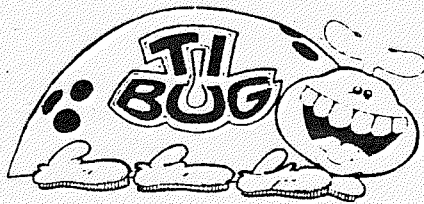


# TIBUG

Texas Instruments  
Brisbane User Group

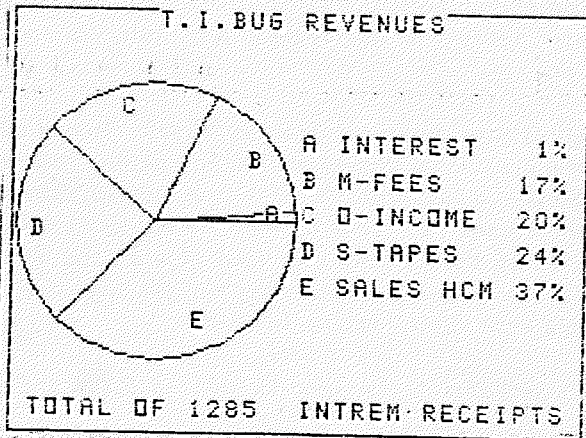


# JUNE

Registered by Australia Post  
Publication No. QBH 3671  
Postal Address P.O. Box 57,  
Aspley, Q. 4034

# 1985

# BUG—BYTES



## CONTENTS

32K MATCHBOX EXP.	3	HI-RES MONITOR	17
NOISELESS PE BOX	6	ELECTRONIC TYPEWRITER	18
PROGRAMMING TUTORIAL	8	PROGRAMMING HINTS	20
ASSEMBLY	9	LIMERICKS	22
BOGGLED	10	SERIES & THE COMPUTER	24
GOLD LOTTO	13	FINANCIAL REPORT	30
MICROWAVE CALCULATOR	16	TRADING POST	32

## NEXT MEETING

28th JUNE 1985  
&  
26th JULY 1985

EAST BRISBANE STATE SCHOOL  
Cnr. Wellington Rd &  
Stanley St.  
EAST BRISBANE

---

## EDITORIAL

Have a read of the article on the next page. It will blow your mind. Our friends from Perth, Bernie Elsner and Phil West have come up with brilliant modification to the console. A 32K memory expansion that fits inside the console in an unbelievably small space. They call it a Matchbox Expansion for that reason. I couldn't fit the whole article in the newsletter, but if you are interested there will be copies of it available at the next meeting. I personally would like to congratulate Bernie and Phil for their fantastic achievement and thank you for being willing to share it with us.

I haven't received many questionnaires back yet. Please remember to fill them in. If the ones I have received are any guide, then more than 90% of Peripheral Expansion Boxes complete with disk drives, 32K and RS232 cards. If there are members out there with only cassette players, get your questionnaires in so that I have some idea of what you want to see in the newsletter. It won't be good for you if I get the idea that everyone has disk drives, and base future issues in that.

We still need a secretary!! The need is getting desperate. Please help if you can. Don't leave it to "someone else". This time there doesn't seem to be a "someone else". Also, I would like someone to help me write a gossip column about club members and activities. Any volunteers?

---

## COMMITTEE

PRESIDENT	NOEL WILSON	205 7407
VICE PRESIDENT	HUMPHREY LINDLEY	263 6161
SECRETARY	?	
ASSISTANT SECRETARY	LES TUTCHINGS	281 8406
TREASURER	HUMPHREY LINDLEY	263 6161
PROGRAM LIBRARIAN	BOB MELKSHAM	271 1968
BOOK LIBRARIAN	NICK LOBODIN	349 3246
EDITOR	SANDRA NICHOLSEN	341 5667
PROJECT OFFICER	MIKE BRYZAK	345 1912

## 32K MATCHBOX EXPANSION

by Bernie Elsner and Phil West, TIUP

The following is a short excerpt from the article in Tit-Bits, TIUP's Newsletter. The full article will be available at the next meeting for anyone who wants one. It is also available by post for \$1 to cover postage and photocopying costs.

Because the TI is no longer produced, hardware and software are becoming available at bargain (DUMPED) prices, therefore the initial awe and fear of damaging expensive equipment has disappeared. (Who's afraid of a \$49.95 computer?). At this stage, like us, you may develop an interest in computer hardware. How it operates, how to fix some of the things that annoy you and how to provide extra features.

Hardware enhancements that we have added include:-

Installation of a great UTILITY device called 'FINGER PRINT' and the addition of five EXTERNAL dip switches for the TI-(MX 80)-PRINTER. (For changing Baud Rate and 7-8 Data Bits quickly.)

Beefing up the Expansion Box power supply to permit the use of two internal, CHINON low power, double sided, slimline disk drives, PLUS, two optional external drives and WITHOUT ANY FAN in ELSNER's 'MEGA-CRAZY' supermod! (Foundation member of 'NOFANS' - The Noise Of Fans is Against Nature Society.)

Installation of a 'GRONOS' load interrupt button. (The MARK 2 'LA-USERS' version.)

Addition of the 'NEW HORIZON'S USERS GROUP' real time (battery backed) clock.

Conversion of a MINI MEMORY to a 'WESTRALIAN INSTRUMENTS' MAXI MEMORY, by replacing the 4K ROM and 4K RAM with an 8K RAM chip. Retaining the MM GROM and battery backed circuit. The ROM data can be restored from cassette with a M-L routine (or from disk, using a CORCOMP DD disk controller card. (To be featured in Part 2.)

Location of a 'CONSOLE ROM' in an 8K (battery backed) RAM CHIP on the 8 bit data bus. Slowed things down by 25% but has some interesting potential.

Expansion of CPU memory with a 'MATCHBOX' 32K CMOS RAM EXPANSION inside the TI-99/4A. (Described in this article.)

These last three mods were made possible by substantial price reductions of a great memory chip, the:-

HITACHI HM6264LP-15

This is a 28 pin- 8K\*8 bit CMOS RAM chip. A big brother to the 2K \* 8 bit HM61161P-4 used in the MINI MEMORY. There are several versions of this chip. The one we used is LP (for low power - which is required if you intend to use battery backup) and 15 (for 150 nanoseconds) which is plenty fast enough for ol' TORTOISE-TI...

A year ago in Perth, the chip would have cost you \$140 and two months ago, it varied between \$42 and \$75. It is now available, in sticks of TEN chips from:-

PROMARK ELECTRONICS

p.o. Box 381,

CROWS NEST, NSW 2065 for \$256 (425.60 per chip) including sales tax and postage)

The HM6264 chip is just made for TI-99/4A hacker. It is STATIC RAM which does not need the complex refresh circuitry etc. of DYNAMIC RAM and the size is just right for the 8K blocks of CPU RAM.

We develop our projects on old equipment and accept the risk that it may be damaged. If you have never taken your computer apart, handled CMOS chips or used a soldering iron, you should NOT ATTEMPT any of the projects described (unless you can obtain assistance from someone who has.)

\*\*\*\*\* WARNING \*\*\*\*\*

\* \* \*

\* If you attempt any of the modifications \*

\* described in this series of articles \*

\* you do so at your own risk!!!! \* \*

\* \* \*

\*\*\*\*\*

32K MATCHBOX EXPANSION works fine with the major software packages. EXTENDED BASIC, TI-WRITER, MULTIPLAN, TI-LOGO 1 & 2, TI-FORTH, EDITOR ASSEMBLER and the RS232, P-CODE, TI-DISK CONT., CORCOMP DD DISK CONT. cards have all been used without any 'APPARENT' problems, however moving into CPU RAM into the computer is a major change to the TI-99/4A expansion system and may cause some software problems. Machine language programs that use SPEECH or are dependant on critical timing MAY NOT operate correctly.

This project should be regarded as a "BUDGET" MEMORY EXPANSION for the unexpanded TI-99/4A user rather than a replacement for existing 32K expansion cards! If you wish to add the 32K expansion but don't have a disk drive or controller, you should also be aware that the extra memory is not fully useable by cassette based users.

#### PARTS REQUIRED

4 Hitachi HM6264LP-15 RAM chips (For M/L use you may add them one at a time as required.)

1 piece of copper strip board (34 strips wide and 23 cm. long, if you intend to add other projects later.)

1 (or 4) 28 Chip sockets.

1 22 uF Tantalum Capacitor.

An edge connector or small plug socket for 30 or more wires. (We will need more wires for future projects, so use larger connector if available.)

Several Non-conducting standoffs or a sheet of insulating material to separate the strip board from the metal shielding around the computer.

Coloured lengths of thin Insulated single conductor wire for use on the stripboard.

Lengths of thin, multi-stranded wire or ribbon cable to provide flexible connections between the board, Grom Extender and main computer board.

Solder, fine tipped iron, etc.

Now, if this sound like something right up your alley, then ring Humphrey right now to reserve your copy of this article complete with full instructions. If you are like me, and like the sound of it, but wouldn't know one end of a soldering iron from the other, don't despair. Lindley and Assoc. have lined up somebody to do the job for you for the complete all up price of \$200. So, if you want a console with 48K RAM, then either get the article and do it yourself, or give Humphrey a ring and have it done for you.

Who wants a Commodore 64? It only has 64K all up, ROM and RAM together. With this expansion, and your Extended Basic module, you have 48K RAM, 26K console ROM and 36K ROM in the Extended Basic module. A grand total of 110K. It makes the Commodore look silly doesn't it?

