# TI\*MES

# T199/4A USER'S GROUP (U.K.) CONTACTS

Chairman: Trevor Stevens. Tel.0623 793077 249 Southwell Rd. East, Rainworth, Notts, NG21 OBN

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: Philip Trotter. Tel.0642 817356

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

Membership Secretary: Alasdair Bryce. Tel.0389 65903

51, Dumbuie Ave., Silverton, Dumbarton, Scotland. G82 2JH

Treasurer: Alan Rutherford. Tel.0625 524642

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", Cemetary Rd., Rhos, Wrexham, Clwd. LL14 2BY

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

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

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

Disk: Stephen Shaw. (Journal Exchange)

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

Publications: Mike Curtis. Tel. 0209 219051

21, Treliske Rd., Roseland Gdns., Redruth, CornwallTR15 1QE

# MAGAZINE CONTENTS

IFC Editorial, AGM, Disclaimer, Date for copy for next issue.

pl Membership News by Alasdair Bryce.

2 Module Library Report by Edward Shaw.

3 Cassette Library Report by Nicky Goddard.

4 Mathematical Problems You Can Solve With Your TI99 by Francesco Lama.

16 Drivel by Mark Wills.

19 Software Company Advertisement by Mark Wills.

20 Software Report by Stephen Shaw.

22 Adding Sound by Mike Goddard.Letter: Assistance Wanted by Peter Hutchison.

24 Hardware Advertisement by Mike Goddard.

25 Review Time by Trevor Stevens.

26 & 42 Disk Library Report by Stephen Shaw.

30 No Comment by Stephen Shaw.

37 You Don't Have To Have It All by Jim Peterson(sjs)

39 Machine Code Tutorial by Mack McCormick(sjs)

41 What's in a Module by Mack McCormick(sjs)

43 Mandelbrot Patterns by Ken Hughes(sjs)

45 Crazy-XB by Wesley R. Richardson(sjs)

49 TipsFrom Tigercub by Jim Peterson(sjs)

52 Care For Cassettes by Stephen Shaw.

55 High Resolution Plotter by Anon.

57 THe Monkey Typist by Walter Allum.

58 Trials of a Factorization Rourine by Walter Allum.

60 Letter from David Duncan.

IBC AGM Location by Stephen Shaw.

ISSUE NO.36

**SPRING 1992** 

# EDITORIAL

Here is the Spring edition of your magazine for your enjoyment! It is a communissue, and once again we are fortunate in having a large contribution from Stephen Shaw. He not only provides us with his selection of articles from the magazines of other groups, but also with his own programs and notes. For this issue he had trouble withthe margins on his own printer, but fortunately walker came to the rescue by printing from Stephen's disk. Unfortunately not have a disk drive controller, so was unable to do it.

As you will see this issue also contains contributions from several of our members, which are more than merely welcome! I worry that we may have to recommend with fewer contributions from overseas, so that we either have to provide our own or make do with a shorter magazine. Since for many of our members the magazine is a major link with the world of the TI99, such a development would be a great pity.

Whilst any contributions are welcome, may I perhaps serve my own interest by suggesting that I for one would be very glad to see an elucidation of the involved operating system under which it labours.

# AGM IN DERBY

The 1992 AGM has been arranged in Derby on 16th. May, at 10 Trinity St. Please see the map provided by Stephen. The Agenda is as follows:

- 1. Apologies for absence.
- 2. Minutes of last AGM (see TI\*MES Summer 1991.)
- 3. Matters arising.
- 4. Reports from Officers. (see list on front.)
- 5. Nomination & Election of Officers for the year.
- 6. Addendum to Constitution.
- 7. B.A.C.C. Membership.

The Show will open at 10am, and there will be demonstrations and equipment to enjoy before and after the formal business of the AGM itself. In order to allow for late arrivals owing to travel difficulties, this latter will start later.

#### DISCLAIMER

The views expressed by contributors to this magazine are those of each contributor and not necessarily supported by the Committee. Contrary opinions are welcomed and will be given space if at all possible. We acknowledge here anything not specifically acknowledged in the text.

#### NEXT COPY DATE

Please provide all copy for the next, Summer, issue of the magazine by lat.June. The most important layout requirement is the width limit of 180mm. I have had copy returned by the printers as too faint, and since they are remote from me the necessary substitution took time and led to a late issue, so please keep it black, or at least a good dense grey!

# TIUG(UK) MEMBERSHIP NEWS

# cassette is breat by Alasdair Bryce

What a time to start! This is my first effort at getting the membership list prepared for a print run and it would have to be the issue where 80 of you are due to renew your subscription. I sincerely hope that most, if not all, of you do sign up for another 12 months or else yours truly will likely get the blame.

Having been a member of the group for nearly 7 years myself now I'm happy to be able to make a positive contribution at long last. To that end I have to thank firstly Stephen Shaw for informing me of the vacant position and secondly Trevor Stevens for his invaluable assistance in providing the programming which runs the group database on TI-BASE v4.1 and for his subsequent patience in helping me with my teething problems. Those of you who already use TI-BASE will be well aware of just how powerful the package is and those who don't should get it now, assuming they have the necessary 32K and disk drive that is.

On the membership front since issue 35 we have been joined by two new members, the first of whom is Suzanne P. Davies — welcome aboard Suzanne. The second is not so much a new member as a late renewal as Mr. A.D. Jones was an original member of TI Exchange back in the early days but let his membership lapse a few years back. So welcome back Mr. Jones and I hope you'll stay with us for some time to come.

Renewals have been coming in steadily. As I said at the start this is the time of year when most of you are due to renew so keep those renewal forms flooding in please. There are a few members who haven't renewed in time to receive issue 36 but as they won't be reading this article any effort to cajole them into renewing does seem a little pointless.

On a personal note it has been a pleasant experience to have personal contact with a number of TI enthusiasts who have phoned me looking for information on the group or to look for a buyer for TI equipment. Your comments on renewal forms are also most welcome so don't be shy. Suggestions and constructive criticisms are appreciated and all comments will be passed on to those who are best able to deal with them. To re-iterate Trevor's comments from the last issue we are able to put you in touch with other TI users in your area if you wish to start up your own local groups so please feel free to contact me for details; with S.A.E. and stamp please.

I hope I've covered all the salient points but as it's my first attempt please bear with me. I look forward to meeting as many of you as possible at Derby for the A.G.M. in May and until then happy programming.

MAII GUARDIAN (J)

# MODULES MODULES MODULES MODULES

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
	ADVENTURE and PIRATE TAPE	5.00	NUMBER MAGIC	4.00
	ALIEN ADDITON	4.00	PARSEC	4.00
	AMAZING	3.50	PACMAN	4.50
	BLACKJACK + POKER	2.50	PROTECTOR	4.50
	BEGINNING GRAMMAR	3.00	SHAMUS	4.50
	CAR WARS	4.50	OTHELLO	4.00
	CONNECT 4	3.50	THE ATTACK	3.5Ø
×		3.00	LOGO I (NO MANUAL)	4.00
*	DISK MANAGER II	3.00		
	EARLY READING	2.50	SPEECH EDITOR	4.00
¥	EDITOR ASSEMBLER / MANUAL	22.50	* TI WRITER / MANUAL	12.00
*	EXTENDED BASIC AND MANUAL	22.50	* TI WRITER MODULE ONLY	2.00
	PERSONAL RECORD KEEPING	3.50	TI INVADERS	.3.50
	PERSONAL REPORT GENERATOR	3.50	TERMINAL EMULATOR II	5.00
	HANGMAN	3.00	TOMBSTONE CITY	3.00
	HUSTLE	4.00	WUMPUS (HUNT THE)	3.5Ø
	HOUSEHOLD MONEY MANAGEMENT	3.00	VIDEO GAMES I	3.50
	HOUSEHOLD BUDGET MANAGEMENT	3.00	VIDEO CHESS	4.50
	HOME FINANCIAL DECISIONS	3.00	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  $4\emptyset$  pence per module.

Application to loan/purchase modules.

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

Since the last issue I have managed to sort through the disks and complete the cassette library list.
Unfortunately I found that several of the programs in the current library list (printed in last issue) are in a disk fromat that is unsuitable for cassette saving. These programs are the following.

	menter titter	G104 MARS MINE LANDER
EIV	MORSE CODED TUTOR	6104 MAKS MINE CHNDER
619	PACKGAMMON	G106 MR. D
650	CONE	G120 POKER
1966	FIRELADY	G125 RAGING RIVER
	FROGLET	G128 ROBIN HOOD
	GEM GRABBER	G129 ROLL 5
	BOLF	G136 STAR PROBE 9
	A KAMIKAZEE	U35 PLANNING CALENDER

You may think oh god what will happen next. Well what will happen next is that I will tell you about the 61 new additions from the 12 disks mentioned earlier.

These new additions will become part of the new cassette library list available 2:6:92 onwards but they are available under their new addition code until 2:6:92.

You can still order any programs from the current list apart from those shown above until 2:6:92 to avoid any confusion.
The first 21 require 32k memory and can only be loaded via Extended

Basic. MARI ADVENTURE MODULE (K) NA12 HOPPER (J/K) NA13 KABOOM (J) NAME A-MAZE-ING (J/K) NA14 PADDLE (J) NAMS ANT EATER (J/K) NA15 PIANO (K) NAM4 CARWARS (J/K) NA16 SHAMUS (J) NAWS CAT AND MOUSE (J) NA17 ST NICK (J/K) NAØ6 CENTIPEDE (J) NA18 STARFORCE (K) NAM7 CONNECT FOUR (K) NA19 THE MISSING LINK (K) NAØ8 CROSS FIRE (J) NAWS DRIVING DEMON (J) NA20 VEGAS (K) NA21 WORM ATTACK (J/K) NAIU FROG (J)

The next 35 are games. 32k NOT required. NA35 GOPHER (EX J/K) NAZZ BOWLS (B K) NA36 HALLOWEEN (EX J/K) NAZS CARS+CARCASSES (B K) NAZ4 CHECKERS (B K) NA37 HANG GLIDER PILOT (EX K) NA38 LIFT ATTENDANT (B/EX K) NA25 COMMANDO (B K) NA26 CUI-OFF (EX J/K) NA39 MAEUSE LABYRINTH (EX J/K) NA27 FEM ON THE GRID (EX K) NA40 MELTDOWN (EX J/K) NAZ8 FERNANDO (EX K) NA41 MINE MAZE (B/EX K) NA29 FLIP FLAP (EX J) NA42 MOOS (EX K) NASO FRAGGLES (EX K) NA43 MOTH MANIA (EX J) NA44 MUNCHER (B/EX J) NAST FROGGER (EX.K) NA32 GANGSTER ALLEY (EX K) NA45 NINEPIN BOWLING (EX K) NA46 NOT ONE (EX K SP(O) NA33 GARBAGE BELLY (EX J) NA47 RACING (EX K) NA34 GOLF/GEER (EX/K)

NA48 ROCK STORM (EX J SP(D)) NA53 SWITCH (B/EX K) NA49 SLITHER (B K) NA54 THE ZONE (B K) MASM SPRING HEELED JACK (EX K) NASS THT (EX K) NASI SPYS DEMISE (EX J) NASA TUNLVISION (EX K)
NASZ SWEATCLOTH (EX K) NAS7 UP SCOPE (EX K)

The last 4 are Education, but NA59 and NA 60 are more suited for young

NASH COMPUTER ALPHABET (B K) NA60 SHUTTLE (EX K) NASY MATH MUNCHER (B K) NA61 TI99ER CRAYON (B/EX K)

All of the New Additions are available at the usual price. B = BASIC EX = EXTENDED BASIC J=JOYSTICK K=KEYBOARD SP=SPEECH (U) = OPTIONAL.

Cassette library prices are the same as usual but if postage goes up any more so will the cassette library prices. Surplus stock cassettes. I have some surplus stock cassettes but you would end up paying usual cassette library prices by the time postage is added on. I will take them to any computer meeting I go to though. Response to the cassette library list being published has been good. I have had several orders but only one person took advantage of the offer published in the last issue of TI\*MES.

All cassette library programs are now available on disk. Any format up to D/Sided D/Density. Please remember that I can fit about 9 programs on a S/Sided S/Density disk, 18 programs on a D/Sided S/Density or a S/Sided D/Density disk and about 35 programs on a D/Sided D/Density disk, to save on postage. 5 1/4 inch disks only please. Prices are £1 if you send a disk £2 if you do not.

Thats about all for now see you next time.

CASSET	TE	REV	IEW	S.		æ		2

----NICKY GODDARD

\* = DREADFUL

\*\* = AVERAGE

\*\*\* = GOOD \*\*\*\* = VERY GOOD

\*\*\*\* = EXCELLENT

BILLY BALL TO THE RESCUE AND PROPERTY OF THE RESCUE Yes here he is again good old billy ball after just having recovered from catching hearts etc he is here once again not to just move about, catch hearts and clobber the little greeny, he is here on a dangerous mission to save someone/thing(I haven't got that far yet!). You start off on the left hand side of the screen facing 3 errm things you have to jump over without them hitting you, You also have to be beware of a nasty little thing above you which if hit when jumping over something causes you to lose a life. The next level is about the same but more difficult. The third level is VERY difficult you have literally hundreds of the thingys flying above you and you have to jump over one to get to you flight up to the top of the wall but carefull just because you are in your flight up to the top doesn't mean that you are immune to the hundreds of thingys flying above ground, choose you time carefully. Afterwards if unknown... Where you are in relation to the wall is shown on the top right hand side of the screen. Keys are A for left L for right Q to jump left P to jump right. A very good extended basic game. STAR RATING \*\*\*\*

Francesco Lama, 14 Granville Court, Cheney Lane, Oxford OX3 0HE. sind out when a function

# MATHEMATICAL PROBLEMS YOU CAN SOLVE WITH YOUR TI-9914A

I Think we all must remember saying "I can't solve that equation!", during our school days. On some occasions a lot might have depended on our being able to do so, but our memory failed us .... we couldn't think of a way!!!!. More often than not we had been given a problem such as:

The interpolation is a decimal to 2 . The interpolation is a second second in the second second in the second second is a second sec

which simply means "Find those Values of X for which this is true" and we either couldn't call to mind the formula which gave us the answer, or our memory had played a trick on us and had put a minus sign where there should have been a plus and viceversa.

However, one way or the other, we all managed to scrape through our school examinations and some "masochists" like me even decided to take

up solving complex mathematical problems as a profession!?!?.

What I am going to describe below are a pair of simple logical algorithms which one is forced to use when the equations one is trying to solve are "transcendental", a big word which means too difficult to find a formula which will give the answer. I must also add that I claim no originality for the methods which have been known for hundreds of years (in times when there were no fast computational ways of inplementing them), and are now to be found in the maths libraries of most pocket calculators.

The first idea one needs to introduce is that of a function of a single variable "X" ; this can be approximately done by saying that f(X) = any combination of the functions present in the computer maths library involving amy number of constants but only one variable "X". For example:

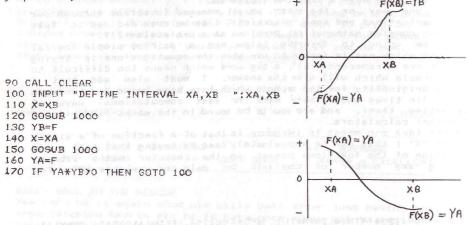
F = SQR(123\*(ATN(4.24\*X)))/0.3695\*X+LN(254.1\*SIN(4\*SQR(34.78\*X)))

The second concept which is needed for the introduction of the methods below is the one of continuity of the function, i.e. for all practical nurposes that nothing funny should happen to it when we try to calculate it like LN of zero or of a negative number, division by zero, SOR of a negative number etc., in other words the function F should be "well behaved" in the region of values of X for which we are trying to calculate it. ample sileogop and notional and recipion XB). The general aim of this procedure is to narrow dayn the interval

want, to know the result!, which you are asked to enter at line 172.

One of the most common problems in physics as well as mathematics is to find out when a function is equal to a certain number, e.g. F=4.532 which is equivalent to determining when F-4.532=0, thereby reducing the problem to finding when a certain function equals zero (it sounds very much like the questions you were set at school doesn't it?!?). So far I have been trying to debunk the complexity of mathematics and it may have been all rather boring, but I am sure you will find the following..... even more boring!!!!

The logical procedure to find the value of X for which a function F is zero relies on the definition of an interval of X-values (defined by its two extremes XA and XP, either of them can be larger) along which F changes sign, i.e. goes from negative values at XA to positive values at XB or viceversa. Therefore lines 90-170 of the program (given below) were written, in order to ensure one of the two situations shown graphically below.



Otherwise it simply requires you to enter a new pair of limits! !!!

The next section of the program can take one of two forms: one is called the bisection method and the other is called Newton's method, the two simply differing in the way they estimate where the zero of the function is likely to be given the two extremes XA and XB. The former simply calculates the value of F in the middle of the interval XA-XB (i.e. at the piont (XA+XB)/2) and if it finds the function to have the same sign as at the point XA it assigns the value of the midpoint to XA otherwise to XB (remember the function has opposite signs at XA and XB). The general aim of this procedure is to narrow down the interval over which a zero is to be found still maintaining XA on one side of zero and XB on the other until the absolute value of the difference between XA and XB is less than the REQUIRED PRECISION (with which you want to know the result), which you are asked to enter at line 172.

Newton's method chooses to approximate F with a straight line between the points XA and XB and finds the point where this straight line is zero. For this purpose it not only has to remember XA and XB but also the values of F at XA and XB (in the program they are called YA and YB). It then takes the point X where this straight line crosses zero as the new intermediate point in the interval XA-XB (directly replacing the midpoint to all intents and purposes).

The two programs only differ in line 410 because of the different chices of intermediate point and in lines 180 and 210 of the main program because of the neccessity for Newton's method to remember the values YA=F(XA) and YB=F(XB).

The latter method is faster at converging to the required accuracy and both are widely used in scientific numerical computations. Sorry if I bored you to death!!!!! (the remaining lines of the programs (written in Extended Basic)follow).

```
172 INPUT "REGUIRED PRECISION ":E
172 INPUT "REQUIRED PRECISION ":E
                       175 GOSUB 400
180 IF Y*YOLD>0 THEN XA=X :: GOTO 175 180 IF Y*YOLD>0 THEN XA=X :: YA=Y :: GOTO 175
                       190 XB=X
190 XB=X
                       200 GOSUB 400
210 IF Y*YOLD>0 THEN XB=X :: GOTO 200 210 IF Y*YOLD>0 THEN XB=X :: YB=Y :: GOTO 200
                       220 XA=X
                       230 GOTO 175
230 GOTO 175
240 PRINT X;Y
                       250 END
250 END
                       400 YOLD=F
400 YOLD=F
                       410 X=XA-YA*(XB-XA)/(YB-YA)
410 X= (XA+XB) /2
                       420 GOSUB 1000
420 GOSUB 1000
                       430 Y=F
430 Y=F
                       440 IF ABS(XB-XA) (=E THEN 240
440 IF ABS(XB-XA) <=E THEN 240
                       450 RETURN
450 RETURN
                       1000 F=((X+1)^10-1)/X/(1+X)^10-7.889
1000 F=((X+1)^10-1)/X/(1+X)^10-7.889
                       1010 RETURN
1010 RETURN
```

The subroutine at line 1000 is user definable and is nothing but the function F. The one given here is only an example and if you run either program with XA=.04 or less and XB greater than .05 you will find a zero at .04561 or thereabouts depending on the precision you choose. The second number which appears with this answer is the value of the function F at that point.

