



MIKE GODDARD COMPUTER SUPPORT

NEW AND USED COMPUTER EQUIPMENT BOUGHT AND SOLD

TI-99/4A SPECIALIST

SPARES

KITS

REPAIRS

COMPONENTS

BOOKS

PRINTER CABLES

RESISTORS

BATTERIES

DISK DRIVES

POWER SUPPLIES

POWER CABLES

PRINTERS

MONITORS

CAPACITORS

DISK DRIVE CABLES

RIBBON CABLE

FOR LATEST LIST CONTACT:

M.G.C.S. "SARNIA", CEMETERY ROAD, RHOS, WREXHAM
CLWYD, LL14 2BY

Tel: (0978)843547.

NBX: 022212529

TI*MES

TI99/4A USER'S GROUP (U.K.) CONTACTS

Chairman & Membership(Records): Trevor Stevens. Tel.0623 793077

249 Southwell Rd. East, Rainworth, Notts. NG21 0BN

Vice Chairman & Programming: Mark Wills.

12 "Rosehill" , Betton St., Shrewsbury, Shropshire. SY3 7YN

General Secretary: Jim Ballinger. Tel.0332 772612

5, Offerton Ave., Derby. DE3 8DU

Publicity Officer & Membership(Contacts): Philip Trotter. Tel.0642 817356

80, Martonburn Rd., Grovehill, Middlesborough. TS4 2TH

Treasurer: Alan Rotherford. Tel.0625 524642. Bus.061 2231301

13, The Circuit, Wilmslow, Cheshire. SK9 6DA

TI*MES Editor & Distribution: Alan Bailey. Tel.081 508 1053

14, Shelley Grove, Loughton, Essex. IG10 1BY

Hardware & Projects: Mike Goddard. Tel.0978 843547

"Sarnia", Cemetery Rd., Rhos, Wrexham, Clwd. LL14 2BY

Librarians: Cassette; Nicky Goddard. Tel.,etc. as above.

Disc & Journal Exchange: Stephen Shaw.

10 Alstone Rd., Stockport, Cheshire. SK4 5AH

Modules: Edward Shaw. Tel.0538 360382(5pm to 8pm)

Crow Holt Farm, Basford, Leek, Staffs. ST13 7DU

Publications : Mike Curtis. Tel.0209 219051

21 Treliske Rd., Roseland Gdns., Redruth, Cornwall.TR15 1QE

MAGAZINE CONTENTS

IFC Editorial, Disclaimer, Next Copy Date.

p1 Module Librarian's Report by Edward Shaw.

2 Letter from Robert Wilson.

3 News From TI by Mark Wills.

4 Drivel no.1 by Mark Wills.

5 Software Advert. by Mark Wills.

9 Multilength Arithmetic by Walter Allum.

Moving Characters on the Screen by Barrie Clark.

12 Cassette Reviews by Nicky Goddard.

13 On the Hardware Front by Mike Goddard.

14 Wrexham & Chester Workshop Report by Mike Goddard.

15 Bulletin Board by Trevor Stevens.

16 Rambles by Stephen Shaw.

17 Disk Library Report by Stephen Shaw.

18 XB by Jim Swedlow.(sjs)

27 TI Bits by Jim Swedlow.(sjs)

31 All Micro Show.(Advt.)

32 Machine Code Tutorial Part 5 by Mack McCormick.(sjs)

36 Fractal by Stephen Shaw.

37 Mandelbrot Plot by Stephen Shaw.

38 The Golden Ratio by Stephen Shaw.

39 Polynomial Equations by Stephen Shaw.

41 Fast Typing & Adding Fractions by Stephen Shaw.

42 Primes by Stephen Shaw.

Continued overleaf.

EDITORIAL

It is with great regret that I have to tell you that Peter Walker feels unable to continue to undertake the very onerous duties of Membership Secretary. We are all in his debt for the many years of work which he has done so well. Thank you Peter! We hope that you will continue to contribute to our magazine and in other ways. Unfortunately the Cuffley Show must now be cancelled. At least for the time being, Trevor Stevens and Phil Trotter will split the Membership Secretary's duties between them, as indicated in the contacts list. You will also see that the position of Vice Chairman has been accepted by Mark Wills, as Stephen Shaw wishes to concentrate on the Disc Library, Journal Exchange and his very considerable contributions to this Journal! As you will see, notice inside the back cover, the 1992 AGM will take place in Derby at a new venue for us! An addition to the Constitution of the group will be proposed at that meeting, designed to regulate the resolution of disputes between members and between members and suppliers. In the interim any advice for or against purchases from any supplier will be referred to the Committee before publishing. If disapprobation is to depend on an evidential process then in fairness to our members, approbation should be treated similarly. Advertisements, where the interest of the advertiser is obvious, are not subject to such review.

DISCLAIMER

All the above makes this disclaimer even more sensitive. In the nature of things no voluntary organisation, relying on the gift of time and effort of unpaid people, is in a position to monitor every contribution for concealed elements. We have to say that all the views expressed are those of the contributors and not necessarily supported by the Committee. We also acknowledge with thanks any items from other publications not specifically attributed in the text.

NEXT COPY DATE

All copy for the next, Winter, issue of the magazine should reach the Editor by 1st December. As you will notice we have 4 pages more than this issue which we can fill, so if you have anything you want to say, please do so.

CG-OPTED COMMITTEE MEMBERS

In order to improve liaison in the TI community in the UK the Committee have asked three representatives of other groups to join them when possible. The following have agreed to do so:
Peter Brooks for the TI International Group. Tel.0865 510822
Gordon Pitt for the TI West Midlands Group. Tel.0922 476373
Trevor Taberner for the Dorset TI Group. Tel.0202 880878

CONTENTS (CONTD.)

- p. 51 Shuffling Around by Stephen Shaw. Show Map.
- 52 Printer Graphics by Stephen Shaw.
- 55 Wanted.
- 56 Count 99999 by Wesley R. Richardson(sjs)
- IBC 1992 Annual General Meeting-Derby

MODULES MODULES MODULES MODULES

APPEAL TO ALL DISK DRIVE OWNERS.....

Do you have any modules that you would consider selling or donating to the module library. Reasonable prices paid. For more information please contact me at the address given below.

The latest list of modules available for purchase follows; please note that cheques should be made payable to "E.H.SHAW". Also members are advised to contact me about the modules that they are seeking as the stock is constantly changing.

ADDITION AND SUBTRACTION 1	2.00	MUSIC MAKER	5.00
ADDITION AND SUBTRACTION 2	3.00	NUMBER MAGIC	4.00
ADVENTURE and PIRATE TAPE	5.00	HOUSEHOLD BUDGET MAN.	3.00
BLACKJACK + POKER	2.50	PROTECTOR	4.50
AMAZING	3.50	SHAMUS	3.50
BEGINNING GRAMMAR	3.00	SLYMOIDS	5.00
HOME FINANCIAL DECISIONS	2.50	OTHELLO	4.00
CONNECT 4	3.50	THE ATTACK	3.50
* DISK MANAGER I and II	3.00	LOGO I (NO MANUAL)	4.00
EARLY READING	2.50	SPEECH EDITOR	4.00
EARLY LEARNING FUN	2.50	* TI WRITER / MANUAL	12.00
* EXTENDED BASIC AND MANUAL	22.50	* TI MULTIPLAN/DISK/MANUAL	25.00
PERSONAL RECORD KEEPING	3.50	* EDITOR ASSEM / MANUAL	22.50
PERSONAL REPORT GENERATOR	3.50	TERMINAL EMULATOR II	5.00
DEFENDER	4.00	VIDEO GAMES 1	3.50
		VIDEO CHESS	4.50
INDOOR SOCCER	4.00	YAHTZEE	4.00

* MODULES MARKED WITH AN ASTERISK REQUIRE DISKS OR 32K RAM OR BOTH. ALSO PLEASE NOTE THAT EARLY READING NEEDS A SPEECH SYNTH TO RUN.

PURCHASING MODULES FROM THE LIBRARY

You may return any module purchased within four weeks and be refunded the purchase price less postage which will be charged at the rate of 40 pence per module.

Application to loan/purchase modules.

Name:..... Modules required:.....
Address:.....
.....
.....
.....

I enclose cheque/PO for £.....(as indicated on the list) & post to
PLEASE MAKE CHEQUES PAYABLE TO E.H.SHAW, MR. E.H. SHAW
Foreign orders can only be accepted if a CROW HOLT FARM
BANKERS DRAFT is enclosed drawn in STERLING BASFORD
on a LONDON bank. It also helps if a little LEEK
extra is added on for postage overseas. STAFFS. ST13 7DU

You will notice that exactly what I write above, this printed text does not contain any italics, as stated above. This is because in the end I printed it out using my trusty old M100. But please, if you can help with the fitment I'd be very pleased to hear from you. Thanks.

23.8.91

Dear Alan,
As a former member, who is going to join up again. I would like to say something about what was said at the agn, about having meetings in different parts of the country. I am all for this idea, as I live on the west coast of scotland. I hope the agn next year is in the midlands somewhere or even further north as I would like to attend. I have missed them in the past as i have been unable to attend because of dates and where they were held.

My main reason for writing is too see if you can put a note in the next copy of the magazine, concerning the fact that some people in the west of scotland would like to arrange a sort of club for T.I members where they can exchange ideas and information with people who are not miles away from them. The people I have spoken to, feel left out as most things seem to take place in london or the midlands, where people with out a car can't reach. If ~~it~~ this is possible we would be most grateful. They can contact me at the above adress.

3 people so far have got together to help each other, and we hope to meet regularly on a monthly basis, like i said we are hoping to extend this with your help.

MR ROBT R WILSON
25 MUIREND ST
KILBIRNIE
AYRSHIRE KA25 7DG
TEL: KILBIRNIE 683743

yours faithfully.

ROBERT R WILSON

NEWS FROM TEXAS INSTRUMENTS - By Mark Wills.

Written 16.8.91

I met with a TI rep at work the other day to discuss some projects my company is working on using some of thier chips, and needless to say we started talking about the old (as he called it) 4A.

He also told me TI are now formally withdrawing the complete range of TMS990 family of processors and family chips. That of course means all of the chips inside the 4A, (Processor, Graphics Processor, GROM chips, GRAM chips TMS9901 I/O chip etc) which means that our machine is now obsolete! Sob! Any one thinking about buying TI chips for replacements for their chips inside their 4A's should do so now, as the price of all chips are rocketing, and some company down south is buying them up by the lorry load! (Honestly!) The price of the TMS9900 microprocessor is now £64.50 from RS components alone! (Most microprocessors (eg. Z80) cost around £2.50 - £3.00 unless you're talking about MEGA powerful chips with cache ram and all that)

Never mind, I suppose there are still plenty of abandoned un used consoles around (tens of thousands were sold in Britain alone), for us to pull parts from when our machines break down, its just finiding them and reaching all those people in this country that own 4A's but don't know about the user group. C'est La Vie!
Lets just be thankful that the TI is such a reliable machine.

Now where did I put that black armband?

Mark

Mark Wills.

CONTD FROM PT.

About now. Think about what I have written, and enjoy your TI. Long may it live.

(To Mike Goddard: This keyboard feels LUUVVERLY! A pleasure to type on!)

Bye for now.

Mark

Mark Wills,
Vice Chairman,
Progammig Officer,
Exhausted.

POST SCRIPT:

You will notice that despite what I wrote above, this printed text does not contain any italics, as stated above. This is because in the end I printed it out using my trusty old MX80. But please, if you can help with the Citizen 120D then please let me know. Thanks.

```

DDD RRR I V V EEEEE L !
D D R R I V V E L !
D D R R I V V E L !
D D RRR I V V EEE L !
D D R R I VVV E L
DDD R R I V EEEEE LLLLL ! NO. 1

```

Whats happening in the TI world by Mark Wills.

This issue sees the start of what hopefully will be a regular column reporting on the goings on in the TI scene. So as not to compete with Stephen Shaws excellent Rambles column, I thought I'd give it a similar but more apt title! Something that reflects the content of the material in it!!

Gladly I am able to report a very active quarter, particularly concerning my own ventures with the TI. I am happy to be able to report that a Shrewsbury company called DJW MicroSystems (0743 244752) have begun researching into making a 512k battery backed RAM DISK including operating system ROM to fit into the PEB. The price is as yet not known, nor are the details as to who it will be sold through - it will probably be sold through one of the British companies that support the TI still, but it is up to me to approach them and I havn't had the chance yet!

In the meantime I hear from various sources that EAR 99ers are working on a hardware project, soon to be released that is going to send shock waves through the international TI community, (hopefully the RAM DISK will also emit such shock waves!) but I think its up to EAR 99ers to tell the world about it! It's nice to see the English getting off thier bottoms and doing something to support the TI instead of just sitting back and letting it all go on around us. A British product also means of course that prices should be lower!

I have also been working closely with the above mentioned company in developing a DTMF radio communications system for the TI-99/4A and a range of other machines such as the Commodore Amiga. The system currently employs standard Citizen Band Radios (CB's) connected to a DTMF encoder/decoder that sits in the PEB and thus connects up to the TI. The user can use the TI as a terminal for holding conversations through the computer keyboard, uploading and downloading files (to and from other machines (not just TI to TI)) and you can choose who you want to talk to - not everyone can receive the text that you send if you don't want them to! It will be a marvellous and very interesting system that employs some rather clever hardware by DJW Micro Systems and some rather clever TMS 9900 machine code programming by yours truly! (If I don't blow my own trumpet, no-one else will!) Things are still in the development stage with all the projects at the moment, but watch this space for continuing updates of whats going on. One of the nicer things about TIMES being a quarterly publication is that three months will have gone by before you next read DRIVEL 2 so I should have more information to report on.

NEW BRITISH SOFTWARE SUPPORT FOR THE TI-99/4A:

No one seems to have picked up on what I thought was a very interesting peice of information so I will report on it, at the risk of sounding like I am advertising my own products...

I have launched my own commercial software house that deals entirely with the TI-99/4A called ABBOTS SOFTWARE. There are currently four software titles available (see the advertisement in this issue) and they all available through Mike Goddard. His address is on the front cover...

Wrexham for their incredibly prompt service. The keyboard in my 4a console finally gave up the ghost when I dismantled my console to fit a Load Interrupt Switch and Master Reset Switch, the wires that connect the keyboard to the main circuit board snapped inside their plastic insulation which meant I had no keyboard! I phoned MGCS on the Friday, and a brand spanking new never been used keyboard arrived on the following Tuesday. Hows that for good service! This is now the THIRD keyboard to go into my console so perhaps this one will last a little longer. A tip... When taking your console apart and keyboard disconnection is necessary, be CAREFUL when disconnecting the keyboard "plug" from the main PCB. They are VERY fragile! I found the hard way!

WANNA BUY A TI?

Just thought I'd mention this one... A sudden un-expected redundancy led to me being in the careers office one afternoon, discussing my options as far as furthering my career went. I took along TMS9900 source code that I had written as examples of my work and competence (well you have to sell yourself!) and in the course of our conversation managed to tell the Careers Officer all about the TI and how wonderful and incredibly priced it was. It seems he was so impressed with what I told him that he has written to me to ask if he can see my system before he buys himself a fully expanded system!! It seems theres still a market for the trusty old TI-99/4A!

GAME WRITERS TOOLKIT UPDATE

I feel a bit quilty writing about my own products but someone has to inform you people out there! (If anyone objects then please say so).

I have decided to bow to pressure and convert The Game Writers Toolkit into a version that is compatable with Extended Basic. I have demonstrated the MINI MEMORY version to many people and they have all been impressed with it, and a few people who have bought it have written to say that they like it, but it seems that everyone would like to see it run in EXTENDED BASIC. So okay I'll start work on it this month. It will be version 4.0 by the time its debugged and finished. It will not be considered an update of the MINI MEM version however. It will be a seperate product in its own right. It will be completely compatable with its TI BASIC+MINIMEM counterpart plus it will support 16 colour 256 x 192 high resolution graphics. Should be a nice (machine code) utility, and cheaper than certain other basic

extensions available from the States. For those who don't know, it is a machine code extension to TI BASIC which allow you to do things like screen scrolling (even windowed scrolling) 32 sprites with movement, all four sizes, and much much more.

TO ALL COMMITTEE MEMBERS

Apoloques for not being able to make the meeting. A last minute problem at work meant I was unable to make that train times. Sorry.

HELP ME PLEASE!

Would anyone who knows ANYTHING about the language PILOT for the TI-99/4a please get in touch with me. I know a fair few people in the computer industry and no-one seems to have heard of it! Was is it? What are its disadvantages/advantages? What equipment is required? Can I get it from the disk library?

HELP ME AGAIN PLEASE!

I have installed a LOAD INTERRUPT switch and master RESET switch to my TI so that I may write some software that uses it. They are very useful devices indeed. When I am programming in machine code (especially the line by line assembler) I set the LOAD INTERRUPT vector at >FFFC to the start address of the line by line assembler before I run my machine code program, then if my program is buqged and crashes irrecoverably, then a simple push on the LOAD switch automatically restarts the assembler most of the time. The software I am writing for the DTMF modem system uses it to restart the program if a crash occurs. Nice for de-buqqing purposes! Should everything else fail then hit the RESET switch and the whole system restarts but all RAM is intact. Handy eh! Anyway I diqress... I wonder is there any software out there that already takes advantage of these devices? I suspect there is because I have seen LOAD and RESET switches on other TI's at meetings etc. If you know of any software that uses these devices then please let me know, or send it to me on a SSSD disk with some docs that tell me what to do, and I'll fill the disk with programs and send it back to you. (Shareware/public domain programs that is!!). Any offers?

HELP ME JUST ONE MORE TIME!

You will notice that the print out of this text seems to consist of italics mixed with normal characters. They should all be normal characters! I have changed printer from a EPSON MX80 to a CITIZEN 120D running on RS232 (serial), but as you can see its not entirely happy. I suspect it is the DIP switch settings on the 120D serial card, but I do not have a manual. Anyone out there using a 120D printer on thier printer? If so send me the DIP switch settings that you are using for either your serial (RS232) card on the 120D or the parallel (PIO) card, if I can find the PIO settings then I will use the parallel card in preference to the serial. Come on, surprise me and show me some feedback!

MORE LAST MINUTE NEWS!

After consulting with Mike Goddard, Mike and myself have decided to produce a MOUSE for the TI with driver software. I am building the MOUSE board which will fill fit into the PEB, and I am also writing the software that drives it. The board will also have a parallel input AND output ports thrown in for good measure. The supporting software will be machine code programs, in ready to run forms for TI BASIC + MINI MEM or EDITOR ASSEMBLER and TI EXTENDED BASIC VERSIONS so that you will be able to use the mouse in your own programs. One CALL LINK at the start of your program will be all you need, from then on the MOUSE arrow movement will be automatic. I am also going to write a GRAPHICS BASED WORKBENCH system for it which will be included with the MOUSE. Anyone who has used an ATARI ST or AMIGA or WINDOWS on the PC range of machines will know what I mean. Basically you'll be able to load/view files by pointing the MOUSE arrow at a picture of the file you wish to load/view and then press the mouse button. No keyboard interaction will be needed. The system will be intelligent in that if you click on a program file then it will load it, if you click on a DISPLAY type file then it will display it. Should be nice. (I am going to be REAL busy over the next few months!!)
The whole package will be marketed through MIKE GODDARD COMPUTER SUPPORT in Wrexham, who is already well established and has a very good reputation, it will of course include the actual MOUSE as well! Contact Mike goddard for details on price and supply on 0978 843547 not too late at night though!

HAS HE FLIPPIN' WELL FINISHED YET?!

Almost!! Bear with me!! Readers may be wondering why we are bothering to produce so much hardware for a machine that is so old? Well, after talking to a few people in the TI community we have all agreed that support for the TI as far as commercial concerns go it is dropping off. Myarc seem dormant, the Geneve is no more it seems (if I'm wrong then please correct me) and Quest have ceased production of the RAM DISK that they made. This, I and many others have found alarming. Without these efforts by third partys the machine would stand still. It is for this reason that we are deciding to make our own peripherals. Hardware made in this country will of course bemuch cheaper (I hope) to buy and to service as well. It is also well known that the UK is a bunch of TAKERS as far as the 4A goes, so we are hoping to change that. Now people abroad will have the opportunity to buy products from the UK for a change! Hooray!! It is the combination of all these things that have led us to make a really hard effort to support the TI. If any of you TI'ers have been thinking of selling up and moving to another computer, I would urge you to wait and see what happens first! I really do think that the TI'er in this country will be much better served soon, it will take time because I intend to take on a lot of these projects on my own (anyone wishing to help in any way they feel they can is welcome to!) I feel sure they will happen, and we together will bring the 4A into the nineties. Stick around and enjoy the ride.

WHEN IS THIS GUY GONNA SHUT UP?!!