-31787,0

NOISELESS PERIPHERAL EXPANSION BOX

by Bob Hubel, MSP 99

(DSK1 LOAD (R))

949113

Are you distracted and disturbed by the tornadic roar of your Peripheral Expansion Box? I engineered the box to provide sufficient cooling capacity for the most strenuous of circumstances -- all 8 card slots occupied and under heavy, continuous usage. Since my use didn't even approach the design limits, I began experimenting with ways to slow the fan down, and I was successful in reducing the noise level down to a barely perceptible purr! I have been testing this change for a sufficient period of time, and I now feel comfortable in recommending this modification to others. In fact, I have even run under light loads for moderate periods of time without any fan at all -- but I don't personally advise going to that extreme.

I had considered 2 options -- purchase a quieter fan, or just slow down the present fan. A quieter fan costs about \$20 or more, so I quickly discarded that option. I could reduce the fan speed with either a special solid state voltage regulation device or merely install a power resistor in series with the fan. Since I have no experience with such solid state devices and the components would likely cost around \$10, I elected for the latter method at a cost of \$0, using components from my electronics junk box. If you had to purchase the components, the cost would probably be around \$2 to \$3.

The PEB fan is rated at 14 watts. I have found that inserting a 500 - 700 Ohm power resistor (10 Watts power dissipation) reduces the speed to a very acceptable level (700 Ohms is my own preference). (Since a 700 Ohm resistor may be hard to find, you can combine 2 or more resistors which add up to 700 Ohms.) This resistor "steals" the energy that was originally intended for the fan and thus the fan doesn't work as hard. However, in doing so, the resistor must shed this extra energy itself, and does so by producing heat. Therefore you must mount the resistor on the OUTSIDE of the PEB, immediately behind the fan, thus allowing the circulating air to cool it. Do NOT mount it inside the cabinet. Although this may be more aesthetically pleasing, it will add unwanted heat inside the cabinet.

Procedure to disassemble the PEB to access the fan lead wires:

1. Unplug power cord.
2. Remove lid.
3. Remove disk drive (4 screws - 2 on top, 2 on bottom; disconnect 2 cables).

4. Remove all slide-in cards.
5. Remove 2 screws on cabinet sides.
6. Remove 6 screws on back (but not the 2 screws holding the lid clips.)
7. Remove 5 screws on the perimeter of the base, plus the one near the middle.
8. Slide base out.

The next step requires a little bit of ingenuity and personal customization. You must disconnect 1 of the fan leads and extend those 2 wires to the outside back of the box through a drilled hole. Splice the resistor/s between those wires, then mount them on the box next to the fan exhaust. I can't tell you exactly how to do this for there are many possible ways. If you are interested in looking at my example, give me a call and I can show you how I did it. Be sure that all wires are insulated as this is a 115V circuit. After that is done, reverse the above steps to reassemble the box.

Since the fan air intake is through the card cage, I have cut foam to fit in the empty slots, at the far forward side of the card cage. This then forces the reduced airflow to flow through the existing cards, thereby increasing their cooling.

The resistors will run warm and possibly be uncomfortable to touch -- this is normal. Don't cheat and use less than 10 Watt resistors as then they may get too hot and cause problems.

If anyone knows of a cheap solid state equivalent which will perform the same function, please let me know, as it would be the preferred method, if inexpensive. You'll be amazed at the considerable reduction in noise pollution.

Good Luck!!

---

#### DEATH OF A COMPUTER

There is a very good article in the April edition of the Melbourne Times, called "Death of a Computer", about as you can imagine, the demise of the TI-99/4A. It is too long for me to publish in this newsletter (there wouldn't be much room for anything else), but if you want to read it, there will be copies of it available at the monthly meeting, or if you can't attend that, it is available by post for \$1 to cover the cost of postage and photocopying.

---

PROGRAMMING TUTORIAL  
by George F. Steffen, LA 99ers

I have always heard the sub-routines should be placed at the beginning of BASIC programs in order to maximise the speed of execution since BASIC searches for lines from the lowest until it gets to the number it wants. While at the TI Fest in San Francisco 2 years ago, I asked one of the TI people there and was told that in TI BASIC short jumps to higher numbers would be fastest. However I finally got around to doing some tests.

I used the program that I wrote to keep the membership list, which is 65 sectors on disk and has approx. 400 lines. I put a 1000 count delay loop at the beginning, middle, and end of the program. In order to eliminate prescan time, I started the small routine with a BREAK and looped back to a BREAK after the delay loop. thus, I could time the loop from a CON command until the next time it broke. There was no measurable difference in the delay loop depending on where it was located.

I then put a GOSUB to a RETURN statement in the middle of each loop. Now there was a significant difference in the loop speed depending on the location of the RETURN statement. When the RETURN was at the beginning of the program, the loop took approximately four times as long to execute as it did without the GOSUB. When the RETURN was the last line in the program, the time difference could not be measured. Of course, I was trying to look at my wrist watch while waiting for the cursor to return, so I could not count time to the exact second. I confirmed this result by getting an intermediate time for GOSUB to RETURN in the middle of the program. This indicates that TI BASIC searches for line numbers in GOSUB statements from the last line forward. Therefore, frequently used sub routines should be placed as close as possible to the end of the program.

Since I was doing the time testing, I decided also to test RESTORE. I put a RESTORE statement and a READ statement inside the loop. I was forced to reduce the loop count to 100 since I got tired of waiting for it to operate 1000 times. This time, the results were exactly the opposite. The earlier in the program the DATA statement being read was located, the faster the loop. Also, if the RESTORE did not point to a DATA statement, there was a significant delay while the program kept looking for the next DATA. This indicates that DATA statements should be grouped at the beginning of the program with the most used ones first. If there are a large number of DATA statements, the first line in the program should be a GOTO to the first executable line.



If you are using the PRESCAN OFF command in your programs, you should start with any variables being initialized to constants, then the GOTO executable line, mention of all variables and sub programs used, one DATA line the PRESCAN OFF followed by the remaining DATA lines.

Although I did not test GOTO statements, they should use the same methods as in GOSUB. The only difference is that there is no return address saved. In this case also, jumps to a higher line number should be faster. The routine for IF - THEN - ELSE also should search the line table in the same order. I suspect that the RESTORE searches in the opposite order because it cannot be sure its search is finished when it gets to the line number it is searching for. It has to continue searching until the line is equal or greater than the line number in the RESTORE statement and the first statement in that line is a data statement.

---

### ATTENTION ASSEMBLY LANGUAGE

PROGRAMMERS!

HAVING PROBLEMS?

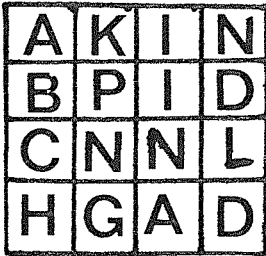
They could be caused by the E.I.S. (Enhanced Instruction Set) recently implemented on the TI 99/4A.....

Listed below are the mnemonics followed by a brief description.

BH	Branch and Hang	BTAD	Branch To Auto Destruct
HCF	Halt and Catch Fire.	JRL	Jump to Random Location
BOI	Byte Operator Immediately	JSP	Jump on Sexy Programmer
CDS	Condense and Destroy System	BSO	Branch on Sleepy Operator
CLBR	Clobber Register	RCI	Read Card and Ignore
BOB	Branch On Bug	RASC	Read And Shred Card
CRN	Convert to Roman Numerals	RCS	Read Card Sideways
CPPR	Crumple Printer Paper and Rip	DO	Divide and Overflow
EIOC	Execute Invalid Op Code	DC	Divide and Conquer
EPI	Execute Programmer Immediately	FCE	Fill Core with Epoxy
EDU	Execute Dissatisfied User	CCP	Clear Core and Proceed
IBP	Insert Bug and Proceed	PCB	Pause for Coffee Break
RPM	Read Programmer's Mind	SPD	Start and Power Down
RSD	On Read Error self Destruct	UER	Update and Erase Record
MET	Misread and Eat Tape	WRTC	Wind Real Time Clock
STJ	Stretch and Tangle Tape	WWTC	Wind Wrong Time Clock
SRSD	Seek Record and Scar Disk	RWOM	Read Write-Only Memory
ED	Eject Disk	WROM	Write Read-Only Memory

With Acknowledgements to 'TACONEWS' May/June 1983, and 'TIUP TIT-BITS' VOL 3 NO 4.