# LINEAR SYSTEMS OF SIMULTANEOUS EQUATIONS

Some of you may remember my previous piece on finding the zeros of a function and may well have been bored to death by it. If so please avoid reading this new, rather heavy going effort of mine! It concerns the rather general problem of solving a linear system of "n" simultaneous equations in "n" unknowns (variables to be determined). where "n" is an integer greater than or equal to 1.

In order to define what the above mathematical jargon means I shall give a concrete example of a simple problem which can be solved by the use of a set of simultaneous equations. Let us imagine three people need to make a payment P=\$800 out of their respective earnings E1=\$500, E2=\$600, and E3=\$1000. However the fraction each of them pays is subject to the conditions that the sum of the income of the first and the second person, after payment, must be an amount A1=5700, and that the income of the second person, after payment, must equal an amount A2-1300. The above statements can be written in mathematical form as follows:

E1 \* X1 + E2 \* X2 + E3 \* X3 = P

E1 \* (1-X1) + E2 \* (1-X2) = A1

TI. MOTSICHA GHAIGSER, ANGRE ELE SEL MOTSICHA GHAIGESE

where X1, X2, and X3 represent the respective fractions of earnings to be paid cut. In order to separate the unknown quantities from the known ones the above can be re-written as follows:

E1 \* X1 + E2 \* X2 + E3 \* X3 = P

E1 \* X1 + E2 \* X2 = E1 + E2 - A1

E2 \* X2 = E2 - A2

The above equations are called simultaneous because the same set of values X1, X2, and X3 must satisfy all three equations. The general form of a system of three simultaneous equations in three unknown.

A11 \* X1 + A12 \* X2 + A13 \* X3 = B1

A21 \* X1 + A22 \* X2 + A23 \* X3 = B2

A31 \* X1 + A32 \* X2 + A33 \* X3 = B3

where all the A's and B's are given constants (numbers) and X1, X2, and X3 are the variables to be determined so that all three equations are satisfied simultaneously. It may therefore be apparent to many of you (if I have not managed to confuse you completely) that the system generated by the little problem at the beginning of the article is the same as the above one with A11=A21=E1, A12=A22=A32=E2, A13=E3, A23=A31=A33=Q, B1=P, B2-E1+E2-A1, and B3=E2+A2.

Substituting the values of the constants into the equations one

500\*X1+600\*X2+1000\*X3=909 500\*X1+600\*X2=400

UKKZ=400

600\X2≈300.

From the last of the equations one gets:

which, when substituted into the second gives:

and therefore using these last two results in the first equation:

x3=0.4.

The solution of this set of simultaneous equations was made particularly easy by the fact that only one variable (x2) was contained in the last equation, only two in the second, and all three finally appeared only in the first equation thereby making the method of substitution used above a viable option. However it may be clear to all readers who have not dozed off yet that the general set shown shove (the one with the A's, and the B's) would not have been as easy to solve there being no obvious point where to start since all three equations look as difficult as each other.

There are a large number of methods for solving systems of "n" simultaneous equations in "n" unknowns (the X's), but one of the most powerful and which can easily be implemented for use in a computer is due to Karl Friedrich Gauss, German mathematician, astronomer, and physicist (Genius!!!!), (1777-1955). Born of a poor family this unusually clever man first came to the notice of his primary school teacher who, one day, feeling rather fed up with the whole thing (nothing seems to have changed, teachers still feel that way!!!!), decided to ask his pupils to do something that would take up the whole morning. He asked them to add up all numbers between I and 100. Our genius, Gauss, was not to be intimidated by the enormous task. He noticed that if you add 50 and 51 together you get 101 which is also the result of 47+52, 48+53 and so on untill you come to 1+100. The little brot also noticed that there are 50 (1.e. 100/2) of these pairs! hence in less than no time he obtained the result (501=5050). Not satisfied with this, he generalized it for the sum of integers up to any number (sum=n\*(n+1)/2). You can imagine how livid this made the teacher!!!!! However, his anger did not prevent him from signalling Gauss to the Duke of Brunswick who became Gauss' patron throughout his studies (sorry for the digression!). previous coefficients sulfigily, and the right-hand side stray coefficients "B"). The IMPUT at line 110 determines, has number at

Let us here state, without proof, a couple of facts about simultaneous equations. A set of "n" such equations in "n" variables (the only ones he are going to be concerned with) admits at least one solution; however if it admits more than one them it has an infinite number of solutions. In this case the system is said to be "Linearly Dependent", meaning that at least one of the equations is redundant in that it is already represented in the remaining ones (i.e. a sum of all or part of the remaining equations multimplied by appropriate constants is equal to this last equation). Gauss' algorithm is capable of determining whether a given set of equations is Linearly Dependent as we shall see in the extended basic program given below.

```
5 REM SOLUTION OF A LINEAR SYSTEM OF SIMULTANEOUS EQUATIONS
           100 DIM A(30,30), B(30), X(30)
          110 INPUT "NUMBER OF VARIABLES=":NX
       120 FOR I=1 TO NX
          130 FOR J=1 TO NX
          140 PRINT "A"; I; J; "=";
           150 INPUT A(I.J)
          160 NEXT J
          170 PRINT "B"; I; "="; constitute to the sid! to settiute
180 INPUT E(I) AND Wolde Alada dash eda Xd Xasa X(Tele
         180 INFO; E(I)

190 NEXT I

200 FOR I=1 TO NX

210 IF A(I,I)<>0 THEN 300

220 L=I

230 L=I +1
          240 IF LYMX THEN PRINT "LINEARLY DEPENDENT SYSTEM" :: STOP
          250 IF A(L,I)=0 THEN 230
          260 FOR M=I TO NX
           270 SWOP=A(L,M):: A(L,M)=A(I,M):: A(I,H)=SWOP
          280 NEXT M
290 SWOP=B(L):: B(L)=B(I):: P(I)=SWOP
          300 FOR J=1 TO NX
310 IF J=1 THEN 370
          310 IF J=I THEN 370

320 RATIO=A(J,I)/A(I,I)

330 FOR K=I TO NX

340 A(J,K)=A(J,K)-A(I,K)*RATIO
         340 A(J,K)=A(J,K)-A(I,K)*RATIO

350 NEXT K

360 B(J)=B(J)-B(I)*RATIO

370 NEXT J

380 NEXT I

390 PRINT
         400 FOR I=1 TO NX
410 X(I)=B(I)/A(I,I)
          420 PRINT "X"; I; "="; X(I)
          430 PRINT "X";I;"=";X(I)
430 NEXT I
          440 END
                                 to remember and the state of th
```

Line 100 of the program allocates space for the two-dimensional array "A" of the above of constant coefficients, the X-array which the previous coefficients multiply, and the right-hand-side array of coefficients "B"). The INPUT at line 110 determines the number of variables (the X's) and therefore the number of equations which are going to be present in the system (referred to as "n" earlier in the text).

Lines 120-190 are simply concerned with the input of all the constants (the A's and the B's). At line 200 begins Gauss' algorithm, Initially it checks whether A(I, I) <>0 (i.e. A(1,1), A(7,2), etc. up to A(NX, NX)). If such a condition is verified it immediately jumps to line 300. Lines 300-370 contain the two nested FOR-NEXT loops which are central to the method. The program subtracts all the constants in the Ith equation multiplied by "RATIO" from all the respective constants (notice in line 340 the second index K is the same in all three of the A's) of every other equation (except itself). RATIO being the particular number it is which depends on I and J as shown in line 320, the effect of this operation is eventually that of making every coefficient in the equations equal to 0 except for A(1,1) in the first, A(2,2) in the second, A(3,3) in the third, ..., A(NX, NX) in the last. By the laws of linear algebra this final set of simultaneous equations is entirely equivalent to the one we had at the beginning, i.e. the values of the X's for which the NX equations are satisfied are the same for both. There is however one great advantage in the latter system in that all the equations it contains can be solved independently of the values determined by the other equations just like in the case of the thick equation of the simple problem given at the beginning of this blurb!!. Therefore the X's are very easy to find (only a matter of dividing the term on the right of the "=" by the coefficient of the X!!!! The program does this for all values of I in lines 400-430 and prints the answers. You can check them for yourself with a calculator if you do not believe it (this can be done by putting the values of the X's supplied by the computer into all the original equations you have given it and verifying that the number you obtain on the left of the "=" is the same as the one you have on the right for each of the equations).

Lines 220-290 are concerned with the case of finding A(I,T)=0. This is clearly a nuisance since no matter what RATIO you multiply zero by, it remains ZILCH!!! All the program does in such an event is swop the Ith equation with one of the remaining "NX-I" equations for which A(I,I) is NOT ZERO. This operation clearly makes no difference to the system of simultaneous equation because the equations only appear in a different order (the conditions they express are all still present!). However, there may arise a case when it is impossible to find a NON-ZERO A(I,I) for that particular value of I. If any one could bear to read thus far I shall here state without proof that this is the case of a linearly dependent system and the answer prompted by line 240 will appear on the screen. No single solution is possible in this case because, as stated (by me if you believe me?!?!) in one of the above paragraphs, there are an infinite

I way be back soon with some more blurb on mathematical problems related to the solutions of simultaneous equations.... that is if no ITUGER will decide to prevent me from doing so before them???????!!!!!!

# POLYNOMIALS

In my long and tedious description of "LINEAR SYSTEMS OF SIMULTANCOLE EQUATIONS" I tried (and probably failed!!) to give a feel (or the sort of problem that simultaneous equations can help solve. Here is another example.

Let us imagine a shop wishes to stock two different items. Item number 1 costs \$100 and takes up two square metres of space in the store-room; item number 2 costs \$300 and takes up 1 square metre. Furthermore the capital available for the purchase of these items in \$2500 and the area available for storage is 25 square metres. How many of each item should the shop therefore buy to combine best of space with optimum use of capital? Put into a mathematical form the problem reads:

\$100\*x + \$300\*y = \$2500 and

sqm2\*x + sqm1\*y = sqm25. The same ballers are entitled to the

The above is a set of simultaneous equations for which the relative quantities of item 1 "x" and item 2 "y" have to be determined. If you feed the values of the coefficients into the program given in the previous article you will get the answer x=10 and y=5.

There are however other more mathematical applications of simultaneous equations, one of which I am going to try and describe here.

At the beginning of my piece on finding the zeros of a function (July 87 issue) I gave an example of a polynomial of second degree (the highest exponent of the "X" being 2). However a more general example of a polynomial is the following:

$$Y = An*X + An-1*X + ... + AZ*X + A1*X + A0$$

in which An, An-1, ..., A2, A1, and A0 are called the coefficients of the polynomial and simply represent numbers, different ones depending on the power of X which they multiply. "X" is, as usual, the variable.

An interesting mathematical problem is that of finding the values of the coefficients such that the polynomial passes through "n+1" given coordinates (pairs of numbers X,Y). Suppose, for the sake of argument, that the specific polynomial given is of second degree and that the three sets of coordinates given are (1,1), (2,3) and (3,4). In this case, substituting the coordinates into the above equation (one pair at a time) one obtains:

```
1 = A2*1 + A1*1 + A0

3 = A2*4 + A1*2 + A0

4 = A2*9 + A1*3 + A0.
```

Many of you will have already recognized that this is a linear set of simultaneos equations in which the variables to be determined are AO, A1, and A2. It is also clear that the multiplying constants are given by the powers of "X". Therefore the simple modified version of the linear simultaneous equations program given below can calculate the polynomial coefficients of the above problem.

```
100 DIM A(30,30), B(30), X(30)
          110 INPUT "NUMBER OF POINTS=":NX
          125 PRINT "X"; I; ", "; "Y"; I; " per aud " segre settles, butter such mer' ! better
          126 INPUT XX, S(I) as a same decade assertion will assert the underlying the same as
          190 NEXT I
          200 FOR I=1 TO NX
          210 IF A(I,I)(>0 THEN 300
          220 L=I deless and he vs . on si sevens sail "Faintee to hedron
          230 L=L+1
240 IF L>NX THEN PRINT "TWO IDENTICAL X'S!!!" :: STOP
           250 IF A(L,I)=0 THEN 230 Manha PA at beyil and antoing damage
          260 FOR MEI TO NX ASSESSMENT TO NX ASSES
          270 SWOP=A(L,M):: A(L,M)=A(I,M):: A(I,M)=SWOP
           280 NEXT Manager as the 9 of beyon total ad warman at a suples pro-
           290 SWOP=B(L):: B(L)=B(I):: B(I)=SWOP
          300 FOR J=1 TO NX / 250 SMI AL hevoleto evabourge was doing
           310 IF J=I THEN 370
           320 RATIO=A(J.I)/A(I,I) areas aid to seek belove one teller
          330 FOR K=I TO NX was sell to mus sell as as illent sel mas seasont
          nch derree which passes through not sulphain reversion
           360 B(J)=B(J)-B(I)*RATIO
           370 NEXT I
           400 FOR I=1 TO NX
           THE PURK I=I TO NX . THE PROBABILITY CAN A SOCIETY OF THE PROPERTY OF THE PROP
           420 PRINT "A";I-1;"=";X(I)
           431 PRINT "DO YOU WISH TO CALCULATE INTERMEDIATE Y VALUES Y/N";
           432 INPUT A$ 1 1 dadt stor gallen the bes I esseted shimers I has
           433 IF AS()"Y" THEN 440 a viscalo el molegarena evoda ent
           434 INPUT "X=?": XX saus dita betauntance at it you edd to seusced
 435 Y=0
436 FOR I=1 TO NX :: Y=Y+X(I)*XX^(I-1):: NEXT I :: PRINT "Y=";Y
           437 PRINT "AGAIN? Y/N"; aidf to dought warvan . 0 22 toubons
would have eade a much better lob of this explanations Turn 88440 ...
made 439 IF A$<>"N" THEN 434 Annual and avirab bas algobia ofsations
paid 440 END slosses istlace as at st (#.C) bas (E.S) (11,1) deponds
```

At line 110 you are asked to enter the number of points through which you wish your polynomial to pass; this also sets the degree of your polynomial to NX-1. Lines 125 and 126 are concerned with the input of the coordinates "X" and "Y" which are read into memory as "XX" and "B(I)" respectively. Note here that except for the input FOR NEXT loops between lines 120 and 190 the present program is identical with the one on linear simultaneos equations down to the variable names!!! The additional lines (481-487) enable you to calculate additional Y values (given arbitrary X's) for the polynomial whose coefficients you have calculated in the preceding part of the program. The condition described in the previous article as a LINEARLY DEPENDENT SYSTEM is brought about in polynomials by an attempt to use the same value of "X" more than once, a process which would create two or more identical equations as can be seen with reference to lines 130-160 and to the practical example given earlier on.

"This is all very well" you may say, "but must one Calways cuse this complicated method to find a polynomial that goes through a given number of points?" The answer is "no". By far the easiest method for this purpose was first thought of by one of the most famous mathematicians of the 18th century: Giuseppe Luigi Lagrange. Of French crigin, he lived in Piedmont (Italy) between 1755 and 1766 where he devoted himself to the study of variational calculus which is widely used today in solving complex quantum problems. After a long soigurn in Germany he later moved to Paris where, during him studies of celestial mechanics he developed the mathematical tools which are newadays employed in the most advanced theories on elementary particles. It is therefore no surprise that such a genius, who devoted some of his spare time to proving that any integer can be written as the sum of the squares of four integers, would come up with a much simpler way of computing a polynomial of nth degree which passes through n+1 points:

which means "SUM all the PRODUCTS multiplied by Yi for the indices and j ranging between 1 and n+1 making sure that i is never equal to j". The above expression is clearly a polymomial in the variable X because of the way it is constructed with sums and products; moreover it will pass through the point (XK,YK) (where K is a number between I and n+1) because when X=XK and i=K the product is 1 and when iCK the product is 0. Anyway, enough of this raving!! I am sure Lagrange would have made a much better job of this explanation!? Let us take a concrete example and derive the second degree polynomial which goes through (1,1), (2,3) and (3,4) as in an earlier example. According to

lagrange's formula we obtain:

Y=1\*(X-2)/(1-2)\*(X-3)/(1-3) + 3\*(X-1)/(2-1)\*(X-3)/(2-3) + 4\*(X-1)/(3-1)\*(X-2)/(3-2) = 2 -0.5\*X + 3.5\*X - 2

which, as you can easily verify by substituting in the X values and calculating the Y values, really does go through the three points given. It is unlikely that Lagrange had computing in mind when he devised this formula (he might have had, though; we'll never know!!!), but the fact remains that this neat piece of mathematics lends itself beautifully to programming and makes the nice little package that follows:

5 REM LAGRANGE INTERPOLATION POLINOMIAL

100 DIM X(30),Y(30)

110 INPUT "NUMBER OF POINTS=?":NP

120 FOR I=1 TO NP

130 PRINT "X";I;",";"Y";I;

140 INPUT X(I),Y(I)

150 NEXT I

160 INPUT "X-VALUE=?":XX

170 PX=0

180 FOR I=1 TO NP

190 PROD=1

200 FOR J=1 TO NP

210 IF I<>J THEN PROD=PROD\*(XX-X(J))/(X(I)-X(J))

220 NEXT J

230 PX=PX+Y(I)\*PROD

240 NEXT I

250 PRINT "P(";XX;")=";PX

260 GOTO 160

290 END

This is proof that a little grey matter goes a long way!!!!!! (I am clearly NOT talking about myself!!!!). Usual apologies about the state of the minds of my readers after wading through this new blurb are in order! SORRY!!!!

7.7.	1.7.	7.7.	4%	1/2	7/2	7/2	7.7.7.7.7.	%	1/4
1/2	7/2	%	%	1/6	%	%	1/4	%	1/4
%	%	%	%	1/2	7.	%	1/4	%	%
%	%	7.7.	12/2	%	.74	7.	7.7.7.	%	1/2
%	1/-	7.	1/4	%	1/2	1/2	%	7.	%
%	1/2	1/2	1/0	1/4	1/.	%	7.	7.	
70/0	77	47	7	=/	*/	7	7.7.7.7.7.	7.7.7.7.7.	%

Mark Wills waffles on about things in the TI line...

Well hello again everyone! I thought i'd better put pin to paper to let you know whats been going on. A reasonably interesting quarter for myself. I have managed to obtain both a RAM DISK and a GRAM KRACKER board for my TI PEB. Nice peices of kit. I am told they were both built by the rather clever Colin Hinson who works at TI in Bedford. All I can say is that he is one CLEVER chap! The Gram Kracker is one clever peice of hardware/software. You can actually copy all your modules to disk!! This I have done because it saves me having to put modules into the GROM port on the front of my TI which normally holds EXTENDED BASIC. My GROM port is now so tired that if I remove my XB module, it normally takes about twenty tries before I can make it work again!! You can see why I'd rather leave XB in place!

The RAM disk is rather nice as well. I have used it while working on programs in Basic/Extended Basic. I just save what I am doing to the ram disk every so often in case of a crash. Trouble is I can't get ANYTHING else to work with it! Disk manager refuses to look at it, so does DSK U. I expect there is a command that makes it compatable but I don't know what it is. I received very sketchy instructions. The only commands I know of are CALL PART(384,96) to partition the RAM Disk, and CALL EMDK(x) to make simulate a particular drive, and CALL RDDIR to obtain a directory. Any one out there know what I'm doing? I'd love to be able to put FUNLWEB or there, but at the moment i'm unable to. It has the MYARC rom in it. Any suggestions??!