**** ADVERTISEMENT ** ADVERTISEMENT ** ADVERTISEMENT ** ADVERTISEMENT ****
A NEW SOFTWARE COMPANY HAS JUST BEEN LAUNCHED FOR THE TI-99/4A COMPUTER

<< ABBOTS SOFTWARE >> << ABBOTS SOFTWARE >> << ABBOTS SOFTWARE >>

WE ARE PROUD TO BE ABLE TO OFFER THE FOLLOWING PROGRAMS FOR THE TI-99/4A

HELIBOMBER - TI BASIC OR EXTENDED BASIC : PRICE - 5.00p
Can you destroy the buildings below you before your helicopter runs out of fuel and crashes? Fun for all the family. Easy to play. Lovely colourful graphics. Ideal for children or adults alike. No extra hardware required.

CREEPY CRAWLIES - TI BASIC OR EXTENDED BASIC : PRICE - 5.00p
Help Barnaby the Bug to get his baby Bugs across the three gardens and fend off those evil spiders, who are trying to eat those poor little bugs. Lovely graphics. Shows what the TI can do in TI BASIC. Children will love this cute colourful game. (Adults too!!) No extra hardware required.

CHARACTER SET/SPRITE EDITOR/ANIMATOR - EXTENDED BASIC + CASSETTE OR DISK (PRINTER OPTIONAL) : PRICE - 6.00p
Design your own character sets or sprites on a large grid on the screen. Rotate them, flip them, magnify them, save them to disk or tape or print the HEX values to a printer. Files created with this program can easily be loaded into your own programs and used with games etc.
Also features an animator. You can animate your sprites on the screen at a selectable speed, return to the editor to make alterations etc then save the lot to use in your own programs!!

THE GAME WRITERS TOOLKIT - MINIMEMORY REQUIRED : PRICE - 8.50p
We believe this represents a breakthrough for the MiniMemory Module. Blow the dust off that MiniMemory and get programming!! How does this sound? 32 Sprites in all the four selectable sizes ala Extended Basic! Four directional WINDOWED screen scrolling! You can scroll the whole screen or just part of it in any direction! Take a photo of any sized section of the screen and bring it back up at any time, INSTANTLY!! A DISPLAY AT command (at last!!) A BYE command to reset the machine from within TI BASIC programs, disable the reset key, true lower case character set AND MORE!! To our knowledge there is NO better BASIC extension in existence for the MINIMEM. Comes complete with manual and demo programs. This program represents around 12 months of continuous development. Get it NOW!!

ALL PROGRAMS AVAILABLE ON TAPE OR DISK (STATE WHICH WHEN ORDERING)

ALL CHEQUES SHOULD BE MADE PAYABLE TO MIKE GODDARD. SEND TO : MGCS,
"SARNIA", CEMETARY ROAD, RHOS, WREXHAM, CLWD. LL14 2BY.

<< ALL PRICES INCLUDE POSTAGE AND PACKING. >>

TECHNICAL ENQUIRIES : M.WILLS, 12 BETTON STREET, SHREWSBURY. SY3 7NY

Santa needs multi-length arithmetic package.

On p.23 TI*MES 32, Stephen mis-remembered the terms of a New Scientist puzzle and so transformed it from something light to a real bonebreaker. Santa's house number, we were told, was the 25th in an increasing sequence of numbers each of which became its own square root upon dropping its first digit. I have found only three such: 25,36,625. The fourth, IF it exists, is $>10^{\wedge}31$. Number theory seems not to do more than provide candidate values for final acceptance/rejection by arithmetic.

So, computer search or theory, we need multi-length arithmetic. The same applies to many more serious problems than Santa's. Can TI*MES readers offer anything with a reasonable execution time using only XPAS? I have routines that will handle numbers with 254 digits but they are painfully slow -- squaring an n-digit number requires approx. $0.5 * n^{1.9}$ sec. Do we have to go to Assembler?

Returning now to Santa, the least demanding (arithmetic-wise) of the three approaches that I have tried is the following (needs multiplication only).

A necessary condition for an n-digit number to be the square root of a number in the Santa sequence is that it reproduces itself in the last n digits of its square. This is not sufficient but it eliminates most numbers. E.g. 25 appears at the end of its square, 625, which happens to be a Santa number. So does 76 in its square 5776 but the latter does not yield 76 by simply dropping the 5 and so is not a Santa number. There are only two such "reproducing" numbers for each n (excluding trivialities like 001). For the first few n, these are 5,6; 25,76; 625,376; 0625, 9376. There is a simple law of formation of successive pairs. (Curiosity: each pair sums to $1+10^{\wedge}n$. Why?) The reproducers are tested by forming their squares which, to be acceptable, must contain only one non-zero digit apart from those of the reproducer itself at the end. Of course, however many numbers are rejected, there is still no proof that further Santa numbers do not exist. The number after the last one tested might be it.

I wonder if I have been using cudsels to break an egg. Maybe someone will write to ask "Why on earth didn't you simply"

Walter Allum

HERE IS AN EXAMPLE OF HOW TO MOVE THINGS AROUND ON THE SCREEN USING TMS 9900 ASSEMBLY LANGUAGE. THE FIRST PROGRAM PLACES AN ASTERISK IN THE CENTRE OF THE SCREEN AND THEN ALLOWS THE USER TO MOVE IT AROUND USING THE ESDX KEYS. THE PROGRAM WILL PREVENT THE ASTERISK FROM GOING OFF THE TOP OR BOTTOM OF THE SCREEN, BUT WILL ALLOW A WRAPAROUND FROM THE END OF ONE LINE ONTO THE BEGINNING OF THE NEXT. THE FCTN = IS ACTIVE IN THIS PROGRAM AND WILL RETURN THE USER TO THE TITLE SCREEN.

PROGRAM 2 IS EXACTLY THE SAME EXCEPT IT ALLOWS KEY REPEAT. THE REPEAT SPEED IS VERY FAST INDEED. TOO FAST FOR THE FCTN = TO WORK! BUT ILLUSTRATES JUST HOW FAST MACHINE CODE IS. THESE PROGRAMS WERE WRITTEN ON THE EDITOR ASSEMBLER, BUT COULD EASILY BE CONVERTED TO WORK WITH THE MINIMEMORY. THEY COULD THEN BE INCORPORATED IN TO TI BASIC GAMES TO MAKE THEM RUN FASTER!

IF THERE'S ANY OTHER 9900 TINKERERS OUT THERE WHO'D CARE TO WRITE, DROP ME A LINE AT THE ADDRESS BELOW. I'M NOW OFF TO INCORPORATE THESE ROUTINES INTO A GAME I'M WRITING. SEE YOU!

N. BARRIE CLARK
SHANTAY
53 CLEEVE DRIVE
CLEEVE AVON
BS19 4NP

```

DEF BAZ
REF VSBW,VMBW,KSCAN
STATUS EQU >837C GPL STATUS BIT
KYPRS EQU >8375 KEYPRESS STORAGE
KEYADR EQU >8374 EMULATES CALL KEY(0)
GPLWS EQU >83E0 EMULATE QUIT
SAVRTN DATA >0000 ENTRY POINT OF PROG
CHAR TEXT '*'
UPKY BYTE >45 E
DNKY BYTE >58 X
LFKY BYTE >53 S
RTKY BYTE >44 D
QUIT BYTE >05
WSPREG BSS >20 ALTERNATE WORKSPACE
BAZ MOV R11,@SAVRTN
LWPI WSPREG
BL @CLEAR
LI R7,367
PUTUP MOV R7,R0 PUT THINGS
LT R1,CHAR ON
LI R2,1 SCREEN
BLWP @VMBW
SCAN CLR @STATUS SCAN KEYBOARD
CLR @KEYADR
BLWP @KSCAN
MOV @STATUS,@STATUS ANYTHING PRESSED?
JEQ SCAN
MOV R7,R0 WRITE SPACE
LI R1,20 TO SCREEN
BLWP @VSBW
CB @KYPRS,@LFKY CHECK FOR S
JEQ LEFT
CB @KYPRS,@RTKY CHECK FOR D
JEQ RIGHT
CB @KYPRS,@UPKY CHECK FOR E
JEQ UP
CB @KYPRS,@DNKY CHECK FOR X
JEQ DOWN
CB @KYPRS,@QUIT CHECK FOR QUIT
JEQ OMEGA
JMP PUTUP REPLACE IF OTHER KEYS PRESSED
LEFT CI R7,0 TOP LEFT OF SCREEN?
JEQ PUTUP
DEC R7
JMP PUTUP
RIGHT CI R7,767 BOTTOM RIGHT OF SCREEN?
JEQ PUTUP
INC R7
JMP PUTUP
UP CI R7,32 TOP OF SCREEN?
JHE PUTUP
LI R1,32
S 1,7
JMP PUTUP
DOWN CI R7,735 BOTTOM OF SCREEN?
JHE PUTUP
LI R1,32
A 1,7
JMP PUTUP
CLEAR CLR R0
CLR R1
LOOP BLWP @VSBW
CI R0,767
JEQ CLEARX
INC R0
JMP LOOP
CLEARX B *R11
OMEGA LIMIT 2
LWPI GPLWS BRANCH TO TITLE
BLWP @>0000 SCREEN
END

```

```

DEF BAZ
REF VSBW,VMBW,KSCAN
STATUS EQU >837C GPL STATUS BIT
KYPRS EQU >8375 KEYPRESS STORAGE
KEYADR EQU >8374 EMULATES CALL KEY(0)
GPLWS EQU >83E0 EMULATE QUIT
SAVRTN DATA >0000 ENTRY POINT OF PROG
CHAR TEXT '*'
UPKY BYTE >45 E
DNKY BYTE >58 X
LFKY BYTE >53 S
RTKY BYTE >44 D
NOKY BYTE >FF
QUIT BYTE >05
WSPREG BSS >20 ALTERNATE WORKSPACE
BAZ MOV R11,@SAVRTN
LWPI WSPREG
BL @CLEAR
LI R7,367
PUTUP MOV R7,R0 PUT THINGS
LT R1,CHAR ON
LI R2,1
BLWP @VMBW
MOV @KYPRS,R3
SCAN CLR @STATUS SCAN KEYBOARD
CLR @KEYADR
BLWP @KSCAN
CB @KYPRS,R3 ANYTHING PRESSED?
JEQ SPACE
CB @KYPRS,@NOKY
JEQ SCAN
MOV R7,R0 WRITE SPACE
LI R1,20 TO SCREEN
BLWP @VSBW
CB @KYPRS,@LFKY CHECK FOR S
JEQ LEFT
CB @KYPRS,@RTKY CHECK FOR D
JEQ RIGHT
CB @KYPRS,@UPKY CHECK FOR E
JEQ UP
CB @KYPRS,@DNKY CHECK FOR X
JEQ DOWN
CB @KYPRS,@QUIT CHECK FOR QUIT
JEQ OMEGA
JMP PUTUP REPLACE IF OTHER KEYS PRESSED
LEFT CI R7,0 TOP LEFT OF SCREEN?
JEQ PUTUP
DEC R7
JMP PUTUP
RIGHT CI R7,767 BOTTOM RIGHT OF SCREEN?
JEQ PUTUP
INC R7
JMP PUTUP
UP CI R7,32 TOP OF SCREEN?
JHE PUTUP
LI R1,32
S 1,7
JMP PUTUP
DOWN CI R7,735 BOTTOM OF SCREEN?
JHE PUTUP
LI R1,32
A 1,7
JMP PUTUP
CLEAR CLR R0
CLR R1
LOOP BLWP @VSBW
CI R0,767
JEQ CLEARX
INC R0
JMP LOOP
CLEARX B *R11
OMEGA LIMIT 2
LWPI GPLWS BRANCH TO TITLE
BLWP @>0000 SCREEN
END

```

!*=DREADFUL **=AVERAGE ***=GOOD
 | ****=VERY GOOD *****=EXCELLENT

TRAFFIC COP

You are a police man controlling the traffic on a crossroads you have the power to stop one car for any period of time.
 At the beginning of the game there are 4 cars one at each road they then start moving in one direction at 8 MPH and your job is to stop them from colliding. So say if you wanted to stop the car which had started at the top of the screen you press down and hold the "E" key and if you wanted to stop the car which started on the right hand of the screen you press "D" and "S" for the left hand car etc etc.
 "Wow easy" you might say but it isn't because the cars increase speed at various stages of the game. Luckily the speed is shown of the top right hand side of the screen and whenever it changes you hear a sound like a car horn.
 The game ends when you have crashed 10 times.
 A good keyboard game for 1 player.
 STAR RATING ***

MARKET SIMULATION

This game simulates competition between 2 companys.
 In each round the player(s) input the production level, the unit cost and the advertisement budget but they must bear in mind the fluctuating wholesale cost per unit the advertising budget, the fixed production cost (£250) and any other things that the computer might come up with for instance strikes, embesilment etc etc.
 You start off with £5000 and the game ends when one of the companys is bankrupt or one of the companys assets reaches £12000.
 An excellent Extended Basic keyboard game for 1 or 2 players.
 STAR RATING *****

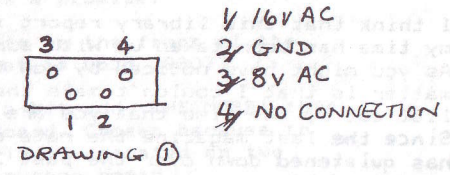
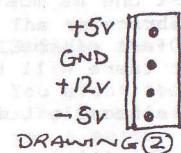
WEBSTER

Webster is a spider and he is very hungry.
 You are webster and you move yourself with the keys S and D for left and right respectively. When you are directly above a fly (an oblong thing with about 5 legs) you spin your web down, catch it and eat it. BUT beware of the scorpion who will kill you if you land on him. You start of with 3 lives and a rumbling tummy!!.
 A good Extended basic game for 1 player.
 STAR RATING ***

Well it outgrew my shed, the loft, the wardrobe, the carport, the spare room, the dining room, and the hamsters cage, What you might ask M.G.C.S did thats what. So in the interests of preseving my wifes sanity and my marriage I have aquired some premises for M.G.C.S also as a bonus there is also a new workshop so hopefully I will be able to provide a better and more efficient service to members in future. And my thanks to all those who have had to put up with delays etc in the past.

Several members have written asking for a fault finding guide for the console I don't recollect seeing one so for the next few issues I will publish several small hints and tips starting here.

The first thing to check when the console goes down is that there is power getting to it so back to basics check the fuse is intact in the mains plug. If that is OK next check the power plug into the console some form of multimeter will be necessary for these tests. Set the meter to a suitable AC range of about 25 volts and test the voltages on the plug see drawing 1. If these aren't correct open the power supply box and test the fuses there if faulty replace with a similar fuse and check voltages again but if everything is OK open up the console and remove the power supply board. Disconnect the fourway connector leading to the main computer board and it should be marked with the various voltages alongside see drawing 2 check these with the multimeter but remember the last voltage is -5v so the polarity will be the opposite way around if all is correct further investigation is required more next time.



- 1/ 16V AC
- 2/ GND
- 3/ 8V AC
- 4/ NO CONNECTION

There will probably be mention of this elsewhere in this issue but the committee have received a request for help for some computer students in Ghana from Professor Erwin Van Meir of Switzerland who discovered these students while there on holiday and actually gave them his TI as they were trying to study computing without any computers at all!!!!. His initial request was for some English manuals which M.G.C.S has agreed to provide at no charge. However Nicky thought it would be a good idea to start an appeal to raise some further equipment to help them out so if anybody has any unwanted modules or cassettes they would like to donate please send them to Nicky anything that is of no immediate use to the students will be kept till later or auctioned to raise cash to provide the necessities.

Mike.

WREXHAM & CHESTER WORKSHOP REPORT.....NICKY GODDARD

PLAS MADOC LEISURE CENTRE

Several members travelled a great distance to see what I thought was a great show .

There was going to be a red dot system where you paid on the door and those who had paid recieved a red dot to stick on their jumper or jumper or whatever but I was the only one who recieved a red dot!!! But that didn't mean that I was the only one who paid.

It was nice to see some of the committee there who were:-

Edward Shaw - Module Librarian.

Mike Goddard - Hardware Consultant and of course myself -

Nicky Buddard - Cassette Librarian.

Mark Hills was on hand to demonstrate the Games Writers Tool Kit.

and I was demonstrating a composite monitor interface on my own T.I. . I think that all of those who attended had a great day and it was nice to greet old and new friends alike.

The leisure center itself contained a squash court a cafe and of course a swimming pool which is heated to 87 degrees F.

So next time you come bring the wife and kids so that you can dump them in the swimming pool which is overlooked by the room the workshop is held in and then get on with some serious TIing.

CASSETTE LIBRARY REPORT.....NICKY GODDARD

I think that this library report is going to be a short one as most of my time has been taken up with sorting the cassette library. As you might have noticed by now there is no list the fact of the matter is that I couldn't make the TI*MES deadline but there will be a list out by the time that you are reading this article.

Since the last magazine the rate of orders went up tremendously but it has quietened down over the past 2 weeks.

As I said if you wanted the list in this magazine I do apologise but there DEFINATELY will be a list in the next issue. If you can't wait that long please send a stamped addressed envolops of a reasonable size to the address on the cover of the magazine.

There is no surplus stock cassettes for sale either as I have sold all of them but I will bring a few along to any T.I. meeting that I go to.

Just to remind you the prices of library cassettes is :-

50p IF YOU SEND A CASSETTE
£1 IF YOU DON'T SEND A CASSETTE

Mike

```
*****
* * * * *
* * * * *
* * * * *
* * * * *
*****
```

BY TREVOR STEVENS (C) 1991

Well I write this time to tell all you modem owners that you now have your very own bulletin board. This board which is run on the program SMALL TALK has the makings of something very big.

At the present I have On-line the following items of interest. Problem page, Private and none private person to person messages, Public Domain Programs. For Sale File, plus much more in the pipe line.

This BBS will be open at 300 baud to start with, depending on the response it will also go on line at 1200/75. Those out there with those posh 2400 may also be catered for if they tell me about it.

Because the board is only just starting it will only be put On-line between 6pm and 10pm Mondays only. This will be increased slowly to other nights, at higher baud rates. but time will tell with use. The board will also welcome other users from other computers such as IBM, C64, BBC, AMIGA. However this BBS will remain totally TI orientated, and thats a promise.

The telephone number for the BBS is on the front of the TI*MES under the Chairmans address. That is 0623 793077.

You might think that the name is a bit odd (THE MOBB) this means the Mansfield On-line Bulletin Board. Chosen because to give you a better service later on, will be operated on two numbers, with two operators running the same board.

Do have a look at the BBS and see what you think. Maybe you could write a read file for us and upload it as per the instructions on the board. If not leave SYSOP (T.STEVENS) a message as to your comments, and ideas.

Please use your board, it is free.....

ALSO:- TI*MES HOT LINE

I have set up for all you people out there who have a problem with your computer, I have now set up the HOT LINE. This is a line that you can call to obtain help on anything that you may have trouble with. It can also be used to express your views directly to me your Chairman so that things can be altered to suit you.

If you have anything to say ring me on 0632 793077 between

Tuesdays and Saturdays (6.30pm - 9.00pm), or write to me on the address given on the front cover. I may not always be available but you will be given a time I will be in as I work shifts. Or you will be given another contact number if applicable.

Please say what you want, as we can only provide what you require.

RAMBLES by Stephen Shaw for TI*MES (UK) #34.

My address, also the address of the Disk Library, is on the front cover, and as always, your comments and suggestions are most welcome. It is helpful if queries requiring a direct reply are accompanied by an SAE.

The disk library details are available by sending four disks and return postage (or for overseas members, just a straight US\$10 note). The disk library catalogue is supplied as a set of DV80 text files on disk for printing out with (almost) any text editor. A short ExBas program can be sent to produce output on disk/printer if required- just ask and a LOAD (XB) program file will be added to the first disk.

First a math program which can take two numbers as input and will output the greatest common divisor and the least common multiple.

For example, if we factor two numbers (list all numbers which multiplied together result in that number):

24 = 1 x 2 x 2 x 2 x 3
78 = 1 x 2 x 3 x 13

Of the divisors listed the largest number common to both is 3 and the least common multiple (the lowest number which divisible by both numbers) is 312 (that is 24 x 13 and also 78 x 4).

```
100 DISPLAY AT(1,1)ERASE ALL:"PROGRAM TO FIND":":":THE GREATEST COMMON
DIVISOR": ":":AND THE":":LEAST COMMON MULTIPLE"
110 DISPLAY AT(8,1):"Spencer Earnshaw/M W Ecker/ S Shaw 1991"
120 DISPLAY AT(14,1):"FIRST NUMBER:":":":SECOND NUMBER:":":
130 ACCEPT AT(14,15)VALIDATE(DIGIT):A
140 ACCEPT AT(16,15)VALIDATE(DIGIT):B
150 AA=A :: GCD=B
160 R=AA-INT(AA/(GCD+ABS(GCD=0)))*GCD
170 IF R<2 THEN GCD=R+GCD*(1-R):: GCD=GCD*ABS(GCD>0)+ABS(GCD=0):: GOTO
190
180 AA=GCD :: GCD=R :: GOTO 160
190 DISPLAY AT(1,1)ERASE ALL:"THE TWO NUMBERS:": " ";A;" ";B
200 DISPLAY AT(4,1):"GREATEST COMMON DIVISOR: ":TAB(13);GCD
210 DISPLAY AT(6,1):"LEAST COMMON MULTIPLE IS: ":TAB(13);A*B/GCD
220 DISPLAY AT(12,1):"ANY KEY FOR ANOTHER"
230 DISPLAY AT(16,12):"< key > " :: CALL KEY(S,K,S):: DISPLAY
AT(16,1):" " :: IF S<>I THEN 230
240 RUN
250 END
301 ! from recreational and educational computing
302 ! $13 sample copies $36 subscription (europe)
303 ! 909 Violet Terrace Clarks Summit PA USA 18411
```

CHRISTMAS PUZZLE and I do have a program that solves this one and will (probably) print it next issue if nobody send me a better solution. The only hint I will give is that the computer will point you in the right direction. (Anyone do cryptic corsswords?).

