS(N)="*" THEN U=202: GOTO 1090
1050 IF FS(N)="#" THEN U=203: GOTO 1090
1060 IF FS(N)="/" THEN U=204: GOTO 1090
1070 IF LEFT$(FS(N), 1)="/" THEN AB=AB+1:
K(AB)=VAL(MID$(FS(N), 2, LEN(FS(N))-2
)): E(LN)=E(LN)+CHR$(100+AB): GOTO
1100
1080 U=VAL(FS(N)): IF U<1 OR U>A THEN 132
0
1090 E(LN)=E(LN)+CHR$(U)
1100 NEXT: GOTO 930
1110 NAMS=RIGHT$(LS, LEN(LS)-14): GOTO 930
1120 TCS=MID$(LS, 17, LEN(LS)-16): IF TCS=
OFF THEN TC=0 ELSE TC=VAL(TCS): IF
TC>0 THEN LC=TC-1
1130 GOTO 930
1140 PRINT "ENTER:"; PRINT "1) DATA"; PRIN
T "2) LOGIC"; PRINT "3) ALL - BOTH L
OGIC AND DATA"; PRINT "4) ABOR
T NEW COMMAND"; PRINT: PRINT
1150 K1$=INKEY$: IF K1$<"1" OR K1$>"4" TH
EN 1150 ELSE ON VAL(K1$) GOTO 1160,
1170, 1180, 930
1160 FOR Z=1 TO 40: FOR Z1=1 TO 13: J(Z, Z1
)=0: NEXT: NEXT: GOTO 930
1170 FOR Z=1 TO 40: E(Z)=0: NEXT: GOTO 92
0
1180 FOR Z=1 TO 40: FOR Z1=1 TO 13: J(Z, Z1
)=0: NEXT: E(Z)=0: NEXT: GOTO 930
1190 PRINT "LOGIC NAME IS": NAMS: PRINT "
TOTAL COLUMN IS": TC: PRINT "LAST COL
UMN IS": LC: FOR Z=1 TO 40
1200 IF D$(Z)>"" THEN PRINT Z: IS=D$(
Z)
1210 IF E$(Z)>"" THEN GOSUB 1230: PRINT S
TR$(Z): IS=IS
1220 NEXT: GOTO 930
1230 IS="": FOR X=1 TO LEN(E$(Z)): U=ASC(M
ID$(E$(Z), X, 1)): IF U=255 THEN RETURN
ELSE IF U>200 THEN IS=IS+N$(U-200

```

```

1240 IF U>100 AND U<200 THEN IS=IS+"(" +S
TR$(K(U-100))+"") ELSE IF U<100 THE
N IS=IS+RIGHT$(STR$(U), LEN(STR$(U))
-1)
1250 NEXT: RETURN
1260 LPRINT "LOGIC NAME IS": NAMS: LPRINT
"TOTAL COLUMN IS": TC: PRINT "LAST C
OLUMN IS": LC: FOR Z=1 TO 40: IF D$(Z)
">"" THEN LPRINT STR$(Z): IS=D$(Z)
)
1270 IF E$(Z)>"" THEN GOSUB 1230: LPRINT
STR$(Z): IS=IS
1280 NEXT: GOTO 930
1290 FOR Z=1 TO A: J(Z, VAL(FS(4)))=J(Z, VA
L(FS(2))): NEXT: GOTO 930
1300 LC=VAL(RIGHT$(LS, LEN(LS)-15)): IF (L
C>TC AND TC=0) OR LC>B THEN PRINT "
LAST COLUMN IS OUT OF RANGE": PRINT:
IF TC>0 THEN LC=TC-1 ELSE LC=B
1310 GOTO 930
1320 PRINT: PRINT "ERROR ** CAN'T UNDERST
AND YOUR ENTRY": PRINT: GOTO 930
1330 RETURN 270
1340 CP=1: GOSUB 1540: IF R=1 THEN RW=RW-1
: IF RW<1 THEN RW=1: GOSUB 360 ELSE G
OSUB 360 ELSE GOSUB 440: R=R-1: GOSUB
430
1350 IF E$(FNR)>"" THEN 1340 ELSE RETURN
1510
1360 CP=1: GOSUB 1540: IF C=1 THEN CL=CL-1
: IF CL<0 THEN CL=0: GOSUB 360 ELSE G
OSUB 360 ELSE GOSUB 440: C=C-1: GOSUB
430
1370 RETURN 1510
1380 IF FNC=LC AND TC=0 THEN 1390 ELSE
IF FNC=LC THEN GOSUB 440: CL=TC-2: C=2
: GOSUB 360: GOSUB 430: GOTO 1390 ELSE
CP=1: GOSUB 1540: IF C=3 THEN CL=CL+
1: IF CL>B-2 THEN CL=B-2: GOSUB 360 E
LSE GOSUB 360 ELSE GOSUB 440: C=C+1:
GOSUB 430
1390 RETURN 1510
1400 CP=1: GOSUB 1540: IF R=10 THEN RW=RW+
1: IF RW>A-9 THEN RW=A-9: GOSUB 360 E
LSE GOSUB 360 ELSE GOSUB 440: R=R+1:
GOSUB 430
1410 IF E$(FNR)>"" AND FNR<A THEN 1400 E
LSE IF E$(FNR)>"" AND FNR=A THEN 13
40 ELSE RETURN 1510
1420 CP=1: GOSUB 1540: RW=RW-5: IF RW<1 THE
N RW=1
1430 GOSUB 360: IF E$(FNR)>"" THEN 1340 E
LSE 1510
1440 CP=1: GOSUB 1540: CL=CL-3: IF CL<0 THE
N CL=0
1450 GOSUB 360: GOTO 1510
1460 IF FNC+3>LC AND TC=0 THEN RETURN EL
SE GOSUB 1540: IF FNC+3>LC THEN GOSU
B 440: CL=TC-2: C=2: GOSUB 430 ELSE CP
=1: CL=CL+3: IF CL>B-2 THEN CL=B-2
1470 GOSUB 360: GOTO 1510
1480 CP=1: GOSUB 1540: RW=RW+5: IF RW>A-9 T
HEN RW=A-9
1490 GOSUB 360: IF E$(FNR)>"" THEN 1400 E
LSE 1510
1500 AS=INKEY$: IF AS="" THEN 1500 ELSE B
E: RETURN
1510 GOSUB 1530: DEF SEG=0: POKE 1050, PEEK
(1052)
1520 K$=INKEY$: IF K$="" THEN 1520 ELSE IF
K$=CHR$(13) THEN GOSUB 1400: GOTO
1520 ELSE IF LEN(K$)>1 THEN GOSUB 1
550: GOTO 1560 ELSE GOSUB 1550: RETUR
N
1530 FOR Z=1 TO 15: KEY(Z) ON: NEXT: RETURN
1540 FOR Z=1 TO 15: KEY(Z) OFF: NEXT: RETUR
N
1550 FOR Z=1 TO 15: KEY(Z) STOP: NEXT: RETU
RN