Thanks go to Mike Goddard and his family in Wrexham for their hospitality in putting me up for the weekend. We had a rather great time just going through boxes and boxes of disks full of programs. Boy was there some interesting programs in there! One of the disks (I brought a big box of them home with me to play with!) has a disk in there entitled Basic Compiler. Homm sounds interesting. I wonder if it converts your programs to Machine Code or Psudo Machine code (both much faster than Basic)? If it does hat will be really something. I havn't had time to look at it yet, but I'll let you all know what it does. If it's any good i'll put it in the disk library for everyone to get there mits on. (And the cassette library if I can get it onto tape.)

# Report On Maidenhead Meeting...

Well, all I can say is i'm glad I went to it. I feel it was problably the most successful meeting since the Chester AGM way back. Can I take the time to pay tribute to Phil Trotter who came all the way down from way up north to Maidenhead and all manner of refreshments with him, at very good prices. Nice to see you again Phil.

There was a much larger attendance than I had gotten used to and I have to say I was pleasantly surprised to see all these people huddled around TI systems ogling the latest software. Talking of which...

Good old Trevor Stevens was there demonstrating a rather nifty GIF File converter from Barry Boone (of Archiver Fame) distributed by Texaments (I think), called GIF MANIA. GIF, i'm lead to belive stands for Graphics Interchange Format and is the standard used a lot on bulletin boards around the world to store picture files under. This program is nice because it takes GIF files and converts them to TI Artist or vice versa, at the same time taking account of colour pallete colour differences between different types of machines and thus approximating them accordingly so that what you get or send is a close as approximation of the original colours as possible. I think it also takes differing resolutions into account as well and can scale accordingly. A nice program. I shall write to Trevor and ask where I can buy it from because I have literally HUNDREDS of GIF files at work on my Amiga, I can convert to TI ARTIST and put them in the disk library or just send them out on to the MOBB for everyone who has a modem. Sendi tilk performater terror research a real ---

Richard Twyning gave me a nice demonstration of his Geneve. I plan to buy one this year hopefully. It has to be said, they are flippin' nice machines. They run probably about twice as fast as a normal TI and have mouse support plus eighty column text as well. He also demonstrated his Windows type Shell program to me, and it is very nice. Unfortuantely it only runs on the Geneve at the moment but hopefully if enough people badger him he will convert it to run on a standard TI, but it would have to be in machine code to run at any reasonable speed. Come on Richard!! My shell program is coming on nicely but its finding the time to do all these things isn't it?

News also of new graphics card for the TI offering eighty columns and very high resolution graphics was also forthcoming at the meeting. Again we have Richard Twyning and his collegue Gary to thank for that. (Sorry Gary, I didn't get your sir name!) Designed by Richard and Gary, it should really vamp up the graphical capabilities of our machine. I'm not going to say too much about that though as I feel its really up to Richard and Gary to publicise it when they feel ready. It is a genuine project though - they were even kind enough to show me the circuit diagrams.

I feel I must pay tribute at this point to people like Richard and Gary. It is really great to see two young, enthusiastic people working feverishly away to try and support the TI as best they can. Just when you begin to wonder "Is anyone bothering with this machine anymore" along come two such people (and others as well) to re vitalise those dwindling spirits and make you think "Yeah go for it!" Keep me posted lads. I'd be really interested to hear how your project is going - and if you think I can help you at all then all you have to do is get in touch. My work number is 0743 244752, my address is on the front cover.

I said in Drivel 1 that save a few of the quys around the country that the Brits are a bunch of TAKERS as far as supporting the TI goes. It s nice to that theres more than me and Mike Goddard trying do something about it!

While I'm on the subject of support, it really doesn't take much

effort. If you write a program that you think is any good. Lob it in the disk or tape library and lob an article across to TI\*MES at the same time. Havn't got a printer? No problem, write it down on paper and I will type it up myself and publish it for you. I can also see that the program is sent to the user groups around the world. All it takes is a little effort and the rewards can very heart warming indeed. You probably have no idea of how good it makes me feel to recieve a peice of software from someone who says "You might not find it of any use but here it is any way, just pass it around." It makes you think "Heres a guy who wants to share his software with the rest of the TI fraternity."...

I do tend to get carried away don't I? Ahem... Straighten Tie, check hair, look cool, carry on!...

It was nice to see the guys from DORTIG there (Dorset TI Group) and especially nice to see Trevor Taberner there, who has been a little under the weather recently. Drop us line when you feel a bit better Trevor. DORTIG have a demo program on disk that runs for about half an hour and is rather funny. The guys were kind enough to give me a copy. If you want a copy then write to: John Murphy, 65 Honeysuckle Lane, Creekmoor, Poole, Dorset, BH17 7YY and enclose a ready formatted blank disk plus SAE and J'm sure they'll give you a copy.

There was such a lot of chin waqqing in general going on that I regret I didn't get the chance to butt in on it all. But suffice it to say that it was surely one of most successful meetings for some time. And I only hope the AGM is as successful.

Thanks go to to Martin Cross in Maidenhead for putting Mike, Nicky and myself up and for forwarding my cables back to me after I had left (spot the terp!). Thanks a lot Martin. Drop us a line.

# Apology and Retraction... 19 19 984 Standard But was and

With reference to Drivel 1, I reported that the makers of the Geneve seem to be dormant and no machines were being shipped. I am happy to hear from Richard that the supply problems are soon to be overcome and we should be able to buy the machines again soon. Great news, and I apologise to Myarc if I have led any potential customers astray. Its a shame we cant make the machines over here under license. Hmmm...

# BASIC Compiler. ... If the story of the study of the stud

I can now report that I have tried the EASIC Compiler and It works. It is a Disk only package I'm afraid. It's written by Peter Krull (C)1984 so I think we can safely assume that the program will not offend anyone if we put it into the disk library. I compiled Creepy Crawlies with and it made it too fast to be playable! I shall forward a cappy to Steven Shaw ASAP.

Well thats it for now. I think i'm going to do some more work on my shell program... Keep on tapping those keys...

Mark Wills. Polynt brabbod as in bos es next eron estent dest of a Vice Chairman.

\*\* 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
Halp Barnaby the Buq 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 them 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

From NOTUNG SOFTWARE, an updated product list and some new products. I previously reported on TI CASINO and this has now been upgraded to Version 3, with improvements to Craps, Blackjack, Baccarat, and a stand up comic to amuse you when you feel jaded. And Notung Software offer previous purchasers a FREE upgrade, all you have to do is return your original disk in a stout mailer, with sufficient return postage.

TI Casino is US\$15, specify whether you have single or double sided drives!

NOTUNG SOFTWARE is Ken Gilliland, and can be found at: 7647 McGroarty Street, Tujunga, Ca, USA, 91042. (A beautifully painted wooden bungalow mostly built by Ken, complete with all-American swinging seat in the veranda!)

New: BRIDE OF THE DISK OF DINOSAURS US\$12, two disks. This set has 13 TI Artist Instances, at least 12 TI Artist pictures, an index of the three dinosaur disk sets available, a list of where to see dinosaurs – bang up to date with the new correct name for St Petersberg! – and of course a Thug animation. Five of the pictures are not merely graphics but seriously educational illustrating defensive strategies, defendisive behaviour etc.

Ken indicates that the animations do not like the Myarc disk controller, but are happy with TI and Corcomp controllers.

COMPLETE SET: Disk of Dinosaurs, Son of Dinosaurs and Bride of Dinosaurs just US\$25.

New: Disk of Horrors. US\$12. Remarkable price for a three disk set! Did I dare to suggest the disk of Pyrates previously released was a little dark? This gothic offering is entirely in the crypt without a candle!

What you get— three short stories by Ken, a brief history of pulp fiction, an animation Ken gives a PG-13 rating to (for violence), twenty TI Artist pictures, eight TI Artist Instances, and a TI Artist creepy Font. Plus Ken's very first TI program (a little rewritten) of music. One of the pictures is of a naked woman ice skating with an alligator, but alas the story to that one is missing!

# Also available:

Fonts and Borders- disks of assorted fonts and borders! Four disks available - disks 1, 2 and 4 are \$7 and disk 3 is \$8, or ANY three for \$20.

Kens Midi Favorites- files to be used with the new Midi hardware now available - a suitable keyboard is also required - two disks, 1 and 2, at just \$5 each.

In view of the low prices, you really should add on a few dollars extra for postage if you order! Four or five dollars would be nice.

# PAIN

This program will cause severe migraine headaches and possibly epileptic fits in anyone who has a tendency to such ailments (especially migraine which is optically induced), when viewed on a 14" screen from about two feet away — such a simple program with such power. BE AWARE and do not use this pattern in your game programs!!!!!!

110 CALL HCHAR ( 32 ,"AAAAAAAAAAAAAAAAA" )

120 CALL COLOR (1 ,2 ,16 )

130 GOTO 130

\_\_\_\_\_

Technically, the difficulty crops up when the frequency of the vertical lines is 3 cycles per degree of visual arc, and the contrast is about 70%.

THIS PROGRAM HAS A HEALTH WARNING. PAY HEED. In the event of any visual distress avert gaze quickly and break program. No responisbility is accepted for any damage or injury. The pattern is used as part of a diagnostic process.

References:
"Brain", #107. p989-1017. "A neurological basis for visual discomfort"
(1984)

"Archives of Neurology" Oct 1989. #46 p1129—1132."Migraine and stripe—induced visual discomfort" (The pattern was noted by British researchers in 1984).

I have been making noises about the potential for porn of computers since 1982 — not making a comment (no longer allowed!) but drawing attention to the possibilities and the weaknesses of the law. In early computing days a consignment of over hyped games software from the USA was seized by Customs, but had all the allure of Space

Invaders.

I have seen nothing since, despite the higher and higher screen resolutions, but noted in the Manchester Evening News of October 29th 1991 the headline "Computer Porn Squad grad discs in house raid". The porn squad may go to court and obtain a search warrant to enter and search premises, and may remove any material they wish — this is what the report referred to. They may not make an arrest, but apply

for an order requiring court attendance. In the article, we are told that 200 discs (sic) "thought to contain pornographic scenes" together with computer equipment, were taken by the police from a house in Bolton. A police spokesman is reported as having said that the use of computers for distributing pornographic material is increasing, the obscene publications squad has been investigating the growing trade for several months.

investigating the growing trade for several months. With high street heavies such as W H Smith and John Menzies selling extreme works such as The Story of O and extracts from The Pearl, it seems unlikely that any textual material will give rise to a prosecution, although the Obscene Publications Act WAS used locally against a book held to be obscenely racist. Note that obscenity is defined as being inclined to deprave or corrupt—sex is NOT mentioned!! This news is brought to you as a public service, as there seems to be a shortage of readily available serious computing press these days!

There are many TI owners using monitors that don't have a sound facility, this is because most monitors were made for use on computers that have built in sound however it is not difficult to connect a small audio amplifier to the audio/video or monitor connection of the console. There are several small amplifiers on the surplus market which are easy to obtain often at less than the component cost. This article will describe how to use the small amplifier currently sold by M.G.C.S at £1.50. The connections given are for the BRITISH PAL and AMERICAN NTSC system consoles and although the amplifier should work on other versions of the console the connections will be different.

first you will have to decide whether you are going to use the 12volt DC power supply avalable at the A/V socket or use a 9volt battery. If the 12volt supply is going to be used the amplifier will have to be modified as per the data sheet also if this option is used the amplifier will give over 2 watts RMS audio output and a suitably rated speaker will have to be used. Another point to remember is that if the speaker is to be mounted in or near the monitor be careful not to site it near to the picture tube as the speaker magnet can cause severe video distortion. The connections are as in the diagrams below and are fairly straight forward use screened audio type cable for the power and audio connections. Taking the power supply from the console has definite advantages over batteries as they will need to be replaced at regular intervals.

The whole unit can be fitted into a small plastic box with the speaker and controls, with flying leads coming out for the connections. If the console has a plug occupying the A/V socket simply solder the new connections to the existing plug connections and bring a seperate cable to the amplifier or simply fit a plug of the correct type to the cable going to the amplifier.

To use on batteries connect +ve and -ve to a suitable 9volt dry battery instead of the console plug, ignore all other connections on the amplifier circuit board.

Mike,

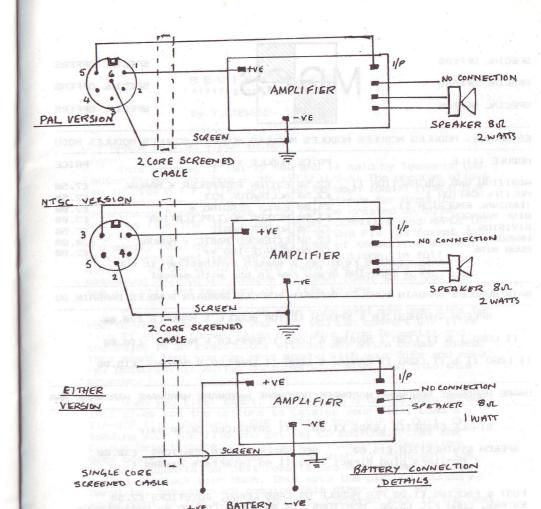
Dear Editor, si nemericus saling A modica in espor a sort saling of

I wonder if you could help me by bublishing this letter because I am desperately looking for some HELP on the Adventure Gnost Town. I have tried Stephen Shaw but I have got no further. The questions I have to ask are:-

- Where can I find the Indian Village and how do 1 get to it?
   (The hints book mentions 'Horse' and nothing eise!)
- 2. Where can I find the charcoal to make the gunpowger?
- 3. Where can I find the \*Chinese go board\* (is it in the sate?)?
- 4. The hints took mentions a fuse beyond the mountains (is this the other telegraph key)?

I have managed to find 7 treasures and managed to get into the jail and the cell and have visited the Teepee in the Hidden Canyon but I am having trouble opening the safe (to do with gunpowder I assume). If anyone has completed Shost Town please write to me soon'''

6 Moorlands View
Free School Lane Yours sincerely,
Savile Park
Hailtax KXI 20X Peter Hutchison



SPECIAL OFFERS

SPECIAL OFFERS

SPECIAL OFFERS



SPECIAL OFFERS

SPECIAL OFFERS

SPECIAL OFFERS

ES MODULES MOD

MODULE TITLE	PRICE	MODULE TITLE	PRICE
ADDITION AND SUBTRACTION II	£2.50	EDITOR ASSEMBLER + MANUAL	£7.50
MULTIPLICATION I	£2.50	ALLIGATOR MIX	£2.50
TERMINAL EMULATOR II	£4.00	EARLY READING *	£2.00
DISK MANAGER II	£3.50	METEOR MULTIPLICATION	£3.00
DIVISION I	£2.50	NUMBER MAGIC	£3.50
DRAGON MIX	£2.50	EXTENDED BASIC + MANUAL	£20.00
MOON MINE	£4.00	A-MAZE-ING **	£2.00
* Early reading red	quires	a speech synthesizer to run	

GAIN BUNDLES BARGAIN BUNDLES BARGAIN BUNDLES BARGAIN BU

\*\* A MAZE-ING Brand new in box with manual

SPEECH SYNTHESIZER & SPEECH EDITOR MODULE & MANUALS £20.00

TI LOGO I & TI LOGO I MANUAL & LOGO I SAMPLER & MANUAL £10.00

TI LOGO II & TI LOGO II MANUAL & LOGO II SAMPLER & MANUAL £15.00

DWARE HARDWARE HARDWARDWARE HARDWARE HARDWARE HARDWARE HARDWARE HARDWARE HARDWARE HARDWARE HARDWARE HARDWARE HA

SINGLE CASSETTE LEADS £1.00 TI JOYSTICKS £4.50 pair

SPEECH SYNTHESIZER £15.00 NEC GREEN SCREEN MONITORS £30.00 (Monitors plug direct into TI no interface needed )

POST & PACKING £1.00 PER MODULE OR CASS LEADS. JOYSTICKS £2.50
ED ASS. LOGO ETC £5.00. MONITORS £10.00 SECURICOR OR BY ARRANGEMENT

LLYS RALLYS RALLYS RALLYS RALLYS RALLYS RALLYS RALLYS RALLYS RALLYS

M.G.C.S WILL BE ATTENDING THE FOLLOWING RALLYS IN 1992.

APRIL 19th Centre of England Rally National Motorcycle Museum Birmingham MAY 4th Mid Cheshire Rally

AUGUST 23rd Red Rose Rally Bolton Sports Centre Silverwell Street Bolton SEPTEMBER 20th Centre of England Rally as above DECEMBER 13th Centre of England Rally as above

Mike Goddard Computer Support. "SARNIA", CEMETERY ROAD, RHOS, WREXHAM, CLWYD, LL14 2BY. TEL 0978 843547.. FAX 0978 845712

REVIEW TIME

By T.STEVENS 1991

GIF MANIA (\$14.95 + P&P \$8.50)

This program is fairly new and is sold by Texments (53 Center Street, Patchogue, NY 11772 U.S.A) The program allows you to view G.I.F files which can be down loaded from B.B.S Boards anywhere in the world. G.I.F stands for Graphics Interchange Format. These files can be created on other machines like the IBM. ATARI, AMIGA, COMMODORE and APPLE to name but a few. The files are all in Dis Fix 128 format and are easily read by our machine. Some of these file are out of this world and have very high resolutions, in full colour. In the World to date there are on files an estimated 100,000 GIF images for down load, so as you can see you will not be short of a file to use.

The program comes with some very good specimen GIFs for you to view. However if you wish you can convert GIF files into TI ARTIST format (Also available from Texments) and the image is an exact clone without any extra colour bleed. This means that you can mess around with your GIFs and make new pictures. TI ARTIST pictures will also load into this program.

When you first start, you boot up onto the main screen. This gives you the options to Catalog your disk, Load a TIA-GIF file, Convert a GIF to TIA, Show GIF or Quit. On loading your GIF file you get set up options. First the Colour Select where you can select intensity, Deviation, Grev Scales, or Monochrome. If you wish to print out, the booklet suggests for best results use Monochrome. The next option is Black Line Mode. This sets the picture boundary handling. Some picture require this other don't so experiment is the word of the day. Condense mode is the next option this allows you to have all or part of the picture on the screen at once. As you may already be aware some computers have the ability to show 640x400 pixels where as the TI can only show 256x192. With Condense ON, the original picture is compressed and placed into 256x192. In proportion I might add. If say you had a picture where you wanted a close up on one part you then leave this option off and go to the next option which allows you to set points up on the screen to view. These are called shift marks. I have a Picture of Neptune which has come from the voyager mission. When you scan onto the planet you can see the cross hairs of the grid reference camara. See (Fig 1 & 2)

The software is fully compatable with all Ram Disks, and Hard Disks, and can be run on a MYARC via EXEC. The system requirements are EXB or EA, 32K and Disk system.

The program is worth its money and gives very good results. However on my copy I have been unable to use the program via Editor Assembler. It loads OK but crashes. However EXB loads perfectly. A friend of mine has a MYARC and it loads into that OK too.

# SOUND FX (\$14.95 + P&P \$8.50)

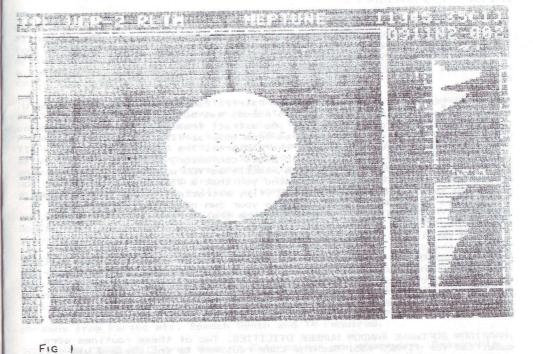
This program again from Texments. This is really new, it has a date on the front screen of 1-11-91. I think that it's possible that my copy was one of the first to be sold to the UK. This is a MEGA program. This is like nothing you have heard before. This program runs Digitized sound files, just like your compact disc does. There is no need for the speech synthesizer only a system with 32k and Disk.