Puzzle- Santa Claus (or if you prefer, Father Christmas) was delivering presents on Christmas Eve, along a loooong street. When he got to house number 125 he noticed that dropping the first digit left 25, a number which is of course divisible by 25. When he got to the 25th house which satisfied this condition, he discovered it was his own. What was the number? All 25 numbers satisfy the one condition: When you remove the first digit, all the remaining digits form a number divisible by 25. Leading zeroes as in 00025 are of course ignored and 00025 is treated as 25 which is divisible by 25.

A garbled version of this puzzle appeared last year- THIS VERSION IS CORRECT.

DISK LIBRARY ADDITIONS SINCE JULY 8TH 1991.

Music disks from HARRISON SOFTWARE:

>LUTE MUSIC OF THE 17TH AND 18TH CENTURIES.

>IL PASTOR FIDO (TWO DISKS) by Antonio Vivaldi.

These are commercial disks which we are licensed to copy. Letters and donations to Harrison are welcome as are suggestions of further works to prepare -taking into account our three voice sound chip!

>FONT DESIGNER 1.1 by John Seager. Design or modify fonts for TI Artist up to a maximum size of 16 pixels wide and 24 pixels high. Transfer the definitions in CHARA1 files to TI ARTIST Font files.

FRENCH GAMES- rules in French but relatively easy to pick up, or short docs in English added (except where noted!!!):

>JEU5: 7 SORCIERS (several random games in one. Bit of gambling too. Neat.); 8 Ball (is this pool?); Alerte; Anor; Asterix; Avenger 2; Awari (three stone version of classic game); Bat; Bouffe; Burgerman.

>JEU6: ANDROID; AVENGER 1; BIG HOTEL (similar to Spy's Demise); Boggle (word game, need rules, computer just roles dice and keeps time); Ceccaldi; Charlot (fun demo); Enfer.

>JEU7: ABORDAGE; ASTRO; AUTOROUTE; CITY/SEA; COMMANDO; CROCODILE; DAFFY; DCA; EMPIRE (Star Wars); EXPLOSION.

>JEU8: ADAM ET EVE; ATOMIC; BALLTRAP; CERBERE; FEU; FLIPP; FORMULE3; JUMPMAN.

>JEU9: COCHON (Pig as in MCP-could possibly be offensive to some); EAU; Herbert; Jevvie (LIFE program); Kong-Bert; Labyrinth; Lasso; Lievre; Loup(excellent graphics on classic wolf v sheep game); Memoire.

>JEU10: PITFALL; PLONGEON; SCROFULSE; SKI; SOSROBIN; STORE; VOLTAGE; ZARBEK. Note that Pitfall appears mildly incomplete-as it appears to be little more than an aimless wander collecting treasure and avoiding sprites (I could be wrong!) this does not seem too disastrous to game play!?

>JEU11: MASTERGOLF; MICROBES; MINIGOLF (Crazy Golf-different, hard and fun); Oeuf; Pilot; Pioneer; Safari; Starship. Some nice games.

>JEU12: Ascenseur (superb game of skill- I had to check there was no machine code in this, very well put together!); Kouilili2- difficult two screen game; TI Lode; Tirsobus

>INTERCEPTEUR. plus MAISON; MISSION; MONKEY; OVERTAKER; PENTAMINOS.

>LUNAR. Lunar Lander (hard but fun- NOT like anything else with this title!), and Lunar Jumper, which looks similar to Moon Patrol, but has lots of French docs, and I have not been able to fathom out what to do (key response appears a little slow), help is requested!

All disks one pound each if you send a blank disk, plus a flat pound per order for post, packing etc. Full catalogue? Just send four disks and return postage.

=====

```
X  X  BBBB  # 19
X  X  B  B
  X  BBBB  By
X  X  B  B  Jim
X  X  BBBB  Swedlow
```

[This article originally appeared in the User Group of Orange County, California ROM]

ON FREWARE

Last year I wrote an Extended Basic program that prints a Multiplan spreadsheet sideways. I wrote it because I was tired of seeing new products for those other computers but not the 99/4A.

After a great deal of thought, I decided to release it as freeware. As I write this I have shipped over 100 copies of my program and, based on user input, revised it twice.

The experience exceeded my expectations. I have had requests from over 30 states and Canada and have received some very nice letters. Its nice to know that there are others out there who still use the 4A.

I'd like to pass on some suggestions should you write for freeware.

IF YOU SEND MONEY: First and foremost, please print clearly. I agonized over deciphering two addresses. Better yet, if you have one, send a printed return address label. You could use this month's program to make some.

IF YOU SEND A DISK: Initialize the disk. This makes sure that it is OK. Sweeping the disk is not sufficient, format it and verify the sectors.

While you are at it, why not put a few public domain programs on the disk? Three folks sent me some programs that were really nice. Do include a note so that the recipient will check the disk, however.

WRITE A NOTE: I released my program to share something with the TI community. The notes I received were worth their weight in gold. On the other hand, the guy who sent me a check without any kind of note did not make any points!

GIVE FEEDBACK: Take the time to let the programmer know what you thought of the program. If you liked it, say so. If it didn't work, let him know. If you have a suggestion, pass it on. If you want a response, include a SASE (Self Addressed Stamped Envelope). The whole thing costs you \$0.44 postage but is important to the recipient. Believe me!

PAY FOR THE PROGRAM: Some freeware/shareware programmers ask you to pay \$10 or so if you like the program. Considering that most commercial software starts at \$20, this is a bargain. Some of the best new items come from freeware/shareware. Without your support, this source just might dry up.

PS: This also applies to freeware/shareware from our users group library. If the program or documentation asks for money, send it. The one pound charge per disk goes to our users group, not the author.

ON PRE-SCAN AND DIM

If you are not familiar with pre-scan, read the supplement to the XB manual!

Your TI does not execute a DIM statement. The only time it reads the DIM is during pre-scan. Therefore, DIM can be part of the pre-scan list.

For those arrays that have ten values, you do not need to DIMention them. You only need to put A0 in the pre-scan list. The parenthesis can be completely empty (assuming a one dimension array, that is).

This, then, is a legal, functional pre-scan list:

```
100 GOTO 110 :: A,B,C#,C0,
D# :: DIM Q(12) :: !@P-
110 ! Program continues
```

This program, of course, uses no subroutines. For another example, see lines 150 to 180 of this months program.

LABEL

I haven't published a program in a few months -- my entry into the freeware arena took up more time than I expected.

I have previously published this program as two separate ones: one that prints disk labels and one that did mailing labels. It got to be a chore changing programs so I combined them.

CUSTOMIZING: Substitute your name and address in the DATA statements in line 170. Be sure and keep the leading and trailing commas (unless you use a four or five line address). Your address lines should be no more than 27 characters long.

Lines 190 to 260 are the printer control codes. They work for most Epson and compatible printers. If your printer uses different codes, change them as necessary.

If your printer name is not PIO, modify line 280 as needed.

DEBUGGING: If you run into problems, delete the << !@P-> at the end of line 180 to keep pre-scan on for the entire program and change <ON ERROR 580> to <ON ERROR STOP> in line 370. This should help you find any errors.

PROGRAM NOTES: Lines 550 and 560 do some interesting things:

ON POS is used to validate the correct key return. The value of I then is used, after returning from the subroutine, as the value in <ON I GOTO . . .>.

The second CALL KEY in line 560 remaps the keyboard as a 99/4A so that you can input lower case for your labels.

By changing 13 and 32 to 80 (or <ENTER> and <SPACE BAR> to <P>) in line 550, you can start printing by pushing P or ENTER or the SPACE BAR.

Enjoy.

```
100 ! LABEL
110 ! Version XB.3.0
120 ! By Jim Swedlow
130 ! 22 Mar 86
140 ! Based on a program by
    Phil Barnes
145 !
150 GOTO 180 :: IN$,SS$,DS$,
PI$ :: DIM Y$(4),A$(5):: C$,
D$ :: DIM F$(144),T$(144)::
B,J,D,E$,I,C,U :: CALL KEY
160 DATA " FIRST CLASS
", " MAGNETIC MEDIA",DO
NOT BEND : DO NOT X-RAY,DO NOT
EXPOSE TO MAGNETISM,
170 DATA "Your Name,Street A
ddress,"City, CA ZIPCD",
180 U=1 :: CALL CLEAR :: FOR
J=0 TO 14 :: CALL COLOR(J,1
6,U):: NEXT J :: CALL SCREEN
(5):: !@P-
190 E$=CHR$(27) ! ESCAPE
200 DS$=E$&"G" !
DOUBLE STRIKE
210 C$=CHR$(15) ! CONDENSED
220 D$=CHR$(14) ! DBL WIDTH
230 PI$=CHR$(18) ! PICA
240 SS$=E$&"S"&CHR$(0) !
SUPERSCRIPT
250 IN$=E$&"U" !
INITIALIZE PRINTER
260 E$=E$&"3"&CHR$(12) !
CHANGE LINE FEED TO
12/144 INCH*<CHANGE THIS!
270 PI$=PI$&C$&D$ :: DS$=DS$
&C$ :: SS$=SS$&E$ :: Y$(U)="
DF" :: Y$(2)="DV" :: Y$(3)="
IF" :: Y$(4)="IV" :: D$="DSK
1" :: C$=""
280 OPEN #1:"PIO" :: DISPLAY
AT(3,9):"LABEL 3.0" :: "Che
ck the position of the lab
els before printing."
```

```
290 DISPLAY AT(20,U)BEEP:"<A
>address":"<C>ustom ":"<D>isk
":"<W>arning"
300 GOSUB 550 :: IF I>4 THEN
300 ELSE B=U :: GOTO 330
310 DISPLAY AT(10,U) :: "
Labels/Disk: ";B:" Dr
ive: ";D$:" Comment:
";TAB(6);C$: :
320 GOSUB 540
330 ON I GOTO 470,500,310,47
0,370,340,600
340 ACCEPT AT(12,20)SIZE(-2)
VALIDATE(DIGIT)BEEP:E$ :: IF
E$="" THEN 340 ELSE B=VAL(E
$):: IF B=0 THEN 340
350 ACCEPT AT(13,20)SIZE(-U)
VALIDATE("12")BEEP:E$ :: IF
E$="" THEN 350 ELSE D$="DSK"
&E$
360 ACCEPT AT(15,6)SIZE(-25)
BEEP:C$ :: GOTO 320
370 ON ERROR 580 :: C=0 :: D
ISPLAY AT(10,U):: DISPLAY AT
(20,U):"Initializing" :: OPE
N #2:D$&".",INPUT ,RELATIVE,
INTERNAL
380 INPUT #2:F$(C),I,I,I ::
T$(C)="FREE "&STR$(I)
390 DISPLAY AT(22,U):F$(C);"
";T$(C):: IF C=127 THEN 42
0 ELSE INPUT #2:F$(C+U),I,J,
J
400 IF F$(C+U)="" THEN IF C>
5 THEN 420 ELSE C=C+U :: GOT
0 390
410 I=ABS(I):: C=C+U :: IF I
=5 THEN T$(C)="Prog" :: GOTO
390 ELSE T$(C)=Y$(I)&STR$(J
):: GOTO 390
420 CLOSE #2 :: ON ERROR STO
P :: DISPLAY AT(20,U):"Print
ing" :: : : FOR B=U TO B ::
J=0 :: D=8 :: E$=""
430 PRINT #U:DS$;F$(0);E$;C$
;E$;T$(0);SS$: :
440 FOR I=J+U TO J+D :: PRIN
T #U:F$(I);TAB(12);T$(I);TAB
(18);F$(I+D);TAB(29);T$(I+D)
;TAB(35);F$(I+2;D);TAB(46);T
$(I+2;D):: NEXT I
450 J=J+24-6:(D=10):: IF C>J
THEN D=10 :: PRINT #U: : :
:: GOTO 440 ELSE PRINT #U:IN
$
```

```
460 NEXT B :: B=B-U :: FOR I
=U TO C :: T$(I),F$(I)="" ::
NEXT I :: GOTO 320
470 IF I=4 THEN RESTORE ELSE
RESTORE 170
480 FOR C=U TO 5 :: READ A$(
C):: NEXT C :: D=10 :: GOSUB
570
490 GOSUB 540 :: ON I GOTO 4
70,500,310,470,530,510,600
500 FOR C=U TO 5 :: A$(C)=""
:: NEXT C :: D=U :: GOSUB 5
70
510 FOR C=U TO 5 :: ACCEPT A
T(C+9,2)SIZE(-28)BEEP:A$(C)
: NEXT C
520 ACCEPT AT(16,19)SIZE(-2)
VALIDATE(DIGIT)BEEP:E$ :: IF
E$="" THEN 520 ELSE D=VAL(E
$):: IF D THEN 490 ELSE 520
530 DISPLAY AT(20,U):"Printi
ng" :: FOR I=U TO D :: FOR C
=U TO 5 :: PRINT #U:PI$;A$(C
):: NEXT C :: PRINT #U:IN$:
:: NEXT I :: GOTO 490
540 DISPLAY AT(20,U)BEEP:"<A
>address <P>rint labels":"<C
>ustom <M>odify defaults":
"<D>isk <Q>uit":"<W>arni
ng"
550 CALL KEY(3,I,C):: IF C<U
THEN 550 ELSE IF I=13 OR I=
32 THEN I=80
560 I=POS("ACDWPMQ"),CHR$(I),
U):: IF I THEN CALL KEY(5,C,
C):: IF B THEN DISPLAY AT(20
,U):: : : : : RETURN ELSE
RETURN ELSE 550
570 FOR C=U TO 5 :: DISPLAY
AT(C+9,U):">";A$(C):: NEXT C
:: DISPLAY AT(15,U):"How
many labels: ";D :: RETURN
580 DISPLAY AT(10,U)BEEP:D$;
" Could not be accessed" ::
ON ERROR 590 :: CLOSE #2
590 ON ERROR STOP :: RETURN
320
600 CALL CLEAR :: CLOSE #U :
: END
```

TIPS FROM THE TIGERCUB

No. 62

Tigercub Software
156 Collingwood Ave.
Columbus, OH 43213

Dec. 1990

My stock of Tigercub Software catalogs is depleted and it would not pay me to reprint it. Therefore I have released all copyrighted Tigercub programs, except the Nuts & Bolts Disks, for free distribution providing that no price or copying fee is charged. All of my Tigercub programs have been added to my TI-PD library and are cataloged, by category, in TI-PD catalog #4.

My three Nuts & Bolts disks, each containing 100 or more subprograms, have been reduced to \$5.00. I am out of printed documentation so it will be supplied on disk.

My TI-PD library now consists of 452 disks of fair-ware (by author's permission only) and public domain, all arranged by category and as full as possible, provided with loaders by full program name rather than filename, Basic programs converted to XBasic, etc. The price is just \$1.50 per disk(!), post paid if at least eight are ordered. TI-PD catalog #4 listing all titles and authors, is available for \$1 which is deductible from the first purchase.

According to Charles Good, running a program containing CALL SAY on a beige console without the speech synthesizer attached will cause a lockup.

On a black and silver console, there is no lockup but program execution can be greatly delayed. To avoid that, CALL PEEK(-28672,@) at the beginning of the program and add IF @=96 before each CALL SAY (remember that, IF causes program execution to skip to next program line if not true!), or IF @<>96 THEN to skip over the CALL SAYs.

In tips #60 I presented a routine to find the lowest power of 7 which contains six 7s in sequence. My version took 24 minutes to find the answer on my TI-99/4A. Several users tried this on a Geneve. The NUTI News of the Nittany UG, Oct 1990 reports that on a 9640 (MDDS 0.97H) with TI XBasic loaded through GPL (speed 5) it ran in 11 min. 33.86 seconds, and with MYARC Advanced Basic V2.99A loaded through GPL it ran in 4 min. 58.62 seconds!

Now, from the TI*MES of England, here is a method using a level of math beyond my comprehension that will solve the problem on an ordinary TI in 6 minutes and 17 seconds!

```
100 ! FASTER WAY John Seager
110 CALL CLEAR :: DIM ELEM(2
6):: ELEM(0)=7 :: POWER,SS=0
:: DISPLAY AT(1,1):"7 TO TH
E POWER OF"
120 ELM=SS :: SS,CARRY=0 ::
POWER=POWER+1
130 DIS$=STR$(ELEM(ELM)): F
OR I=ELM-1 TO 0 STEP -1 :: D
IS$=DIS$&RPT$("0",10-LEN(STR
$(ELEM(I))))&STR$(ELEM(I)):
NEXT I
140 DISPLAY AT(1,19):STR$(PO
WER);"=" :: :DIS$
150 FOR I=6 TO LEN(DIS$)STEP
```

```
6 :: IF SEG$(DIS$,I,1)<>"7"
THEN 190
160 FOR J=I-5 TO I :: IF SEG
$(DIS$,J,6)<>"777777" THEN 1
80 ELSE DISPLAY AT(24,1):"AN
Y KEY TO CONTINUE"
170 CALL KEY(O,K,S):: IF S=0
THEN 170 :: DISPLAY AT(24,1
):: J=I
180 NEXT J
190 NEXT I
200 ELEM(SS)=ELEM(SS)*7+CARR
Y :: IF ELEM(SS+1)=0 AND ELE
M(SS)<1.E+10 THEN 120
210 CARRY=INT(ELEM(SS)/1.E+1
0):: ELEM(SS)=ELEM(SS)-CARRY
*1.E+10
220 SS=SS+1 :: GOTO 200
```

And if you think that is fast, the Autumn '90 edition of TI*MES contains a Mini-memory program to solve the program in 2 SECONDS! And an assembly version that will search to the 10,000 power and find 52 strings of six 7's in an hour and a half!

Here's a puzzler for you. Can you figure out why that 1000-microsecond CALL SOUND is cut short?

```
100 CALL CLEAR
110 DISPLAY AT(12,1):"Filena
me? DSK" :: ACCEPT AT(12,14)
BEEP:F$
120 ON ERROR 130 :: OPEN #1:
"DSK"&F$ :: STOP
130 GOSUB 140 :: RETURN 110
140 CALL SOUND(1000,110,0,-4
,0):: DISPLAY AT(24,1):"CAN'
T OPEN FILE" :: RETURN
```

I recently programmed a diskfull of gospel songs, and in each one I used this formula to set up an array containing the frequencies for 3 octaves?

```
DIM N(36) :: F=110 :: FOR J
=1 TO 36 :: N(J)=INT(F*1.059
463094^(J-1)+.5):: NEXT J
```

At the end of each selection I put CALL INIT :: CALL LOAD(-31961,149) I don't remember where I learned that one, but it clears the screen, sets all colors and characters to default, deletes sprites, and looks for a LOAD program on DSK1.?

The LOAD program has a routine to play each song one after another, but one song crashed with a BAD VALUE error even though it had previously been OK. I found that this was the only song that actually used N(1). The value should have been 110 but it had somehow changed to 24263 which the program line multiplied by 2, therefore out of range.

I found that the routine was correctly giving N(1) a value of 110 the first time but after the CALL LOAD it always had the 24263 value. Substituting other values for 110, I found that any value was being multiplied by 220.5727273, rounded off.

Further experimentation revealed that the problem was being caused by the ^ (exponentiation sign, shift 6 on your keyboard, in case someone prints this through the Formatter!). So I wrote this little routine to experiment with:

```
100 FOR J=1 TO 10 :: PRINT
2^J :: NEXT J :: CALL INIT
:: CALL LOAD(-31961,149)
```

I saved that as DSK1.TEST and then wrote another one 100 RUN "DSK1.TEST", saved that as DSK1.LOAD, and then entered RUN "DSK1.TEST".

It printed out the proper values time after time, so I changed the 2^J to read 2^(J-1). The first time

around, the first value was 1 as it should be - the computer will consider any number to the power of 0 to have a value of 1. But, the next time around, the first value was F0.57000101 !

That was not even a valid numerical representation, so I changed the formula to $2^{(J-1)*2}$, expecting it to crash. Instead, it gave me a value of 441.140002 !

Further experimentation showed that $2^{(J-1)+1}$ gave a value shown as $1 < 1.570001$.

Changing the +1 to +10 gave $1 = 0.570001$ and to +100 gave $2 < 0.570001$!

So, poking a value of 149 into -31961 will cause any number taken to the power of zero to have a value of 220.5727273, which will be represented on screen in some apparently undocumented format - it's not even radix 100. I wonder if the fellows who built this computer could explain that!

ATTENTION all newsletter editors! If you print the above through the Formatter PLEASE transliterate the caret sign!