BOGGLED by BATZ In the ATICC



This program duplicates the famous word game of the same name, which you play with dice. The object is to find as many words of three or more letters made up of adjoining letters, vertically horizontally or diagonally. There is a time limit, so see how you go!

```

100 REM *****
110 REM *   BOGGLED   *
120 REM *   BY BATZ   *
130 REM *   IN THE ATICC*
140 REM *   13.12.84   *
150 REM *   BOTH BASICS *
160 REM *****
170 CALL CLEAR
180 CALL SCREEN(16)
190 PRINT TAB(12);"BOGGLED":
   : : :
200 PRINT TAB(14);"BY": : :
   :
210 PRINT TAB(7);"BATZ IN TH
E ATICC": : : : :
220 PRINT "DO YOU REQUIRE IN
STRUCTIONS?": :
230 PRINT TAB(12);"(Y/N)"
240 CALL KEY(0,K,S)
250 IF S=0 THEN 240
260 IF K=89 THEN 1320
270 IF K<>78 THEN 240
280 CALL CLEAR
290 TM=180
300 XX=1
310 V=9
320 H=13
330 D=26
340 CALL COLOR(11,2,1)
350 FOR ST=1 TO 8
360 CALL COLOR(ST,6,6)
370 NEXT ST
380 CALL CHAR(42,"007E7E7E7E
7E7E00")
390 CALL CHAR(81,"6090959595
B29864")
400 CALL CHAR(97,"0000001F1F
18181818")
410 CALL CHAR(98,"000000F8F8
181818")
420 CALL CHAR(99,"1818181F1F
")
430 CALL CHAR(100,"181818F8F
8")
440 CALL CHAR(101,"000000FFF
F")
450 CALL CHAR(102,"181818181
8181818")
460 CALL CHAR(103,"181818FFF
F181818")
470 CALL CHAR(104,"181818FFF
F")
480 CALL CHAR(105,"000000FFF
F181818")
490 CALL CHAR(106,"181818F8F
8181818")
500 CALL CHAR(107,"1818181F1
F181818")
510 CALL CHAR(108,"0")
520 CALL CHAR(112,"007E7E7E7

```

```

E7E7E00")
530 PRINT : : : : : : : :
540 PRINT TAB(10);"aeieiefeb
"
550 PRINT TAB(10);"f|f|f|f|f|f
"
560 PRINT TAB(10);"kegegegej
"
570 PRINT TAB(10);"f|f|f|f|f|f
"
580 PRINT TAB(10);"kegegegej
"
590 PRINT TAB(10);"f|f|f|f|f|f
"
600 PRINT TAB(10);"kegegegej
"
610 PRINT TAB(10);"f|f|f|f|f|f
"
620 PRINT TAB(10);"cehehehed
"
630 PRINT : : :
640 PRINT TAB(5);"0";TAB(11)
;"1";TAB(17);"2";TAB(23);"3"
650 PRINT : :
660 CALL HCHAR(22,8,42,18)
670 PRINT TAB(10);"TIME LEFT
"
680 DIM US(30),LR(30)
690 FOR Z=1 TO 16
700 FOR I=1 TO 6
710 READ A(I)
720 NEXT I
730 AB=INT(RND*6)+1
740 LR(Z)=A(AB)
750 NEXT Z
760 RANDOMIZE
770 AA=INT(RND*16)+1
780 US(XX)=AA
790 FOR CY=0 TO XX-1
800 IF US(CY)=US(XX) THEN 770
810 NEXT CY
820 CH=LR(AA)
830 CALL HCHAR(V,H,CH)

```

```

840 H=H+2
850 IF H<=19 THEN 890
860 H=13
870 IF H<>13 THEN 890
880 V=V+2
890 XX=XX+1
900 IF XX<17 THEN 770
910 CALL HCHAR(21,7,112)
920 FOR ST=1 TO 8
930 CALL COLOR(ST,16,1)
940 NEXT ST
950 CALL SCREEN(14)
960 CALL HCHAR(22,8,101,18)
970 TM=TM-1
980 AA=TM/10
990 IF AA=INT(AA) THEN 1120
1000 FOR DEL=1 TO 270
1010 NEXT DEL
1020 IF TM>0 THEN 970
1030 CALL COLOR(11,10,1)
1040 CALL SOUND(999,600,0)
1050 PRINT "        YOUR TIME
IS UP": :
1060 PRINT "PRESS E TO EXIT"
:"PRESS ANY KEY TO PLAY AGAI
N"
1070 CALL KEY(O,K,S)
1080 IF S=0 THEN 1070
1090 IF K=ASC("E") THEN 1570
1100 RESTORE
1110 GOTO 280
1120 D=D-1
1130 CALL HCHAR(21,D,32)
1140 GOTO 1000
1150 DATA 73,65,84,89,66,76
1160 DATA 65,77,79,81,66,74
1170 DATA 76,69,75,85,89,71
1180 DATA 87,71,76,82,85,73
1190 DATA 85,68,79,78,84,85
1200 DATA 82,79,70,73,66,88
1210 DATA 77,79,82,65,72,83
1220 DATA 82,65,76,69,67,83
1230 DATA 69,83,85,84,76,80

```

1240 DATA 69,72,89,70,69,73  
 1250 DATA 77,80,68,67,69,65  
 1260 DATA 65,73,79,65,84,67  
 1270 DATA 80,73,78,69,83,72  
 1280 DATA 86,69,90,65,68,78  
 1290 DATA 84,73,86,69,71,78  
 1300 DATA 83,78,87,68,79,69  
 1310 GOTO 280  
 1320 CALL CLEAR  
 1330 PRINT TAB(5);"BOGGLED I  
 NSTRUCTIONS"  
 1340 PRINT TAB(5);"\*\*\*\*\*  
 \*\*\*\*\*".  
 1350 PRINT "WHEN THE GAME ST  
 ARTS, EACH PLAYER SEARCHES  
 THE LETTERS ON THE BOARD FOR  
 WORDS OF 3"  
 1360 PRINT "OR MORE LETTERS.  
 WHEN A WORD IS FOUND BY A PL  
 AYER, HE/SHE WRITES IT DOWN."  
 1370 PRINT "THE WORDS ARE FO  
 RMED FROM ADJOINING LETTER  
 S. LETTERS MUST JOIN IN THE  
 PROPER"  
 1380 PRINT "SEQUENCE, HORIZO  
 NTALLY VERT-ICALLY OR DIAGON  
 ALLY, TO THE LEFT RIGHT, OR U  
 P AND DOWN"  
 1390 PRINT "NO LETTER MAY BE  
 USED MORE THAN ONCE IN EAC  
 H WORD."  
 1400 PRINT : : : "(PRESS ANY  
 KEY TO CONTINUE)"  
 1410 GOSUB 1590  
 1420 PRINT "ANY WORD, PROVID  
 ED THAT IT CAN BE FOUND IN  
 A STANDARD ENGLISH DICTIONA  
 RY MAY BE"  
 1430 PRINT "USED. WORDS WITH  
 IN WORDS ARE ALSO PERMISSIBLE  
 ;E.G.-SPARE spa,par,are,spar  
 ,pare."  
 1440 PRINT "WHEN THE TIMER R

UNS OUT, EVERYONE STOPS W  
 RITING. EACH PERSON THEN READ  
 S HIS/HER"  
 1450 PRINT "LIST ALOUD, AND  
 ANY WORD APPEARING ON MOR  
 E THAN ONE PLAYER'S LIST IS  
 CROSSED OFF"  
 1460 PRINT "ALL THE LISTS.":  
 "AFTER ALL THE PLAYERS HAVE  
 READ THEIR LISTS, EACH PERS  
 ON THEN"  
 1470 PRINT "SCORES THE REMAI  
 NING WORDS:": : "NO OF LETTER  
 S: 3 4 5 6 7 8+ -----  
 -----"  
 1480 PRINT "POINTS:  
 1 1 2 3 5 11": : "(PRESS ANY  
 KEY TO CONTINUE)"  
 1490 GOSUB 1590  
 1500 PRINT "THE WINNER IS TH  
 E PLAYER WHOSE SCORE IS T  
 HE HIGHEST OR WHO FIRST REA  
 CHES AN"  
 1510 PRINT "AGREED TARGET SC  
 ORE, 50, 100 POINTS ETC.": : "  
 MULTIPLE MEANINGS OF THE  
 SAME WORD DO NOT EARN EXTRA"  
 1520 PRINT "CREDIT.": : "THE  
 SAME WORD FOUND IN A DIFF  
 ERENT AREA DOES NOT COUN  
 T FOR EXTRA CREDIT": :  
 1530 PRINT "QU COUNTS AS TWO  
 LETTERS": : "SINGULAR AND PL  
 URAL FORMS EARN CREDIT PRO  
 VIDED THAT"  
 1540 PRINT "THAY ARE LISTED  
 AS SEPARATE WORDS.": : TAB(3)  
 ;"(PRESS ANY KEY TO START)"  
 1550 GOSUB 1590  
 1560 GOTO 280  
 1570 CALL CLEAR  
 1580 STOP  
 1590 CALL KEY(O,K,S)