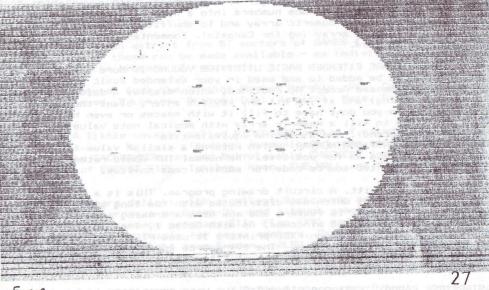
When you run the program for the first time you can't believe what you hear. I bought with my disk a collection of sound files (\$9.95). I loaded in the Cartoon sound disk and wow you hear the character and the back ground music from the film. It is difficult to describe sound to anyone so if you wish to see or should I say hear this in action I will have it at the AGM in Derby. Some of you will also have heard it at Maidenhead

The program when it runs has a master screen with five options. Load a FX File. Hear a FX file. Convert a FX File Disk Catalog and Quit . FX Files are also available in this country as are GIFs from BBS Boards, again there are plenty to choose from. If you down load a RAW data file from any IBM Soundblaster VOC file, Macintosh SND or Amiga SND file, you can convert them to our format FX by running the convert option. This can also help you once converted. If the file is slow when played you just reconvert and use the amplification option to bring up the speed. When you run the sound file the program also gives you the chance to slow or speed up the file in its loaded form. at add. If say you had a picture where you

# To moverview was increased and an analysis of paging and paging an

I love this little program but at the present sounds can only be used from it and you can't produce your own at





## DISK LIBRARY:

The following are some newish additions to the disk library. The disk library offers copies of these disks, which are almost all "fullish" for one pound per disk side (not per little program!) if you send the blank disk, plus an extra pound per order for handling, post, carriage

TIPS MANIPULATOR... version 2.1 from Patrick Powell. At last. a program to edit those huge TIPS files which nobody has bothered to send for.... this very useful program will let you extract from each TIPs file those graphics YOU want. collect pictures from several TIPs files into one file, sort that file, and rename the pictures. The TIPs files in the library will remain as original to avoid confusion. but with this utility you can order the graphics exactly as YOU want. As you don't have TIPs, just to remind you that a TIPs disk contains maybe 120 small graphics all in ONE file. extracted by the TIPs program as you need them. Now you can arrange your own graphics collections. TIPS and TIPS GRAPHICS are available from the library.

>FUNLWEB Version 4.40, on FOUR DISKS, partly archived. This set INCLUDES the 80 column files for 9938 vdo owners, but excludes disk patch and dm1000- if you want these please send an extra disk and ask for them! Thanks to Dortig for sending these disks, and to Mike Goddard for decoding the double density disk Tony McGovern sent to Dortig! As a mere 40 column techno-klutz I cannot discern any major change here. the only minor change noted is a rather slower key response to Show Directory!

>HARRISON SOFTWARE RANDOM NUMBER UTILITIES. Two of these routines were submitted for JI\*MES #35. Machine code routines to include and use in your Extended Basic programs, to properly SEED the random number generator, to place a random integer within a range into a variable: to similarly place several random numbers into a numeric array: to shuffle a deck of cards into a numeric array and to shuffle several decks of cards into a numeric array (eg for Canasta). Commented source code for machine code novices.

>HARRISON SOFTWARE EXTENDED BASIC UTILITIES VOLUME 1. More utilities in machine code to be added to and used in your Extended Basic programsincluded are amended Accept At routines, which display a default value (number or string) and if you hit any key but enter. blank the whole input field, so you don't have to fill it with spaces or even hit ERASE. Also a routine to fill a numeric array with musical note values (5 octaves in a second). And a routine equivalent to B=INT(ABS(NUMVAR))\*SGN(NUMVAR) which returns a similar value for negative fractions as for positive. The normal INT would return -5 for INT(-5.3). Commented source code for machine code novices.

>GOCAD2 by D G Hewitt. A circuit drawing program. This is a rewrite in c99 of Dave's Gocadl which was distributed with the long gone 4FRONT diskazine. Version 2 is faster, and now uses a drawing area of about 6 screens, overlapping and windowed. As with Gocad 1, I am unable to print from it with my Epson FX80 printer. It uses PIO only, and Dave says it is for an IBM printer. The c99 source code is included to make a very full disk and you could perhaps alter the printer details? An XB loader is included.

>STAR TREK PARODY TEXT: CLASSIC MEETS THE NEXT GENERATION.

>UTIL 30... Utilitiy to check if alpha lock is up or down; 14 character display 6 memory calculator from Jim Peterson; Ansi-Tool 5.2 which allows you to see on (TI Writer) screen special IBM printer codes -only for IBM-emulating printers!: Rip Off which transfers from ExBas to TI Artist font special character (8x8) definitions; routine to DISPLAY AT a string of up to 255 chars using all 32 columns; an iterative formula calculator; and a long division cryptogram puzzle.

NUTIL 31... A very useful utility from Jim Peterson to print almost anything, including 28 column wide Triton XB files, text up to 5 columns per page (NB: format first with TI Writer!)(Vn 1.6); V1.1 of E P Rebel's Screen Utilities, lots of CALL LINKs for your XB programs; TIA-LINK which prepares machine source code from a TI Artist instanceafter assembling you can quickly link to it in XB; two sort routines; an excellent routine to store one XB screen display and put another one up- quickly put up a menu then return to calling screen or do a disk directory then return to calling screen etc, most useful.

>VOLLYBALL (USVBA). Machine code game for one or two from Arcade Action Software. I do not know the status of this one- no ads seen for it and they were not at Chicago in November. Play volleyball with your TI!

>SPEECH EDITORS PLUS. A selection of speech oriented programs including two speech editors which allow you to trim the ends off inbuilt sounds, then add them together to make new words. The disk library already has a speech editor which allows you to make new XB words from the TE2 like Text to Speech program- these are not as strong but are offered for educational purposes. Disk includes Bert and Ern (Sesame St) and some phrases from Parsec etc. Speech Synth and XB required.

>COMPUTER CRAPS VN 2 from Ramsoft Enterprises now shareware. Reasonable simulation of the dice game. There is a nicer version in Notung Softwares TI CASINO if you enjoy it!

The following is an extract from 87 sectors of GAMES files in the library. Many of these can be made available - as individual titles from the cassette library if required.

>COLLINS. Programs from the Collins Packs- Airship, Bat, Cards, Commando, Croses, Dicerace, Dragon, Duel, Logicol, Maze, Racetrack, Target.

>GAMES-2 : A little novelty called PICKACARD and an X rated graphics demo called .... by .... (someone famous!). And some good games: A very playable BACKGAMMON, an odd but playable CHECKERS (=draughts), a new version of FOREST FIRE, Imhotep, Keno, THE WALL and MOTOR CROSS and RALLY CROSS.

>GAMES-3.:A gamers delight with excellent Backgammon program (a revision of the program on Games-2), and a tough gomoku program. Also includes Roland Truemans programs Flip Flap (XB) and Maketracks (Super TIB), plus Giant and Dwarfs for mensa members, Crystal Palace Towers from Italy, and Left/Right from Sweden. ALSO Mad Dogs AND UFOGAME. AND Bonkers AND Mazzo. Only two sectors spare... as odf daide of avon much

que san rejumen ent menero to trassens

\_\_\_\_\_

>GAMES-4: A biorhythm program, a TIB Othello program, R'bert, three programs based on Rubik's cube, TI-Poker, White Holes, Superjot, Spiderbop.



by Stephen Show

Welcome to the second No Comment. If you have any queries or comments, these are welcome and should be sent to me at the address on the front cover. I really like to know what YOU like to read about!

Back in the Autumn the BBC presented an interesting series of programs on Computers, and one topic touched upon was artificial intelligence. In particular we saw a very very early (and expensive) computer learning how to play noughts and crosses.

TI have already provided (in the LOGO manual) a "what animal am I" learning program which builds up a library of classifications so that it will — in the end! — learn to identify animals — or it can easily be modified to learn any other group.

One of our members sent to me an extract from a very early Readers Digest Annual, (1963 Readers Digest Young Peoples Annual), which contained an article adapted from one of Martin Bell's Scientific American math articles.

The Readers Digest ascribed the invention of the machine (about to be described) to Martin, but he invented the game, which led to a simplified machine, the original having been invented in Edinburgh back in 1961 by Donald Michie.

The original machine learned to play noughts and crosses and you can read about it in Penguin Science Survey 1961 Volume 2. You can catch up on Martin's work in the Pelican edition "Further Mathematical Diversions".

Here we go then - and the purpose of this note is firstly for you to PROGRAM the machine Martin described, and then to develop it further.

Our game is played in a very small grid, just 3 x 3, making 9 squares. You and the computer have three chess pawns, yours at the top, the computers at the bottom. Normal moves apply— move in one direction only, straight towards opposite side, or if making a capture, diagonally forward. You always go first. You win by placing one of your pawns on the far side; by capturing all the computer's pawns, or by preventing the computer from making a move.

The small size of the board allows for only a very limited number of positions, even fewer if you consider reflection about the vertical column.

The game machine Martin described was built with matchboxes (arrays?). Each matchbox had a picture on it showing the possible positions after your move in which the computer can still play. For any one arrangement of pieces the computer has up to four possible moves. Each of the four possible moves is contained in the matchbox

by means of coloured markers. Decree 3000 000 001 9373A good 1000

When the computer makes a play, it remembers which marker it has used. If on its next move it loses then that coloured marker is removed from the box and play begins again with that possible play removed. It follows by the way, that if after selecting a move, your move leads to a matchbox which has already been emptied, then the marker which led to the empty box is removed, so the computer will need to remember at least two moves back!

There are many ways of programming this, but I would not say any of them was unduly simple. Allow for display of the board and human input. At the end of one game you start again, with a different set of "possible" moves for the computer due to its learning experience.

Once you have mastered the 3x3 grid, try a grid 3 high by 4, 5, 6 etc wide. When you go to four high the number of possible positions begins to bloom...

Can you make a self learning program from this description? Send it in for us to print!

We have some fairly good simulation programs for the TI99/4A for sports such as golf, bowls, ten pin bowling, tennis, baseball (for MBX), Volleyball... would anyone care to program a game of CROQUET? I have not yet seen a good simulation of billiards, pool or snooker, which seem to involve rather more complex math than can be readily handled in real-time. Hint to anyone writing a billiard program: on a standard table, hitting a ball with absolute maximum force across the width of a table, how many times does it hit a cushion before halting? If you hit a ball hard against a cushion at an angle of 45 degrees at what angle does it bounce off? Is it the same angle as if you gave it a very gentle push? When I see those questions reflected in a snooker program I may consider it!

The FOG INDEX is a measure used to define the reading age for the text, or the other way round, to define how badly the author writes! A high fog index would indicate someone who has communications problems. Can anyone write a program which will read a DV80 file to evaluate a fog index for the work? The program needs to count the words in sentences, which it has to keep track of by watching for punctuation, and to evaluate the number of syllables.

A product of the Gunning-Mueller Clear Writing Institute, the fog index is the average sentence length plus the percentage of words having three or more syllables, multiplied by 0.4 to give the number of years schooling. Add four to obtain a reading age.

R B Nottingham tackled this as a program in the long gone magazine CREATIVE COMPUTING. He derived a formula to count the number of syllables in a word, as follows:

Any word having three or more of the letters AEIOUY has three or more syllables if:

two vowels appearing next to each other are counted as one vowel.
 a final letter E or final letters ED are not counted as vowels unless they form part of the groups ded,ted,le.

Not perfect, but pretty close for most texts, give or take the odd decimal.

Now how about a program to read a DV80 file and output a fog index? In general you will only need to read to the end of the first sentence AFTER 100 or 200 words. Sentences do not always finish with a full stop!

\*Exception: in some styles of writing, sentence breaks are hidden with odd use of punctuation- for example forcing a continuation with a hyphen- but such tricks are quite correctly going to lead to a high foo index as the method of construction, while perhaps echoing a line of thought, is NOT as crystal clear as it could be- just count the number of words in this sentence!

Please send your programs to me for printing!! Assume text is for the Editor not the Formatter, and may contain control codes.

BIOMORPHS - as illustrated in a program adapted from Jose E Murciano-I learn that Biomorphs were discovered by our friend Clifford Pickover, as a result of a bugged program!! This learned through A K Dewdney, who also passes on the following gem of popcorn....

# POPCORN 1 POPCORN 1

- 2 ! from Clifford Pickover 3 ! via A K Dewdney
- 3 ! via A K Dewdney
- 4 ! for TI99/4A + The MIssing Link
- 5 ! stephen shaw 12/91 protection to the state of the sta
- 6 ! 100 CALL LINK("CLEAR") 110 H=0.05 120 FDR J=1 TO 26

- 130 FOR K=1 TO 26 OF THE OTHER CHARGE THE SUPERING AND ADDRESS OF THE OTHER
- 140 X=-4,3+J/3
- 150 Y=-4.3+K/3
- 170 FOR N=1 TO 50
- 190 YY=Y-H\*SIN(X+TAN(3\*X))
  200 X=XX :: Y=YY

- 210 CALL LINK("PIXEL", X+90, Y+96) 220 NEXT N :: NEXT K :: NEXT J
- 220 NEXT N :: NEXT K :: NEXT J
- 230 6010 230
- 1 : PUPCURN Z 100 CALL LINK("CLEAR")
- 110 H=0.03 120 FOR J=1 TO 12 130 FOR K=1 TO 12 140 X=-2.1+J/3
- 150 Y=-2.1+K/3
- 160 REM

- 170 FOR N=1 TO 80 180 XX=X-H\*SIN(Y+TAN(3\*Y))
- 200 X=XX :: Y=YY
- 210 CALL LINK("PIXEL", X+88, Y+98)
  220 NEXT N :: NEXT K :: NEXT J

- 1 ! POPCORN 3
- 100 CALL LINK ("CLEAR")
- 110 H=0.05
- 120 FOR J=1 TO 12 383 (7804) TM 283 (7804)
- 130 FOR K=1 TO 12
- 140 X=-2.0+J/3 150 Y=-2.1+K/3 160 REM
- 170 FOR N=1 TO 150
- 180 XX=X-H\*SIN(Y+TAN(3\*Y))
- 190 YY=Y-H\*SIN(X+TAN(3\*X))
- 200 X=XX :: Y=YY
- 210 CALL LINK("FIXEL", X+99, Y+90)
- 220 NEXT N :: NEXT K :: NEXT J
- 230 GOTO 230
- 110 H=0.05 100 CALL LINK ("CLEAR")
- 120 FOR J=1 TO 15 STEP 0.5
- 130 FOR K=1 TO 12 STEP 0.5
- 140 X=-1.0+J/3
- 150 Y=-1.1+K/3 160 REM
- 170 FOR N=1 TO 90
- 180 XX=X-H\*SIN(Y+TAN(3\*Y))
- 190 YY=Y-H\*SIN(X+TAN(3\*X))
- 200 X=XX :: Y=YY
- 210 CALL LINK("FIXEL", X+90, Y+50)
- 220 NEXT N :: NEXT K :: NEXT J
- 230 GOTO 230

Even Micropendium has entered the world of fractals. I have several books which mention cellular automata, but I have not really followed them! Now Micropendium to the rescue, with a good program, modified by me below. This is one dimensional Life with each successive screen line representing a generation. If on pixels disappear, life has ended! If you see a black triangle pointed up, this represents life slowly expanding to a catastrophe. And so on!

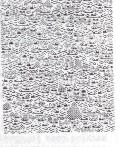
In my variation I have the rules of Life set at random, and the initial cell positions set at random. The results rarely look like the book illustrations for this topic!

100 ! LINUS 110 ! HARRY WILHELM MICROPENDIUM 12/91 120 ! EX BAS + TML 130 CALL LINK("CLEAR"):: CAL L LINK("PRINT", 40,1, "PRESS") 140 CALL LINK("PRINT", 50, 1, " C FOR CONTINUOUS"):: CALL LI NK("PRINT".60.1."K for KEY T O PROCEED")

150 CALL KEY(5, K, S):: IF K<> 80 AND K<>67 THEN 150

160 IF K=67 THEN FLAG=1

170 RANDOMIZE 180 CALL LOAD(-24576,2,1,160 ,42,2,129,160,58,20,14,192,1 ,2,2,131,74,140,176,22,6) 190 CALL LOAD(-24556, 140, 176 ,22,4,140,176,22,2,192,48,4, 80, 2, 33, 0, 8, 16, 239, 4, 96) 200 CALL LOAD(-24536, 32, 90, 8 3,84,82,73,78,71,160,58,76,7 3,78,69,65,82,160,116,4, 32) 210 CALL LOAD(-24516,60,18,1 6,240,162,87,2,0,162,88,2,1, 0,240,4,194,208,144,2,13 0)



220 CALL LOAD (-24496, 32, 0, 21 ,3,220,32,160,114,16,2,220,3 2,160,115,6,1,22,245,4,2 24) 230 CALL LOAD (-24476, 161, 60, 4.96.0.106.0.14.0.14.0.2.0.5 ,0,1,2,224,161,52) 240 CALL LOAD(-24456, 2, 132, 0 ,191,22,5,7,4,2,224,131,224, 4,96,0,106,5,132,22,21) 250 CALL LOAD (-24436, 2, 4, 0, 1 28.2.0.40.0.2.1.163.72.2.2.1 6.0.4.32.32.44) 260 CALL LOAD (-24416, 2, 0, 32, 0,4,32,32,36,2,0,48,0,4,193, 2.2.8.0.6,160) 270 CALL LOAD(-24396,58,246, 2,0,161,94,2,1,162,88,2,3,0, 240, 204, 49, 6, 67, 22, 253) 280 CALL LOAD(-24376, 200, 227 .162.88.162.78.200,227,163,6 2.161.84.5.195.2.131.0.10.22 ,246)290 CALL LOAD(-24356, 2, 7, 161 .94.194.7.4.192,112,24,19,3, 194,96,160,106,16,2,194, 96) 300 CALL LOAD(-24336,160,108 .194.160,160,112,98,32,160,1 10, 176, 56, 6, 10, 22, 253, 6, 192, 5.128) 310 CALL LOAD (-24316, 9, 9, 24, 4,217,224,160,114,0,250,16,3 .217.224.160.115.0.250,5,135 320 CALL LOAD (-24296, 2, 135, 1 62.78.22.225,4,197,209,165,1 62.88.19.2.4.32.47,188,5,133 330 CALL LOAD(-24276, 2, 133, 0 ,240,22,247,16,162,"",8192,1 60,0) 340 WS="5" :: AONE=10 :: AZE RO=20 350 CALL LINK("CLEAR"):: CAL L LINK("PRINT", 1, 93, "L I N U 360 CALL PEEK(-31956, A, B):: B\$="3579" :: CALL VALID(A, B) 370 W=INT(RND\*8+2):: IF W/2= INT(W/2)THEN 370 380 CALL CONVERT(85, W, AONE) 390 CALL CONVERT(130, W, AZER 400 ! 410 !