This one requires the TEII module and the Speech Synthesizer. Want to make the computer so mad it will fuss and fume and cuss and mutter? Run this program and answer the prompt with 1.

```
100 CALL CLEAR
110 OPEN #1:"SPEECH",OUTPUT
120 INPUT X
130 PRINT #1:"//"&STR$(X)&"
    "&STR$(X*3.17)
140 PRINT #1:"THIS IS THE SE
    CRET METHOD OF MAKING THE CO
    MPUTER SPEAK IN A WHISPER"
150 GOTO 120
```

Want to make it whisper to

you? Answer the prompt with 0 or -10.

Why did I get an INPUT ERROR when the strings in this routine got too long?

```
100 CALL CLEAR :: X=1
110 X=X*2 :: A#=RPT$("A",X):
    : B#=RPT$("B",X):: C#=RPT$("
    C",X):: D#=RPT$("D",X):: PRI
    NT A#:B#:C#:D#
120 OPEN #1:"DSK1.TEST",VARI
    ABLE 254,OUTPUT :: PRINT #1:
    A#:B#:C#:D# :: CLOSE #1
130 OPEN #1:"DSK1.TEST",INPU
    T :: INPUT #1:A#,B#,C#,D# ::
    PRINT A#:B#:C#:D# :: CLOSE
    #1 :: GOTO 110
```

Thanks to Irwin Hott for the answer to that one. I don't think it's in the books anywhere, but the TI won't input multiple records in a single INPUT if the total number of bytes is too high - less than 154 for two records to less than 144 for six records.

I still think computers should be fun, so here is a quickie for the kids, or for the kid in you -

```
100 PRINT TAB(9);"QUICK DRAW
    " :: : " How good a gunslin
    ger are":"you?": " Can you
    outdraw":"Deadeye Joe?":
110 PRINT " Watch the countd
    own from 1":"to 10.": " Wait
    for the gun...": " Then
    hit any key FAST!! - ": " -
    and HOLD IT DOWN":
120 PRINT " I got down to 20
    once - can":"you beat that?
    ": " Press any key to start
    "
```

```
130 CALL KEY(O,K,ST):: IF ST
    =0 THEN 130
140 CALL CLEAR :: S@=300 ::
    CALL CHAR(58,"009F9191919191
    9F"):: CALL CHAR(42,"0000FCF
```

```
E171F0707")
150 CALL KEY(O,K,ST):: IF ST
    =-1 THEN 150
160 CALL CLEAR :: FOR M=1 TO
    10 :: CALL HCHAR(12,16,M+48
    ):: FOR N=1 TO 100
170 NEXT N :: CALL KEY(O,F,X
    ):: IF F=70 THEN 330
180 NEXT M :: CALL CLEAR ::
    FOR J=1 TO 500
190 NEXT J :: IF F=70 THEN 3
    30
200 CALL KEY(O,K,ST):: IF ST
    <>0 THEN 330
210 CALL HCHAR(12,16,42):: F
    OR D=1 TO S@
220 NEXT D :: CALL KEY(O,Z,X
    ):: IF X=0 THEN 240
230 GOTO 270
240 CALL CLEAR :: PRINT :: P
    RINT "YOU'RE DEAD!"
250 FOR D=1 TO 200
260 NEXT D :: GOTO 160
270 PRINT "OUCH!" :: IF S@<5
    1 THEN 290
280 S@=S@-50 :: GOTO 320
290 IF S@<31 THEN 310
300 S@=S@-5 :: GOTO 320
310 S@=S@-1
320 PRINT S@ :: GOTO 250
330 PRINT "YOU CHEATED!" ::
    GOTO 150
```

I always wondered about those recipe programs. Does the cook lug the computer out to the kitchen to read the screen, or use a printer to make a hardcopy of a file that was keyed in from a hardcopy in the first place?

Anyway, some of those programs do convert quantities for different servings, so here is a little program to do that. It provides input and output in fractions instead of decimals, because that is the way recipes are written.

```
100 DISPLAY AT(3,6)ERASE ALL
    : "RECIPE CONVERTER"
110 DISPLAY AT(6,1):"Enter f
```

ractional quantities separated by a space from whole quantities."

120 DISPLAY AT(9,1):"For instance, to enter three and one-half, type 3 1/2"

130 DISPLAY AT(12,1):"Results will be rounded to the nearest 8th."

140 DISPLAY AT(24,7):"press any key" :: DISPLAY AT(24,7) : "PRESS ANY KEY" :: CALL KEY(O,K,S) : IF S=0 THEN 140

150 DISPLAY AT(12,1)ERASE ALL : "TURN PRINTER ON!"

160 OPEN #1:"PIO" :: PRINT #1:CHR\$(27):"@" :: CALL CLEAR

170 DISPLAY AT(5,1):"Name of recipe?" :: ACCEPT AT(7,1) : M# :: PRINT #1:M#:"":

180 DISPLAY AT(3,1)ERASE ALL : "Recipe is for how many servings?" :: ACCEPT AT(4,11)VALIDATE(DIGIT)BEEP:R

190 DISPLAY AT(6,1):"You want to cook how many serving s?" :: ACCEPT AT(7,11)VALIDATE(NUMERIC):S :: X=S/R

200 DISPLAY AT(10,1):"Name of ingredient? (just enter if finished)" :: ACCEPT AT(13,1)BEEP:A# :: IF A#="" THEN STOP

210 DISPLAY AT(15,1):"Unit of measure?" :: ACCEPT AT(17,1)BEEP:M#

220 ON ERROR 310 :: DISPLAY AT(19,1):"Quantity in recipe ?" :: ACCEPT AT(21,1)BEEP:AX # :: A=VAL(AX#)

230 Q=X*A :: J=INT(Q):: P=Q-J :: IF P=0 THEN X#=STR\$(J) : Y#="" :: GOTO 290

240 IF J=0 AND P<=.0625 THEN X#="" :: Y#="" : "less than 1/16" :: GOTO 290 ELSE IF P<=.06

25 THEN X#=STR\$(J):: Y#="" : GOTO 290

250 IF P>.9375 THEN X#=STR\$(J+1):: Y#="" :: GOTO 290

260 DATA .8125,7/8,.6875,3/4,.5625,5/8,.4375,1/2,.3125,3/8,.1875,1/4,.0625,1/8

270 RESTORE 260

```

280 READ M,N# :: IF P>M THEN
Y#=#N# :: X#=#STR$(J)ELSE 280
290 IF J<1 THEN X#=#
300 PRINT #1:A#&" "&X#&" "&Y
#&" "&M# :: GOTO 200
310 P=POS(AX#," ",1):: Q=POS
(AX#,"/",1):: IF Q=0 THEN 34
0
320 ON ERROR 340 :: IF P=0 T
HEN A=0 ELSE A=VAL(SEG$(AX#,
1,P-1))
330 B=VAL(SEG$(AX#,P+1,Q-1-P
)):: C=VAL(SEG$(AX#,Q+1,255
)):: A=A+B/C :: RETURN 230
340 DISPLAY AT(24,1):"OOPS!
TRY AGAIN" :: CALL SOUND(1,1
10,0,-4,0):: RETURN 220

```

And here is an oldie - a utility to get the bugs out of your programs.

```

100 ! MOSQUITO #2 by Jim Pet
erson from a PEEK by Crag Mi
ller
110 CALL CLEAR :: CALL SPRIT
E(#1,42,2,100,100)
115 DISPLAY AT(22,1):"Don't
let the mosquito get":"out o
f the TV!":"Press any key ->
WICK!"
120 RANDOMIZE :: CALL PEEK(-
31808,A,B):: CALL MOTION(#1,

```

```

A-128,B-128):: CALL KEY(0,K,
S):: IF S=0 THEN 120
130 CALL CLEAR :: CALL COLOR
(1,2,8):: CALL SCREEN(2):: C
ALL CHAR(32,"FF888888FF88888
8"):: GOTO 120
Long live the TI-99/4A!

```

Jim Peterson
The Tiger Cub
~~~~~

TI BITS \* Number 24  
By Jim Swedlow

[This article originally appeared in the User Group of Orange County, California ROM]

#### EA DISK ERROR CODES

Some programs (like Archiver) display a number when they encounter a disk error (something like "IO ERROR #7"). The numbers by themselves are of little use. Here is what they mean:

- 0 UNKNOWN DEVICE - Could not find the specified drive.
- 1 WRITE PROTECTED - The disk is write protected.
- 2 BAD OPEN ATTRIBUTE - One or more OPEN options were illegal or didn't match the file characteristics.
- 3 ILLEGAL OPERATION - The book says that this code should not be generated!
- 4 OUT OF SPACE - The disk is full or you are trying to open more files than are allowed (127).
- 5 END OF FILE - You are trying to read beyond the end of the file.
- 6 DEVICE ERROR - The disk is not initialized, the disk is damaged, the disk drive is broken (oh no!), the drive door is open, etc.
- 7 FILE ERROR - The file doesn't exist or you are trying to read a BASIC file as if it were data.

These are the same as the second digit in the BASIC disk error codes.

#### TI LIVES

The December, 1988 issue of PC Computing (a magazine normally dedicated to MS DOS machines) has an article entitled "Gone But Not Forgotten". There is coverage of most of the orphans (TI, Osborne, Eagle, PCjr, etc). By the time you read this, you won't be able to buy that issue, but it might be worth checking out in your library. A couple of interesting quotes:

"The 99/4A and PCjr were early experiments in the home computing market. They weren't nearly as fast or powerful as other computers of their time, yet in many homes they continue to fulfill the role that visionaries once predicted for them: they've evolved from somber, cold pieces of machinery to tools that are useful and fun."

"Fans of the TI 99/4A are legion, and they form a network of users that's lively. TI user groups number 300 with the Chicago group the largest at nearly 550 members. Other TI organizations are small but spirited. The 14 member north New Jersey group works with other New York area computer groups to sponsor the TI Computer Fest which each year draws about 300 people to attend workshops and hear speakers."

[ edited by Stephen Shaw].

## HINTS, TIPS & ANSWERS (HTA)

### TI-WRITER (or equivalent)

(1F refers to a tip for the FORMATTER while 2E refers to a tip for the Editor section. Equivalents include Funlweb).

- 1F Outdenting- This is the reverse of "indenting". It will allow first line of a paragraph to be start farther to the left than the remaining lines in the paragraph.  
eg - ".LM12;IN7;RM71" causes the first line to start at column 7 and subsequent lines to begin at column 12.
- 2E When using REPLACE STRING you should use word wrap OFF if the document lacks carriage returns or you will end up with one huge paragraph. With word wrap ON, Replace String will reformat from the amended word to the end of the paragraph.
- 3E This is for use in Funnelweb Text Editor mode (not TI Writer) Want to convert the character case (Upper to Lower; Lower to Upper)?  
i) Upper to Lower - depress CTRL and "." (period)  
ii) Lower to Upper - depress CTRL and ";" (semi-colon)  
By keeping the keys depressed the autorepeat function will take affect and every character the cursor passes over will be converted (you only need hold the , or ; once auto repeat takes over- you can let go of CTRL.).
- 4E To save part of a document insert in front of the filename the first line number you wished saved then either a comma or a space and then the last line number you wished saved.  
eg - 5 30 DSK1.MINUTES or 5 30 DSK1.MINUTES - will save the lines 5 to 30, inclusive, onto disk drive #1 under the name MINUTES.
- 5E To "get rid" of the line numbers on the left of the screen press FCTN 0 (zero). To get them back press FCTN 0 again.
- 6E If you erase a line in error, press CTRL 1 (known as "OOPS!") and your line will be restored. Note: This will work ONLY if no other keys were pressed between the erasing (or editing) and
- 7E When using the "Search" command remember that the search is only from the point that the cursor is located. Therefore to search the total document the cursor must be on line 1 before you go to the Command mode.
- 8E To backspace beyond the left margin press CTRL Y . This will temporarily disable the left margin. It will also disable the right margin in latest Funlweb ONLY. In both cases the cursor should be next to the margin.
- 9F When using the Header or Footer command with the page number it is possible to have NO value printed (such as for the introduction, etc) by using the .PA format command with a value of zero. The page numbering will begin on the following page. A .PA at the end of each page will delay the numbering further.

- 10E TI-Writer can save a file in other than the normal D/V80 format by using the PF command and either putting a "F" in front of the filename (ie F DSK1.MYFILE) or by putting a "C" in front of the filename. "F" causes a file to be created in a Display/Fixed 80 format. "C" strips any control characters from the file as it is sent.
- 11E TI-Writer can be used as a database. Each line must be a record and set up exactly the same. For example if the data was names, addresses and phone numbers then all names must start in the same column; all addresses must start in the same column and all phone numbers must start in the same column. There can be no lines which are blank or which have other type of information on them (ie - titles). Then using the program SORT UTILITY (by D R Romer & J Clulow) you can sort this file. Once sorted, which is done very quickly, titles can be added if you are printing it out.
- 12E There are CTRL keys equivalents to most FCTN keys, plus a few others. For example to tab to the right you can go FCTN 7 or CTRL W (Funlweb also has CTRL Z) ; to tab back (to the left) you can go CTRL T (there is no equivalent FCTN key).
- 13E If you must go to the bottom of your text (and it is very long), instead of paging down simply go to the Command Line and press "S" for "show line" and at the prompt "enter line number" just type "E" and press ENTER. "E" is a valid line number for the last line (end) of a document. This feature is active in all the commands requiring you to enter a line number.
- 14F If you wish to prevent the form feed at the end of printing when using the Formatter then make the last line of your text ".PL 1". This will suppress the form feed, but note: do not forget to reset PL if you have another document to print.
- 15F You can string the formatter commands on the same line separated with a semicolon. eg - LM 10;RM 70;IN +5;FI;AD
- 16F If you are having problems with formatter commands, make sure they are UPPERCASE letters.
- 17E To avoid a BUFFER FULL notice you just save the files as it gets larger, then use the SD command to see the file size. The BUFFER is usually becoming full at 92 sectors. [due to use of Run Length Encoding in the buffer area, a full buffer may occur at very different file sizes depending on nature of text. Stephen].
- 18F When using the .CE command you MUST also use the .LM and .RM commands because .CE centres the text between the SET margins, not necessarily the middle of your paper.
- 19F The Formatter also ensures that you have two spaces after each period. To control this when you are typing such things as "Mrs. E Smith" or "1023 N. Queen Street" then use the "^" sign in place of the space after the period.  
eg - Mrs.^E Smith; 1023 N.^Queen Street
- 20F If you must have a dot in column one of your text, transliterate it. ".TL 124:46" will allow FCTN A to print out a period.



21. To create a file with out line feeds yet Formatted, do the following:
- use the FORMATTER to Print the text to disk
  - go back to EDITOR and do a Print File (PF) replacing PIO with C DSKn.filename.
22. If you wish to include a program listing in your document instead of retyping it into TI-Writer just LIST the program to the diskette using the following command LIST "DSKn.filename". This will save the program in DISPLAY/80 format which allows it to be read by TI-Writer. You can now load this file into TI-Writer and place the carriage return character at the end of each program line.
- 23E If you wish to place a Carriage Return at the end of a line of text (an line without the return will usually occur when you have inserted blank lines in the text and then put text on them) then place the cursor at the end of the text and press CTRL B. This will place a Carriage Return where you want it and insert a blank line below. If this line is not wanted you can delete it with FCTN 3.
- 24E You can get a print out of your file WITH LINE NUMBERS when printing out of the Editor mode by placing an "L" and a space before the printer name in the command instruction.  
eg - L PIO  
This will eliminate the last 6 characters at the end of each line (#75 to #80) therefore keep your line lengths to a maximum of 74.
- 25E Did you know that you can type anything you want after a carriage return ON THE SAME LINE and it won't print out; but it will SAVE. This is great for text notes for screen reading.
- 26E When typing up a document which uses certain long words or phrases frequently then a time saver (and also added insurance against typing errors) is to type the words in shortform or initials (ie - TI-ARTIST could become TIA; Ottawa TI-99/4A Users Group could become OTIUG). When you are finished with the document use the Replace String function (RS) to change the words back to the full spelling (eg - /TIA/TI-ARTIST/). Care must be taken in three forms when using this:
- the search only begins from the spot that the cursor is on so to do the whole document insure that the cursor is on line 1 before starting.
  - the search will locate all occurrences of the string. Therefore if the string searched for is "at" it will find word "at" and also the "at" in "cat" and "that", etc. So before telling the machines to "Change all occurrences automatically be sure this program can't arise. If you are not confident of this it is best to walk through and change each separately as it is found.
  - as a reformatting will be done wherever a change is made it would be wise to review the document after to be sure that it is still formatted correctly.
- 27E When using the FIND STRING command you can specify which column range to search.  
eg - 5 15 /text/ will look for the string "text" in the columns 5 through to 15 inclusive.  
[also available with Replace String- sjs]

- 28F If your printer does not have a slashed zero and you want to print it out that way use the following Transliteration:  
.TL 48:48,8,47  
This will cause the normal zero (48) to be printed; then backspace (B); and then a printing of a slash (47).
- 29E Two files can be loaded into the Editor (assuming the total size is not too large for memory) by loading in the first file then doing a LF and entering E DSKn.YYY (where n=disk # and YYY=second file). This will load the second file after the end of the first file.
- 30E If you don't like the windowing when using the 80 column format then set the margins for 0 and 40 and turn off the line numbers (FCTN 0). When you are finished reset the left and right margins to what you desire and reformat each paragraph.
- 31E You can merge sections of a second file into the current document by the following entry using the LF command:  
25 7 12 DSK1.YYY - This will load lines 7 to 12 (inclusive) from file YYY to the current document after line 25.
- 32E If you are using FUNNELWEB 4.1 or greater after you have the directory on the screen (using the SD command) you will be able to see how many lines are in a file by marking the file and then requesting (V)iew. The line count will be shown at the bottom of the screen as you read through the document.



- ★ SUPPORTING LOCAL CHARITIES ★  
AA Signposted from Junction 14 on M6  
Hardware - Software - Books - TV & Video - Components  
Media - Discs - Paper - Ribbons - Radio - Printers - Monitors  
and more  
All Micros including -  
Amstrad - Sinclair - Einstein - Oric - QL - Dragon - Z88  
Atari 8-bit - Commodore - SAM - IBM PC - Amiga - ST  
★ FREE PARKING ★ T1997/A  
★ FREE COURTESY BUS FROM STAFFORD BR STATION ★  
★ BAR and REFRESHMENTS ★  
★ ELECTRONIC BRING & BUY SALE ★  
Entrance on the day - £2 Adult - £1 Children  
Advance tickets by post just £1 from the organisers -  
Sharward Services, Upland Centre, 2 Upland Road, Ipswich IP4 5BT  
Tel 0473 272002 - Fax 0473 272008

## MACHINE CODE TUTORIAL PART FIVE.

### SPIRITED SPRITES

by Mack Mc Cormick

Definition: Any shape or color. Can occupy screen positions independent of any character already present. Once set into motion, can move independently of direct program control. You can magnify or make double size.

How they can be used: Up to 32 sprites on the screen at any one time. Can be used in GRAPHICS and MULTICOLOR modes. Also can be used in BIT MAP mode but not the automatic motion feature (according to TD) Sprites cannot be used in the TEXT mode.

There are three tables which contain all the information needed to use sprites:

1. SPRITE ATTRIBUTE TABLE
  - a. Sprite Position
  - b. Sprite Color
2. SPRITE DESCRIPTOR TABLE
  - a. Sprite Pattern Identifier
  - b. Specify magnified or double sized sprites.
3. SPRITE MOTION TABLE
  - a. Define X and Y velocities of Sprites.

#### DEFAULT LOCATIONS OF SPRITE TABLES

| Table                   | Table Begins at this VDP address |
|-------------------------|----------------------------------|
| SPRITE ATTRIBUTE TABLE  | >0300                            |
| SPRITE DESCRIPTOR TABLE | >0400                            |
| SPRITE MOTION TABLE     | >0780                            |

Sprites are numbered from 0 to Here's how the screen is set up. 31. Here's how the screen is defined for Sprites: Columns are labeled starting from the left from 0 to 255 (>00 to >FF). Rows are numbered from top left, the first row is numbered 256 (>100), followed by the numbers 0 to 190 (>0 to >BE). Each screen location defined in this manner is referred to as a pixel. A pixel is the smallest area of the screen you can turn on or off. Heres the way it looks:

Pixel 1 is in row >100 column >02.

P4 is in row >BE column >01.

Here are the formulas to convert row and column locations to pixel locations:

#### GRAPHIC TO PIXEL CONVERSIONS

GRAPHIC ROW TO PIXEL ROW  $GR*8-7=PR$   
 GRAPHIC COLUMN TO PIXEL COLUMN  $GC*8-7=PC$

PIXEL ROW TO GRAPHIC ROW  $INT((PR+7)/8)=GR$   
 PIXEL COLUMN TO GRAPHIC COLUMN  $INT((PC+7)/8)=GC$

### SPRITE ATTRIBUTE TABLE

Begins at VDP >0300 by default. Contains the present position of sprites and their colors. Each sprite takes up four bytes in the table. The first byte is the row or Y position of the sprite. The second byte is the column or X position. The third byte references the pattern of the sprite as to where it is located in the Sprite Descriptor Table. The fourth byte is the early clock attribute and also codes for the color of the sprite.

When the computer moves sprites it updates the information in the sprite attribute table. The more sprites it has to update the longer it takes to execute the program. To shorten this time place a value of >D0 as the Y location of the lowest numbered non-moving sprite. Always let the final unused sprite be undefined by specifying the Y location as >D0.

The third byte references a pattern in the Pattern Descriptor Table. Can range from >00 to >FF. For example if the third byte contained >80 it would point to >0400 through >0407 in the Sprite Descriptor Table.

The fourth byte controls the early clock and color. The first four bits control the early clock. If the last bit (3) is reset to zero the early clock is off and the location of the sprite is said to be it's upper left hand corner. This means the sprite will fade in and out on the right hand side of the screen. If bit 3 is on the sprites location is shifted 32 pixels to the left. The sprite can then fade in and out from the left side of the screen.

Bits 4-7 of byte four contain the color. Same as other VDP colors 0 to >F.

Heres an example Sprite Attributes:

```

                Sprite 0   Sprite 1
                | |       | |
SAL  DATA  >3356,>8001,>A828,>B10F,>D0 -- 3rd Sprite Undefined
                // //
                Y X / color
                pattern
    
```

### SPRITE DESCRIPTOR TABLE

Just like the pattern descriptor table for characters. Usually begins at >0400. Addresses >0400 through >0407 are defined as sprite pattern >80.

You can also make sprites magnified or double sized by writing a value to the two least significant bits of VDP register 1.

### SPRITE MOTION TABLE

Describes the X and Y velocities of each sprite. This table begins at >0780. Before a sprite can be placed into motion several conditions must be met. Your program must allow interrupts using LIM1 2 but before accessing VDP RAM you must disable interrupts with a LIM1 0.

You must indicate how many sprites will be in motion by placing a value at CPU address >837A. For example if sprites 2, 5, and 7 are in motion you must place an >8 at address >837A which will allow motion of 0 through 7. A description of the motion must be placed in the Sprite Motion Table. Each sprite takes up four bytes in the table. The first byte is the Y velocity, the second byte is the X velocity. The third and fourth bytes are used by the interrupt routines, just be sure you leave space for them. The following are allowed as values for X and Y velocities:

A value of >01 will cause the sprite to move one pixel every 16 VDP interrupts. About once every 16/60 of a second.

A thought: Have you ever seen a screen dump program that would dump sprites? It could be done by obtaining their location and pattern and converting to printer bit graphics. Have fun!

\*\*\*\*\*

```
*
*          CALL SPRITE
* PROGRAM PLACES A HELICOPTER
* SPRITE IN MOTION BY ENABLING
* INTERRUPTS. PRESS ANY
* KEY TO ALTER MAGNIFICATION
* BY MACK MCCORMICK
*****
```

```
DEF START
REF VMBW,VWTR,KSCAN
```

```
*
HELI DATA >007F,>0000,>0107,>0E0E HELICOPTER PATTERN DESCRIPTION
      DATA >1EBE,>FFBF,>0F07,>020F BLOCK 2
      DATA >00FF,>80B0,>C0FB,>04C2 3
      DATA >DACA,>FEFC,>FBEO,>40FB 4
SDATA DATA >70B0,>800B INITIAL SPRITE DATA
      DATA >D000 >D0 PREVENTS GHOST SPRITES
```

```
*
SPEED DATA >0A0F,>0000 SPRITE SPEED FOR AUTO MOTION
STATUS EQU >837C GPL STATUS BYTE
VDP DATA >01E0 INITIAL VALUE OF VDP REGISTER 1
MYREG EQU >8300 MYREG IN 16 BIT HIGH SPEED AREA OF MEMORY
```

```
*
START LWPI MYREG
      CLR @>8375 KEYBOARD DEVICE = 0. SCAN ALL.
      MOV @VDP,R6
      LI R0,>0400 LOAD (BASE ADDRESS OF SPRITE DESCRIPTOR TABLE)
      LI R1,HELI SPRITE
      LI R2,32 DESCRIPTOR
      BLWP @VMBW TABLE
```

```
*
      LI R0,>0300 LOAD (BASE ADDRESS OF SPRITE ATTRIBUTE TABLE)
      LI R1,SDATA SPRITE
      LI R2,6 ATTRIBUTE
      BLWP @VMBW TABLE

*
      LI R0,>07B0 LOAD (BASE ADDRESS OF SPRITE MOTION TABLE)
      LI R1,SPEED SPRITE
      LI R2,4 MOTION
      BLWP @VMBW TABLE

*
      LI R1,>0100
      MOV B R1,@>837A ONE SPRITE IN MOTION

*
LOOP CLR @STATUS
      BLWP @KSCAN
      MOV B @STATUS,@STATUS HAS KEY BEEN PRESSED?
      LIMI 2 ENABLE INTERRUPTS FOR AUTO MOTION
      LIMI 0 DISABLE INTERRUPTS SO VDP IS NOT AFFECTED ON READ/WR
      JEQ LOOP

*
CHECK INC R6 R6 IS USED AS A COUNTER TO KEEP
      CI R6,>01E4 TRACK OF WHICH MAGNIFICATION
      JLT G0 LEVEL (1 TO 4) WE ARE ON.
      MOV @VDP,R6

*
G0 MOV R6,R0 LOAD R0 WITH DATA TO LOAD INTO VDP R1
      BLWP @VWTR CHANGE THE VDP REGISTER
      B @LOOP
      END
```

\* FOR EXTRA PRACTICE ADD A ROUTINE THAT SHOWS THE X AND Y POSITION OF THE SPRITE  
 \* ON THE SCREEN AS IT MOVES. HINT: Y LOCATION IS 1ST BYTE IN SPRITE ATTRIBUTE  
 \* LIST. X SECOND BYTE. READ THEM, CONVERT TO ASCII DECIMAL AND REDISPLAY WITH  
 \* APPROPRIATE TEXT. WHO'LL BE FIRST?

# FRactal...

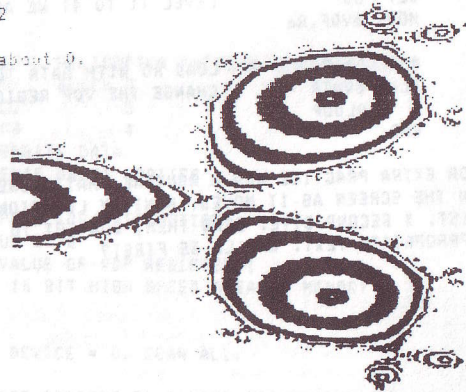
Here is a truly fractal program, which, using The Missing Link and Extended Basic ONLY takes about 12 hours or so to plot a graphic of 120 x 200 pixels! The total graphic is vertically reflected about 0, so if 0 is at screen centre you can plot top and bottom parts at the same time to half plotting time...

This program is all about "attraction basins" which I do not understand, and while the program can probably be modified for other functions, I do not know how! What I know is that it plots a truly fractal shape which we can zoom in on.

```

100 ! TO DRAW AN ATTRACTION BASIN USING NEWTON FORMULA ON
110 ! F(z)=z^(-3)-1
120 !
130 !
140 ! J C TOPHAM ! Fractal Report 13 ! Feb 1991
150 ! for titttl by s shaw ! march 1991
160 !
170 !
180 !
190 !
200 WIDE=242 :: HEIGHT=190
210 !
220 !
230 CALL LINK("CLEAR")
239 ! actual pixels plotted:
240 ACROSS=200 :: DOWN=120
246 ! area to be plotted
247 ! whole image is within X=-2 to +2
248 ! Y= -1.75 to +1.75
249 ! Y is vertical and is reflected about 0.
250 XMIN=-1.1 :: XMAX=0.9
260 YMIN=-1.70 :: YMAX=0.30
270 !
280 !
290 RY(1)=-SQR(3)/2
300 RY(2)=SQR(3)/2
310 RY(3)=0
320 !
330 DX=(XMAX-XMIN)/ACROSS
340 DY=(YMAX-YMIN)/DOWN
350 XPOS=INT((WIDE-ACROSS)/2)
360 YPOS=INT((HEIGHT-DOWN)/2)
370 !
380 FOR YP=0 TO DOWN
390 FOR XP=0 TO ACROSS
400 CALL LINK("PRINT",181,12,STR$(YP)&" ":"&STR$(XP))
410 YN=YMIN+YP*DY
420 XN=XMIN+XP*DX
430 FOR ITER=1 TO 30
440 GOSUB 650
450 XM=(A*C+B*D)/9
460 YM=(B*C-A*D)/9
470 IF XM*XM+YM*YM>1000 THEN 620
480 !
490 !
500 IF ABS(YN-YM)>1E-2 THEN 590
    --->continued---

```



```

510 IF ABS(XN-XM)>1E-2 THEN 580
520 FOR I=1 TO 3
530 IF ABS(RY(I)-YM)>1E-2 THEN 560
540 CALL LINK("PRINT",1,200,STR$(ITER)&" "):: IF ITER/2<>INT(ITER/2) THEN
ITER=30 :: GOTO 570
550 CALL LINK("PIXEL",YPOS+DOWN-YP,XPOS+XP):: ITER=300
560 !
570 NEXT I
580 !
590 !
600 XN=XM :: YN=YM
610 NEXT ITER
620 NEXT XP
630 NEXT YP
640 END
650 ! INVERT Z3
655 X2=XN*XN :: X3=X2*XN :: X4=X2*X2 :: V2=YN*YN :: Y3=Y2*YN :: Y4=Y2*Y2
660 A=4*XN-X4+6*X2*Y2-Y4
670 B=4*YN-4*X3*YN+4*XN*Y3
680 C=3 :: D=0
690 RETURN

```



## MANDELBROT PLOT

This is EXCEEDINGLY SLOW if you want a detailed plot! However, a minimum detailed plot over a limited screen area is not TOO slow! Maximum detail and you could be looking at SEVERAL DAYS to complete a screenful.

THIS PROGRAM DOES WORK! and cries out for some machine code! Some means of saving the picture, to disk or printer, would be required- that's perhaps the hardest bit!

User should input area to be plotted, and possibly be given a choice of what method is to be used for deciding if a pixel is to be plotted or not- see comments in listing! - and the degree of inner detail required!

```

100 ! MANDELBROT PLOTTER
110 ! VERY SLOW!
120 ! FOR EX BAS + THE MISSING LINK
130 !
140 CALL LINK("CLEAR")
150 !
160 ! DESIGN LIES IN AREA
170 ! PMIN -2.25 PMAX 0.75
180 ! QMIN -1.50 QMAX +1.50
190 !
200 ! use smaller area for more detail- ensure
there IS detail there though!
210 !
220 ! Concentrate on areas very close to central creature!
230 !
240 A=240 :: B=180 ! maximum values
250 !
260 ! A=width plotted B=height plotted in pixels
270 !
280 ! M= number of iterations per point. Needs to be enough to reach
CMAX at centre of beast.
290 ! 100 is often used but lower values can usually be safely used.
    ----> continued ---->

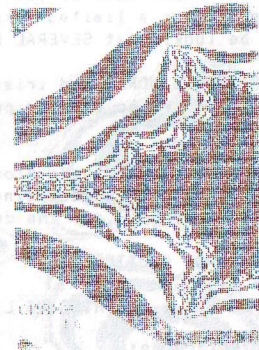
```



```

300 !
310 !
320 A=160 ! A=width
330 B=150 ! B=height in pixels
340 M=71 ! MAX ITERATIONS
350 ! P=REAL Q=IMAG
360 PMIN=-.250 ! DEFINE AREA
370 PMAX=-.20 ! TO BE
380 QMIN=-0.83 ! DRAWN
390 QMAX=-0.79 !
400 !
410 ! CMAX must be a power of 2, eg 2,4,8,16,32,64,128...
420 ! The higher the value the more detail close to the beast.
430 ! 16 gives minimum acceptable detail, 32 is reasonable for
an odd/even test, while 64 and higher give maybe too much detail.
440 ! If using 64 or higher you must use a logarithmic scale with greater gaps
as K gets larger... eg
450 ! 64...pixel on.          53 to 63...pixel off      K= 44 to 52...pixel
on          K= 36 to 43...pixel off      K= 29 to 35...pixel on...
460 !
470 ! or use          IF INT(K^.8)/2=INT(INT(K^.8)/2) THEN pixel on...
or some other fractional power
480 ! note that processing time really does get longer as CMAX increases!!!!!!
490 !
500 !
510 CMAX=64 ! DETAIL, 8,16,32,64,128,256 ETC
520 !
530 DP=(PMAX-PMIN)/(A-1)
540 DQ=(QMAX-QMIN)/(B-1)
550 FOR NP=1 TO A
560 FOR ND=1 TO B
570 P=PMIN+NP*DP :: Q=QMIN+ND*DQ :: K,X,Y=0
580 ! LOOP
590 !
600 !
610 XN=X*X-Y*Y+P
620 Y=2*X*Y+Q
630 X=XN :: K=K+1
640 !
650 IF X*X+Y*Y>M THEN 690
660 IF K=CMAX THEN 690
670 GOTO 580
680 !
690 IF K/2=INT(K/2) THEN CALL LINK("PIXEL",ND,NP)! SEE COMMENTS ON CMAX ABOVE!
700 !
710 CALL LINK("PRINT",170,200,STR$(ND)&STR$(K)&" ")
720 NEXT ND
730 NEXT NP
740 CALL LINK("SAVER","DSK2.PIC2")
750 GOTO 750

```



**THE GOLDEN RATIO**

1 ! THE GOLDEN RATIO 1: 1.618...

2 ! removing a square from a rectangle with sides in this ratio leaves a smaller rectangle which also has sides in the Golden Ratio

3 ! -points dividing sides lie on a logarithmic spiral which can be found in shells, and in art works by da Vinci, Dali- even in the Parthenon.

4 ! the Parthenon.

5 ! Fractal in nature- it keeps getting smaller or larger depending on which way you go...

```

6 ! program written by Ashley Tilling for JBM103 converted by
y S Shaw for TML.
7 !
8 ! Due to our not having square pixels the rectangle on screen is
NOT in the golden ratio as viewed. Ah well...
9 !
99 ! remove ! from line 100 for use with JBM103.
100 ! call load(-31890,56,0) :: CALL LOAD(-31964,56,0)
110 AX=12 :: AY=8
120 CL=200 ! side length
130 CALL LINK("CLEAR")!for jbm103 CALL LINK("SCR2") also.
140 RL=INT(CL/1.618)
150 FOR I=0 TO 1 :: FOR J=0 TO 1
160 CALL LINK("LINE",AX+RL*I,AY+CL*I,AX+RL*J,AY-CL*(J=0))
162 ! CALL LINK("LIGNE",16,AX+RL*I,AY+CL*I,AX+RL*J,AY-CL*(J=0))
170 NEXT J :: NEXT I
180 FOR K=1 TO 3
190 M=CL-RL :: N=INT(CL*.236):: P=INT(CL*.146)
195 CALL LINK(A,X,AY,AX,AY)
200 CALL LINK(RL,M,1,M)
210 CALL LINK(N,1,N,M)
220 CALL LINK(1,N,N,N)
230 CALL LINK(P,M,P,N)
240 AX=AX+P :: AY=AY+N :: CL=P :: RL=INT(CL/1.618)
250 NEXT K
260 GOTO 260
270 SUB LINE(A,B,C,D)
280 REM ! LINES for JBM103 INSTEAD of CALL LINK("LINE".. and CALL LINK("PIXEL"..
.
290 IF (A=C)*(B=D) THEN AX=A :: AY=B :: SUBEXIT
300 CALL LINK("LINE",AX+A,AY+B,AX+C,AY+D)
310 ! CALL LINK("LIGNE",16,AX+A,AY+B,AX+X,AY+D)
320 IF B=D THEN 350
330 L=ABS(B-D)-1
340 ST=-PI*(B>D):: FI=PI/2+PI*(D>B):: GOTO 370
350 L=ABS(A-C)
360 ST=-PI/2-PI*(A>C):: FI=PI*(C>A)
370 IF L<3 THEN SUBEXIT
380 FOR J=ST TO FI STEP -1/L
390 Y=AY+D+INT(.5+L*SIN(J))
400 X=AX+C+INT(.5+L*COS(J))
410 CALL LINK("PIXEL",X,Y)
420 ! CALL LINK("POINT",16,X,Y)
430 NEXT J
440 SUBEND
450 END

```



**POLYNOMIAL EQUATIONS...**

1 ! Newtons method for solving polynomial equations.

2 ! eq find z when  $4Z^6-2Z^3+z-1=0$

3 ! sorry about the input format, you just need to enter the multiplicand for each power of z and its sign if negative (6z or -6z)

4 ! enter the default zero if that power is not in the equation.

5 ! one numeric answer will usually be given but some formulae may have more than one answer- try varying the seed

6 ! in line 290, value of X

7 ! to produce different answers.

8 ! a few equations MAY not be solvable with this program.

9 ! amend input method if using multipliers over +99 or under -9 or if you wish to use higher powers (remember to DIM the array).

----> continued ---->

```

10 !
11 !
100 CALL CLEAR :: V$="0123456789-+"
110 DISPLAY AT(1,1):"NEWTONS METHOD TO SOLVE      Polynomial Equations"
120 DISPLAY AT(3,4):" Dr M Ecker 1987.          S Shaw for TI 1991"
130 DISPLAY AT(6,1):"Your equation is in the form 5Z^5 + 0Z^4 -2Z^3 +0Z^2
Z +9 =0"
140 DISPLAY AT(12,1):" 0 Z^9 0 Z^8 0 Z^7 0 Z^6"
150 DISPLAY AT(13,1):" 0 Z^5 0 Z^4 0 Z^3 0 Z^2"
160 DISPLAY AT(14,1):" 0 Z 0 = 0"
170 ACCEPT AT(12,2)SIZE(-2)VALIDATE(V$):A(9)
180 ACCEPT AT(12,8)SIZE(-2)VALIDATE(V$):A(8)
190 ACCEPT AT(12,14)SIZE(-2)VALIDATE(V$):A(7)
200 ACCEPT AT(12,20)SIZE(-2)VALIDATE(V$):A(6)
210 ACCEPT AT(13,2)SIZE(-2)VALIDATE(V$):A(5)
220 ACCEPT AT(13,8)SIZE(-2)VALIDATE(V$):A(4)
230 ACCEPT AT(13,14)SIZE(-2)VALIDATE(V$):A(3)
240 ACCEPT AT(13,20)SIZE(-2)VALIDATE(V$):A(2)
250 ACCEPT AT(14,2)SIZE(-2)VALIDATE(V$):A(1)
260 ACCEPT AT(14,6)SIZE(-2)VALIDATE(V$):A(0)
270 FOR I=0 TO 9 :: IF A(I)<>0 THEN N=I
280 NEXT I
290 TLD,TL=.000000000001 :: X=.800 ! INITIAL GUESS
300 FOR K=0 TO N
310 Y=Y+A(K)*X^K
320 NEXT K
330 IF ABS(Y-0)<TL THEN 410
340 FOR K=1 TO N
350 YD=YD+K*A(K)*X^(K-1)
360 NEXT K
370 IF YD=0 THEN X=X+.01 :: GOTO 400
380 X=X-Y/YD
390 DISPLAY AT(22,1):"Next iterate:"X;"
400 Y,YD=0 :: GOTO 300
410 DISPLAY AT(20,1):"***DONE***:"<any key for another>"
420 DISPLAY AT(16,3):"Z=";X
430 CALL HCHAR(18,6,42,6)
440 CALL KEY(5,P,B)
450 IF B=1 THEN RUN
460 CALL HCHAR(18,6,32,6)
470 CALL HCHAR(18,6,45,6)
480 GOTO 430
491 ! Recreational & Educational Computing
492 ! 909 Violet Terrace
493 ! Clarks Summit PA
494 ! USA 18411
495 ! 1991 sub Europe US$36 or
496 ! send $12 up for sample copies
500 END

```



I gather this method is best for orders up to ^6 but better methods are available for higher orders... so I am told...

=====

I gather this method is best for orders up to ^6 but better methods are available for higher orders... so I am told...

=====

I gather this method is best for orders up to ^6 but better methods are available for higher orders... so I am told...

=====

## HOW A TWO YEAR OLD CAN TYPE AT 20 W.P.M.

### FAST TYPER...