1560 IF RIGHTS(K$, 1)=CHR$(71) THEN 1420
1570 IF RIGHTS(K$, 1)=CHR$(73) THEN 1440
1580 IF RIGHTS(K$, 1)=CHR$(81) THEN 1460
1590 IF RIGHTS(K$, 1)=CHR$(79) THEN 1480
ELSE RETURN
1600 LOCATE 25, 1: PRINT "*** ERROR": ERR
="IN LINE": ERL="": CHR$(7): FOR E
R=1 TO 1000: NEXT: LOCATE 25, 1: PRINT
" "; RESUME NEXT
1610 DATA LOGIC NAME IS: TOTAL COLUMN IS:
NEW LIST: PRINT: END: LAST COLUMN IS:
DATA + " " " " " " " " " " " " " " " " " LAG
1620 GOSUB 1550: CLS: LOCATE 1, 1: PRINT CHR
$(7): CHR$(7): ARE YOU SURE YOU WANT
TO EXIT THE PROGRAM (Y/N)?":
INPUT EX$: IF EX$="Y" THEN END ELSE
RETURN 270

```

END OF ARTICLE 

REGIONAL GROUP REPORTS

Meeting Summary For OCTOBER

Central Coast 14/10/95 Saratoga
 Glebe 12/10/95 Glebe
 Hunter Valley 15/10 22/10/95
 Illawarra 10/10/95 Keiraville
 Liverpool 13/10/95 Yagcona West
 Sutherland 20/10/95 Jannali

CENTRAL COAST Regional Group

Regular meetings are normally held on the second Saturday of each month, 6.30pm at the home of John Coulton, 34 Mimosa Ave., Saratoga, (043) 69 3990. Contact Russell Welham (043)92 4000.

GLEBE Regional Group

Regular meetings are normally on the Thursday evening following the first Saturday of the month, at 8pm at 43 Boyce Street, Glebe. Contact Mike Slattery, (02) 692 8162.

HUNTER VALLEY Regional Group

The Meetings are usually held on the second or third Sunday of each month at members homes starting at 3pm. Check the location with Geoff Phillips by leaving a message on (049) 428 617. Please note that the previous phone number (049) 428 176 is now used exclusively by the ZZAP BBS which also has TI support. Geoff.

ILLAWARRA Regional Group

Regular meetings are normally held on the first Tuesday of each month after the TISHUG Sydney meeting at 7.30pm, at the home of Geoff Trott, 20 Robsons Road, Keiraville. A variety of investigations take place at our meetings, including Word Processing, Spreadsheets and hardware repairs. Contact Geoff Trott on (042) 29 6629 for more information.

LIVERPOOL Regional Group

Regular meeting date is the Friday following the Tishug Sydney meeting at 7.30 pm. Contact Larry Saunders (02) 644-7377 (home). After 10.30 PM or at work (02)602 3312 Liqueurland Liverpool West for more information.

*** ALL WELCOME ***

13th OCTOBER 1995 : MY PLACE

10th NOVEMBER 1995 : MY PLACE

Bye for now Larry.
 Liverpool Regional Co-Ordinator

SUTHERLAND Regional Group

Regular meetings are held on the third Friday of each month at the home of Peter Young, 51 Jannali Avenue, Jannali at 7.30pm. Peter Young.

TISHUG in Sydney

Monthly meetings start promptly at 2pm on the first Saturday of the month. They are held at the MEADOWBANK PRIMARY SCHOOL, on the corner of Thistle Street and Belmore Street, Meadowbank. Cars can enter from Gale Street and park in the school grounds. Regular items include news from the directors, the publications library, the shop, and demonstrations of monthly software.

OCTOBER MEETING - 7th OCTOBER

NOVEMBER MEETING - 4th NOVEMBER

The cut-off dates for submitting articles to the Editor for the TND via the BBS or otherwise are:

NOVEMBER 14th OCTOBER

These dates are all Saturdays and there is no guarantee that they will make the magazine unless they are uploaded by 6:00 pm, at the latest. Longer articles should be to hand well before the above dates to ensure there is time to edit them.

FAMOUS(?) QUOTES(S)

A man in love is incomplete until he is married.
 Then he is finished.

Zsa Zsa Gabor

One father is worth more than a hundred schoolmasters.

George Herbert
 (1592-1633)

Never have children, only grandchildren.

Gore Vidal