420 AONEM=INT(AONE/256):: AO NET.=AONE-AONEM\*256 :: AZEROM =INT(AZERO/256):: AZEROL=AZE RO-AZEROM\*256 430 CALL LOAD (-24470, AONEM, A ONEL, AZEROM, AZEROL, O, INT(W/2 ), O, W) 440 CALL LINK("CLEAR"):: CAL L LINK("CHAR", 46, "8") 450 ! 460 RND\$="" :: FOR I=1 TO 60 :: RND\$=RND\$&SEG\$("^ ,1+(RND\*64 AND 252),4) 470 NEXT I 480 CALL LINK("CLEAR"):: CAL L LINK("CHSIZE",1,1):: CALL LINK("PRINT", 1, 1, RND\$):: CAL L LINK("STRING", RND\$) 500 ! 510 CALL LINK("LINEAR") 520 IF FLAG=1 THEN 360 530 CALL KEY(5, K, S):: IF S<1 THEN 530 ELSE CALL LINK("CHS IZE",6,8):: GOTO 360 540 550 SUB CONVERT(R, W, X) 560 ! 570 580 A\$="" :: FOR I=W-1 TO 0 STEP -1 :: IF X>=2^I THEN X= X-2^I :: A\$=A\$&"1" ELSE A\$=A 600 CALL PEEK(-31956, A, B):: BS="01" :: CALL VALID(A,B) 610 A\$="" :: FOR I=1 TO W:: A\$=A\$&STR\$(RND<RND-.25):: NE XT I 620 X=0 :: FOR I=W-1 TO 0 ST EP -1 :: IF SEGS(AS, W-I, 1)=" 1" THEN X=X+2^I 630 NEXT I 640 IF X<1 THEN X=5 650 SUBEND 660 SUB VALID(A,B):: AD=A\*25 6+B-65530 :: CALL LOAD(9649,0 ."", 9652, INT(AD/256), AD ):: 670 SUB UNVALID :: CALL LOAD (9649.6."", 9652, 38, 40):: SUI

Careful how you key in those CALL LOAD lines, which as you suspected do contain machine code to speed things up a mite.

ALLIGATOR ALLEY - TI BASIC: 510 GOTO 550 BASIC 110 REM FROM PROGRAMS FOR TH E TI HOME COMPUTER 120 REM COPYRIGHT (C) 1983 B Y STEVE DAVIS 130 DIM A(24,32) 140 RANDOMIZE 150 CALL CLEAR 160 PRINT "I'M HIDING THE LIGATORS": "STAND BY 170 FOR R=1 TO 24 180 FOR C=1 TO 32 190 A(R,C)=INT(RND\*20)+1 200 NEXT C 210 NEXT R 220 A(12,16)=0 230 CALL CHAR(96, "FFFFFFF FFFFFF") 240 CALL COLOR(9, 16, 1) 250 PRINT : "YOU ARE IN THE M IDDLE OF A SWAMP. YOU MUST REACH THE EDGE WITHOUT GET TING EATEN BY AN ALLIGATOR." 260 PRINT : "MOVE YOUR MARKED TO ONE SIDEBY PRESSING THE ARROW KEYS: S=WEST, D=EAST, X=SOUTH. E=NORTH' 270 PRINT : "ALLIGATORS ARE G 6) REEN SO THEYARE CAMOUFLAGED IN THE SWAMP": : "HOPE YOU DO N''T GET EATEN!" 280 PRINT : "PRESS ANY KEY TO START" 290 CALL KEY(O, KEY, STATUS) 300 IF STATUS=0 THEN 290 310 CALL CLEAR 320 CALL SCREEN(3) 330 CALL HCHAR(12, 16, 96) 340 R=12 350 C=16 360 M=0 370 X=INT(RND\*20)+1 380 CALL KEY(O, KEY, STATUS 390 IF STATUS=0 THEN 380 400 IF (KEY<>69)\*(KEY<>101)7 HEN 440 410 IF R=1 THEN 610 430 GOTO 550 440 IF (KEY<>88)\*(KEY<>120) 450 IF R=24 THEN 610

460 R=R+1

470 GOTO 550

480 IF (KEY<>83)\*(KEY<>115)

520 IF (KEY<>68)\*(KEY<>100)T HEN 380 530 IF C=32 THEN 610 540 C=C+1 550 CALL HCHAR(R,C,96) 560 M=M+1 570 IF A(R,C)=X THEN 660 580 FOR D=1 TO 20 590 NEXT D 600 GOTO 380 610 CALL SOUND(1000, 262, 0, 33 0,0,392,0)620 PRINT "WHEW! YOU MADE IT IN": M+1; "MOVES!": "WANT TO T RY AGAIN? (Y/N)" 630 INPUT YS 640 IF (Y\$="Y")+(Y\$="y")THEN 310 650 STOP 660 ROW=R 670 COL=C 680 IF ROW(24 THEN 700 690 ROW=23 700 IF COL>2 THEN 720 710 COL=3 720 IF C<29 THEN 740 730 COL=28 740 CALL CLEAR 750 CALL SOUND(900,110,2,-7 760 CALL SCREEN(16) 770 CALL COLOR(10,13,1) 780 CALL CHAR(104, "000000000 000703F") 790 CALL CHAR(105, "OF03071F3 8") 800 CALL CHAR(106, "000000000 00000E3") 810 CALL CHAR(107, "7FFFFFFFO F070307") 820 CALL CHAR(108, "00007C3F0 70100E") 830 CALL CHAR(109, "FCFFFFFFF FFF8081") 840 CALL CHAR(110, "000000COF OFC7E3E") 850 CALL CHAR(111, "3FFFFEFEF 860 CALL HCHAR (ROW, COL, 104) 870 CALL HCHAR (ROW+1, COL, 105 880 CALL HCHAR (ROW, COL+1, 106 890 CALL HCHAR (ROW+1, COL+1, 1 07) 900 CALL HCHAR (ROW, COL+2, 108 910 CALL HCHAR (ROW+1, COL+2, 1 920 CALL HCHAR(ROW,COL+3,110)
930 CALL HCHAR(ROW+1,COL+3,1
11)
940 FOR I=1 TO 5
950 CALL CHAR(104,"000000000
0000001")
960 CALL CHAR(105,"3F033F7F"
))
970 FOR D=1 TO 10
980 NEXT D
990 CALL CHAR(104,"000000000

000703F")
1000 CALL CHAR(105, "OF03071F
38")
1010 FOR D=1 TO 10
1020 NEXT D
1030 NEXT I
1040 CALL CLEAR
1050 PRINT "OH NO! YOU WERE
EATEN!": :"WANT TO TRY AGAIN
? (Y/N)"
1060 GOTO 630

This program was keyed in by my 7 year old son from the book detailed at the start, so no complaints about it being too long to key in!!! Although there is a long pause at the start, thereafter it only takes a little while to reset.

Of special note: The program was listed to disk in 28 column format, using Triton Extended Basic, which created a DV28 file. This disk file was then read by Jim Petersons excellent PRINT ALL program, modified to open an output disk file as DV80 (instead of DV254) and PRINTALL then produced the double column spread you see above, ON DISK so that I could add this text below. PRINTALL is on library disk UTIL31.

NO you do not need to buy Triton XB... that was just one thing Jim's program can do... it can also print in 5 columns of 28 characters! or any other combination, but with those choices you have to print direct to paper or to a disk format that Funlweb cannot handle- DV254.

Without Triton XB, you can produce the above effect with Tony McGoverns excellent CO LIST program, also in the disk library. With this you LIST your program in the normal way to disk, then run the file through CO-LIST. Unlike Printall, Co-List will not split a line, but would print the end of the program like this:

930 CALL HCHAR(ROW+1,COL+3,1 11) 940 FOR I=1 TO 5 950 CALL CHAR(104,"000000000 0000001") 960 CALL CHAR(105,"3F033F7F") ) 970 FOR D=1 TO 10

1030 NEXT I 1040 CALL CLEAR 1050 PRINT "OH NO! YOU WERE EATEN!": :"WANT TO TRY AGAIN ? (Y/N)" 1060 GOTO 630

Can you see the difference? The two columns are not exactly equal in length!

Co-List can also insert blank column lines to make rems stand out more.

CORRECTION : TI\*MES 35 PAGE 55 : RIEMANN SPHERE:

LINE 520 should end: NEXT KC not NEXT C

LINE 540 has become tagged onto line 530 and needs separating.

LINE 540 should end NEXT XC and not NEXT C.

The text from the middle of page 56 is witg reference to the earlier Riemann Inverse program on page 49.

# CORRECTION PAGES 2-7 MOON BASE ALPHA:

Where you see a z sign you need to type a z instead Trevors printer is set to "UK Character Set" which puts a finstead of a z Line 200 is followed by 210, not 10! Line 1340 is followed by 1350 not 350. Line 1090 has become attached to line 1080 and needs detaching! Line 1540 can be safely omitted.

# You don't HAVE to have it all!

# by Jim Peterson

Do the conversations at your user group meeting sound like a coffee break in Silicon Valley? Are you confused by talk of GROMs and GRAMs, puzzled by references to HFDCs, intimidated by discussions of megabytes and frightened by talk of burning EPROMs? Well, join the crowd, buddy — so am I!

There are basically three types of people interested in computers. First, there are those who use a computer to run programs, to accomplish something useful or just to have fun. I believe that those people are still in the great majority, although we don't hear much from them.

Then, there are those who get their kicks out of writing programs, of creating software for others to use.

There aren't too many of those left in the TI world.

And finally, there are those who like to tinker with the computer, soup it up, plug in doohinkies and thingama— jigs, and talk in that strange language I mentioned above. I don't know how many of those folks there are, but they are certainly the most knowledgeable, active, and interested, and they tend to dominate the conversations and the printed material in the TI world nowadays.

I presume that those fellows also do actually run programs on their souped up systems. And, some of them must be skilled programmers, because many of their hybrid hardware creations would be useless without specialized soft— ware. I'm very glad that those people are around. Once in a while they invent something that I actually find useful, and they are a lifesaver when my equip— ment breaks down.

But, don't be intimidated by all that high-tech talk, and don't think that the computer world is passing you by. There are so many things to do with a computer that no one could possibly find time to do them all. Do your own thing and don't worry about the rest.

I have operated a TI software company for seven years, and I also spend a lot of time writing programs, using the computer as a word processor, etc. I probably spend more time on my TI than 90% of the users. So, what does my equipment consist of?

I have a console with the Extended Basic module plugged in, attached to a P-box which contains a TI disk controller, two double-sided drives, the 32k card, RS232 card, and a Horizon Ramdisk. Also plugged into the RS232 card is an old Gemini 10X printer and an Avatex 1200 baud modem. I also have a Speech Synthesizer, a pair of TI joysticks, a TEII module and an Editor Assembler module, all of which I plug in occasionally when I need them; also, a cassette recorder and cable which hasn't been used in a long time.

I use Triton's Super Extended Basic module because it has some editing features which are useful when programming. It also has some limited plotting capability which I have never used — and have never heard of anyone who has Isjs—we can supply Bill Harms CLASS which uses this feature! J. If you don't program, it would hardly pay to switch from the old TI Extended Basic. I also have the Mecha—tronics module but never got around to trying it.

I had a Gram Kracker but soon sold it and bought a Ramdisk instead. The Gram Kracker has fantastic capabilities if you have the skill and knowledge to take advantage of them, but most users don't seem to have done much beyond personalizing the title screen.

36

I had a widget, and I guess it is still collecting dust around here some place. It was a nuisance, and since I use XBasic 99% of the time I didn't need it. There are now widgets or "module expanders" that allow you to access more than one module from within a program. That is, if you have the skill to write such a program. I don't know that anyone has released such programs to the public domain, and I can't think of any practical use except to access TEII speech from XBasic — but you can do that with the Text-To-Speech disk.

The ram disk is the one tool that I would not be without. In order to assemble my TI-PD catalog, I screened over 4000 programs, debugged and modified, merged in help files, con- versions to XBasic and loaders, and assembled over 400 disks of programs. It took me hundreds of hours of work — without a ram disk it would have taken thousands of hours and I would not even have attempted it.

The ram disk enables me to switch from one program to another almost instantly, and with John Johnson's Boot program I can just as quickly catalog a disk or view a file. Mine has 256k of memory. I could get one with much more memory but I see no reason to do so; I have every program on it that I am apt to use even once a month, and it is only half full. That leaves plenty of room for temporary storage and downloading. However, if you only use your computer to play games, do a little word processing and a bit of record keeping, a ram disk would be an expensive convenience rather than a necessity.

Since my ram disk is only half full, I would consider a hard drive to be about as useful as the mammalian appendages on a swine of the masculine persuasion. If I was running a BBS, sure — or if I was doing a lot of work with those memory-gobbling graphics and needed everything quickly accessible.

My old Gemini printer has been a faithful workhorse, although the hood over one sprocket wheel has lost its spring and is being held down by a loop of elastic cord. I will have to give it up soon, because the Gemini printer codes are becoming obsolete and I need to be able to write and test Epson codes. But, I hate to give up these 79-cent typewriter ribbons and start getting ripped off on \$2.50 cartridges! As for a color ribbon, the temperature will have to go way down, down under, before I pay for one of those.

Once in a while, when someone sends me a double-density diskfull of stuff, I wish I had a CorComp disk controller. Otherwise, with diskettes selling for a quarter or less, it wouldn't pay to change. If I ever get around to subscribing to GENie or Delphi, it will pay me to get a 2400 baud modem.

I can't think of anything else I need, and I don't want what I don't need. If I really wanted to play joystick games, I would certainly get something better than the TI joystick. And if that MIDI interface cable becomes a reality, I wil be sorely tempted.

I can't see any advantage in putting the 32k under the hood, or anyplace other than where it is now. If I used speech a great deal, it would be nice to get rid of the synthesizer — but I know only one user who uses speech that much. I don't need a clock built in because I have a watch on my wrist. If I really did a lot of serious writing, an 80-column card would be wonderful. But then I would have to buy a monitor capable of displaying 80 columns. I certainly don't want to give up color, and high-resolution color monitors cost more. I would still want to use my old monitor for programming, because I like to write programs for folks who have basic equipment. I don't have room on my computer desk for two monitors, so I think I'll pass.

I'm a three-finger typist, so a RAVE keyboard wouldn't speed up my typing very much. If I really wanted an IBM keyboard and 80-column capability, I would throw in a few bucks more and get a Geneve.

Bo, what about the Geneve? If I had an irresistible urge to run the few great programs that have been written for it, or if I wanted to explore its great programming capabilities, I would get one. But, I like to write programs for other people to use. When so few are interested in programs that I write for a computer that sold in the millions, why would I write programs for a computer purchased by a couple of thousand people?

I am sure that many folks will disagree with what I have written. That's why I wrote it. I hope they will disagree so strongly that they will immediately boot up Funlweb and compose a blistering reply. But don't send it to me - send it to your newsletter editor. The newsletters are badly in need of more articles by more writers!

\* HACHINE CODE / COMMENTED SOURCE CODE \*

\* Mack McCormick. It shows you \*

\* how to redefine characters, place \*

\* them on the screen, and change the \*

\* color of each character set \*

\* Entry Point -> START \*

\* Must be run from E/A or MM Load & Run \*

\* RO-R2 Used for general VDP access \*

\* R3 Used as a general purpose counter \*

\* R11 contains the return address on \*

\* entry. \*

\_\_\_\_\_

DEF START DEFINE THE ENTRY POINT OF THE PROGRAM
REF VMBW, VSBW, VWTR SYSTEM UTILITIES WE WILL USE

\* DATA STATEMENTS \*

SAVRTN DATA 0 SET ASIDE A WORD OF MEMORY TO SAVE THE RETURN ADDRESS BORDER DATA >8080,>2020,>2020,>2020 CHAR >80 WILL BE OUR BORDER DATA >2020,>2020,>2020,>2020 DATA >2020,>2020,>2020,>2020 DATA >2020,>2020,>2020,>2020

PATRN DATA >3C7E, >FFFF, >FFFF, >7E3C

STATUS EQU >837C LOCATION OF THE GPL STATUS BYTE

\* THIS EQUATES THE LABLE STATUS TO THIS ADDRESS
WS BSS >20 SET ASIDE 32 BYTES OF MEMORY FOR OUR WORKSPACE

MSG TEXT 'THIS IS A TEST' 14 BYTES LONG
EVEN FORCES THE PROGRAM COUNTER TO AN EVEN MEMORY ADDRESS.
\* AS A GENERAL RULE ALWYS USE EVEN AFTER THE LAST BYTE, BSS, OR TEXT OPCODE.

START MOV R11,@SAVRTN MOV THE ADDR IN R11 TO THE WORD OF MEMORY AT SAVRTN LWPI WS

TELL THE HARDWARE WORKSPACE REGISTER WHERE YOUR WS IS

\*--PUT A BLUE BORDER AROUND THE SCREEN---\*

LI RO.>0705

BLWP AVWTR

07 IS VDP REG 7 OR THE SCREEN BACKGROUND COLOR 05 IS THE BACKGROUND COLOR. (LT BLUE)

- \* THIS SETS THE TOP AND BOTTOM OF THE SCREEN TO LT BLUE.
- \* SEE SECTION 16.1 E/A MANUAL FOR MORE INFORMATION ON VWTR.

\* NOW SET CHAR >80 (128 decimal) TO A BLUE ON BLUE SQUARE

LI RO,>390 THIS IS THE POS IN VDP RAM COLOR TABLE FOR CHARS >80->87

LI R1,>5500 BLUE FOREGROUND/BACKGROUND IN MSB OF R1

BLWP @VSBW REMEMBER RO ALWAYS IS THE ADDR IN VDP. R1 ALWAYS CPU.

\* SEE SECTION 21.2.2 FOR MORE INFORMATION ON THE COLOR TABLE

\* NOW LET'S MAKE ALL THE CHARACTER SETS BLACK ON WHITE

LI RO.>383 LI R1.>1F00

START WITH CHAR SET >18 1=BLACK FG, F=WHITE BKGND

LOOP2 BLWP AVSBW INC RO

POINT TO NEXT VDP RAM COLOR TABLE ADDRESS LAST COLOR TABLE ADDRESS (CHARS >78->7F)

CI RO.>390 JNE LOOP2

\* NOW WRITE THE BORDER TO THE SCREEN AND CLEAR THE SCREEN AT THE SAME TIME

24 ROWS TO WRITE

RO R1.BORDER

BEGINNING OF SCREEN IMAGE TABLE ADDRESS OF ONE ROW OF BORDER DATA

LI R2.32

LOOP

32 BYTES TO WRITE

BLWP QVMBW

WRITE A ROW

AI RO.32 POINT TO BEGINNING OF NEXT ROW

\* NOW WE'LL PUT THE MESSAGE ON THE SCREEN CENTERED

\* REMEMBER THAT THE SCREEN IMAGE TABLE IS FROM 0 TO 767 IN VDP RAM

\* TO DETERMINE THE ADDRESS FROM ROW AND COLUMN WE USE THE FOLLOWING FORMULA

\* ADDR=((ROW-1)\*32)+(COLUMN-1) IF WE WANT OUR MESSAGE AT ROW 12 COLUMN 9 THE VDF

\* ADDRESS WOULD BE 360

LI RO, 360 R1 . MSG VDP ADDRESS IN SCREEN IMAGE TABLE

LI R2.14 ADDRESS OF THE DATA IN CPU RAM 14 BYTES LONG

BLWP QVMBW

WRITE IT TO THE SCREEN

- \* NOW SUPPOSE WE WANT TO MAKE CHAR >7F A BALL SHAPE AND PLACE IT UNDER THE TEXT
- \* THE PATTERN TABLE IN E/A IS LOCATED AT >0800. TO CALCULATE THE LOCATION OF A
- \* PARTICULAR CHARACTER MULTIPLY IT'S HEX VALUE BY 8 AND ADD THE RESULT TO >800.