```

1 REM SILLY PROG BY S SHAW      MARCH 1991
2 ! did you see COMPUTER WARS-the film? It is said that the star, who was required to type fast into a computer
3 ! could not type, so a program just like this one was used to give a good effect!
4 ! now adjust it how you wish and show your friends how fast you can type
5 ! at end of text string program will just stop with this listing but can be modified to do anything you wish!
6 !
100 A$="This is how a non-typist can produce information on      screen quickly,
without
110 A$=A$&"having to look at what keys are being bashed! Just bash keys and watch how perfect text appears no matter what you press"
120 PRINT A$ : : : :
130 CALL KEY(5,A,B):: IF B<1 THEN 130
140 C=C+1 :: PRINT SEG$(A$,C,1):: IF C=LEN(A$)THEN 160
150 GOTO 130
160 GOTO 160

```



```

1 ! ADDING FRACTIONS
2 ! result is not reduced      eg 12/16 would usually      be shown as 3/4
100 CALL CLEAR
110 DISPLAY AT(10,5):"--- + --- = ---"
120 ACCEPT AT(9,5)SIZE(3)VALIDATE(DIGIT):A
130 ACCEPT AT(11,5)VALIDATE(DIGIT)SIZE(3):B
140 ACCEPT AT(9,11)SIZE(3)VALIDATE(DIGIT):C
150 ACCEPT AT(11,11)SIZE(3)VALIDATE(DIGIT):D
160 GOSUB 230
170 DISPLAY AT(9,16):USING "####":N
180 DISPLAY AT(11,16):USING "####":L
190 DISPLAY AT(14,1):"ENTER KEY FOR ANOTHER"
200 DISPLAY AT(1,1):"NORMAL RESULT=";A/B+C/D
210 ACCEPT AT(24,12):A$ :: GOTO 100
220 STOP
230 FOR X=2 TO B*D
240 IF INT(X/B)<X/B THEN 260
250 IF INT(X/D)=X/D THEN 270
260 NEXT X
270 L=X
280 N=INT(L/B)*A+INT(L/D)*C
290 RETURN

```



Now write a program which will reduce the answer to a more normal format- and also one which can take away as well as add. This is not a trivial test! and I will list a suitable program later- see if you can do it first!

=====

Now write a program which will reduce the answer to a more normal format- and also one which can take away as well as add. This is not a trivial test! and I will list a suitable program later- see if you can do it first!

=====

PI

PI is the ratio of a circles radius (distance from centre to edge) and its circumference (distance all the way round).

In Extended Basic we have a fixed variable PI we can use. In TI Basic we can use the function ATN to derive PI. In machine code or most other languages we are on our own! WE could approximate to 3.14, but if greater accuracy is required? This little program derives PI to as many places as our language can support, without using any trigonometry, and is quite remarkably fast!

You may note the use of AND 2, which is using AND as a "logical operator", not too well covered in the manual. In Extended Basic, type in:

```
FOR T=1 TO 20 :: PRINT T;T AND 2 :: NEXT T and inspect the result!!!
```

```
1 ! CALCULATE PI
2 ! QUICKLY AND ACCURATELY
3 ! from W F DOSSETT, AUSTIN, TEXAS, 25 feb 88
4 ! for ti99/4a s shaw 3/91
5 ! from John Machin, 1706
6 !
7 ! n=7 for 13 digit accuracy of ti99/4a basic
8 ! n>7 if language goes beyond 13 digits!
9 ! eg n=35 for 53 digits
10 !
100 N=7 :: K=-1
110 K=K+1
120 I=-((2 AND 2*K)-1)
130 U=U+4*I/((2*K+1)*5^(2*K+1))
140 V=V+I/((2*K+1)*239^(2*K+1))
150 IF K=N THEN 110
160 W=U-V :: PI2=4*W
170 ! DONE
180 PRINT PI2,PI
190 PRINT (PI2-PI)*1000,! COMPARE with on board variable
200 PRINT (PI2-(4*ATN(1)))*1000 ! compare with TI Basic's calculation
210 PRINT " ";(PI2-3.14)*100;:: DISPLAY AT(24,2)SIZE(4):"3.14"
! displays ALL the digits actually in the variable!!!=13 incl dec point.
220 PRINT " 3.14159265358979 -etc"
! more accurate actual value for comparison!
230 END
999 !from THE REC NEWSLETTER March 1988 Vol 3 #2
```



The program is not derived from the standard mathematical series, stretching off to infinity, but from an approximation published around 1706, a little before computers were around! This -courtesy of Mr Dossett- has yielded a rather fast program!

PRIMES . . .

Continuing with math related topics, prime numbers- that is numbers which are divisible only by themselves and 1. Asked to write a program to list prime numbers I suspect most of us would start from 1 and then try dividing every number by every number to see if it was divisible...

The first listing is fsairly straight forward, but uses a few tricks to speed things up. In particular, if we know that a number is NOT divisible by two, it cannot possibly be divisible by any higher power of two, such as 4- or indeed any multiple of two. So we can check just the odd numbers, and also not bother testing for divisors greater than the first power of a known prime.

```
100 ! FIRST 100 PRIMES -NEATLY-
110 ! from Loren Krienke via THE REC NEWSLETTER March 1988 Vol 3 #2
120 ! for ti99/4a s shaw
130 L=100 ! number required
----> continued ---->
```

```
140 DIM P(100)! P(L)-holds found primes, used as divisors
150 DIM S(100)! S(L)-squares of primes, reduces number of divisions needed
160 P(1)=2 ! first prime
170 PRINT P(1);
180 T=1 ! initialise loop
190 FOR I=2 TO L ! test non-even numbers:
200 T=T+2
210 FOR J=1 TO I-1
220 CALL MOD(T,P(J),R):: IF R=0 THEN 200
230 IF T>S(J)THEN 235 ELSE 240
235 NEXT J
240 P(I)=T ! found prime, store it
250 S(I)=T*T ! GREATER SPEED
260 PRINT T;I*PRIME
270 NEXT I ! LOOP
280 !
290 ! 18 JULY 1986
300 ! Loren Krienke
310 ! San Diego, Ca.
320 !
330 SUB MOD(A,B,C)! C=A MOD B becomes CALL MOD(A,B,C)
340 C=INT(A-B*INT(A/B))
350 SUBEND
```



A similar but very much more opaque program follows, which has no obvious connection with the above, but it produces the same output!!! Check out the speed comparison between the two programs. Any difference? You may also wish to try looking for a larger number of primes!

```
100 ! FIRST 100 PRIMES -QUICKLY-
110 ! Dr H B Phillips from THE REC NEWSLETTER March 1988 Vol 3 #2
120 DIM P(300),X(12)
130 A=0 :: B=1 :: D=0.5 :: E=180
140 M=100 :: L=3 :: F=0
150 ! increase M for more- also increase DIMs.
160 PRINT 2;:: C=B :: IF M=B THEN END
170 L=INT((M/C)*L+F):: N=L+L+B
180 FOR I=B TO INT((SQR(N)-B)*D):: PP=P(I)
190 IF PP=B THEN 230
200 IF PP=A THEN PP=I+I+B :: PRINT PP;: P(I)=PP :: C=C+B :: IF C=M THEN END
210 IF X(I)=A THEN X(I)=(PP*PP-B)*D
220 FOR J=X(I)TO L STEP PP :: P(J)=B :: NEXT J :: X(I)=J
230 NEXT I :: IF F=0 THEN S=I
240 FOR I=S TO L
250 IF P(I)=A THEN PP=I+I+B :: PRINT PP;: P(I)=PP :: C=C+B :: IF C=M THEN END
260 NEXT I :: F=(M-C)*L/E :: S=L+B
270 GOTD 170
```



And no, I have no idea how this program works! When faced with a problem, the programmer usually has more than one way of attacking it, and it also usually happens that by restating the problem a faster solution can be found.

Now for a graphic program requiring The Missing Link.

```
100 ! RABBITS AND FOXES
110 ! Dolores Garcia,Spain
120 ! Dr M Ecker, PA, USA
130 ! s shaw for ti99/4a      ex bas+tml  april 91
140 CALL LINK("CLEAR"):: A,B=5
150 CALL LINK("PRINT",A,B,"Rabbits and Foxes"):: A=A+12 :: B=5
160 CALL LINK("PRINT",A,B,"Please input the variables requested below. Suggested
  values are given for a 'typical' response. Try others!")
170 CALL LINK("PRINT",70,12,"RABBIT BIRTH RATE:")
180 CALL LINK("PRINT",80,12,"RABBIT DEATH RATE (Excl eaten ones!):")
190 CALL LINK("PRINT",100,12,"RATE OF EATEN RABBITS:")
200 CALL LINK("PRINT",110,12,"FOXES DEATH RATE:")
210 CALL LINK("PRINT",120,12,"FOXES BIRTH RATE:")
220 CALL LINK("INPUT",70,150,A,4,".04"):: IF A<.00001 THEN A=0.4
230 CALL LINK("INPUT",90,150,B,6,".00005"):: IF B<1E-9 THEN B=.0005
240 CALL LINK("INPUT",100,150,C,4,".002"):: IF C<.00001 THEN C=.002
250 CALL LINK("INPUT",110,150,D,4,".03"):: IF D<.00001 THEN D=.04
260 CALL LINK("INPUT",120,150,E,5,".0002"):: IF E<1E-8 THEN E=.0002
270 CALL LINK("PRINT",140,12,"RABBITS START AT:")
280 CALL LINK("PRINT",150,12,"FOXES START AT:")
290 CALL LINK("INPUT",140,120,R,4,"300")
300 CALL LINK("INPUT",150,120,F,4,"30")
310 CALL LINK("CLEAR")
320 P=F*R
330 R=(1+A-B*R)*R-C*P
340 F=(1-D)*F+E*P
350 P=F*R
360 IF ER>500 THEN CALL LINK(180,180,"-.-.-"):: GOTO 360
370 IF NOT(F>0 AND R>0 AND F<200 AND R<640)THEN ER=ER+1 :: GOTO 330
380 CALL LINK("PIXEL",2+4*F,R*.7+2):: CALL LINK("PRINT",181,1,"RABBITS:"&STR$(IN
  T(R))&" ")
390 CALL LINK("PRINT",181,110,"FOXES:"&STR$(INT(F))&" ")
400 ER=0
410 GOTO 330
420 END
=====
```

This is a tricky little program - for The Missing Link, in this format it is a demo program, but if you leave the worm invisible all the time (by leaving Pen Erase on all the time) then you have a nasty little trick - especially if you disable break and quit. This is the fun version...

```
100 A,B=10
110 CALL LINK("PRINT",A,B,"The Missing Link is a disk based utility which allows
  the Extended Basic programmer to make use of many ")
120 CALL LINK("PRINT",A,B,"enhanced commands, including true bit map graphics, w
  hich in turn allows more powerful PRINT commands, and of ")
130 CALL LINK("PRINT",A,B,"course, full access to sprites.")
140 CALL LINK("PRINT",A,B,"This program demonstrates graphics outside a small wi
  ndow which, worm fashion, eat up the text! Enjoy..")
150 ! WORM
160 ! SCIENTIFIC AMERICAN DEC 87 by S L Wentworth
170 ! TI99/4A BY S SHAW 91
180 ! EXT BAS + THE MISSING LINK
190 DIM XC(25),YC(25)
200 XC(2),YC(2),XC(1),YC(1)=100
210 D=0
220 T=1
230 RANDOMIZE
```

-----> more more ---->

```
240 CALL LINK("WINDOW",80,80,110,120,1)
250 CALL LINK("PRINT",7,7,"WORM HOLE")
260 CALL LINK("REVWIN")
270 WT=T
280 CALL MOD(T,25,T):: T=T+1
290 CALL LINK("PE")
300 CALL LINK("CIRCLE",XC(T),YC(T),4,0)
310 CALL LINK("PD")
320 C=RND
330 IF C<.5 THEN D=D+.1745 ELSE D=D-.1745
340 X=XC(WT)
350 Y=YC(WT)
360 NX=X+4*COS(D)
370 NY=Y+4*SIN(D)
380 IF NX>189 THEN NX=NX-189
390 IF NY>240 THEN NY=NY-240
400 IF NX<2 THEN NX=NX+189
410 IF NY<1 THEN NY=NY+240
420 XC(T)=NX
430 YC(T)=NY
440 !
450 CALL LINK("CIRCLE",NX,NY,4,0)
460 CALL LINK("PD")
470 GOTO 270
480 SUB MOD(A,B,C)
490 C=INT(A-B*INT(A/B))
500 SUBEND
```

This is the promised fractional plus and minus program. I won't tell you how long it took me to get the displays like that....

```
100 ! FRACTIONAL + AND -
110 ! R CALDWELL JAN 91
120 ! FOR TI99/4A BY
130 ! S SHAW APR 91
140 DIM Q(102)
150 DISPLAY AT(1,1)ERASE ALL:"FRACTIONAL + & -"
160 DISPLAY AT(7,1):"-----"
170 DE,X=1
180 ACCEPT AT(6,2)VALIDATE(DIGIT,"+-")SIZE(4):N(X)
190 IF N(X)=0 THEN 310
200 ACCEPT AT(8,2)VALIDATE(DIGIT)SIZE(4):D(X)
210 IF D(X)=0 THEN 310
220 DP(X)=N(X):: DE=DE*D(X)
230 CALL HCHAR(6,1,32,12)
240 CALL HCHAR(8,1,32,12)
250 DISPLAY AT(5,1):"PLUS:"
260 DISPLAY AT(3,1):"ENTER 0 TO TOTAL"
270 X=X+1 :: DISPLAY AT(7,15):"ITEM";X :: DISPLAY AT(8,15):"-MAX 10-"
280 IF X=11 THEN 310
290 GOTO 180
300 REM
310 F,X=X-1
320 FOR S=1 TO F :: FOR J=1 TO F
330 Z=Z+1 :: K=S+J-1
340 P(S,J)=K-F*INT(K/F):: IF P(S,J)=0 THEN P(S,J)=F
350 Q(Z)=P(S,J):: NEXT J :: NEXT S
360 FOR X=1 TO F :: Y=F*X-F+1 :: FOR C=1 TO F-1
370 Y=Y+1 :: DP(X)=DP(X)*D(Q(Y)):: NEXT C
```

----> more ----> more





```

380 NU=NU+DP(X):: NEXT X :: DD=2
390 DISPLAY AT(10,1):"SUM IS:" :: DISPLAY AT(11,5):NU;" / ";DE
400 IF NU/DD=INT(NU/DD)AND DE/DD=INT(DE/DD)THEN NU=NU/DD :: DE=DE/DD :: GOTO 400
410 DD=DD+2+(DD=2)
420 IF ABS(NU)>DE THEN A=ABS(NU)ELSE A=DE
430 IF DD<=SQR(A)THEN 400
440 IF NU/DE=INT(NU/DE)THEN 540
450 IF NU>0 THEN WN=INT(NU/DE):: NU=NU-WN*DE :: PN=1
460 IF NU<0 THEN WN=INT(NU/DE)+1 :: NU=ABS(NU-WN*DE):: PN=2 :: IF WN=0 THEN NU=-
NU
470 REM
480 DISPLAY AT(13,5):USING "#####":NU
490 IF PN=1 THEN DISPLAY AT(14,1):WN :: NU=NU+WN*DE
500 IF PN=2 THEN DISPLAY AT(14,1):WN :: IF WN<0 THEN NU=WN*DE-NU
510 PN=0
520 DISPLAY AT(14,5):"-----"
530 DISPLAY AT(15,5):USING "#####":DE
540 DISPLAY AT(18,4):NU/DE
550 DISPLAY AT(24,4):"ANY KEY FOR ANOTHER"
560 CALL KEY(5,G,H):: IF H<1 THEN 5&0
570 RUN
580 END
=====
CRYPTO CALC...

```



In this puzzle the computer will ask you to form a target number by using the four main math operators ( add, minus, divide, times) and some numbers it will give you. You may select the number of numbers you wish to use, the computer will select the numbers and the target, and then the computer will find the order of operations. You will probably beat the computer!

Let's say the target is 1 and the numbers are 10 10 6 and 8.  
 $10/10=1$  and  $8-6=2$  and then  $2-1=1$   
 The computer will display this in the form:  
 $((8-6)-(10/10))$  adding brackets to make the order of operations plainer.

From time to time, unusually! the computer may beat you to it or even find a solution you cannot - but it may be painfully slow! Can you speed this up?

```

100 CALL CLEAR :: RANDOMIZE
110 PRINT "CRYPTO CALC":"Ray McClanahan 1989":"for TI99/4A 1991 S Shaw":"":"base
d on Krypto by":"MBH Games Co"
120 PRINT "":"from REC Newsletter":" Vol 4 #7":"
130 !
140 !
150 PRINT "How many cards? 4-9" :: :
160 DISPLAY AT(23,9):"5" :: ACCEPT AT(23,9)SIZE(-1)VALIDATE("456789"):C :: W=C-1
170 !
180 V=INT(RND*C*5)+1 :: PRINT "":"TARGET VALUE CALCULATED"
190 F(1)=1 :: FOR A=2 TO C :: F(A)=A*F(A-1):: NEXT A
200 PRINT "Computer now selecting numbers to use..."
210 RANDOMIZE
220 FOR A=1 TO C
230 I(A)=INT((INT(RND*RND*C*10+1)+INT(RND*C*4+1)+INT(RND*RND*12+1))/3):: IF I(A)
>C*5 THEN 230
240 IF I(A)=I(A-1)THEN I(A)=I(A)+4 :: IF I(A)>25 THEN 230
250 PRINT I(A):: NEXT A
260 FOR A=1 TO 500 :: NEXT A :: CALL CLEAR
270 FOR U=1 TO F(C):: D=C :: J=U :: GOSUB 660
continued---->

```



```

280 FOR A=1 TO C :: M(A)=I(R(A)):: NEXT A
290 FOR E=1 TO F(W):: D=W :: J=E :: GOSUB 660
300 DISPLAY AT(4,1):"TARGET VALUE:";V
310 DISPLAY AT(6,1):"Make this value using only + - * / and these numbers:"
320 FOR A=1 TO C :: DISPLAY AT(A+8,24):I(A):: NEXT A
330 DISPLAY AT(16,1):"There MAY be no solution- computer will tell you..."
340 FOR A=1 TO W :: Q(A)=R(A):: NEXT A
350 FOR A=1 TO W :: Z(A)=1 :: NEXT A
360 FOR A=1 TO C :: N(A)=M(A):: NEXT A
370 FOR A=1 TO W :: P(A)=Q(A):: S(A)=Z(A):: NEXT A
380 FOR A=1 TO W :: L=C+1-A
390 ON S(P(A))GOTO 400,410,420,430
400 N(P(A))=N(P(A))+N(P(A)+1):: GOTO 450
410 N(P(A))=N(P(A))-N(P(A)+1):: GOTO 450
420 N(P(A))=N(P(A))*N(P(A)+1):: GOTO 450
430 IF N(P(A)+1)=0 THEN 570
440 N(P(A))=N(P(A))/N(P(A)+1)
450 IF A=W THEN 550
460 IF P(A)=L-1 THEN 510
470 FOR B=P(A)+1 TO L-1
480 N(B)=N(B+1):: NEXT B
490 FOR B=P(A) TO W-1
500 S(B)=S(B+1):: NEXT B
510 FOR B=A+1 TO W
520 IF P(A)>P(B)THEN 540
530 P(B)=P(B)-1
540 NEXT B
550 NEXT A
560 IF ABS(V-N(1))<.001 THEN 760
570 FOR A=W TO 1 STEP -1
580 IF Z(A)=4 THEN 600
590 Z(A)=Z(A)+1 :: GOTO 650
600 Z(A)=1
610 NEXT A
620 DISPLAY AT(10,1):"OP Order Perm No";E;" failure" :: NEXT E
630 DISPLAY AT(12,1):"Number order Perm No";U;" failure" :: NEXT U
640 DISPLAY AT(14,1):"ALL POSSIBILITIES CHECKED":"NO MATCH" :: CALL SOUND(900,20
0,0):: RUN
650 GOTO 360
660 FOR H=1 TO D :: T(H)=H :: NEXT H
670 FOR X=1 TO D-1
680 Y=D-X
690 G=INT((J-1)/F(Y)+.00001)
700 R(X)=T(G+1)
710 J=J-G*F(Y)
720 IF G=Y THEN 740
730 FOR K=G+1 TO D-1 :: T(K)=T(K+1):: NEXT K
740 NEXT X
750 R(D)=T(1):: RETURN
760 FOR A=1 TO C :: O$(A)=STR$(M(A)):: NEXT A
770 FOR A=1 TO W :: P(A)=O(A):: S(A)=Z(A):: NEXT A
780 FOR A=1 TO W :: L=C+1-A
790 ON S(P(A))GOTO 800,810,820,830
800 O$(P(A))="("&O$(P(A))&"+"&O$(P(A)+1)&")" :: GOTO 840
810 O$(P(A))="("&O$(P(A))&"-"&O$(P(A)+1)&")" :: GOTO 840
820 O$(P(A))="("&O$(P(A))&"*"&O$(P(A)+1)&")" :: GOTO 840
830 O$(P(A))="("&O$(P(A))&"/"&O$(P(A)+1)&")" :: GOTO 840
840 IF A=W THEN 940
continued---->

```



```

850 IF P(A)=L-1 THEN 900
860 FOR B=P(A)+1 TO L-1
870 O$(B)=O$(B+1):: NEXT B
880 FOR B=P(A) TO W-1
890 S(B)=S(B+1):: NEXT B
900 FOR B=A+1 TO W
910 IF P(A)P(B) THEN 930
920 P(B)=P(B)-1
930 NEXT B
940 NEXT A
950 PRINT O$(1)
960 DISPLAY AT(15,1):O$(1):"": "ANY KEY FOR ANOTHER"
970 CALL KEY(5,A,B):: IF B<1 THEN 970
980 DISPLAY AT(12,2)ERASE ALL: "ONE MOMENT..."
990 RUN
1000 END

```



This very long top-down program could be made shorter - can you do it? - and is a trick based upon Pascals Triangle. Details of operation in the listing- if you like, try to think of it this way...

The computer first establishes rules which determine how it will deal with six numbers. These rules are fixed. A target result is set, you input two numbers, and based upon these the computer will forecast the other numbers you are going to input, and put in a balancing number itself- before you input your extra numbers- in order to achieve the desired result. It is an interesting puzzle....