1600 IF S=0 THEN 1590  
1610 CALL CLEAR

1620 RETURN

### GOLD LOTTO

By John Holland, modified by Sandra Nichelsen

This program picks your numbers for you, saves them on cassette or disk, gives you a printout of your numbers if you have a printer, and then checks to see if you have won anything when you enter the results on Saturday night. I took two of John Holland's programs that have been already published and modified them. Any errors therefore are mine. Don't complain to John.

```
100 REM *****
*
110 REM * LOTTO DRAW &
*
120 REM * RESULTS
*
130 REM * BY J HOLLAND
*
140 REM * MODIFIED BY
*
150 REM * SANDRA NICHELSEN
*
160 REM *****
*
170 CALL CLEAR :: CALL SCREE
N(11):: FOR COL=3 TO 14 :: C
ALL COLOR(COL,13,11):: NEXT
COL
190 DIM B(10,6),X$(10,6),N(4
0)
200 PRINT "DO YOU WANT TO:":
:TAB(7);"1. SELECT NUMBERS"
: :TAB(7);"2. CHECK RESULTS"
: :TAB(7);"3. QUIT": : : :
210 CALL KEY(0,K,S):: IF S=0
THEN 210
220 IF K=50 THEN 880
230 IF K=51 THEN 1160
240 IF K<>49 THEN 210
250 CALL CLEAR :: PRINT TAB(
8);"$$$$$$$$$$$$$":TAB(8);"
$ $"
260 PRINT TAB(8);"$ THE
$:TAB(8);"$ $
"
270 PRINT TAB(8);"$ GOLD LOT
TO $":TAB(8);"$ $
"
280 PRINT TAB(8);"$ DRAW
$:TAB(8);"$ $
"
290 PRINT TAB(8);"$$$$$$$$$
$$$": : : : : :
300 PRINT "HOW MANY GAMES WI
LL YOU PLAY": :
310 INPUT " <1 TO 10>
":S
320 IF (S<1)+(S>10)THEN 3276
7
330 CALL CLEAR :: RESTORE
340 CALL COLOR(10,2,11)
350 CALL COLOR(10,2,11):: FO
R I=128 TO 142 :: READ A$::
CALL CHAR(I,A$):: NEXT I
360 CALL CHAR(104,"COE070381
COE0703",105,"03070E1C3870EO
CO")
370 DATA 000000FFFF,30303030
30303030,0C0C0C0C0C0C0C0C,00
00003F3F303030,000000FCFC0C
```

```

COC
380 DATA 080808080808FFFF,08
08080808080808,000000FF,0808
08FF08080808,000000000000FFF
F
390 DATA 3030303030303F3F,0C
0C0C0C0C0C0CFCFC,000000FFFF080
808,3030303F30303030,0C0C0CF
C0C0C0C0C
400 FOR A=1 TO 5 :: CALL CLE
AR
410 PRINT TAB(6);"1 2 3 4
5 6 7": : : :TAB(6);"8
9 10 11 12 13 14"
420 PRINT : : :TAB(5);"15 16
17 18 19 20 21": : : :TAB(5
);"22 23 24 25 26 27 28"
430 PRINT : : :TAB(5);"29 30
31 32 33 34 35": : : :TAB(5
);"36 37 38 39 40"
440 CALL HCHAR(1,7,128,20)::
CALL VCHAR(2,6,129,23):: CA
LL VCHAR(2,27,130,23):: CALL
HCHAR(1,6,131)
450 CALL HCHAR(1,27,132):: C
ALL HCHAR(24,7,137,20):: CAL
L HCHAR(24,6,138):: CALL HCH
AR(24,27,139)
460 FOR R=5 TO 21 STEP 4 ::
CALL HCHAR(R,6,141):: CALL H
CHAR(R,27,142):: CALL HCHAR(
R,7,135,20):: NEXT R
470 FOR C=9 TO 24 STEP 3 ::
CALL HCHAR(24,C,133):: CALL
HCHAR(1,C,140):: CALL VCHAR(
2,C,134,22):: NEXT C
480 FOR R=5 TO 21 STEP 4 ::
FOR C=9 TO 24 STEP 3 :: CALL
HCHAR(R,C,136):: NEXT C ::
NEXT R
490 S$="GAME "&STR$(A):: FOR
I=1 TO LEN(S$):: CALL VCHAR
(I+2,3,ASC(SEG$(S$,I,1)))::

```

```

NEXT I
500 FOR G=1 TO 40 :: N(G)=0
:: NEXT G :: RANDOMIZE :: FO
R I=1 TO 6 :: FOR DEL=1 TO 5
0 :: NEXT DEL
510 P=INT(RND*40)+1 :: IF N(
P)=1 THEN 510
520 N(P)=1 :: R2=0 :: R1=P
530 IF R1<=7 THEN 570
540 FOR L=1 TO 6 :: R1=R1-7
:: R2=R2+1
550 IF R1<=7 THEN 570
560 NEXT L
570 R=R2*4+2 :: Z=P :: IF Z<
=7 THEN 610
580 FOR J=1 TO 6 :: Z=Z-7
590 IF Z<=7 THEN 610
600 NEXT J
610 C=Z*3+4 :: FOR T=220 TO
550 STEP 100 :: CALL SOUND(-
100,T,0,-3,10):: NEXT T
620 CALL HCHAR(R,C,104):: CA
LL HCHAR(R,C+1,105):: CALL H
CHAR(R+1,C,105):: CALL HCHAR
(R+1,C+1,104)
630 P$=STR$(P):: X$(A,I)=P$:
: IF LEN(P$)=1 THEN 640 ELSE
650
640 P$=" "&P$
650 FOR C=1 TO 2 :: CALL HCH
AR(I+2,C+29,ASC(SEG$(P$,C,1
))): : NEXT C :: NEXT I
660 FOR DEL=1 TO 500 :: NEXT
DEL :: NEXT A
670 CALL CLEAR :: PRINT "SAV
E NUMBERS Y/N?" :: CALL SOUN
D(50,110,0)
680 CALL KEY(0,K,ST):: IF ST
=0 THEN 680
690 IF K=89 THEN 780
700 IF K<>78 THEN 680
710 FOR DEL=1 TO 100 :: NEXT
DEL :: CALL SOUND(50,110,0)

```

```

:: PRINT "PRINT NUMBERS Y/N?
"
720 CALL KEY(0,K,ST):: IF ST
=0 THEN 720
730 IF K=89 THEN 820
740 IF K<>78 THEN 720
750 PRINT TAB(8);"GOOD LUCK"
: : : : :
760 FOR A=1 TO S :: PRINT "G
AME";A;" : " :: FOR I=1 TO 6 :
: PRINT VAL(X$(A,I));:: NEXT
I :: PRINT :: NEXT A
770 RESTORE :: GOTO 200
780 INPUT "ENTER DEVICE NAME
":DV$
790 OPEN #1:DV$,INTERNAL ,OU
TPUT
800 PRINT #1:S :: FOR A=1 TO
S :: FOR I=1 TO 6 :: B(A,I)
=VAL(X$(A,I)):: PRINT #1:B(A
,I):: NEXT I :: NEXT A
810 CLOSE #1 :: GOTO 710
820 INPUT "ENTER PRINTER NAM
E ":PRT$
830 FOR A=1 TO S :: FOR I=1
TO 6 :: B(A,I)=VAL(X$(A,I)):
: NEXT I :: NEXT A
840 IMAGE " GAME ##: ##
## ## ## ## ##"
850 OPEN #2:PRT$,OUTPUT
855 FOR A=1 TO S
860 PRINT #2,USING 840:A,B(A
,I),B(A,2),B(A,3),B(A,4),B(A
,5),B(A,6)
865 PRINT #2 :: NEXT A
870 CLOSE #2 :: GOTO 750
880 CALL CLEAR :: CALL CHARS
ET :: FOR COL=1 TO 14 :: CAL
L COLOR(COL,13,11):: NEXT CO
L :: INPUT "DEVICE NAME? ":D
V$
890 IF SEG$(DV$,1,1)="C" THE
N 930

```

```

900 IF LEN(DV$)>4 THEN 930
910 INPUT "ENTER FILENAME ":
FL$
920 DV$=DV$&"."&FL$
930 OPEN #1:DV$,INTERNAL ,IN
PUT
940 DATA 1st,2nd,3rd,4th,5th
,6th
950 FOR I=1 TO 15 :: READ AR
$::: NEXT I
960 INPUT #1:S
970 RESTORE 940
980 FOR C=1 TO S :: FOR A=1
TO 6 :: INPUT #1:B(C,A):: NE
XT A :: NEXT C :: CLOSE #1
990 FOR C=1 TO 6 :: READ Z$(
C):: NEXT C :: FOR C=1 TO 6
1000 DISPLAY AT(1,5):"GOLD L
OTTO RESULTS":TAB(5);"==== =
==== ====="
1010 DISPLAY AT(C+3,4):Z$(C)
;" NUMBER DRAWN?" :: ACCEPT
AT(C+3,22)BEEP:D(C):: NEXT C
:: DISPLAY AT(C+3,4):" SUPP
LEMENTARY #?"
1020 ACCEPT AT(C+3,22)BEEP:E
:: DISPLAY AT(12,1):RPT$("=
",28):: FOR C=1 TO S :: FOR
A=1 TO 6
1030 FOR F=1 TO 6 :: IF B(C,
A)=D(F)THEN G=G+1
1040 NEXT F :: IF H=1 THEN 1
060
1050 IF B(C,A)=E THEN B$="+S
UP" :: H=1 ELSE B$=""
1060 NEXT A :: IF G=6 THEN C
$="DIVISION 1" :: GOTO 1120
1070 IF G=5 AND B$="+SUP" TH
EN C$="DIVISION 2" :: GOTO 1
120
1080 IF G=5 THEN C$="DIVISIO
N 3" :: GOTO 1120
1090 IF G=4 THEN C$="DIVISIO

```

```

N 4" :: GOTO 1120
1100 IF G=3 AND B$="+SUP" TH
EN C$="DIVISION 5" :: GOTO 1
120
1110 C$="bad luck"
1120 DISPLAY AT(C+13,2):"GAM
E";C;G;B$;TAB(18);C$:: IF C$
="bad luck" THEN 1140
1130 FOR I=14 TO 5 STEP -1 :
: CALL SCREEN(1):: CALL A ::
NEXT I
1140 G=0 :: B$="" :: H=0 ::
NEXT C :: DISPLAY AT(23,1):R
PT$("=",28):: DISPLAY AT(24,
5):"PRESS ANY KEY TO END"
1150 CALL KEY(0,J,K):: IF K=
0 THEN 1150 ELSE CALL CLEAR
:: GOTO 190
1160 END
1170 SUB A :: FOR A=330 TO 7
30 STEP 100 :: CALL SOUND(-5
0,A,0):: NEXT A :: SUBEND