RO.>OBF8

(>7F\*8)+>800=>BF8

LI R1, PATRN PATTERN TO DEFINE >81 TO FROM CPU RAM

LI R2,8

PATTERNS ALWAYS 8 BYTES

BLWP QVMBW

>81 IS NOW THE SHAPE OF A BALL

LI RO, 431

SCREEN IMAGE TABLE TWO ROWS BELOW TEXT CENTERED WRITE >81 (BALL) TO THE SCREEN

LI R1,>7F00 BLWP @VSBW

PUT IT UP

\* EXPERIMENT WITH THIS PROGRAM UNTIL YOU ARE COMFORTABLE WITH VDP RAM ACCESS

MOV @SAVRTN, R11 RESTORE R11 TO THE ADDRESS YOU WISH TO RETURN

LIMI 2 JMP \$

ENABLE INTERRUPTS SO QUIT KEY WILL WORK

LOCK UP THE COMPUTER (SAME AS 100 GOTO 100)

# WHATS IN A MODULE???? MACK MCCORMICK

This is a text file discussion of the ROM cartridge port for the TI-99/4A. It represents information I have been able to obtain from various references. Cartridge programs must operate from >6000 to >7FFF. When the computer is RESET or turned on, the power up routine looks for a Header or Control block at location >6000 in the cartridge port. This contol block establishes the linkage into your cartridge program and allows you to have multiple entry points. Here is an example contol block used to provide one entry point;

0000	AA01		DATA	>AAO1	6000	ID FOR BOOT
0002	0000		DATA	>0000	6002	TO TON DOOT
0004	0000		DATA	>0000	6004	
0006	000C		DATA	CHAIN	6006	ADDRESS OF MENU LIST
8000	0000		DATA	>0000	6008	THE WAS CHARLES OF THE ROLL OF
The state of the s	0000		DATA	>0000	600A	
		CHAIN	DATA	>0000	6010	CHAIN POINTER
000E	0020		DATA	SLOAD	6012	ENTRY POINT
0010	OF		BYTE	SLOAD-\$-1	6014	LENGTH OF MENU TEXT
0011	54		TEXT	'CARTRIDG		TO A MARKET OF THE PARTY OF THE
0020	0460	SLOAD	В	START		
0022	092E					

Let's examine the control block. If the TI operating system finds >AA at >6000 it knows a cartride is plugged in the port. The next byte must be a >01 at location >6001. This informs the operating system that the code in the cartridge is executable machine language. Other codes are used for GROM, but that's another discussion. Was of a edd (is espheric bos wolled not parties

The data at location >6002 - >6005 is zero. Location >6006 must contain a word pointer to a list which identifies the menu text and associated entry point when that item is selected. This location usually contains a >600C. Locations >6008 - >600B must be zero. an oafa Ilim nov .nolla resource sylenoviero

The chain list at >600C contains the following:

Bytes 1 & 2 = chain pointer to the next menu list - or 0000 is this is the last list in the chain.

Bytes 3 & 4 = entry point associated with this menu selection.

Byte 5 = length of the menu text.

Bytes 6 - N = Menu Text - this is displayed on main menu.

Remember all dynamic data must be in RAM usually in the >8300 area. This area is used for registers plus VDP RAM is used for variable storage. Cartridges cannot REFerence any lable or routine outside the cartridge. This means the cartridge program must provide it's own VSBW, VSBR, VMBW, and VMBR routines which are normally loaded from the Editor Assembler cartridge. Examples of what these routines look like may be found in the Tombstone City game. Armed with this information, it possible to disassemble code to see how the program works. Hope you find this information useful. See ya around the boards... Hack McCormick a sample to be be A . Horst wineh we call he a

Dave Hewitt wrote to confirm that SOMEONE was programming in the UK and prepared to share their efforts. Way back when in the good old days, Harry Pridmore produced a few diskazines called 4FRONT, and one issue contained GOCAD by Dave. He has now produced GOCAD2, in c99. This is a circuit drawing program, which now allows your circuit to occupy almost 6 screens, by means of overlapping windows. Alas, like the original Gocad, this does not produce output to my Epson FX80 printer - Dave says it is written for an IBM printer. It is also directed to PIO only, although the c99 source code is provided for those of you clever enough to amend it. The very full disk is in the disk library. -----

Plan 129
And extracted from 343 sectors of UTILITIES listings (some of these will not be suitable for tape): A place over I malignated a processor II

>24PIN. An up to date offering of programs which will print pictures using 24 pin printers- produces pictures from files which have been printed to disk by MacFlix or PagePro -not from their usual picture files. Disk includes a 265 sector picture of MM ready to print. Uses files. Disk include: [ESC]\*(39)(n1)(n2) format.

>ASSEMBLY UTILITIES BY COOK AND JOHNSON: (important-quote all this in FULL! when ordering so I can trace it!) TWO DISKS of utilities as source code to incorporate into your own assembly programs, such as BLWPCLR, BLWPGCHAR, or DISPLAYAT.

>BA WRITER- TI Writer again this time from Paolo Bagnaresi. TWO DISKS.

>BEAXS by Paolo Bagnaresi of Italy. "Better Editor Assembler" for XB loading. TI's EdAs amended for loading with XB module, with one or two utility files added. Vn 3

>BIBLE TRIVIA Vn 3.01 by Steven de Geare, with several categories, a no-lose scoring system and attractive windows!

>BOOT DISK CHANGER by Scott Morrow, Version 2.1 - goes through a disk looking for DSKn. and changes all the n to any number you want- NOW you can run TI Runner from Drive 2!

>C99 by Clint Pulley. THE language! FIVE DISKS PLEASE which contain the compiler and a number of library and demonstration files, as well as comprehensive documentation. You will also need a book on C. This is a fairly complete implementation of small C. You prepare your c source code with say the TI Writer Editor, then compile using this package. then assemble using TIs Editor Assembler. Then you have a machine code program ready to run! This is version 4.0, sometimes referred to as REL Even 3 s - seeing committee of the control of the month of the control of the con 4.

>C99 TUTOR. From D L Mahler and the Boston Computer Society, this is not so much a tutorial as a large number of sample files, includes brief instructions on use and HOW TO use c99.

>C BITS. FROM MIKE CAVANAGH, a disk containing a complete-ish library for C programs for use with RAG's RAGLIB and Linker Librarian, with a modified control file for same; modified C optimiser, modified C string library file, and a logo-type graphics program- C code- needs two inputs, try for example 5,81

>CATLIB Vn 1.5 by Marty Kroll. A disk catalogue program with a capacity of 123 disks/900 files per data file. Small data files can be merged. Printouts can be of disks, files or standard format, in 1 2 or 3 columns. Printer control characters are replaced, so printouts stay neat. Global search on disk/file names.

MUCH MORE available. For a complete library listing please send four disks and return postage.

NEW PROGRAMS ARE ALWAYS WELCOME. rected to PiDeonapa atthoughtone recent a cone is provided for those

```
which expands the TI's capability to deal with Mandelbrot patterns.
 The Mandelbrot series involves a very large number of calculations, and from
 eac output value you decided whether to turn a pixel on or off. For example,
 turn on all pixels which relate to an odd value. There are however many more
 ways of patterning Mandelbrots, and it is NOT necessary to recalculate the
 values for each set up.
 While my machine code and c99 programming is poor, I do have Turbo Pasc 99,
 which allows fairly easy file handling. I therefore put together the following
program, which accepts three inputs and then sends the data for each of 160
 times 160 pixels to a disk file.
 I could not allow my ancient disk drives to grind away for several hours, and
 have therefore used my ram disk for output, but changing the device name is
 fust a question of typing in something else!
 The BOTTOM variable you input will actually be plotted from the top of the
 screen, due to our rows starting at 1 at the top- this is consistent with
 Fractal Explorer. The BOTTOM value is the LOWEST value plotted, similarly the
 LEFT input is the lowest column value.
An input well worth trying is BOTTOM: -1.9412614 LEFT: 0.0018350 SIDE: 0.000033
  or BOTTOM: 0.0029 LEFT:0.2700
                                                                                                                                                                                                                             SIDE:0.0026
Saving the data to RAM disk, plotting 160 x 160 pixels with an average value of
30 (768 thousand iterations) takes about FIVE HOURS!
 CONST STATE OF THE PROPERTY OF
 hs=160; se to do strange things and the second of the seco
vs=160;
it=100;
VAR f:STREAM[80];
             acnr,bcnr,gap,i,sd,x,y,nx,ny,p,q,x2,y2,xy2,vgap:REAL[8];
             v, h, count: INTEGER;
             scount:STRING[4];
 PROCEDURE ask(VAR acnr:REAL[8]; VAR bcnr:REAL[8]; VAR sd:REAL[8]);
   cls; And Alotta and the continue path and the continue to the 
      cursor(2,2);
      write("bottom value");
cursor(5,2);
      write("left value");
      cursor(3,12);
      READ(acnr); a for the different appropriation. The limited for the party of the control of the c
      cursor(6.12); appear that only the 10 is in the drawing, on the TV or ind the core
      read(bonr); run, it will function exectly like -x21 and -x25. ((a.5.26)2232) lay-n chr
      cursor(8,2); are how the program spoil acrear after you true 1,187. Off attend off
     write("side length");
cursor(9,12);
read(sd);

write("side length");
cursor(9,12);
cur
             {for julia set add: }
            { cursor(12,2); } (48, 432/48*) MHI LIMD :: (48, 4,064, 479/381*) MHI LIMD 005
            { write("point in M-set bottom: left:"); }
              { cursor(13,17); }
              { read(p); }
             { cursor(13,25); } bouley neve Join 1 085 MEHT(S\M)THICS\M TI 085
                                                                                                                                                                                                lashes xelomos eros rol 43
             { read(q);
   END:
```

Member Ken Hughes has very kindly sent in an idea from the world of Atari

```
ask(acmr,bcmr,sd); triefing formishment ditte laeb of villidades a lit edd therefore do Ldw cls;
   open(f, "DSK2.DAT", output):
   gap:=sd/hs; The results ere statil ten fav had no at a state daths afacts (is no erec
   vgap:=gap; wit elakualaunteetewannenen choken het Lines de Lond lebetel an lare it an ter aus
    FOR V:= 0 TO VS DO BEGIN
       FOR h:= 0 TO hs DO BEGIN
        i:=i*vgap; of wild radiance ing wasterned by politional atta many virtue mundic make
         p:=bcnr+i; {for julia set use nx:= instead of p:=}
         q:=acnr+i; {for julia set use ny:= instead of q:=}
         nx:=0; { omit for julia set}
         ny:=0; { omit for julia set}
         count := 0; da la second, sorto beforesed vilsatos like fecol goy eldarray MONTOS add
                 stream, due to our rows starting at 1 at the top- this is consistent with
         flag: A refusileten har selector agley ERENOL and an agree activities and property and the complete and the 
         count:=count+1; 0008400 0017830 8150160 r- smaller at meteria direction for
         x:=nx: 00.0 3018 = 4.0055, 6: 7021 + 6. 0405.0 1707308 - 70 4505.0 1707308
        x2:=x*x; | VIA Ve N. O. E. Itsucou NVIVIDeeds assist (entitless) t bearsoned Sect of
        y2:=y*y; ada kan ayaban and artractive adapter
        nx:=x2-y2+p;
ny:=xy2+q;
        IF (count<it) AND (x2+y2<=4) THEN GOTO flag;</pre>
                                                count:=0; and a number of Alberta and Department at the country of the country of
       END; makemative documentations You will also need a datak on the Chiteograph
    END; I'v complete implementation of small C. You or spars you a course
    put(f,scount);
    put(f, scount);
    close(f);
  END.
The above code, compiled and assembled into a PROGRAM format file occupies just
10 sectors. The program to print the data occupies only 3 sectors, and below is 10000
given for Extended Basic and The Missing Link, although any language allowing
pixel graphics can be used... a typical plot might take about 40 minutes from
ram disk: an earchard for these seals seem to seal a seas a respective model of the
```

```
100 OPEN £1: "RD.DATA", INPUT
110 H,V=4
120 LINPUT £1:A$
130 FOR C=1 TO 77 STEP 4
 140 N=VAL(SEG$(A$,C,4))
 160 NEXT C . Printer control characters and roplaced (datassinehed) addiss
170 IF EOF(1)THEN 180 ELSE 120 (SL.9)TORTOS ((SL.9)TORTOS 
189 ! line 190 allows pic to be saved to a P file.
190 CALL LINK("INPUT",180,1,A$):: CALL LINK("SAVEP",A$)
200 GOTO 200
210 H=H+1
 220 IF N=100 THEN 270
 230 IF N/2<>INT(N/2)THEN 280 ! plot even values
 240 ! three lines
 250 ! for more complex codes!
  44
```

```
270 CALL LINK("PIXEL", H, V)
200 IF H>164 THEN H=4 :: V=V+1
290 RETURN on the program within works
```

Perhaps I should mention here that the DATA file for a 160 x 160 pixel picture occupies 433 sectors, so if you only have a single density single sided disk drive of 358 sectors, you will need to plot a smaller picture, and modify the values in both programs above.

I look forward to someone sending me in something faster!!!! The second secon

Only one response to my request for data on the RANDOMIZE feature working or not when loading another program - affirming my previous note of the bug. My present console may be unique in NOT having it! One response is hardly statistically significant though...

WAR ZONE 2...ASGARD.

Some time ago I reviewed a fairly simple game called WAR ZONE, well now we have WAR ZONE 2, this time with a much higher standard of presentation, although the game is essentially the same, a shoot them-before-they-shoot-you event, with a cross hair to be moved around to destroy sundry graphics.

War Zone was playable, War Zone 2 is not only playable, but more interesting

If you enjoy blasting things, I think you will enjoy War Zone 2, which starts deceptively easy and gradually gets quite wicked!

The following article, mildly clarified by sjs, will i) show how Basic programs are stored on disk and in memory ii) show you how to use this knowledge to do strange things!

by WESLEY R. RICHARDSON NORTHCOAST 99ER'S, CLEVELAND, OH

The purpose of this article is to describe how an Extended BASIC (XB) program is stored on disk and how a program can have line numbers out of sequence, or even have hidden lines, yet still run properly. The intent is inform programmers so they can attempt to restore programs which have been altered.

The program CRAZY-XB1 is a very simple program which prints 'LINE 40' 'LINE 50' and so on to the screen. The listing for CRAZY-XB2 shows how the program can be altered to have descending line numbers. Note that line number 7 is for two different instructions. The listing for CRAZY-XB3 would appear that only line 10 is in the program, yet when CRAZY-XB3 is run, it will function exactly like -XB1 and -XB2. These listings are how the program would appear after you typed LIST.

```
30 REM NORTHCOAST 99ER'S, CLEVELAND, OH
40 PRINT "LINE 40"
50 PRINT "LINE 50" 03 = $800
60 PRINT "LINE 60" 08 08 8800
70 PRINT "LINE 70"
70 PRINT "LINE 70"
80 PRINT "LINE 80"
```

90 PRINT "LINE 90"

100 END

- 10 REM CRAZY-XB2
- 20 REM WESLEY R. RICHARDSON, FEB 1990
  30 REM NORTHCOAST 99ER'S, CLEVELAND, OH 9 PRINT "LINE 40"
- 8 PRINT "LINE 50"
- 7 PRINT "LINE 60" of w odd a rot size and size and said that and solines bisods I agadize
- 7 PRINT "LINE 70" hate viruseb elonia a sved vice sev il de arrobem see aploure
- 6 PRINT "LINE 80" Jemiola veliage a Jeld of been like mor , avoices 600 do gyrat
- 5 PRINT "LINE 90"
- 100 END : For this estimonation dissaid of of dulings accompa of bisyrot sloot

# 10 REM CRAZY-XB3

To understand how these programs work, we must first look at the Extended BASIC representation for the program. If you refer to the CRAZY-XB1 ASCII code sector listing (below), you will see that the

lines are listed in reverse order.

The disk sector listing has line 90, then 80 and so on, ending with the CRAZY-XB1 statement.

Note that if you edit a line or add a line, then that line gets moved to the beginning of the file. If line 40 is edited, then it will be in the file (and located in memory) before line 90.

The line number table in memory enables the computer to run the program in the correct sequence, even if the processor has to expend some time chasing around all over memory to do so!

on the screen will be in proper order, but internally they will be quite mixed. If you have a program, for example PROGNAME1, in which a special soul you have made several changes, the lines can be re-ordered by the following steps:

- 1) SAVE "DSK1.PROGNAME2", MERGE
- 2) NEW

3) MERGE "DSK1.PROGNAME2"
4) SAVE "DSK1.PROGNAME3"

40 MALEVALD AND TRANSPORTED TO THE PROGRAMMES TO T I suggest using different filenames in case you make an error, then you can recover using the original file. When creating a program, do all of you debugging and modifications and when your program is finished, then use the MERGE routine to organize the internal program lines.

rtent is isine more programment a social single Now that we understand the BASIC lines can be out of order in the file, how do we modify the line numbers? If you refer to the CRAZY-XB1 hex code sector listing, we will see how XB keeps track of the line numbers. In the first row, locate the 0064, that is line 100. Also in the first row is 005A (hex 50=decimal 80+ hex A=decimal 10), that is line 90. We can see the old line numbers in hexadecimal and decimal.

	and decimal.	
OLD LINE £	NEW LINE £ 5 1824 BELL Y CORRES NO 17368 11 to 11 1727 B1 2	
	HEX TY DEC Tells teeping block margard eds wed era conlis	
0064 = 100		
	0005 = 5	
0050 = 80	MESTER B. RICHARDSON, FEB 1990 6 = 6000	
0046 = 70	0007 = 7 - Navad Sa a 180 (1884 1884 1) . 8 1822 TEACORTEON 1	
003C = 60	0007 = 7	
0032 = 50	0008 = 8	
0028 = 40	0009 = 9	
001E = 30		
0014 = 20		
000A = 10		

Using a sector editor, I changed the old line number hex values to those indicated under new. If you examine rows one, two and three in the CRAZY-XB2 hex code sector listing, you will see these changes. But wait, how can the program still work?

Extended BASIC executes instructions according to memory location, not to line numbers (the actual ine numbers found in the line number table are not important in running a program, their place in the table is important).

When we list the CRAZY-XB2 program, it appears on the screen as I listed it previously. If you try to edit the program by typing 10 then FCTN X, you will be able to see lines 10, 20 and 30, but when you go to line 9, the old line 40, XB will tell you "LINE NOT FOUND." The program will still run correctly. THE STATE OF THE S

If we make one more change, we can hide some lines. By changing the sector row one value of 0064 for line 100 to a value like 0001, you will produce CRAZY-XB3. Now only line 10 can be viewed when listed, but the program still works fine.

Line numbers in XB range from 1 to 32767, or hex 0001 to 7FFF. If we change the line number to a value in the range of 8000 to FFFF, it will cause a BREAK in the program when that line is executed. For example, if the program reached the line number 83E8, the line number would then have the value of 8000 subtracted, leaving 03E8 and the message "BREAKPOINT IN 1000" would be displayed.