```

10 ! math trick from          REC NEWSLETTER Vol 4 #7      after Michael W Ecker
    from Prof H I Ridge
20 ! for ti99/4a by s shaw    april 1991
40 PRINT "Math Trick": "S Shaw April 1991": ""
100 REM
110 PRINT "We will first choose six numbers across the top of the screen"
120 PRINT "Then we will add up each pair of adjacent numbers."
130 PRINT "The result of this sum will be divided by five until only a remainder is left."
140 PRINT "This we will place below the pair we have added. We continue to do this until only one number is left."
150 PRINT "You will choose all the top numbers except one. Before you have chosen all your numbers I will tell you the final number."
160 PRINT "Aren't us computers clever!"
170 PRINT "": "press any key to start"
180 CALL KEY(5,A,B)
190 IF B<1 THEN 180
200 CALL CLEAR
210 PRINT "No, tell you what, YOU choose what that final number should be: 0, 1, 2, 3, or 4:":
220 ACCEPT AT(24,26)SIZE(1)VALIDATE("01234"):A# :: IF LEN(A#)<1 THEN 220
230 TGT=VAL(A#)
240 CALL CLEAR
250 DISPLAY AT(1,1):"-----"
260 DISPLAY AT(23,1):"INPUT ONE OR TWO DIGIT NUMBERS AS INDICATED ABOVE"
270 DISPLAY AT(20,1):"TARGET=":A#
280 ACCEPT AT(1,1)SIZE(2)VALIDATE(DIGIT):N1
290 ACCEPT AT(1,10)SIZE(2)VALIDATE(DIGIT):N4
    -----> more more ----->

```

```

300 Z=N1
310 IF Z>5 THEN Z=Z-5 :: GOTO 310
320 N6=TGT-Z :: N6=N6+10*INT(RND*8+1)
330 DISPLAY AT(1,15):N6
340 ACCEPT AT(1,4)SIZE(2)VALIDATE(DIGIT):N2
350 ACCEPT AT(1,7)SIZE(2)VALIDATE(DIGIT):N3
360 ACCEPT AT(1,13)SIZE(2)VALIDATE(DIGIT):N5
362 CALL HCHAR(23,1,32,64)
370 DISPLAY AT(17,1):N1;"+";N2;"=";N1+N2 :: Z=N1+N2
380 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
390 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N1=Z2*5
400 DISPLAY AT(2,2)BEEP:N1 :: FOR T=1 TO 1900 :: NEXT T
410 DISPLAY AT(17,1):N2;"+";N3;"=";N2+N3 :: Z=N2+N3
420 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
430 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N2=Z2*5
440 DISPLAY AT(2,5)BEEP:N2 :: FOR T=1 TO 1900 :: NEXT T
441 !
442 !
450 DISPLAY AT(17,1):N3;"+";N4;"=";N3+N4 :: Z=N3+N4
460 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
470 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N3=Z2*5
480 DISPLAY AT(2,8)BEEP:N3 :: FOR T=1 TO 1900 :: NEXT T
481 !
482 !
490 DISPLAY AT(17,1):N4;"+";N5;"=";N4+N5 :: Z=N4+N5
500 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
510 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N4=Z2*5
520 DISPLAY AT(2,11)BEEP:N4 :: FOR T=1 TO 1900 :: NEXT T
521 !
530 DISPLAY AT(17,1):N5;"+";N6;"=";N5+N6 :: Z=N5+N6
540 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
550 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N5=Z2*5
560 DISPLAY AT(2,14)BEEP:N5 :: FOR T=1 TO 1900 :: NEXT T
561 !
570 DISPLAY AT(17,1):N1;"+";N2;"=";N1+N2 :: Z=N1+N2
580 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
590 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N1=Z2*5
600 DISPLAY AT(3,3)BEEP:N1 :: FOR T=1 TO 1900 :: NEXT T
610 DISPLAY AT(17,1):N2;"+";N3;"=";N2+N3 :: Z=N2+N3
620 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
630 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N2=Z2*5
640 DISPLAY AT(3,6)BEEP:N2 :: FOR T=1 TO 1900 :: NEXT T
650 DISPLAY AT(17,1):N3;"+";N4;"=";N3+N4 :: Z=N3+N4
660 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
670 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N3=Z2*5
680 DISPLAY AT(3,9)BEEP:N3 :: FOR T=1 TO 1900 :: NEXT T
690 DISPLAY AT(17,1):N4;"+";N5;"=";N4+N5 :: Z=N4+N5
700 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
710 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N4=Z2*5
720 DISPLAY AT(3,12)BEEP:N4 :: FOR T=1 TO 1900 :: NEXT T
721 !
730 DISPLAY AT(17,1):N1;"+";N2;"=";N1+N2 :: Z=N1+N2
740 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
750 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N1=Z2*5
760 DISPLAY AT(4,5)BEEP:N1 :: FOR T=1 TO 1900 :: NEXT T
761 !
770 DISPLAY AT(17,1):N2;"+";N3;"=";N2+N3 :: Z=N2+N3
780 DISPLAY AT(18,1):Z;" / ";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
790 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N2=Z2*5
800 DISPLAY AT(4,8)BEEP:N2 :: FOR T=1 TO 1900 :: NEXT T
    -----> continued ----->

```

THE ARTIST + SMARTCOPY

```

801 !
810 DISPLAY AT(17,1):N3;"+";N4;"=";N3+N4 :: Z=N3+N4
820 DISPLAY AT(18,1):Z;"/";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
830 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N3=Z2*5
840 DISPLAY AT(4,11)BEEP:N3 :: FOR T=1 TO 1900 :: NEXT T
841 !
850 DISPLAY AT(17,1):N1;"+";N2;"=";N1+N2 :: Z=N1+N2
860 DISPLAY AT(18,1):Z;"/";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
870 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N1=Z2*5
880 DISPLAY AT(5,7)BEEP:N1 :: FOR T=1 TO 900 :: NEXT T
881 !
890 DISPLAY AT(17,1):N2;"+";N3;"=";N2+N3 :: Z=N2+N3
900 DISPLAY AT(18,1):Z;"/";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
910 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N2=Z2*5
920 DISPLAY AT(5,10)BEEP:N2 :: FOR T=1 TO 900 :: NEXT T
921 !
930 DISPLAY AT(17,1):N1;"+";N2;"=";N1+N2 :: Z=N1+N2
940 DISPLAY AT(18,1):Z;"/";5;"=";Z/5 :: Z2=Z/5-INT(Z/5)
950 DISPLAY AT(19,1):Z2;"*";5;"=";Z2*5 :: N1=Z2*5
960 DISPLAY AT(6,9)BEEP:N1 :: FOR T=1 TO 900 :: NEXT T
970 DISPLAY AT(6,9)BEEP:N1 :: FOR T=1 TO 900 :: NEXT T
980 DISPLAY AT(9,1):"TARGET ACHIEVED"
990 DISPLAY AT(23,1):"ANY KEY TO DO IT AGAIN"
1000 CALL KEY(5,A,B)
1010 IF B<1 THEN 1000
1020 DISPLAY AT(12,3)ERASE ALL:"ONE MOMENT..."
1030 RUN
1040 END
=====

```



```

1 ! Bloggs Store is giving away a free playing card with
every purchase. The y have the same number of each of the 52
cards.
2 ! Cards are given at "true" random.
3 ! on average how many purchases do I need to make to get a
full set of cards?
4 ! Phillips 86. Shaw 91.
100 ITEMS=52
110 FOR K=1 TO ITEMS
120 S=S+1/K
130 NEXT K
140 PRINT "BUY ";S*ITEMS
150 STOP

```

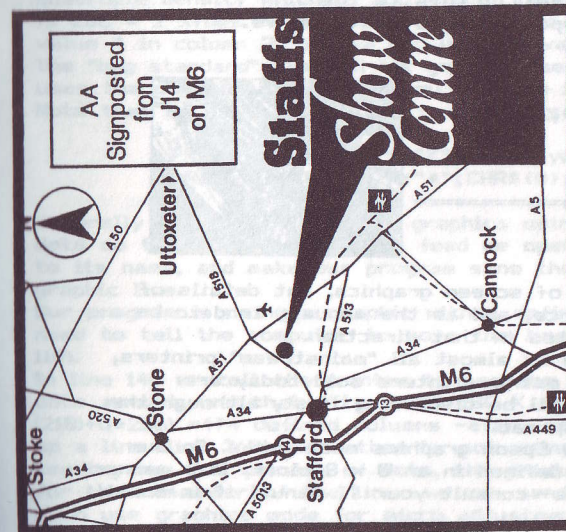
## SHUFFLING AROUND...

These programs have a direct use in card games, and can replace shuffles used in the various programs you may have that are slower than these! You may find other reasons for shuffling values of course! (BINGO anyone).

```

1 ! FASTEST & FAST 52 CARD SHUFFLE
2 ! no excuse for taking longer than 11 seconds!
3 ! improve your XB card games today
4 !
5 !
6 RANDOMIZE
100 DIM C(52)
101 ! 10.2 SECS:
110 FOR X=1 TO 52 :: C(X)=X :: NEXT X
120 FOR X=52 TO 2 STEP -1
130 I=INT(RND*X+1)
140 A=C(I):: C(I)=C(X):: C(X)=A
150 NEXT X
160 FOR T=1 TO 52 :: PRINT C(T);: NEXT T
170 PRINT : : :
180 !
190 !
200 ! using strings slows things down:
5000 ! 15.5 secs:
5010 FOR T=1 TO 52 :: A$=A$&CHR$(T):: NEXT T
5020 FOR T=1 TO 52 :: P=INT(RND*LEN(A$)+1)
5030 C(T)=ASC(SEG$(A$,P,1))
5040 A$=SEG$(A$,1,P-1)&SEG$(A$,P+1,52)
5050 NEXT T
5060 FOR T=1 TO 52 :: PRINT C(T);: NEXT T
=====

```



Exhibitors include -

**A1 Electrical Services**  
**A1 Electronics**  
**Capital Products**  
**Carrs Electronics**  
**Chic Computer Club**  
**Dean Garraghty**  
**Digital Precision Limited**  
**DX Electronics**  
**Hollywood Business Supplies**

**ID Computers**  
**MFM Supplies**  
**Micro Discount**  
**Nipco - Frank Martin**  
**Original Media Company**  
**Page 6 Publishing**  
**Tekdata Limited**  
**Willis Enterprises**  
 etc etc

I have used the DEF function to prepare COS and SIN functions COSD and SIND which expect to find degree measures instead of the usual radians. It is a good way of finding out how efficient DEF is too— just try replacing the SIND and COSD functions and replacing them with the ordinary COS and SIN functions but amend the variable in accordance with the DEF - eg instead of COS(A) use COS(A/180\*PI) in every SIND and COSD line. Any faster?

```

1 REM ANALYTIC COMPUTER ART
2 ! JOE JACOBSON
3 ! APT 1009 675 E STREET RD
4 ! WARMINSTER PA USA 18974
5 ! fluted scallops
6 ! The Journal of Chaos &
7 ! For TI99/4A plus ExBas +
Stephen Shaw UK Jan 91
8 !
100 DEF SIND(X)=SIN(X/180*PI)
110 DEF COSD(X)=COS(X/180*PI)
120 OS=99 :: RANDOMIZE
130 L=INT(RND*60)+3
140 CALL LINK("PRINT",180,220,STR$(L))
150 FOR B=16 TO 76 STEP 2
160 FOR A=0 TO 360 STEP 5
170 GOSUB 250
180 IF A>0 THEN 210
190 CALL LINK("PIXEL",OS+X,Y+OS):: OLDX=X :: OLDY=Y
200 GOTO 220
210 CALL LINK("LINE",OS+OLDX,OLDY+OS,OS+X,OS+Y)::
OLDX=X :: OLDY=Y
220 NEXT A
230 NEXT B
240 GOTO 290
250 R=B*(1+.25*ABS(SIND(L*A)))! this is the
function plotted:R=f(x):=polar coord inate curve.
260 X=R*COSD(A)
270 Y=R*SIND(A)
280 RETURN
290 CALL LINK("PRINT",180,20,"+")
300 CALL LINK("DUMP")
305 CALL LINK("CLEAR")
310 RUN

```

Graphics August 1988  
The Missing Link by



#### PRINTER GRAPHICS.

We have had a number of programs of screen graphics, but details of printer graphics seem a little scarce, and as the manuals tend to be a little unhelpful, this item is directed in that direction. The EPSON standard is now covered by almost all "mainstream" printers, and to some extent almost all dot matrix printers sold today are compatible with the commands we shall be discussing first, although the second method may not be so widespread. Firstly let us look at the way the Epson graphics mode works. On our computer characters are typically defined in an 8 x 8 block, in a manner fairly well known to most TI owners— consult your TI manual if in doubt!  
more---->

The TI defines its characters in two horizontal blocks, so that a straight line character could be defined as "FF00000000000000" while a solid 8x8 block would be "FFFFFFFFFFFF FFFF" or a straight vertical line "8080808080808080". The printer has a print head with 9 pins arranged vertically, and characters are defined in a vertical manner to allow faster printing.

On the Epson, characters are nominally 8 dots high, with an option of shifting them down one row to use the ninth (bottom) printing pin. We shall here discuss only the use of the "normal" eight pins, and refer to the eighth pin from the top as the "bottom" pin— eight pin bit image graphics by definition do not use the ninth pin! The bottom pin is allocated a value of 1, the next one up a value of 2, then 4 then 8 then 16 then 32 then 64 and finally the top pin a value of 128. We add up the values of all the pins we wish to strike in a single column and send this value to the printer— so a straight vertical line would require us to send a value of 255, eg CHR\$(255). Sending several columns to print a character needs a little rethinking from the TI approach however, as the Epson graphic standard uses a half column. Imagine that the usual TI columns are numbered 1,3,5,7 and so on. These print exactly next to each other. The Epson standard allows us to print dots half-under each other, like this:

| TI:             | EPSON:              |
|-----------------|---------------------|
| XXX XXX XXX XXX | XXX XXX XXX XXX XXX |
| XXX XXX XXX XXX | XXX XXX XXX XXX XXX |
| XXX XXX XXX     | XXX XXX XXX XXX     |
| XXX XXX XXX     | XXX XXX XXX XXX     |
| XXX XXX         | XXX XXX XXX XXX     |
| XXX XXX         | XXX XXX XXX XXX     |

Thus the printer allows characters to be a little less square than the TI forces us to be. Note however that the intermediate half column pins cannot be used as well as an adjacent normal column pin when using quadruple density which is ESC \* 3 or double speed double density which is ESC \* 2 ! (Thus if we use pin value 4 in column 1 we cannot use pin value 4 in column 2, but we can use pin value 2 or 8 in column 2). The "bog standard" Epson 8 bit graphic uses 480 dots along each line and uses the command ESC K... as you will see in the program which follows. Note that an exact equivalent to ESK K is ESC \* 0— so instead of:

```

PRINT #1:CHR$(27);"K";... you could use:
PRINT #1:CHR$(27);"*";CHR$(0);...

```

Generally you will be printing graphics using more than 80 columns of dots so to avoid an auto line feed we open the printer up by adding .CR to its name, and make our program sane the line feeds at the end of each graphic line.

Our program is using a graphic with a defined shape in a 6x6 grid, so we need to tell the computer to move the line up 6 dots at a time (Line 110).

In line 140 we tell the printer to use the 8 pin bit image mode with 480 dots across, and that after each ESC K sequence we are going to send (21B)+(1\*256) =474 defined columns —after which we will tell it to move up a line (line 260). Note that by not using the full width, and by sending say 200 columns of dots, the printer then switched to text mode for the remainder of the line— we can send mixed text and graphics, or even use graphics mode for micro adjustment of character positions.  
----> continued---->

The program is based on a Truchet tile generation.

```

100 OPEN #1:"PIO.CR"
110 PRINT #1:CHR$(27);"A";CHR$(6);
120 REM lines are 6 dots high
130 RANDOMIZE
140 A$=CHR$(27)&"K"&CHR$(218)&CHR$(1)
150 REM sets up for 474 dot columns to follow.
160 [$=CHR$(8)&CHR$(16)&CHR$(32)&CHR$(65)&CHR$(2)&CHR$(4)
170 ]$=CHR$(8)&CHR$(4)&CHR$(2)&CHR$(65)&CHR$(32)&CHR$(16)
180 PRINT #1
190 PRINT #1:A$;
200 FOR WIDTH=1 TO 79
201 REM 6 x 79 = 474
210 IF RND<.5 THEN 240
220 PRINT #1:[$;
230 GOTO 250
240 PRINT #1:]$;
250 NEXT WIDTH
260 PRINT #1:CHR$(10);CHR$(13);
270 GOTO 190
280 END

```



Please note that this graphics program is written in pure TI BASIC and only requires a printer and appropriate TI->Printer interface to operate. Instead of ESC K try using instead ESC \* 0 to ESC \* 6 or any other ESC \* modes your printer may support. The above program does not use adjacent horizontal dots and so all modes may be used.

The Epson range of printers can be found with "definable fonts" which allows you to redefine the character definitions held in the printer. This is a little more arduous. The TI Basic program below prints about the same pattern as the above program, but instead of using different ESC \* modes you may use character modes such as condensed, expanded, Pica, Elite, and so on. Your printer must support the commands ESC & as well as ESC %. On the FX80 printer it is necessary to switch OFF the on board print buffer, as this is used to store the new definitions. See your printer manual under "download characters".

```

100 OPEN #1:"PIO.CR"
110 PRINT #1:CHR$(27);"A";CHR$(6);
120 PRINT #1:CHR$(27);"p";CHR$(1);
121 REM proportional mode required to get correct horizontal spacing for
6x6 graphic.
130 RANDOMIZE
140 A$=CHR$(27)&"&"&CHR$(0)
150 B$=CHR$(0)&CHR$(0)&CHR$(0)&CHR$(0)&CHR$(0)
160 [$=A$&"["&CHR$(6)&CHR$(8)&CHR$(16)&CHR$(32)&CHR$(65)&CHR$(2)&CHR$(4)
&CHR$(8)&B$
170 ]$=A$&"J]"&CHR$(6)&CHR$(8)&CHR$(4)&CHR$(2)&CHR$(65)&CHR$(32)&CHR$(16)
&CHR$(8)&B$
180 PRINT #1:[$&]$.
181 REM redefines characters [ and ]
190 PRINT #1:CHR$(27);"Z";CHR$(1);CHR$(0);
191 REM tells printer to use new definition instead of usual character
200 FOR WIDTH=1 TO 131

```

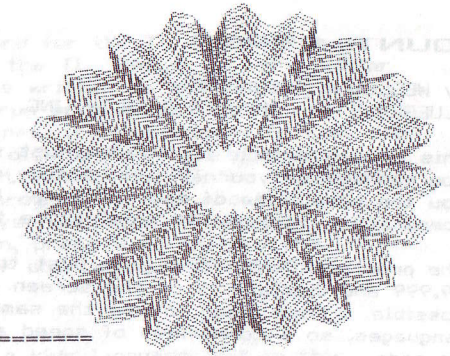
----> continued----->