```

MICROWAVE TIME/WEIGHT CALCULATOR  
by Les Tutchings, TIBUG

This short program calculates the time required to cook your roast in a microwave oven. Extended basic is required, but the program could easily be rewritten in Basic. Note line 160. This is used to centre the word "Microweight" on the screen. This same technique can be used for any word or message even when you don't know its length as in the case of a string variable.

```

100 REM *****
110 REM * MICROWAVE TIME/ *
120 REM *WEIGHT CALCULATOR*
130 REM * BY LES TUTCHINGS*
140 REM *****
150 CALL CLEAR :: CALL SCREE
N(4):: FOR COL=1 TO 14 :: CA
LL COLOR(COL,2,1):: NEXT COL
160 X=LEN("MICROWEIGHT"):: D
ISPLAY AT(12,(28-X)/2)BEEP:"
MICROWEIGHT"
170 FOR DE=1 TO 400 :: NEXT
DE :: DISPLAY AT(1,3)ERASE A
LL:"ENTER DATA AS REQUESTED"
:: DISPLAY AT(4,2):" WEIGHT
FACTOR(GRAMS) 500"
180 DISPLAY AT(9,2):" TIME F
ACTOR(MINUTES)" :: DISPLAY A
T(14,2):"WEIGHT OF ITEM(GRAM
S)"
190 ACCEPT AT(4,24)VALIDATE(
NUMERIC)SIZE(-5)BEEP:WF :: A
CCEPT AT(9,24)VALIDATE(NUMER
IC)BEEP:TF :: ACCEPT AT(14,2
4)VALIDATE(NUMERIC)BEEP:WI
200 NT=(TF/WF)*WI :: DISPLAY
AT(20,1):"COOKING TIME IS";
NT;"MINUTES"
210 DISPLAY AT(24,1)BEEP:"AN
Y MORE DATA WANTED (Y/N)?"
220 CALL KEY(3,K,S):: IF S=0
THEN 220
230 IF K=89 THEN 190 :: IF K
<>78 THEN 220 ELSE CALL CLEA
R :: END

```



# HIGH RESOLUTION MONITOR

by Steve Wilkinson, TIUP

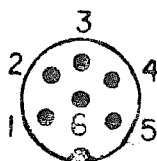
Having expanded my collection of 99/4A's to 2 and also having an old HI-resolution monitor ex an X-Ray Medical TV system, I decided to connect the black and white monitor to the "Y" signal, on the socket of J201 (J201 is the 6 pin socket that feeds the UHF or VHF modulator). The "Y" signal contains all the necessary sync. and luminance levels to run a monochrome monitor.

Modification completed, I was confronted with a good picture that would not stay synchronised, rolling vertically or horizontally with the slightest change in picture content.

On examination of my UHF modulator, I noted that the "Y" connection does not correspond with the circuit diagram. Reconnecting the plug J201 as per the following table produced a good stable picture.

J201

cct. diagram	* correct connection
1 = + 12v	1 = + 12v
2 = R-Y	2 = Y
3 = Audio	3 = R-Y
4 = Y	4 = B-Y
5 = B-Y	5 = Audio
6 = Ground	6 = Ground



Rear view of plug.

\* This is true for the PAL 99/4A's, but may not be valid for the NTSC version.

Plugs are easily obtainable from Atkins Carlyle. Plug type is DP6, costs approx \$1.86.

Remember that you will need an Audio connection also, if you are going to have the dulcet tones to remind you that you have just "BAD VALUE"ed again.

ELECTRONIC TYPEWRITER

By R.T.Tamashiro, Sydney News Digest

Here is an interesting program, made up of just CALL LOADs. It needs Extended Basic and Memory Expansion, as well as a printer to run. Change line 520 to suit your printer.

100 ! ELECTRONIC TYPEWRITER	9,82,32)
110 ! BY ROY T.TAMASHIRO	250 CALL LOAD(12432,42,32,69
120 ! XB/MEM EXP/PRINTER	,110,116,101,114,32,97,32,10
130 ! FCTN S CURSOR LEFT	8,105,110,101,32,111,102,32,
140 ! FCTN D CURSOR RIGHT	116,101,120,116,58,32)
150 ! FCTN 3 DELETE LINE	260 CALL LOAD(12456,40,84,12
160 ! FCTN 2 DELETE SPACE	1,112,101,32,69,78,68,32,119
170 ! FCTN 1 DELETE CHAR	,104,101,110,32,100,111,110,
180 CALL INIT :: CALL LOAD(8	101,46,41,32,2,224)
196,63,248):: CALL LOAD(1637	270 CALL LOAD(12480,131,224,
6,84,89,80,69,32,32,48,190)	2,1,240,129,216,1,131,212,21
190 CALL LOAD(12288,0,0,0,1,	6,1,140,2,6,193,216,1,140,2,
48,36,255,0,255,0,255,0,255,	2,1,244,135)
0,255,0,255,0,255,0,255,0,49	280 CALL LOAD(12504,216,1,14
,148)	0,2,6,193,216,1,140,2,2,224,
200 CALL LOAD(12312,255,0,25	48,0,4,192,2,1,128,0,4,32,32
5,0,255,0,255,0,2,48,13,0,80	,32)
,32,32,32,32,32,32,32,32,32,	290 CALL LOAD(12528,5,128,2,
32,32)	128,3,192,22,250,2,0,0,7,2,1
210 CALL LOAD(12336,32,32,32	,138,0,4,32,32,32,5,128,2,12
,32,32,32,32,32,32,32,32,32,	8)
32,32,32,32,32,32,32,32,32,3	300 CALL LOAD(12552,0,32,22,
2,32,32)	250,2,0,0,87,4,32,32,32,5,12
220 CALL LOAD(12360,32,32,32	8,2,128,0,112,22,250,2,0,0,4
,32,32,32,32,32,32,32,32,32,	7)
32,32,32,32,32,32,32,32,32,3	310 CALL LOAD(12576,4,193,4,
2,32,32)	194,208,98,48,120,2,33,96,0,
230 CALL LOAD(12384,32,32,32	4,32,32,32,5,128,5,130,2,130
,32,32,32,32,32,32,32,32,32,	,0,25)
32,32,32,32,32,32,32,32,32,0	320 CALL LOAD(12600,22,245,2
,255,0)	,1,32,0,2,2,0,1,216,129,48,3
240 CALL LOAD(12408,42,32,69	6,5,130,2,130,0,81,22,250,2,
,76,69,67,84,82,79,78,73,67,	0)
32,84,89,80,69,87,82,73,84,6	330 CALL LOAD(12624,1,184,4,

```

193,4,194,208,98,48,146,2,33
,96,0,4,32,32,32,5,130,5,128
,2,130)
340 CALL LOAD(12648,0,22,22,
245,2,0,1,224,4,194,208,98,4
8,168,2,33,96,0,4,32,32,32,5
,130)
350 CALL LOAD(12672,5,128,2,
130,0,22,22,245,2,0,2,48,200
,0,48,32,6,160,50,240,192,32
,48,32)
360 CALL LOAD(12696,2,1,126,
0,4,32,32,32,2,0,5,0,216,0,1
31,116,4,193,2,0,32,0,2,2)
370 CALL LOAD(12720,255,0,4,
32,32,28,144,32,131,124,19,2
6,144,160,131,117,19,243,2,3
,0,5,6,3)
380 CALL LOAD(12744,2,1,9,19
2,6,1,22,254,4,32,32,28,144,
32,131,124,19,11,144,160,131
,117,19,228)
390 CALL LOAD(12768,4,32,32,
28,192,195,22,239,152,32,48,
34,131,117,22,220,216,32,131
,117,48,34,4,193)
400 CALL LOAD(12792,208,96,1
31,117,2,129,8,0,22,15,4,194
,192,160,48,32,2,130,2,48,22
,2,4,96)
410 CALL LOAD(12816,49,144,6
,2,200,2,48,32,6,160,50,240,
4,96,49,144,2,129,9,0,22,2,4
,96)
420 CALL LOAD(12840,50,178,2
,129,13,0,22,2,4,96,50,206,2
,129,7,0,22,2,4,96,48,190,2,
129)
430 CALL LOAD(12864,4,0,22,2
0,192,32,48,32,2,32,253,210,
2,2,0,80,192,194,6,3,216,163
,48,36)
440 CALL LOAD(12888,48,36,6,
2,128,3,22,249,2,1,32,0,216,
193,48,36,4,96,49,144,2,129,
3,0)
450 CALL LOAD(12912,22,20,19
2,96,48,32,2,33,253,209,192,
193,2,0,0,80,5,131,216,99,48
,36,48,36)
460 CALL LOAD(12936,5,129,12
8,3,22,249,2,2,32,0,216,194,
48,36,4,96,49,144,2,129,32,0
,26,251)
470 CALL LOAD(12960,2,129,12
6,0,27,248,192,224,48,32,2,3
5,253,209,216,193,48,36,6,16
0,50,240,192,96)
480 CALL LOAD(12984,48,32,2,
129,2,127,17,2,4,96,49,144,5
,129,200,1,48,32,4,96,49,144
,120,32)
490 CALL LOAD(13008,131,124,
131,124,4,192,2,1,0,1,2,2,48
,36,4,32,32,16,120,32,131,12
4,131,124)
500 CALL LOAD(13032,2,224,13
1,224,4,96,0,112,2,0,2,48,2,
2,0,1,4,193,208,98,48,36,2,3
3)
510 CALL LOAD(13056,96,0,4,3
2,32,32,5,128,5,130,2,130,0,
81,22,245,4,91)
520 OPEN #1:"PI0.LF"
530 CALL LINK("TYPE",W$):: I
F SEG$(W$,1,3)="END" THEN CL
OSE #1 :: CALL PEEK(2,1,B)::
CALL LOAD(-31804,A,B)
540 PRINT #1:W$:: GOTO 530

```

## PROGRAMMING HINTS

by John Luck, Sydney News Digest

Example #1 - Tired of zapped spacecraft just being a coloured spot on your screen?. The craft is moving across the screen left to right. Now press the 'H' key. The craft turns yellow, then divides into 4 sections which move away at random directions and speeds. The sections then rotate 360 degrees in a clockwise direction. Needs Extended Basic.