In the hex code sector listing for CRAZY-XB1, in lines 1 to 3, there are 2 byte or four digit numbers such as 373B, 373E, 374A, and 3756, after each line number. These refer to the memory location for the XB

The difference between adjacent values is the number of bytes used for the XB instruction. The format for each instruction is XXYYY...YYY00. The XX is the number of bytes used for the instruction, not including

Since the maximum value which can be represented is FF, the longest line length in XB is 255 bytes. Depending upon the statements which you use, this 255 byte length can have different ASCII lengths which you see when entering an XB program. The XB statements are stored in token format, for example PRINT is 9CC7. The word PRINT takes 5 ASCII bytes, but to XB, only requires 2 bytes to store 9CC7.

Some information such as the text contained in print statements is in the same format when saved to disk. For example the characters LINE 50 are stored on disk in the readable form as shown in line 7 of the ASCII code sector listing for CRAZY-XB1.

The third format which XB uses on disk for program files is for CALL statements. Memory must be reserved for variables and CALL statements. One way to find the tokens for each of the XB commands is to write a program using each of the commands on a separate line, and then look at the hex codes using a sector editor. Be sure to use the MERGE technique listed above if you wish to keep the sequence of lines in order when the program is saved to disk.

As I indicated earlier, I do not agree with using hidden instructions in XB programs. If you encounter one of the modified programs, perhaps now you will have some idea about how they were modified and the meaning of the values of an XB program stored on disk.

#### CRAZY-XB1 - ASCII CODE SECTOR LISTING

# CRAZY-XB1 - HEX CODE SECTOR LISTING

. + 79 7 . 7 . . d 7; . Z 7> P 7 J . F 7 V . < 7 b . 2 7 n 

002B 3739 3712 37D7 0064 373B 005A 373E 0050 374A 0046 3756 003C 3762 0032 376E 0028 377A 001E 3786 0014 37AA 000A 37CC 028B 000B 9CC7 074C 494E 4520 3930 000B

...LINE 80 ....L IN E 70 . . . . L IN E 60 . . . . L IN E 50 . . ...LINE 40.£. NO 9CC7 074C 494E 4520 3830 000B 9CC7 074C 494E 4520 3730 000B 9CC7 074C 494E 4520 3630 000B 9CC7 074C 494E 4520 3530 000B

RT HC OAST 99ER'S, C LE VE LA ND, OH.! . WESLEY R. RICH ARDSON, FEB 1990

9CC7 074C 494E 4520 3430 0023 9A20 4E4F 5254 4843 4F41 5354 2039 3945 5227 532C 2043 4C45 5645 4C41 4E44 2C20 4F48 0021

... CRAZY-XB1..? . . . . . . . . . . . R A Z Y - X 

9A20 5745 534C 4559 2052 2E20 5249 4348 4152 4453 4F4E 2C20 4645 4220 3139 3930 000C 9A20 4352 415A 592D 5842 3100 AA3F FF11 0300 0000 0600 01C3 5241 5A59 2D58 4231 2000 0000 0000 0100 0000 0000 0000

7777-751 A.C. ( some service travers to a

CRAZY-XB2 - HEX CODE SECTOR LISTING

. + 79 7. 7. . d 7; . . 7> .. 7J .. 7V .. 7b .. 7n 72...7...7. ..... L I N E 90 ..

CRAZY-XB2 - ASCII CODE SECTOR LISTING

002B 3739 3712 37D7 0064 373B 0005 373E 0006 374A 0007 3756 0007 3762 0008 376E 0009 377A 001E 3786 0014 37AA 000A 37CC 028B 000B 9CC7 074C 494E 4520 3930 000B

...LINE 80 ..... INE 70...LINE 60 . . . . L IN E 50 . . ...LINE 40.£. NO RT HC OAST 9 9 E R'S. 9CC7 074C 494E 4520 3830 000B 9CC7 074C 494E 4520 3730 000B 9CC7 074C 494E 4520 3630 000B 9CC7 074C 494E 4520 3530 000B 9CC7 074C 494E 4520 3430 0023 9A20 4E4F 5254 4843 4F41 5354 2039 3945 5227 532C

CLEVELAND, OH! . WESLEY R. RICH ARDSON, FEB 1990 ... CRAZY-XB2. ?

2043 4C45 5645 4C41 4E44 2C20 4F48 0021 9A20 5745 534C 4559 2052 2E20 5249 4348 4152 4453 4F4E 2C20 4645 4220 3139 3930 000C 9A20 4352 415A 592D 5842 3200 AA3F

. . . . . . . . . . . R A Z Y - X B 2 ED JECTUSE DE LESCHER ENGLANDES DE SANCO DE

FF11 0300 0000 0600 01C3 5241 5A59 2D58 4232 2000 0000 0000 0100 0000 0000 0000 

# CRAZY-XB3 - ASCII CODE SECTOR LISTING

# CRAZY-XB3 - HEX CODE SECTOR LISTING

. . 7 J . . 7 V . . 7 b . . 7 n . . 7 z . . 7 . . . 7 . . . 7 . ..... L I N E 90 ... . . L I N E 80 . . . . L INE 70 . . . . LINE 60 . . . . L I N E 50 . . ...LINE 40.£. NO RT HC OAST 99ER'S. CLEVELAND, OH.! . WESLEY R. RICH ARDSON, FEB 1990 ... CRAZY-XB3...? . . . . . . . . . R A Z Y - X

B 3

. + 7 9 7 . 7 . . . 7; . . 7 > 002B 3739 3712 37D7 0001 373B 0005 373E 0006 374A 0007 3756 0007 3762 0008 376E 0009 377A 001E 3786 0014 37AA 000A 37CC 028B 000B 9CC7 074C 494E 4520 3930 000B 9CC7 074C 494E 4520 3830 000B 9CC7 074C 494E 4520 3730 000B 9CC7 074C 494E 4520 3630 000B 9CC7 074C 494E 4520 3530 000B 9CC7 074C 494E 4520 3430 0023 9A20 4E4F 5254 4843 4F41 5354 2039 3945 5227 532C 2043 4C45 5645 4C41 4E44 2C20 4F48 0021 9A20 5745 534C 4559 2052 2E20 5249 4348 4152 4453 4F4E 2C20 4645 4220 3139 3930 000C 9A20 4352 415A 592D 5842 3300 AA3F FF11 0300 0000 0600 01C3 5241 5A59 2D58 4233 2000 0000 0000 0100 0000 0000 0000 

TIPS FROM THE TIGERCUB

No. 64

Tigercub Software 156 Collingwood Ave. Columbus, OH 43213

\*\*\*\*\*\*

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 on disk.

My TI-PD library now has over 500 disks of fairware (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 #5 will probably be printed by the time these Tips appear. and is available for \$1 which is deductible from the first order.

Back in the days of David Ahl's Creative Computing magazine, when computers were too expensive for hardware hacking and had memory too small to run much of a program, the emphasis was on "recreational computing". and the British TI'ers carry on that tradition. A recent issue of their excellent TI\*MES newsletter had this challenge - write a program to set up a circle of any chosen number of objects; starting at one, count them off by 10's, removing every 10th object. What are the numbers of the last two left?

This is my solution. It is not the best one, but it does show how strings can be used to perform math.

100 INPUT "NUMBER?": N 110 FOR J=1 TO N :: N\$=N\$&CH R\$(J):: NEXT J :: IF N<10 TH EN 140 120 N\$=SEG\$(N\$,11,255)&SEG\$( N\$,1,9):: IF LEN(N\$)>9 THEN 140 FOR J=1 TO 10 :: N\$=SEG\$ (N\$,2,255) &SEG\$ (N\$,1,1):: NE XT J :: N\$=SEG\$(N\$,1,LEN(N\$) -1):: IF LEN(N\$)>2 THEN 140 150 FOR J=1 TO 2 :: PRINT AS

A 12 Day BERESESS DARROUSE METAL TE Which reminds me that I forgot to give you the answer to that short CALL SOUND puzzler in Tips #62. A CALL SOUND, even with a positive duration, will be interrupted by a BEEP.

C(SEG\$(N\$,J,1)):: NEXT J

\$5 15 2(1-1)s3 (secannesss

Here's a bit of nonsense I worked up from an idea by Tim Brooks. Save this by SAVE DSK1.BUGS.MERGE . Then when you get a chance, load one of your friend's favorite programs, add this to it by MERGE DSK1.BUGS, and in the middle of the program somewhere put a line with 

32000 !@P+ Text = ded a A H 24 RT 24 27 = 20 32001 SUB BUGS # 81 745 987 810 32002 CALL CLEAR :: CALL CHA RSET :: CALL DELSPRITE(ALL): : CALL SOUND(225,220,0):: PR INT "\*ERROR 4 IN LINE 150": : PRINT "\*BUGS IN PROGRAM" 32003 CALL SCREEN(8):: FOR A =1 TO 500 :: NEXT A :: A\$(1) ="997E3CFF3C7EBD99" :: A\$(2) ="DB3CBD7E3CFFBD99" ;: X=1 : : CALL CHAR(96,A\$(X)) 32004 RANDOMIZE :: CALL MAGN IFY(2):: FOR T=1 TO 2 :: FOR A=1 TD 20 :: X=X+1+(X=2) \*2 :: CALL CHAR(96.A\$(X)):: FOR D=1 TO 20 :: NEXT D 32005 CALL SPRITE (#A.96.2.19 5.RND\*240,-5.0):: NEXT A :: NEXT T :: CALL CLEAR :: CALL DELSPRITE(ALL):: SUBEND \*\*\*\*\*\*\*\*\*\*

Here is a puzzle game for you brainy types. I worked it up from a game by Jack Sughrue -

100 ! PSYCHO by Jim Peterson 110 CALL CLEAR :: RANDOMIZE :: CALL SCREEN(2):: FOR S=1 TO 12 :: CALL COLOR(S.2.16): : NEXT S :: CALL VCHAR(1,31, 31,96):: CALL KEY(3,K,S) 120 RANDOMIZE :: Y\$(1) = "+" : : Y\$(2)="-" :: Y\$(3)="x" :: Y\$ (4) = "/" 130 CALL VCHAR(1.3.32.672):: D\$="" :: Y(0), X=INT(10\*RND+

140 DISPLAY AT (2.11): "PSYCHO ": " Enter P(lus). (M)inus.

(T) imes or (D) ivided by" 150 FOR J=1 TO 4 :: Y(J)=INT (10\*RND+5): Z(J-1)=INT(4\*RN)

160 IF Z(J-1)=1 THEN X=X+Y(J ):: GOTO 180 ELSE IF Z(J-1)= 2 THEN X=X-Y(J):: GDTO 180 E LSE IF Z(J-1)=3 THEN X=X\*Y(J-1)):: GOTO 180

170 IF X/Y(J) = INT(X/Y(J)) THE N X=X/Y(J)ELSE Z(J-1)=INT(3\*)RND+1):: GOTO 160

180 NEXT J :: R=6 :: FOR J=0 TO 3 :: DISPLAY AT(R.12):Y( J):: R=R+2 :: NEXT J :: DISP LAY AT(R.12):Y(4)

190 DISPLAY AT(R+1,12):" ":: DISPLAY AT(R+3,12):X 130 DISPLAY AT(15,1): "Output 200 FOR J=0 TO 3 :: D\$=D\$&ST R\$(Y(J))&Y\$(Z(J)):: NEXT J:: D\$=D\$&STR\$(Y(4))&"="&STR\$(

X):: FOR J=1 TO 4 210 ACCEPT AT (J\*2+5,12) SIZE (

1) VALIDATE ("PMTD"): A\$ :: IF A\$="" THEN 210

220 ON POS("PMTD", A\$, 1) GOSUB 270,280,290,300 4 41 28084 38184

230 DISPLAY AT(J\*2+4.12):"" :: DISPLAY AT(J\*2+6,12):Y(J) 240 NEXT J

> 250 IF Y(4)=X THEN DISPLAY A T(19,9): "RIGHT!" :: GOTO 260 ELSE DISPLAY AT(19.7): " WR ONG! OFF BY": ABS(X-Y(4)):: D

ISPLAY AT(21.3):D\$ 260 DISPLAY AT (23,2): "PLAY A

GAIN? Y/N" :: ACCEPT AT(23.1 B) SIZE(1) VALIDATE ("YN") : Q\$ : : IF Q\$="N" THEN CALL CLEAR :: STOP ELSE 130 4343 3 440 444 1 1934

270 Y(J)=Y(J-1)+Y(J):: RETUR

280 Y(J)=Y(J-1)-Y(J):: RETUR

290 Y(J)=Y(J-1)\*Y(J):: RETUR

300 Y(J)=Y(J-1)/Y(J):: RETUR

When text files are reformatted to a shorter line length, using the Funlweb Formatter, there are often long gaps at the ends of the lines, or between words if Fill and Adjust is used, due to long words which would have been hyphenated if the text had been originally typed in the shorter length. This little program will reformat text (containing carriage returns) to any shorter length and allow you to optionally hyphenate words which do not fit at the end of a line.

> 100 CALL CLEAR :: CALL SCREE N(5):: FOR SET=0 TO 12 :: CA LL COLOR(SET. 2.16):: NEXT SE and The an engineer artists account to the second

110 CALL CLEAR 120 DISPLAY AT(12,1): "Input filename?": "DSK" :: ACCEPT A T(13.4) BEEP: IF\$ :: OPEN #1:" DSK"&IF\$.INPUT

filename?":"DSK" :: ACCEPT AT(16,4)BEEP:OF\$ :: OPEN #2: "DSK"&OF\$,OUTPUT

140 DISPLAY AT(18,1): "Reform at to what length?" :: ACCEP T AT(18.26) SIZE(2) VALIDATE(D

150 IF EOF(1) THEN 270 :: CAL L CLEAR :: LINPUT #1:M\$ :: M \$=P\$&M\$ :: P\$=""

160 L=LEN(M\$)+(POS(M\$,CHR\$(1 3).1)<>0):: IF L<=R AND POS( M\$.CHR\$(13),1)<>0 THEN PRINT #2:M\$ :: GOTO 150 ELSE IF L

(R THEN P\$=M\$&" " :: GOTO 15 0 170 C\$=SEG\$(M\$.1.R):: CALL L

ASTPOS(C\$." ".Y) 180 IF Y(>0 THEN 190 ELSE PR

INT #2:C\$ :: M\$=SEG\$(M\$,R+1, 255):: GOTO 160

190 IF R-Y<3 THEN C\$=SEG\$(M\$ .1,Y):: M\$=SEG\$(M\$,Y+1,255): : PRINT #2:C\$ :: GOTO 160

200 X=POS(M\$," ",Y+1):: IF X =0 THEN X=LEN(M\$)ELSE IF X=R +1 THEN PRINT #2:C\$ :: M\$=SE G\$(M\$,Y+2,255):: GOTO 160

210 DISPLAY AT(2,1):M\$ :: DI SPLAY AT(8,1): SEG\$ (M\$,1,R) 220 DISPLAY AT(12,1):SEG\$(M\$ .Y+1.R-Y-1)&"-"&SEG\$(M\$.R.X-R+1):: Z=R-Y 230 DISPLAY AT(15.1): "Hyphen ate?" :: ACCEPT AT(15.12)SIZ E(1) VALIDATE("YNVn"):Q\$ :: I F Q\$="N" DR Q\$="n" THEN 260 240 ACCEPT AT (18.1) SIZE (Z) : H \$ :: IF POS(H\$."-".1)=0 THEN 250 C\$=SEG\$(C\$.1.Y)&H\$ :: M\$ =SEG\*(M\*,Y+1+LEN(H\*)-1,255):: PRINT #2:C\$ :: GOTO 160 260 PRINT #2:SEG\$(C\$.1.Y):: M\$=SEG\$(M\$.Y+1.255):: GOTO 1 270 CLOSE #1 :: CLOSE #2 :: STOP 280 SUB LASTPOS (A\$, B\$, Y):: X 290 X=POS(A\$.B\$.X+1):: IF X> 0 THEN Y=X :: GOTO 290 300 SUBEND -----------

I really think that all program listings should be published in 28-column format, as my Tips from the Tipercub have always been published, because that is how they appear on screen. making it much easier to key them in accurately. However, if you absolutely MUST reformat them, I think that this program will accurately reformat to/from any length up to 79 PROVIDING that you first put a carriage return at the end of every program line.

100 DISPLAY AT (3.6) ERASE ALL : "PROGRAM RELISTER": "": Wi 11 reformat a LISTed XBas ic program from any lineleng th to any other length." 110 DISPLAY AT(8.1):" Each propram line (not file li ne) must end in a carriag e return." 120 DISPLAY AT(12.1): "Input filename?": "DSK" :: ACCEPT A T(13,4): IF\$ :: DISPLAY AT(15 ,1): "Output filename?": "DSK" :: ACCEPT AT(16,4):OF\$ 130 DISPLAY AT(18,1): "Presen t line length?" :: ACCEPT AT (18.22) SIZE (2) VALIDATE (DIGIT 140 DISPLAY AT(20.1): "Reform at to what length?" :: ACCEP T AT (20, 26) SIZE (2) VALIDATE (D IGIT): X :: IF X=A THEN 130 150 DPEN #1: "DSK"&IF\$. INPUT :: OPEN #2: "DSK"&OF\$.OUTPUT :: IF X<A THEN 230 160 IF EOF (1) THEN 270 :: LIN PUT #1:M\$ :: L=LEN(M\$):: IF POS (M\$.CHR\$(13).1)=0 THEN 18 170 IF P+L<X+1 THEN PRINT #2 :M\$ :: P=0 :: GOTO 160 ELSE

PRINT #2:SEG\$ (M\$,1,X-P):SEG\$ (M\$.X-P+1.255):: P=0 :: GOTO 160 180 IF L(A THEN M\$=M\$&RPT\$(" ".A-L):: L=A 190 IF P=0 THEN PRINT #2:M\$; :: P=L :: GOTO 160 200 IF P+L<X THEN PRINT #2:M \$::: P=P+L :: GOTO 160 210 IF P+L=X THEN PRINT #2:M \$ :: P=0 :: GOTO 160 220 PRINT #2: SEG\$ (M\$.1.X-P): SEG\$ (M\$, X-P+1, 255)::: P=LEN( SEG\$(M\$.X-P+1.255)):: GOTO 1 230 IF EDF(1) THEN 270 :: LIN

PHT #1 . M . 240 L=LEN(M\$):: IF L+P>X THE N FRINT #2: SEG\$ (M\$, 1, X-P):: M\$=SEG\$(M\$, X-P+1, 255):: P=0 :: GOTO 240 250 IF M\$=CHR\$(13) THEN 230 260 IF POS(M\$, CHR\$(13),1)<>0

THEN PRINT #2:M\$ :: P=0 :: GOTO 230 ELSE PRINT #2: M\$::: P=LEN(M\$):: GOTO 230 270 CLOSE #1 :: CLOSE #2

MEMORY FULL

Jim Peterson



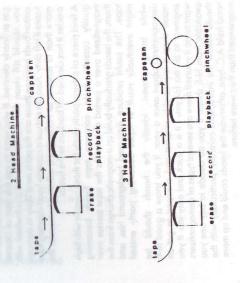
When a tape is played, some of its oxide coating will become loose and stick to the various parts of the machine. This will happen to **all** tapes even if they are well looked after (poor handling and storage will make this effect much worse - see section on 'Looking after your Cassettes').

Some cassette owners probably know they ought to clean their players but perhaps do not know why, or how to do it. Others simply forget or don't bother. Yet **most** tape faults are caused by the **player** and not by anything wrong with the cassette itself.