```

210 IF RND<.5 THEN 240
220 PRINT #1:"L";
230 GOTO 250
240 PRINT #1:"J";
250 NEXT WIDTH
260 PRINT #1:CHR$(10);CHR$(13);
270 GOTO 200
280 END

```

Have fun. Stephen Shaw 1990



WANTED

MBX Expansion System (Box, special joystick, & microphone) with following modules: Terry Turtles Adventure, I'm Hiding, Honeyhunt & Soundtrack Trolley, Championship Baseball.

Offers to A.Cougenat, 66 Rue de Repos, 69007 Lyon, France.

## COUNT 99999

by WESLEY R. RICHARDSON  
BLUEGRASS 99 COMPUTER SOCIETY, INC.

This lengthy article illustrates just some of the many languages you can use on your TI99/4a, and by using a simple project shows you the basic idea of each language. Would you like more comparative listings like this? Write to Stephen Shaw.

The program COUNT 99999 does just that, counts from 00,000 to 99,999 and displays it on the screen as it goes. As much as is possible, I have tried to use the same algorithm for all of the languages, so a comparison of speed and program statements could be made.

The TI-99/4A now has several languages available to program in, including BASIC, EXTENDED BASIC, 99 FORTRAN, UCSD PASCAL, LOGO, PILOT, FORTH, c99, and 9900 ASSEMBLY. A comparison of the run times is shown below. Please keep in mind that the relative speeds shown apply only to this program and that different programs would have different relative times.

The languages which support interfacing with assembly are 99 FORTRAN, Extended BASIC, PASCAL, FORTH, and c99. This means that in an application where speed is important, you could write the speed limiting routines in assembly and yet have the advantage of ease of programming for the rest of the program in the language of your choice.

### PILOT

PILOT has not been included as one of the program listings, I believe there are three versions of PILOT for the TI available.

### BASIC/EXTENDED BASIC

These languages are probably used by the greatest number of TI-99/4A programmers because of the ease of writing. In a speed contest, they are the slowest.

### 99 FORTRAN

FORTRAN on the TI-99/4A is the newest member of the programming family and has many advantages in formatting of the output and in data manipulation. I did expect a faster running time, but this may have been due to the way in which I wrote the program. All functions of editing, compiling, linking, and running the program must be done from the 99 FORTRAN program environment. First EDIT the program, then SAVE it as 99999-H.

COMPILE the program, giving DSK2.99999-H as the input file and DSK2.99999-I as the object file. Use either CRT or PIO as the listing file, or just press (enter). I used scratch disk as 2 and took no options. After compiling with no errors, select the LINK function and give DSK2.99999-I as the object file. For the second object file just press (enter).

At SCAN LIBRARY ? put the FORTRAN LIBRARY DISK into one of the drives and answer Y and press (enter). After scanning press (enter) again and then select the SAVE function and save as file DSK2.99999-J. Then you can RUN the file, or later LOAD and then RUN the DSK2.99999-J file.

### UCSD PASCAL (P-CODE CARD):

Very few people have the p-CODE card for the TI, and even fewer people write programs in PASCAL on the TI. PASCAL is very similar in structure to c99. The program is written as a TEXT file, and then is compiled to a CODE file to run or execute. Since COUNT uses the SUPPORT routines, it is necessary to copy the file SYSTEM.LIBRARY to the ED-FILR: disk both for the compiling, and to run the program. The SYSTEM.LIBRARY file can be (Transfer) from the COMPILR: disk. After the program has run about 10 min, the screen will blank because no keys have been pressed. If you intend to have it count again, press Y to restore the screen, if you want it to stop after counting, press enter to restore the screen.

### LOGO

Many people think of LOGO as being a kids language, yet in this case it runs faster than Extended BASIC and PASCAL. LOGO has many features such as the ability to test procedures by building from the bottom up, that is is very easy to write and debug. In design, LOGO and FORTH are very similar.

In both languages you start defining simple procedures or words and then build on them by defining new words which use the previously defined words. This process continues until the final procedure or word does the function desired of the program. After entering the LOGO procedures definitions, save the PROCEDURES to disk. Give START as the calling word to start the program.

### FORTH

To enter the FORTH routine, you will need -EDITOR as your load option and -PRINT if you are going to list the screen. To run the program, you will need at least -TEXT to be loaded, then type 48 LOAD and the program will autostart. After COUNT 99999 runs, pressing N will exit the program, then type MON for the TI colorbars screen.

### c99

To enter the c99 version of COUNT 99999, use the Editor/Assembler Editor to enter the program, and save it to disk as 99999;C Use E/A #5 to run the c99 compiler C99C and answer Y to the first two questions. Give DSK1.99999;C as the input file and DSK1.99999;S as the output file. (These can also be done from drive DSK2 ). Note that the file CONIO must be on DSK1 while compiling. After compiling with no errors, load the E/A Assembler and give DSK1.99999;S as the source code and DSK1.99999;O as the object code for output. Do not take any options. After assembling with no errors, the program can be run. Use E/A #3 to LOAD AND RUN. First load DSK1.99999;O then load DSK1.CSUP and then give START as the PROGRAM name to start it running. CSUP and CONIO are on the c99 disk. To change the program COUNT 99999 to PROGRAM format, you will need the E/A routine called SAVE and the c99 programs called C99PFI and C99PFF. Use E/A #3 and first load DSK1.C99PFI then load DSK1.99999;O then load DSK1.CSUP then load DSK1.C99PFF then load DSK1.SAVE then press enter. Give SAVE as the PROGRAM name and DSK1.99999;V as the output filename. You will then be able to run 99999;V from E/A #5 without needing the other files.

### TMS9900 ASSEMBLY

Assembly is the overall winner in terms of speed of execution. By specifying the workspace registers in the highspeed area at hex 8300, the program was speeded up by 3 seconds. Further speed improvements could be made by moving some of the code into this high speed area also.

The statement COUNT ? YN could have been written with a shorter VMBW routine but I wanted to preserve the same methods used in the other languages. Use E/A Editor to enter the listing and then save as the file 99999-S for source. When assembling, be sure to use the R as an option and give 99999-0 as the object file for output.

The program will then run from E/A #3 by loading 99999-0 and giving SFIRST as the program name. To convert this to PROGRAM image format, select E/A #3 and load 99999-0 then load SAVE and then press enter. Give SAVE as the program name and 99999-V as the output file. Then you can use E/A #5 to run the program 99999-V.

#### TURBO PASC 99

Stephen Shaw of the U.K. TI Users Group in England wrote that he could not believe the slow timing for PASCAL, so he tried COUNT 99999 in TURBO PASC 99. He found that TURBO PASC 99 took only 7.5 percent of the time that PASCAL did to count to 99,999. The source code is very similar, with the main difference being that PASCAL uses GOTOXY(COLUMN,ROW) while TURBO PASC 99 uses CURSOR(ROW,COLUMN) to locate the character for printing to the screen.

In a comparison of standard benchmarks of computer speed, such as INTMATH, REALMATH, TRIGLOG, TEXTSCRN, GRAFSCRN, and STORE, Stephen Shaw has found that TURBO PASC 99 usually runs faster than PILOT 99, MYARC XB, TI BASIC, and TI XB. His data, as well as the COUNT 99999 data show that c99, FORTH, and TURBO PASC 99 all run in similar times. Stephen Shaw's conclusions, which I agree with are that while one language may be fast at one thing, inevitably another language is fast at another. Some languages are easier to use than others, and some can do things others cannot.

Below is an updated comparison of the languages that have run COUNT 99999.

| LANGUAGE:  | COUNT 99999 | RUN TIME | RATIO: |
|------------|-------------|----------|--------|
| BASIC      | 1 HR 51 MIN | 39.4 SEC | 295.1  |
| 99 FORTRAN | 0 HR 55 MIN | 21.9 SEC | 146.3  |
| XBASIC     | 0 HR 44 MIN | 13.1 SEC | 116.9  |
| PASCAL     | 0 HR 37 MIN | 32.5 SEC | 99.2   |
| LOGO       | 0 HR 35 MIN | 28.8 SEC | 93.8   |
| TURBO PASC | 0 HR 02 MIN | 48.0 SEC | 7.4    |
| FORTH      | 0 HR 01 MIN | 36.5 SEC | 4.3    |
| c99        | 0 HR 01 MIN | 28.0 SEC | 3.9    |
| ASSEMBLY   | 0 HR 00 MIN | 22.7 SEC | 1.0    |

#### FORTRAN:

##### PROGRAM COUNT

```

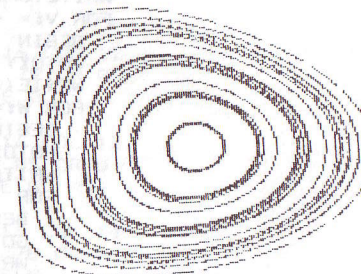
C COUNT 99999 TI-99/4A 99 FORTRAN
C COUNT 99999-H SOURCE CODE
C COUNT 99999-I OBJECT CODE
C COUNT 99999-J EXECUTABLE CODE
C WESLEY R. RICHARDSON DEC 1987
C BLUEGRASS 99 COMPUTER SOCIETY, INC.
C 99 FORTRAN RUN TIME 55 MIN 21.9 SEC
INTEGER V, W, X, Y, Z, YKEY, STAT
CALL CLEAR
YKEY=89
DO WHILE (YKEY.EQ.89)
WRITE(6,9100)
STAT=0

```

```

DO WHILE(STAT.EQ.0)
CALL KEY(0,YKEY,STAT)
ENDDO
IF (YKEY.NE.89) THEN
CALL EXIT
ENDIF
CALL CLEAR
WRITE(6,9200)
DO 2000 V=48,57
WRITE(6,9300) V
DO 2000 W=48,57
WRITE(6,9400) W
DO 2000 X=48,57
WRITE(6,9500) X
DO 2000 Y=48,57
WRITE(6,9600) Y
DO 2000 Z=48,57
WRITE(6,9700) Z

```



```

2000 CONTINUE
ENDDO
9100 FORMAT('+',M16.11,'COUNT ? YN')
9200 FORMAT('+',M12.15,',')
9300 FORMAT('+',M12.13,CN)
9400 FORMAT('+',M12.14,CN)
9500 FORMAT('+',M12.16,CN)
9600 FORMAT('+',M12.17,CN)
9700 FORMAT('+',M12.18,CN)
END

```

(\* COUNT 99999-P TI-99/4A PASCAL \*)

(\* WESLEY R. RICHARDSON DEC 1987 \*)

(\* BLUEGRASS 99 COMPUTER SOCIETY, INC \*)

(\* PASCAL RUN TIME 37 MIN 32.5 SEC \*)

PROGRAM COUNT;

USES SUPPORT;

VAR V, W, X, Y, Z : INTEGER;

YKEY : CHAR;

BEGIN

PAGE(OUTPUT);

YKEY := 'Y';

WHILE YKEY = 'Y' DO

BEGIN

GOTOXY(11,16);

WRITE(CHR(67));

WRITE(CHR(79));

WRITE(CHR(85));

WRITE(CHR(78));

WRITE(CHR(84));

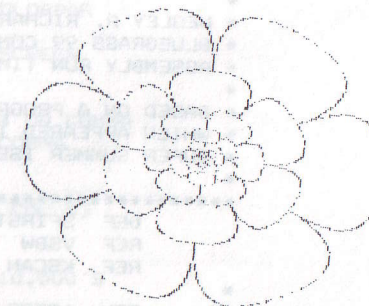
WRITE(CHR(32));

WRITE(CHR(63));

WRITE(CHR(32));

WRITE(CHR(89));

WRITE(CHR(78));



```

READ(YKEY);
IF YKEY <> 'Y' THEN EXIT(COUNT);
PAGE(OUTPUT);
GOTOXY(15,12);
WRITE(CHR(44));
FOR V:= 48 TO 57 DO
  BEGIN
  GOTOXY(13,12);
  WRITE(CHR(V));
  FOR W:= 48 TO 57 DO
    BEGIN
    GOTOXY(14,12);
    WRITE(CHR(W));
    FOR X:= 48 TO 57 DO
      BEGIN
      GOTOXY(16,12);
      WRITE(CHR(X));
      FOR Y:= 48 TO 57 DO
        BEGIN
        GOTOXY(17,12);
        WRITE(CHR(Y));
        FOR Z:= 48 TO 57 DO
          BEGIN
          GOTOXY(18,12);
          WRITE(CHR(Z));
          END;
        END;
      END;
    END;
  END;
END;
END;

```

END.

**9900 MACHINE CODE-SOURCE:**

```

*****
*
* 99999-S SOURCE CODE TI-99/4A
* 99999-O OBJECT CODE ASSEMBLY
* 99999-V E/A #5 RUN PROGRAM FILE
*
* WESLEY R. RICHARDSON DEC 1987
* BLUEGRASS 99 COMPUTER SOCIETY, INC.
* ASSEMBLY RUN TIME 0 MIN 22.7 SEC
*
* BASED ON A PROGRAM BY S. PEACOCK
* WHICH APPEARED IN CHICAGO TIMES
* SUPER SUMMER ISSUE AUGUST 31, 1987
*
*****
DEF SFIRST,SLAST,SLOAD
REF VSBW SINGLE BYTE WRITE
REF KSCAN KEY SCAN
*
ST EQU >837C STATUS AT >837C
KEYV EQU >8375 KEY VALUE
YES BYTE >59 YES KEY VALUE
*
SFIRST LWPI >8300 HIGH SPEED AREA
SLOAD EQU SFIRST
LI R3,>0030 CHAR 48 OR DEC 0
LI R4,>003A CHAR 58 OR :
*

```

```

BL @CLS CLEAR SCREEN
*
RSTRT LI R0,490 ROW 16 COL 11
LI R1,>4300 CHAR 67 OR C
BLWP @VSBW WRITE C
*
LI R0,491 ROW 16 COL 12
LI R1,>4F00 CHAR 79 OR O
BLWP @VSBW WRITE O
*
LI R0,492 ROW 16 COL 13
LI R1,>5500 CHAR 85 OR U
BLWP @VSBW WRITE U
*
LI R0,493 ROW 16 COL 14
LI R1,>4E00 CHAR 78 OR N
BLWP @VSBW WRITE N
*
LI R0,494 ROW 16 COL 15
LI R1,>5400 CHAR 84 OR T
BLWP @VSBW WRITE T
*
LI R0,496 ROW 16 COL 17
LI R1,>3F00 CHAR 63 OR ?
BLWP @VSBW WRITE ?
*
LI R0,498 ROW 16 COL 19
LI R1,>5900 CHAR 89 OR Y
BLWP @VSBW WRITE Y
*
LI R0,499 ROW 16 COL 20
LI R1,>4E00 CHAR 78 OR N
BLWP @VSBW WRITE N
*****
SCAN CLR @ST CLEAR STATUS
BLWP @KSCAN KEYBOARD SCAN
MOVB @ST,@ST KEY PRESSED?
JEQ SCAN KEY NOT PRESSED
CB @KEYV,@YES KEY = Y ?
JEQ COUNT
BLWP @>0000 GOTO COLORBAR
*****
COUNT BL @CLS CLEAR SCREEN
*
LI R0,366
LI R1,>2C00 CHAR 44 OR COMMA
BLWP @VSBW WRITE COMMA
*
MOV R3,R5 R5=48
VL LI R0,364 ROW 12 COL 13
MOV R5,R1
SWPB R1
BLWP @VSBW WRITE 10,000'S
*
MOV R3,R6 R6=48
WL LI R0,365 ROW 12 COL 14
MOV R6,R1
SWPB R1
BLWP @VSBW WRITE 1,000'S
*

```



```

XL   MOV R3,R7      R7=48
     LI R0,367     ROW 12 COL 16
     MOV R7,R1
     SWFB R1
     BLWP @VSBW    WRITE 100'S PLACE
*
     MOV R3,R8      R8=48
YL   LI R0,368     ROW 12 COL 17
     MOV R8,R1
     SWFB R1
     BLWP @VSBW    WRITE 10'S PLACE
*
     MOV R3,R9      R9=48
ZL   LI R0,369     ROW 12 COL 18
     MOV R9,R1
     SWFB R1
     BLWP @VSBW    WRITE 1'S PLACE
     INC R9
     MOV R9,R2
     S R4,R2       ZL LOOP END?
     JLT ZL        DO AGAIN
*
     INC R8
     MOV R8,R2
     S R4,R2       YL LOOP END?
     JLT YL        DO AGAIN
*
     INC R7
     MOV R7,R2
     S R4,R2       XL LOOP END?
     JLT XL        DO AGAIN
*
     INC R6
     MOV R6,R2
     S R4,R2       WL LOOP END?
     JLT WL        DO AGAIN
*
     INC R5
     MOV R5,R2
     S R4,R2       VL LOOP END?
     JLT VL        DO AGAIN
*
     B @RSTRT     TO RESTART
*****
CLS  LI R1,>2000   SPACE CHAR 32
     CLR R0
     BLWP @VSBW    WRITE SPACE
CLS1 INC R0
     INC R0        NEXT SCREEN PLACE
     CI R0,>0300
     JNE CLS1
     RT
*
SLAST END

```

### TURBO PASC 99

```

{ COUNT 99999-T TP99 }
{ TURBO PASC 99 - L L Conner }
{ Stephen J. Shaw, U.K. TI Users Group }
{ January, 1988 }
{ TP99 run time 2 min 48.0 sec }

```

```

PROGRAM count;
LABEL exit;
VAR v,w,x,y,z : INTEGER;
     k : STRING[1];
BEGIN
  cls;
  k:="Y";
  WHILE (k="Y") DO
    BEGIN
      cursor(16,11);
      write(chr(67));
      write(chr(79));
      write(chr(85));
      write(chr(78));
      write(chr(84));
      write(chr(32));
      write(chr(63));
      write(chr(32));
      write(chr(89));
      write(chr(78));
      read(k);
      IF k<>"Y" THEN GOTO exit;
      cls;
      cursor(12,15);
      write(chr(44));
      FOR v:=48 TO 57 DO
        BEGIN
          cursor(12,13);
          write(chr(v));
          FOR w:=48 TO 57 DO
            BEGIN
              cursor(12,14);
              write(chr(w));
              FOR x:=48 TO 57 DO
                BEGIN
                  cursor(12,16);
                  write(chr(x));
                  FOR y:=48 TO 57 DO
                    BEGIN
                      cursor(12,17);
                      write(chr(y));
                      FOR z:=48 TO 57 DO
                        BEGIN
                          cursor(12,18);
                          write(chr(z));
                          END;
                        END;
                      END;
                    END;
                  END;
                END;
              END;
            END;
          END;
        END;
      END;
      exit;
    END.

```

### C99

```

/* COUNT 99999;C TI-99/4A c99 */
/* WESLEY R. RICHARDSON DEC 1987 */
/* BLUEGRASS 99 COMPUTER SOCIETY, INC */
/* c99 RUN TIME 1 MIN 28.0 SEC */

```

```
#include dsk1.conio
```

```

main()
{ /* COUNT */
int v,w,x,y,z;
char ykey;

putchar (FF);
ykey=89;
while(ykey=89)
{ locate(16,11);
putchar(67);
putchar(79);
putchar(85);
putchar(78);
putchar(84);
putchar(32);
putchar(63);
putchar(32);
putchar(89);
putchar(78);
ykey=getchar();
if (ykey!=89)exit(7);
putchar(FF);
locate(12,15);
putchar(44);
for(v=48;v<=57;v++)
{ locate(12,13);
putchar(v);
for(w=48;w<=57;w++)
{ locate(12,14);
putchar(w);
for(x=48;x<=57;x++)
{ locate(12,16);
putchar(x);
for(y=48;y<=57;y++)
{ locate(12,17);
putchar(y);
for(z=48;z<=57;z++)
{ locate(12,18);
putchar(z);
}
}
}
}
}
} /* end COUNT */

```

#### FORTH:

```

SCR #48
0 ( 99999-F TI-99/4A FORTH -TEXT )
1 ( WESLEY R. RICHARDSON DEC 1987 )
2 ( BLUEGRASS 99 COMPUTER SOCIETY, INC. )
3 ( FORTH RUN TIME 1 MIN 36.5 SEC )
4 BASE->R DECIMAL : IT ;
5 : ZL 58 48 DO 18 12 GOTOXY I EMIT LOOP ;
6 : YL 58 48 DO 17 12 GOTOXY I EMIT ZL LOOP ;
7 : XL 58 48 DO 16 12 GOTOXY I EMIT YL LOOP ;
8 : WL 58 48 DO 14 12 GOTOXY I EMIT XL LOOP ;

```

```

9 : VL 58 48 DO 13 12 GOTOXY I EMIT WL LOOP ;
10 : CNT 11 16 GOTOXY 67 EMIT 79 EMIT 85 EMIT
11 78 EMIT 84 EMIT 32 EMIT 63 EMIT 32 EMIT
12 89 EMIT 78 EMIT KEY ;
13 : CNTY CNT 89 = IF CLS 15 12 GOTOXY 44 EMIT
14 VL ELSE QUIT ENDIF ;
15 : START CLS BEGIN CNTY AGAIN ; START R->BASE

```

#### T.I.99/4a USER GROUP(UK)

#### ANNUAL GENERAL MEETING 1992

The AGM has been fixed for Saturday May 16th and will be held at the:-  
Princess Anne Training Centre (St John Ambulance Training Centre)

10, TRINITY STREET, DERBY

The centre will be open for exhibitors at 9.00am and ready for business at 10.00am. As usual, entry will be free, and an invitation is extended to any TI enthusiasts, Groups, or organisation wishing to attend or to mount an exhibit.

Anyone intending to display items please contact the General Secretary, who will explain the procedure for using the very convenient loading facilities, and arrange for you to be met.

Visitors will meet no difficulty in finding the venue. Reference to the map of central Derby printed in the TI\*MES Issue no 20 is ideal.

Entering Derby on the A6, on the railings outside the Derby Royal Infirmary you will find a notice pointing to Trinity Street opposite the DRI. Driving down Trinity St, watch out for the hall clearly labelled on your right, immediately before a sharp left curve.

You will find that there are adequate parking facilities, and if arriving by train or bus, you will find it a very easy walk to the venue indeed.

The hall is a nice size, less than the last one used in the city for our AGM, but rather bigger than we usually have for our AGMs of late.

A bar will be open during licensins hours, and soft drinks will be available at all times (supplied by Phil Potter) and a kitchen is included from which tea will be available (supplied by my wife at a nominal cost), and hot water if required.

The area has many restaurants, cafes, shops offering snacks, and a fish and chip shop in the immediate area. I look forward to seeing you all at Derby. The meeting is scheduled to close at 4.30 pm, but extra time can easily be arranged if required.