Example #2 - Did you ever need a Nuclear Explosion and didn't have one on hand? Well, now your problems are over. The following sub-routine provides one in DOUBLE-SIZED sprites and is about 8 seconds from start to finish. So have a go! If you have to ZAPP 'EM, do it properly.

### EXAMPLE #1

```

100 REM *****
110 REM *   ZAPPED   *
120 REM *   SPACECRAFT *
130 REM *BY J.LUCK TISHUG*
140 REM *****
150 CALL CLEAR :: CALL SCREEN(2):: CALL CHAR(96,"182442D5AB817F00")
160 CALL CHAR(99,"102040D",100,"8040205",101,"00000000A0807",102,"00000000B010E00")
170 CALL CHAR(103,"0000000010A0408",104,"00000000A040201",105,"1020508",106,"000000005040502")
180 CALL CHAR(107,"007080D",108,"0C0A020A",109,"3040503",110,"000E0105",111,"00000000020A020C")
190 CALL CHAR(112,"080C0209",113,"00000000B020408",114,"000000009040301")
200 CALL SPRITE(#1,96,5,96,124,0,8)
210 CALL KEY(0,K,S):: IF S=0 THEN 210
220 IF K=72 THEN 230 ELSE 210
230 CALL MOTION(#1,0,0):: CALL POSITION(#1,A,B)
240 CALL SPRITE(#3,99,11,A,B,#4,100,11,A,B,#5,101,11,A,B,#6,102,11,A,B)
250 CALL DELSPRITE(#1):: RANDOMIZE :: CALL SOUND(700,-7,0)
260 FOR Q=3 TO 6 :: Z=INT(RND*15)-7 :: S=INT(RND*15)-7 :: CALL MOTION(#Q,Z,S):: NEXT Q
270 FOR D=1 TO 150 :: NEXT D :: CALL PATTERN(#4,103,#6,106,#5,109,#3,112):: FOR D=1 TO 150 :: NEXT D
280 CALL PATTERN(#4,104,#6,107,#5,110,#3,113):: FOR D=1 TO 150 :: NEXT D :: CALL PATTERN(#4,105,#6,108,#5,111,#3,114)
290 FOR D=1 TO 150 :: NEXT D :: CALL PATTERN(#4,100,#6,102,#5,101,#3,99):: FOR D=1 TO 150 :: NEXT D
300 FOR Q=3 TO 6 :: CALL DELSPRITE(#Q):: NEXT Q :: GOTO 200

```

EXAMPLE #2

```

100 REM *****
110 REM *NUCLEAR EXPLOSION*
120 REM *BY J. LUCK TISHUG*
130 REM *****
140 CALL CLEAR
150 CALL CHAR(96,"0000000000
0000008004000B3FFF7F1F000000
0000000000100200E4FEFFFEF8")
160 CALL CHAR(100,"000000000
123010307830307071F1F0700000
01000C0E0C0C280C8E0F0FCF8F0"
)
170 CALL CHAR(108,"810103010
123010307830307071F1F078080C
090C2C0E0C0C280C8E0F0FCF8F0"
)
180 CALL CHAR(104,"03377FFFF
FFF5F0F002200002000004060F8F
EFFFEFFFCF00020080020040000"
)
190 CALL CHAR(112,"000000000
000000000000040E8BFFF3D00000
000000000000000000133FFF6FC"
)
200 CALL CHAR(116,"000041030
F070B01000000000000000000008
0A0F9F0F0C0000000000000000")
210 CALL SCREEN(4):: FOR N=2
319 TO 880 STEP -20 :: CALL

```

```

SOUND(-99,N,2):: NEXT N
220 CALL SOUND(500,-7,0):: C
ALL MAGNIFY(4):: CALL SPRITE
(#3,112,2,96,124):: CALL DEL
(150)
230 CALL PATTERN(#3,96):: CA
LL SOUND(500,-7,5):: CALL DE
L(100)
240 CALL COLOR(#3,7):: CALL
DEL(75):: CALL PATTERN(#3,10
0):: CALL DEL(200):: CALL SO
UND(500,-7,10)
250 CALL SPRITE(#2,116,2,96,
124):: CALL DEL(200):: CALL
PATTERN(#2,104,#3,108):: CAL
L MOTION(#2,-1,0):: CALL DEL
(750)
260 CALL PATTERN(#3,100):: C
ALL DEL(200):: CALL SPRITE(#
1,96,2,96,124):: CALL DEL(20
0):: CALL DELSPRITE(#1)
270 CALL PATTERN(#3,96):: CA
LL DEL(100):: CALL PATTERN(#
3,112):: CALL DEL(200):: CAL
L DELSPRITE(ALL):: CALL DEL(
500):: GOTO 210
280 SUB DEL(A):: FOR I=1 TO
A :: NEXT I :: SUBEND

```

DOODLEBUG from Sydney News Digest.

```

100 CALL CLEAR
110 RANDOMIZE
120 CALL CHAR(96,"81423C1818
3C4281")
130 DISPLAY AT(2,10):"DOODLE
BUG"
140 CALL SPRITE(#1,96,2,90,1

```

```

25)
150 A=INT(RND*20)-INT(RND*20
)
160 B=INT(RND*20)-INT(RND*20
)
170 CALL MOTION(#1,A,B)
180 GOTO 150

```

LIMERICKS from Sydney News Digest

Here is a program that will give you 32,760 different limericks. If you get sick of these, it would be simple enough to change the data statements to give you another 32,000. If you have a Speech Synthesizer and Terminal Emulator II it will also speak the limericks. You can also print any on a printer if you want. See how you go.

```

100 REM
110 REM LIMERICKS
120 REM BY TONY FALCO
130 REM 1241 MAIN ST, WORCE
STER, MA. 01603
140 CALL CLEAR
150 PRINT TAB(11);"LIMERICKS
": : : : " 32,760 POSSIBLE
POEMS": : : : : :
160 INPUT "DO YOU HAVE SPEEC
H AND TERMINAL EMULATOR
II(Y/N) ":X$
170 CALL CLEAR
180 IF SEG$(X$,1,1)<>"Y"THEN
200
190 OPEN #1:"SPEECH",OUTPUT
200 RANDOMIZE
210 DIM W$(7,4)
220 CALL SCREEN(4)
230 RESTORE 550
240 IF INT(RND*2)-1 THEN 260
250 RESTORE 640
260 READ P$,Q$,R$,D$,F$
270 FOR T=1 TO 7
280 FOR X=1 TO 4
290 READ W$(T,X)
300 NEXT X
310 NEXT T
320 S$=W$(1,INT(RND*4)+1)
330 A$=P$&S$
340 B$=Q$&W$(2,INT(RND*4)+1)
350 B$=B$&" "&R$&W$(3,INT(RN
D*4)+1)
360 C$=" "&W$(4,INT(RND*4
)+1)
370 E$=W$(5,INT(RND*4)+1)
380 E$=E$&" "&W$(6,INT(RND*4
)+1)
390 G$=W$(7,INT(RND*4)+1)
400 CALL CLEAR
410 PRINT P$: : " ";S$;:
:B$: :C$: :D$: :E$: :F$: : "
";G$: : : :
420 CALL SCREEN(11)
430 IF SEG$(X$,1,1)<>"Y"THEN
450
440 PRINT #1:A$: :B$: :C$: :
D$: :E$: :F$: :G$
450 INPUT "PRINT IT? Y/N ":Y
N$
460 IF SEG$(YN$,1,1)<>"Y"THE
N 500
470 OPEN #2:"PIO.LF",OUTPUT
480 PRINT #2:P$: " ";S$;:
B$:C$:D$:E$:F$: " ";G$: :
: :
490 CLOSE #2
500 INPUT "ANOTHER ONE? Y/N
":YN$
510 IF SEG$(YN$,1,1)="Y"THEN
220
520 IF SEG$(X$,1,1)<>"Y"THEN
540
530 CLOSE #1
540 STOP
550 DATA "THERE WAS A YOUNG
MAN FROM ","WHO ","HIS "
560 DATA "ONE NIGHT AFTER DA
RK","AND HE NEVER FOUND OUT
"

```

570 DATA TASHKENT,TRENT,KENT  
 ,GHENT  
 580 DATA WRAPPED UP,COVERED,  
 PAINTED,FASTENED  
 590 DATA HEAD,HAND,DOG,FOOT  
 600 DATA IN A TENT,WITH CEME  
 NT,WITH SOME SCENT,THAT WAS  
 BENT  
 610 DATA IT RAN OFF,IT GLOWE  
 D,IT BLEW UP,IT TURNED BLUE  
 620 DATA IN THE PARK,LIKE A  
 QUARK,FOR A LARK,WITH A BARK  
 630 DATA WHERE IT WENT,ITS I  
 NTENT,WHY IT WENT,WHAT IT ME  
 ANT  
 640 DATA "THERE WAS A YOUNG  
 GIRL ","WHO ","A "

650 DATA "ONE NIGHT AFTER DA  
 RK ","AND SHE WOKE UP IN A "  
 660 DATA FROM KOWLOON,FROM R  
 ANGOON,FROM THE MOON,BORN IN  
 JUNE  
 670 DATA DATED,STAYED WITH,F  
 OLOWED,DREAMED OF  
 680 DATA WRINKLED,HAIRY,SILL  
 Y,CHUBBY  
 690 DATA BUFFOON,BABOON,OLD  
 GOON,OLD PRUNE  
 700 DATA THEY KISSED,THEY DA  
 NCED,THEY WALKED,THEY SANG  
 710 DATA IN THE PARK,FOR A L  
 ARK,ON THE MARK,BY A SPARK  
 720 DATA MONSOON,BALLOON,SPO  
 ON,SALOON

---

#### WITH OR WITHOUT SPEECH SYNTHESIZER

by Harold Hoyt, St Louis 99er's "BRIDGE" via ATICC

Programs are greatly enhanced by adding speech. A "CALL SAY" statement is added after every displayed message in the program. The program should function well with or without a speech synthesizer attached. When the speech unit is not attached, encountering a CALL SAY statement causes long program delays. If you insert REM before each line with a CALL SAY statement, the information is preserved for later use without slowing program execution speed. An even better way is to add logic to the program and steer it past CALL SAY statements if your synthesizer is disconnected. Consider the following program segment:

```
100 CALL PEEK(-28672,SP)
110 PRINT "HELLO, HOW ARE YOU?"
120 IF SP=0 THEN 140
130 CALL SAY("HELLO HOW ARE YOU")
```

Line 100 returns 96 if the speech synthesizer is attached and 0 if it is not. The PEEK function will operate since Extended Basic, Minmemory, and other modules that allow speech also allow CALL PEEK. Line 120 logic bypasses the CALL SAY statement. A similar bypass can be added to every CALL SAY statement in the program.