# LOOKING AFTER YOUR CASSETTE PLAYER A) Why you should clean it

The heads are mounted at a very exact angle to the tape so as to get the best possible sound when recording or playing back. If the heads become loose, or get knocked out of position, then the sound auality from the machine will be poor, and the heads will need re-setting by an engineer.

The record/playback heads will have a highly polished metal face with a tiny gap in the middle - so tiny you may not be able to see it. The gap allows the player to record the high pitched sounds which are found in all types of music. The erase head is built in much the same way but the gap is always wider because this head is not needed to record or replay anything it only wipes the tape clean when a new recording is made.



Low to mid-price machines will usually have 2 heads - an erase head, and a combined record/playback head. More expensive hi-fi machines may have 3 heads - one each for erasing, recording and playback.

All cassette players work by pulling the tape across a number of 'heads'. These heads carry ou the erasing, the recording and the playback of the sounds on the tape. The pulling is done by the capstan (a revolving metal spindle) and the pinchwheel (a rubber roller which presses the tape evenly against the spindle when you are recording or playing back).

THE CASSETTE PLAYER

CASSETTE USERS....
We have many members using cassettes for storage, and as it is some time since we passed on any tips for cassete use, here are some courtesy of Manchester City Council. Cassettes are often taken for granted, resulting in reduced life...

Cassettes seem much less fragile and much easier to look after than records - in fact some people appear to think they don't need any looking after at all. But to get the best from them, your cassettes and cassette player both need a certain amount of care and attention. Without it, your tapes can be damaged by your player, and your player can be damaged by your player, and your player can be damaged by your tapes.

Some of the information which follows is quite detailed, so to help, here is a list of **DO's** and **DON'T's** for easy checking. If you want to know more about why you should or shouldn't do certain things you will find the answers in the sections that follow the list.

clean your machine regularly. This means cleaning the heads, the capstan and the pinch-wheel. (If you are not sure what these are look in the section called The Cassette Player'.) It is best to use cotton buds and alcohoi - tape head cleaning cassettes can also be useful.

Months of the properties of the pro

**DO** .....Keep cassettes away from children and pets.

**DON'T...**Store cassettes where they will get hot, cold or damp.

dashboard or near the ignition. If the weather is either very hot or very cold, don't store them in cars at all.

**DON'T**...Store cassettes on or near any piece of electrical equipment.

DON'T... Use C120's unless your machine is clean, and don't fast-wind/rewind them unless it is absolutely necessary. Also please don't fast-wind/rewind double (2LP) tapes.

**DON'T** . . . Drop, bang, or otherwise knock cassettes about.

# THE CASSETTE TAPE

A cassette basically contains a long piece of very thin tape which needs to run extremely smoothly from one spool to another. To do this properly, and to sound as good as it should, both the tape and the cassette player need to be in good condition.

The tape inside a cassette is a plastic ribbon which has been coated on one side with a layer of iron oxides. These oxides become magnetized when the tape is recorded, and the magnetic pattern on them is turned back into sound when the cassette is re-played. The thickness of the tape will depend on whether it is a C60, C90 or C120. It is perhaps better to think of 'thinness' rather than 'thickness' because a C60 tape is finer than a human hair, and a C120 is very much thinner.

C120's quite often cause problems in low to mid-price players because the mechanical parts of these machines find it difficult to cope with the extreme thinness of these tapes. The most common faults are jamming, tape spilling into the machine, stretching and snapping. Even C90's and pre-recorded double play (2LP) cassettes can suffer from the same problems. If you are using any of these types of tape it is **very** important that your machine is clean and the tapes are well looked after.

3) **Erase Head** - this head is used less often than the record/playback heads, and doesn't wear as quickly. Even so, it should be kept clean. Bad oxide build-up will mean that when it is used it won't be able to erase properly. Faint sounds from an earlier recording can be left on the tape and these may well spoil a new recording.

The best way to keep all these parts clean is to use cotton budseither dipped in alcohol or head-cleaning fluid. **Don't use any other fluids** as you could damage your player. (Isopropyl alcohol, which we recommend for stylus cleaning and spot-cleaning records is **not** recommended for cleaning cassette players. It can cause the pinchwheels on some machines to swell. Methylated spirits can be used safely.) B) How you should clean it be clean in order to press evenly on the tape as they pull it pas' the heads. This allows the tape to wind smoothly onto the take-up spool. Oxide build-up on these parts causes the pinchroller to become out of shape, and this means the pressure on the tape will be uneven. C60's will not be seriously affected unless the oxide build-up is very bad, but C120's, 2LP affected unless the oxide build-up is very bad, but C120's, 2LP affected unless the oxide build-up is very bad, but C120's, 2LP affected unless the oxide build-up is very bad, but C120's, 2LP affected unless the oxide build-up is very bad, but C120's, 2LP are recorded tapes, and C90's will be much more likely to jam, pre-recorded tapes, and C90's will be much more likely to jam, snap, or get 'chewed up'. If this happens then, at best, the tape will be runned, and at worst the tape will spill into the machine and sometimes expensive to get it out!

2) Record and Piayback Heads - these need to be clean so that the tape has the best possible contact with the be clean so that the tape has the best possible contact with the bed lead as oxide build-up on it, the oxides will be head. If a head has oxide build-up on it, the oxides will be head against the head as the tape passes through. This will ground against the head as the tape passes through. This will ground against the head will no longer be able to player to sound dull because the head will no longer be able to record or play back the brighter, higher pitched sounds.

In most machines the best way to get at the heads for cleaning is to open the lid and press the play button, but with the power switched off. This will bring the heads and pinchwheel forward so you can reach them with a cotton bud.

Dip one end of a cotton bud in the fluid. Rub the heads, pinchwheel and capstan in turn. Dry off each part with the dry end of the cotton bud. Use a different cotton bud for each part. The part will be clean when you can see no brown/black stain left on the cotton bud.

If you do this regularly, say once a week or so, or every 10-15 hours' playing time, then your machine and tapes will continue to sound good, and will last a lot longer.

It is possible to buy cassette head cleaners that look just like a normal cassette. You put one in your machine and play it through in the normal way. These cleaners do work, but only really on the heads - the capstan and the pinchwheel do not get properly cleaned. If you want to use this type of cleaner you should still clean the capstan and the pinchwheel with cotton buds because these are the parts most likely to damage tapes if left dirty. Also, do not use a cassette head cleaning tape more than 12 times or so - after that it will start to transfer the oxides it has cleaned off back on to the heads.

A final piece of advice, particularly for those people with high quality cassette decks, concerns tape heads becoming magnetized over a period of time. As the tape passes over them, the heads gradually build up a magnetic charge from the tape. This could produce increased background hiss, and might eventually result in tapes becoming partly erased while being played. Ideally the heads should be demagnetized about once a year. If you know a good hi-fi repair shop it is probably simplest and cheapest to get an engineer to do this for you - it is a quick and easy job. If you want to do it yourself there are several types of demagnetizers should be defended to be approached to be yone you should follow the instructions carefully, particularly with the h

50 REM : "HIGH-RESOLUTION X"Y GRAPH PLOTTER"

60 CALL CLEAR

70 GOSUB 960
80 REM : (CAPTIONS)
90 CALL CLEAR

100 RE12
1110 REM :
100 REM 590 J=8\*D(L,5,K)+4\*D(L,6,K)+2\*D(L,7,K)+D(L,8,K)
600 H(17-2\*L)=N
610 H(18-2\*L)=J
620 NEXT L
630 FOR L=1 TO 16
640 IF H(L)<10 THEN 660
650 ON 16-H(L)GOTO 680,700,720,740,760,780 660 I\$(L)=STR\$(H(L)) 670 GOTO 790

```
680 I$(L)="F"
690 GOTO 790
700 I$(L)="E"
710 GOTO 790
720 I$(L)="D"
730 GOTO 790
740 I$(L)="C"
750 GOTO 790
760 Is(L)="B"
770 GOTO 790
780 I$(L)="A"
790 NEXT L
800 \ P\$=I\$(1)\&I\$(2)\&I\$(3)\&I\$(4)\&I\$(5)\&I\$(6)\&I\$(7)\&I\$(8)\&I\$(9)\&I\$(10)\&I\$(11)\&I\$(12)\&I\$(12)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&I\$(13)\&
)&I$(13)&I$(14)&I$(15)&I$(16)
810 CALL CHAR(M.P$)
820 CALL HCHAR(D(9,8,K),A+C,M)
830 M=M-1
840 IF M=33 THEN 950
850 FOR L=1 TO 9
 860 FOR J=1 TO 8
 870 D(L,J,K)=0
 880 NEXT J
 890 NEXT L
 900 NEXT K
 910 FOR L=1 TO 8
  920 G(L)=0
  930 NEXT L
  940 NEXT A
  950 CALL KEY(P,Q,R)
  960 IF R >1 THEN 950
  970 END
  980 REM : (CAPTIONS)
  990 CALL CHAR(64,"00FF000000000000")
  1000 FOR K=2 TO 8
  1010 CALL COLOR(K, 1, 4)
  1020 NEXT K
  1030 READ A$
  1040 IF AS="X" THEN 1070
  1050 PRINT AS
  1060 GOTO 1030
  1070 FOR K=2 TO 8
  1080 CALL COLOR(K, 2, 4)
  1090 NEXT K
  1100 FOR K=1 TO 3000
  1110 NEXT K
  1120 RETURN
                                                                                                                                                                                                                                                       X/Y GRA
  1130 DATA "
                                                          HIGH-RESOLUTION
  PH PLOT
                                                                                                                                ",,,EDIT 300 TO SPECIFY Y=F(X) @@@@@@@@
  1140 DATA "
                                                           1150 DATA EDIT 100 TO LOCATE X AXIS
                                                                                                                                  . " . 99999999
                                                                                                                                                                                                                (ROWS 1 TO 24)"
                                                                                                                                   @@@@@@@@,"
                                                                                                                                                                                                          (COLUMNS 3 TO 32)",
  1160 DATA EDIT 110 TO LOCATE Y AXIS
  1170 DATA RUN 90 TO BYPASS CAPTION
                                                                                                                                   @@@@@@, , , X
```

The author of this program wishes to remain anonymous. Since it appeared in TI\*HOME some time ago we have agreed to this request. Ed.

# THE MONKEY TYPIST.

Here is an oddment inapired by the Royal Institution Xmas Lectures 1991. Richard Dawkins used a variant of the monkey on a typewriter idea to make a point about natural selection but confined himself to two extreme cases: 1) where the selection of letter was random throughout and 2) where a randomly-chosen letter that matched the target letter was preserved in all subsequent attempts. I thought it would be interesting to explore the middle ground where "correct" letters still stood a chance of being changed. This would correspond to a "favourable" mutation not being able to withstand local pressures every time it arose.

Specifically, after the first appearance, the correct letter acquires probability V of being reconsidered in the next attempt. If it survives, the probability falls to Val authorized to Val Val etc. If it does not survive, random selection takes over until another correct letter appears and probability V applies again.

On my TI, each attempt at the 13-letter phrase took about 1.2 sec. The program

in XB for convenience but needs only simple Basic resources.

```
code, Except, that is, for one surprished that a man the surprished
100 OPTION BASE 1
110 DIM TXTE(13), TARGE(13), MODF(13), CHE(28)
120 D1=13 :: D2=28
130 TI£="TI OR NOT TI?" :: C£=" ABCDEFGHIJKLMNOPQRSTUVWXYZ?"
140 FOR I=1 TO D1 :: TARGE(I)=SEGE(TIE,I,1) :: NEXT I
150 FOR I=1 TO D2 :: CH£(I)=SEG£(C£,I,1) :: NEXT I
160 PRINT :: INPUT "VULNERABILITY INDEX OF A MATCH (O TO 1) ":V
170 N=O :: RANDOMIZE
180 FOR I=1 TO D1 :: MODF(I)=1 :: NEXT I
200 FOR I=1 TO D1 :: IF MODF(I)=0 THEN 240 :: IF MODF(I)=1 THEN 230
210 IF RND>=MODF(I) THEN 220 :: MODF(I)=1 :: GOTO 230
220 MODF(I)=MODF(I)*V :: GOTO 240
230 TXTE(I)=CHE(INT(1+RND*D2)) :: IF TXTE(I)=TARGE(I) THEN 220 :: CNT=CNT+1
240 NEXT I
250 N=N+1 :: T£=TXT£(1)
260 FOR J=2 TO D1 :: T£=T£&TXT£(J) :: NEXT J
270 PRINT N,TE :: IF CNT>O THEN 190
280 PRINT :: PRINT "FINISHED WITH V=";V
290 END
```

With V=1, we have raw random selection. The probability of matching the target phrase in one attempt is then about 1 in  $6.5*10_{\rm A}18$  so that the expectation for run time is of order  $10_{\rm A}11$  years! With V=0 (guaranteed survival for correct letters), the theoretical expectation of the number of attempts (according to me) is about 87.9. In ten runs, the actual numbers ranged from 37 to 124, averaging 88.2. I cannot manage the theory for middling V but tests indicate that more than 0.7 will entail very long runs.

You could think up some other laws to replace those in statements 200-230. What about giving "near-misses" an enhanced probability of later coming on target? Or a facility for gradually learning to exclude a letter from a position where it is off-target?

Walter Allum

# TRIALS OF A FACTORIZATION ROUTINE.

# mandamenter, winds the seemed now Walter Allum care horse and see in the large transfer.

It is common knowledge that factorizing a "large" integer in an acceptable time can be difficult or impossible. Good for the users of

Public Key Cryptography; a hindrance otherwise.

Because of this, powerful procedures for factorizing have been developed. Some, the "probabilistic" methods, on the principle of half a loaf being better than none, buy relative simplicity at the price of occasional failure. None of them is likely to be of practical value in the modest environment of the TI but I wanted to see them working, even if in only a shadowy version. Unfortunately, from my limited access to the literature, the underlying math seems a bit too daunting for me to code. Except, that is, for one surprisingly simple procedure, due to Pollard. I have learned a lot from trying it out and I am sure that many members would find something to interest them, whether in making it better suited to TI or understanding its operation.

Here is a no-frills version for factorizing integers not larger than about  $10_{\rm A}6$ . In XB for compactness but only simple Basic statements

needed.

100 DIM X(1400)

110 PRINT :: INPUT "INTEGER TO FACTORIZE ":N :: INPUT "INTEGER PARAMETER ":A

120 RANDOMIZE :: X(0)=INT(RND\*N) AND OR(1) MORE HAR IN ON ANY ROPE OF A

- 130 FOR I=1 TO 1400 :: X(I)=X(I-1)\*X(I-1)+A :: X(I)=X(I)-N\*INT(X(I)/N)
- 140 FOR J=I-1 TO 0 STEP -1 :: T=ABS(X(I)-X(J)) :: AA=N :: GCD=T

150 IF GCD=0 THEN 190

160 IF GCD=1 THEN 180 :: R=AA-INT(AA/GCD)\*GCD :: IF R=O THEN 200 :: IF R=1 THEN 180

170 AA=GCD :: GCD=R :: GOTO 160

- 180 NEXT J :: NEXT I :: PRINT :: PRINT "PROGRAM OUT OF SPACE" ::
- 190 PRINT :: PRINT "ROUTINE FAILS FOR N= "&STR\$(N):" WITH PARAMET ER=";A:"AND SEED=";X(0) :: GOTO 210
- 200 F2=N/GCD :: PRINT :: PRINT "FACTORS OF "&STR\$(N):" ARE";GCD; F2

210 END

- Note 1: A is integer chosen arbitrarily, magnitude < N. I have mostly used 1. I find a note saying that A should not be 0 or −2 but no indication why. As A=−M behaves just as A=N-M, I don't see much point in using negative A.
  - Note 2: Neither of the factors GCD and F2 will necessarily be prime.

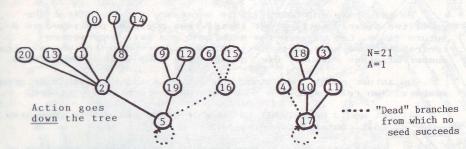
The routine exploits the fact that Euclid's algorithm lets us determine whether two integers have a common factor without needing to factorize either of them. We apply it to N and a succession of positive test integers  $T(\blacktriangleleft N)$  until we strike lucky. So long as the messages in statements 180 & 190 have not appeared, there is hope. (Of course, if N is prime, no factors will ever be found so you should do a primality test first.) The problem is getting a suitable T as soon as possible.

Take N=741317 as an example; of all the integers N, only about 1 in 430 has a common factor with it, beleating T randomly, we would have a 50% chance of acceeding with 124 to 595 attempts. In 10 trials, my Pollard program required from 34 to 250 attempts, averaging 117. By the way, with the 78-digit Fermal number that succumbed to Pollard's method, the proportion of suitable T is only about 1 in 10x15. At 1 microsecond per attempt, random selection would entail an expectation of run length about 32 years, day and night. I read that it actually took 2hr on a University 1100/42. Somehow, the simple procedure in sn.130 and 140 finds the needle in this haystack.

Fair comparisons with other procedures are difficult because each is at its best in different circumstances. With integers under 10.6, and on time alone, Pollard's method(P) is usually much inferior to trial division(TD) or Fermat's method(F). My F program factorized 74117 in lace, TD did it in 6.5 sec and P in average 73 sec. I do not know at what size of number the advantage turns to P, as it must from

its employment on the Fermat number.

Why is P only probabilistic? Clearly, there are no more than N distinct values of X available so that, sooner or later, the sequence must loop. Then, if a suitable T has not yet emerged, it never will with that particular seed. Actually, I have never seen the full range of X developed from one seed. The X values are grouped on a "tree" or "trees" as is illustrated below for the trivial case of N=21. Looping then sets in earlier. There can even be immediate failure, as witness seeds 5 and 17 in the illustration.



While a suitable T usually arises before X runs out, several seeds may have to be tried. I do not know whether there are A values that make this less likely. Nor can I recommend a specially good choice of new seed when seed s has just failed, except to suggest avoiding N-a.

Extending the program to cope with integers up to about  $10 \text{\AA} 12$  is pretty straightforward. We need a modification of sn.130 to evaluate the remainders (mod N) of the X squares without having to store the squares themselves. For still larger numbers, I imagine there would be difficulties needing something more than XB. Some compensating economies of space might be had by reducing the X array, remembering that m X values furnish  $\frac{1}{4}$ m(m+1) attempts.

15 Inglewood Close Darlington Co Durham DL1 2TX Tel (0325) 480889

Dear Alan,

I write with regards to the views expressed in TI\*MES which, I am sure we all know, are the views of individuals. This is quite clearly set out in the disclaimer on the inside front cover of each issue.

I, and I would think a number of other group members, am sick of the numerous comments in the last few issues about the views being those of certain people, Stephen Shaw's name seems to crop up time after time.

Surely we live in a Country where the right to free speech exists, and as long as it is not defamatory or libelous why not let them speak. It is just this sort of back billing and censorship which will lead to members leaving the group, goodness knows we need all the members and support we can get. If I was an outsider reading the magazine it would certainly dissuade me from joining the group.

So lets have an end to it, if necessary a comment from the committee clearly setting out their views could be published in a future issue. Let us keep the excellent and well experienced contributors we have, after all we don't want to go the way a large number of other computer user groups have gone — out of existence.

On anther point, how about publishing a list in TI\*MES of group members. Two reasons spring to mind, there are probably others.

- It would be nelpful to know of other members in your area so that contacts could be made.
- If travelling away from home or moving house, meetings or visits could be arranged with members in that area.

Waluncan stages of the same and the same and

Yours Sincerely (1991) About the real & beet note book yes 10

David Duncan