## SERIES AND THE COMPUTER

by Garry Christensen

Prior to the advent of calculators, books of tables were used to find values for sine, cosine, tangent, etc. Those who remember the size of these tables will quickly realize that all the data could not be stored in a calculator or a home computer. How, then, is the value for say sine of 23.6354 calculated? The answer is a mathematical tool called a series.

An infinite series is a sum of terms that continue to infinity. In the series that we will be considering, each successive term is smaller than the term preceding it. As each term is added, the result approaches the final answer.

For example:

$$\sum 1/2^{(n-1)} = 1 + 1/2 + 1/4 + 1/8 + 1/16 \dots$$

This series continues to infinity with each successive term approaching zero. The sum of these terms converges on the value two, that is, it approaches two but never gets there.

The symbol  $\sum$  means "add all the terms". Throughout the text I have included both the general term for use in programming and the expanded series to demonstrate the appearance of the final product.

Another concept needed at this stage is that of factorial numbers. The value for factorial three (3!) is  $1*2*3=6$ . Factorial 5 (5!) is  $1*2*3*4*5=120$ . Factorial ten (10!) is  $1*2*3*4*5*6*7*8*9 = 3,628,800$

One method of calculating trigonometric functions is by using a special set of series called "MaLauren's series".

The value for Sine is given in general terms by  $\sum (-1)^{n-1} * (X^{2n-1}) / (2n-1)!$

$$\text{or } \sin X = X - X^3/3! + X^5/5! - X^7/7! + X^9/9! - \dots$$

..... where X is in radians. (1 Radian = 180/PI Degrees)

Minus one raised to an even power is positive while minus one raised to an odd power is negative.  $(-1)^{n-1}$  merely ensures alternating signs.

The following is the programme to calculate Sine. You will notice that the general term is used. Following it is a listing of the results for an input of 1.570798 (radians). "LOOP" is the number of the loop(N), "RESULT" is the progressive answer and "ADD" is the amount added to produce the new result.



```

100 INPUT X
110 RESULT=0
120 FOR N=1 TO 10
130 FACTORIAL=1
140 FOR F=1 TO 2*N-1
150 FACTORIAL=FACTORIAL*F
160 NEXT F
170 ADD=(-1)^(N-1)*(X^(2*N-1)/FACTORIAL)
180 RESULT=RESULT+ADD
190 PRINT N,RESULT,ADD
200 NEXT N

```

LOOP	RESULT	ADD
1	1.570798	1.570798
2	.9248318383	-.6459661617
3	1.004524889	.0796930507
4	.9998430999	-.004681789
5	1.000003543	.0001604427
6	.9999999437	-3.59889E-06
7	1.000000001	5.69225E-08
8	1.	-6.68814E-10
9	1.	6.06705E-12
10	1.	-4.37715E-14

As you will note, the answer converges on one very quickly. 1.570798 radians is, in fact 90 degrees and the Sin 90 deg. is exactly 1.

(It is interesting to note the number handling capability of the TI-99/4A computer. Exponential values range from -127 to 128, compared to -40 to 40 for the Apple, for example. Try the above programme with an input of 200. The numbers involved grow very quickly. The result will climb very high then eventually fall but even the TI-99/4A can't handle the numbers involved.)

The value for Cosine can be calculated by  $\sum (-1)^{n-1} * (X^{2n-2}) / (2n-2)!$

$$\text{or } \cos X = 1 - X^2/2! + X^4/4! - X^6/6! + X^8/8! - \dots$$

For Cosine the programme is identical except that line 170 uses the new general term and the value for factorial is changed in line 140:

```
140 FOR F=1 TO 2N-2
```

$$170 \text{ ADD} = (-1)^{(N-1)} * (X^{2N-2}) / \text{FACTORIAL}$$

The values for Tan, Cot, Sec and Cosec can then be obtained by using Sin/Cos, Cos/Sin, 1/Cos and 1/Sin respectively. (Please note that Cot, Sec and Cosec are not built-in functions in the TI-99/4A)

PI (3.1415..) is stored in memory but have you ever wondered how it was calculated in the first place. In fact, a series is used again.

$$1/4 * \text{PI} = \sum (-1)^{n-1} * 1/(2n-1) \text{ or } 1/4 * \text{PI} = (1 - 1/3 + 1/5 - 1/7 + 1/9 - \dots)$$

100 ANSWER=0

110 FOR N=1 TO 500

120 ADD=(-1)^(N-1)\*1/(2\*N-1)

130 ANSWER=ANSWER+ADD

140 PRINT ANSWER\*4

150 NEXT N

assemble the code

This particular series converges very slowly. After 1.5 million terms the answer is only accurate to 6 decimal places. A faster method is given by:

$$\text{PI} = 16 * \text{ArcTan}(1/5) - 4 * \text{ArcTan}(1/239) \text{ where}$$

$$\text{ArcTan } X = \sum (-1)^{n-1} * (X^{2n-1}) / (2n-1)$$

$$\text{or ArcTan } X = X - (X^3)/3 + (X^5)/5 - (X^7)/7 + (X^9)/9 - \dots$$

By taking this series to 6 terms the value for PI is accurate to 7 decimal places. (Watch future editions for PI accurate to several thousand decimal places.)

Hyperbolic functions are also not built-in functions in the TI-99/4a (see page 202 of extended basic manual for their definitions) but armed with series we can create our own. Sinh (Hyperbolic sine, pronounced "shine") is given by

$$\sum (X^{2n-1}) / (2n-1)! \text{ or Sinh } X = X + (X^3)/3! + (X^5)/5! + (X^7)/7! + \dots$$

The value for Cosh (Hyperbolic Cosine, pronounced as it is spelt) is given by

$$\sum (X^{2n-2}) / (2n-2)! \text{ or Cosh } X = 1 + (X^2)/2! + (X^4)/4! + (X^6)/6! + \dots$$

For Sinh, use the Sine programme with the new general term,

```
170 ADD=X^(2*N-1)/FACTORIAL
```

and for Cosh, use the Cosine programme with

```
170 ADD=X^(2*N-2)/FACTORIAL
```

Values for Tanh, Coth, Sech and Cosech can be calculated as with their trigonometric counterparts. Alternatively, Sinh and Cosh can be found by the use of their definitive equations. These are :

$$\text{Sinh } X = (e^X - e^{-X})/2$$

$$\text{Cosh } X = (e^X + e^{-X})/2$$

where  $e^X$  is given by  $\sum (X^n)/(n!)$

$$\text{Thus } e^X = 1 + X + (X^2)/2! + (X^3)/3! + (X^4)/4! + \dots$$

$$\text{and } e^{-X} = 1 - X + (X^2)/2! - (X^3)/3! + (X^4)/4! - \dots$$

Incidentally, when  $X=1$  then  $e^X = e$  so the value  $e$  is given by  $\sum 1/(n!)$

$$e = 1 + 1 + 1/2! + 1/3! + 1/4! + \dots = 2.718..$$

The value for  $e$ , however, is a constant and is best stored in memory rather than calculated each time.

I should point out that the series discussed may not be the most efficient series to the job. Before computers, the work on series had been limited as all calculations had to be done manually but since then a considerable amount of research using computers to do the hard work has produced many new applications for series.

One other problem has surfaced with the use of repeated calculations as used in series. This is a problem of "round-off error". Answers given by the TI-99/4A are rounded off to fourteen decimal places even though only nine are displayed. After repeated calculations the accumulation of round-off error can drastically alter the answer. Using the programme for  $\pi$  to 1.5 million terms, the answer climbed above the real value of  $\pi$  in the seventh decimal place even though the error was only in the fourteenth. Look once again at table of results for the sine of 90 degrees. The answer of 1 cannot be absolutely correct because very small numbers are being added and subtracted but the answer has been rounded off to 1. When using a series that does not converge quickly beware of round-off error when approaching accuracy to seven or eight decimal places.. The faster a series converges the less important is the round-off error.

WORD SQUARE    BUG-BYTES/4

E E O U R A U R E V O L P S P	BROLGA	KITE
T M H L L E E T A K C O C T I	COCKATEEL	KOOKABURRA
I U A E R A L C H I A I S T G	CRAKE	LORIKEET
K H W R E H R A O A F O L H E	CROW	LYREBIRD
R D K T R O M O L R L D L C O	CUCKOO	MARTIN
E O E E W U K B O L R A U N N	DOTTEREL	OWL
H T L P O C B G R I E C G I F	EAGLE	PELICAN
S T W O U S M A B O R S A F N	EMU	PETREL
I E R C R O W E K A L G O G A	FINCH	PIGEON
F R E R U I R I K O R G E R C	FROGMOUTH	PLOVER
G E N T A Y K E F E O T A N I	GALAH	ROSELLA
N L H U L M B E B T E K R O L	GREBE	STINT
I N I T R A M E E F N A W S E	GULL	SWAN
K T N I T S B E A T U T I F P	HAWK	SWIFT
U L W O L B I R E L G A E D S	KINGFISHER	WREN

HOW DO MEMBERS USE THEIR COMPUTERS?

Here is the first letter in this series. It comes from Cheryl Bailey, who is one of the club's foundation members. Thank you for your letter, Cheryl. I would like to hear from other club members about the use to which they put their computers. Don't forget their is a prize of some free club software for the best letter received by 15th July. So get your letters in.

Dear Sandra,

The incentive of a prize induces me to write a short note about my experiences and frustrations of being an android. After 3 years, the initial enthusiasm has waned and with my free time limited, I find my computer has become a trusted workhorse which very seldom lets me down and doesn't answer back. As I inspected many homes in view of purchase, the real estate agents found it hard to believe that a computer could be given such preferential treatment as its own room, cosy but well ventilated, with a power supply as first priority.

TI-WRITER receives the most wear and tear and definitely gets my vote as the most useful program I use. I have written many articles for TIBUG, stored recipes, produced orchid catalogues and written job applications. My grocery list is done weekly and dossiers on all m

ancestors are recorded and updated using TI-WRITER. School projects become much easier to do with TI-WRITER's terrific editing facilities. Labels for all sorts of purposes are easily reproduced. Sport Training Sheets are stored and reprinted each season.

The other purposes our computer is used for is education, and PLATO receives top billing. As a tutorial and drill program, it is excellent, but limited by the expense of the packages. Here I feel modules are the best value for money as children can operate the computer easier.

Games serve an excellent recreational pastime and EXTENDED BASIC is a must because of our little friends "sprites". Some games modules are good, but I think I prefer many of the disk based programs written by amateurs.

As far as programming goes, I am still struggling for success. I keep saying one day I will have time to learn, as BASIC, EXTENDED BASIC, LOGO, FORTH and ASSY. LANGUAGE are all awaiting my urgent attention, a hoarder I must be.

When I feel like getting away from it all, I fire up my MUSIC and GRAPHICS disks and lap up the sound and colour capabilities of the TI. However, the lure of the chase always attracts me in. Every new program listing in any books or magazines, starts my fingers tapping frantically as I wait for the magical moment when it's ready to RUN.

Finally, the greatest benefit of being a TI owner has been the introduction to the many enthusiasts by member of the user group.

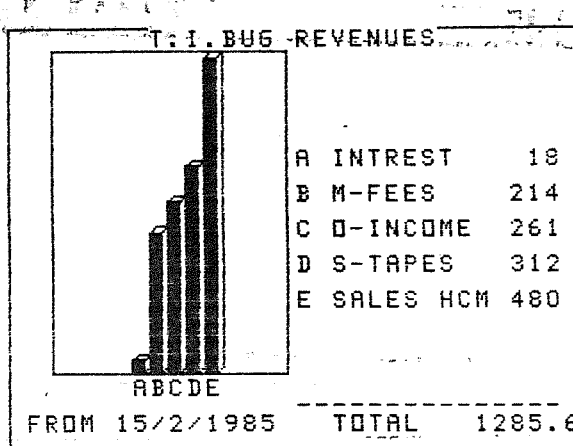
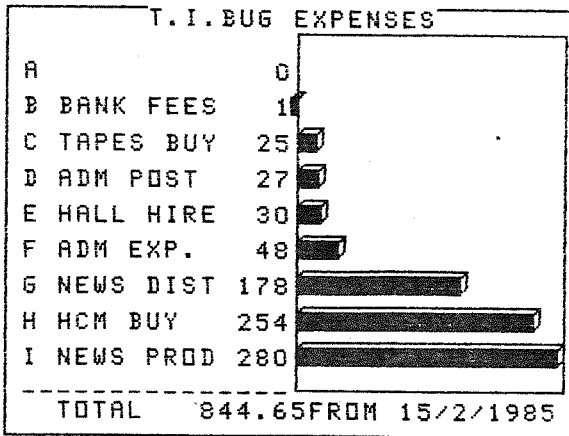
Yours sincerely,

Cheryl Bailey

---

#### VIATEL

At the next meeting, we hope to have a demonstration of Telecom's new system Viatel. This system will allow for electronic mail, purchasing from shops etc all without leaving your computer keyboard. It should be a really interesting meeting, a glance into the future! Don't forget to come along.



DRAWN BY CHART MAKER  
 AND DRAW 'N PLOT  
 FROM LINDLEY & ASSOC.

T. I. BRISBANE USER GROUP  
 INCOME STATEMENT  
 FOR THE PERIOD TO 5/31/85

\*\*\* REVENUES \*\*\*

	MON TO DATE	YEAR TO DATE
STOCK-CLOSING	00.00	317.00
NEWSCRIPES (SUBS)	00.00	74.00
SALES COMMERCIAL	00.00	480.00
SALES TAPES & DISKS	00.00	312.00
DONATIONS ACCEPT.	00.00	00.00
INTREST RECEIVED	00.00	18.68
OTHER INCOME	00.00	261.20
	-----	-----
TOTAL REVENUES	00.00	2,162.28

\*\*\* EXPENSES \*\*\*

STOCK-OPENING	00.00	317.00
PURCHASES-COMMERCIAL	00.00	254.77
PURCHASES-TAPES&DISKS	00.00	25.00
DONATIONS GIVEN	00.00	00.00
GEN & ADM EXPENSES	00.00	43.72
BANKCARD/VISA CHARGES	00.00	00.00
ACCOUNTANCY	00.00	00.00
ADMIN, STAMPS & STATIONERY	00.00	27.80
ADVERTISING	00.00	00.00
CLEANING	00.00	00.00
HALL HIRE	00.00	30.00
BANK FEES	00.00	00.00
INSURANCE	00.00	00.00
OTHER EXPENSES	00.00	00.00
NEWSLETTER PRODUCTION	00.00	290.38
NEWSLETTER DISTRIBUTION	00.00	178.61
TRAVEL & ENTER	00.00	00.00
REPAIRS & MAINTENANCE	00.00	00.00
POSTAGE OF SALES	00.00	(117.45)
	-----	-----
TOTAL EXPENSES	00.00	1,646.52
	=====	=====
NET INCOME	00.00	454.34

\*\*\* ASSETS \*\*\*

CASH IN BANKS	568.98	
CASH ON HAND	48.00	
I. B. B.	500.00	
DEBTORS	40.00	
LESS: ALLOWANCE FOR BAD DEBT	00.00	40.00
	-----	
EQUIPMENT & LIBRARY	2,890.00	
LESS: EQUIP ACCUM DEPRECIATION	00.00	2,890.00
	-----	
STOCK ON HAND	817.00	
TOTAL ASSETS		\$ 4,863.98
		-----

\*\*\* LIABILITIES \*\*\*

CREDITORS	00.00	
PREPAYMENTS M'SHIP	18.00	
PREPAYMENTS SALES	00.00	
LOANS	00.00	
TOTAL LIABILITIES		\$ 18.00
		-----

\*\*\* OWNER'S EQUITY \*\*\*

CAPITAL	4,389.62	
ACCUMULATED PROFITS	00.00	
* INTERIM EARNINGS *	454.34	
TOTAL OWNER'S EQUITY		\$ 4,845.96
		-----



## For Sale — Wanted

FOR SALE :- PERIPHERAL EXPANSION BOX complete with DISK CONTROLLER CARD, single sided single density DISK DRIVE, 32K MEMORY EXPANSION card, and the following software, EDITOR ASSEMBLER, TI WRITER, TI LOGO II, 10 DISKS with programs and 1 Disk Box. \$900

Phone 343 5164

FOR SALE :- PERIPHERAL EXPANSION BOX complete with DISK CONTROLLER CARD, TI DISK DRIVE and 32K MEMORY EXPANSION card. Has had little use \$700 ono. Phone 284 0967

### CLUB SOFTWARE

This month, we have two new releases, Extended Basic 5 on tape, and 99er #3 for disk only.

If you have purchased the last Issues of HCM you will have noticed the program "The Organizer" and probably given up on typing it in when you saw how much of it there was. Well, we have had it typed up to save you the trouble as well as the Quiz Construction Set and Personal Loan Calculator from Vol 4 No 5. You will still need your HCM however, as the instructions are quite long and involved. These programs are on 99er #3 for \$8

Extended Basic #5 is on cassette tape for \$5 and includes the following programs:

Side A: Binary, Bulls & Cows, Doodling, Glad/Mom, Mastermind, Meteor, Qld Towns, and Starguard.

Side B: Challenger, Jawbreaker, S\*A\*M, Tablut, Tease Time, World Money, and Zippy #2

These two items will be on sale at the next meeting, or by mail. Include \$1 for postage. Don't forget we take Bankcard and Visa.

### TI FORTH

Col Christensen is trying to put together a disk full of FORTH programs. If you have any programs that you have written in FORTH, even if they are only small, give Col a ring on 284 7783.

### ASSEMBLY LANGUAGE

If you are into Assembly then this is for you! Garry Christensen is trying to organise an ASSEMBLER workshop. If you are interested in attending, or helping with the organization, let Garry know as soon as possible. Phone 284 1841