

# TI\*MES

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

Chairman: Trevor Stevens. Tel. 0623 793077  
249 Southwell Rd. East, Rainworth, Notts. NG12 OBN  
Vice Chairman & Programming: Mark Wills.  
12 "Rosehill", Betton St., Shrewsbury, Shropshire. SY3 7YN  
General Secretary: Richard Twynning. Tel. 0623 27670  
24 Peel Rd., Mansfield, Notts. NG19 6HB  
Publicity Officer & Module Librarian: Philip Trotter. Tel.0642 817356  
80 Martonburn Rd., Grovehill, Middlesborough. TS4 2TH  
Membership Secretary & Past Issues: Alasdair Bryce. Tel. 0389 65903  
51 Dumbaie Ave., Silvertown, 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.  
Disk:Stephen Shaw.(Journal Exchange)  
10 Alstone Rd., Stockport, Cheshire. SK4 5AH  
Publications:Mike Curtis. Tel. 0209 219051  
21 Treliske Rd., Roseland Gdns., Redruth, Cornwall. TR15 1QE

## MAGAZINE CONTENTS

IFC Editorial, Disclaimer, Date for copy for the next issue.  
1 Membership News by Alasdair Bryce.  
4 From the Chairman's Chair by Trevor Stevens.  
8 News and Reviews by Richard Twynning.  
31 Article by Stephen Shaw.  
33 Further Report on Lima by Stephen Shaw.  
35 Chaos Catalogue, Mancala, TI Writer Tip, Program by Stephen Shaw.  
36 Date by Wesley R. Richardson, Algorithm Design by G.J.Christensen(sjs)  
40 Myarc Ceases Trading. Micropendium(sjs)  
41 Harrison's Easy Sort reviewed by Charles Good(sjs)  
42 Letter from Australia by Tony McGovern(sjs)  
43 Supercarts by Andy Frueh(sjs)  
45 Harrison's Code Breakers by Charles Good(sjs)  
46 Programming Music the Easy Way by Jim Peterson(sjs)  
49 TI Writer and Clones, Abbreviated Information byStephen Shaw.  
53 Disk Fix by Wesley R. Richardson(sjs)  
55 The Art of Assembly- Part 1 by Bruce Harrison(sjs)  
57 Disk Library Report by Stephen Shaw.  
60 Tips from the Tigercub nos.4,5,6,27,28,&66by Jim Peterson(sjs)

ISSUE NO. 38

AUTUMN 1992

#### EDITORIAL

Welcome to the Autumn issue of your magazine! It is a big one , to keep you happy as the days shorten.

This issue contains the minutes of the A.G.M. , as well as the rest of a long article by our new General Secretary Richard Twynning. I think you will find it fascinating! In response to a number of requests our Membership Secretary Alasdair Bryce has provided area listsof those members who permit. The Chairman has contributed instructions for converting Romox modules to add 8 or possibly later 16k of extra RAM to the 4A. Of course you do seem to need an E/A Grom and a full system.

Finally Stephen makes his usual massive contribution , including the repeat of early Tigercub columns , surely new to later members , and K.F.Hughes reviews Stoll's "The Cuckoo's Egg".

#### DISCLAIMER

The views expressed by contributors to this magazine are strictly their own , and do not necessarily represent those of the Committee. Contrary opinions are welcome and will be given equal prominence if at all possible.

Attributions not made or made in error will be corrected on request.

#### NEXT COPY DATE

Lots of copy for the big Winter issue to reach me by 1st. December please.

Once again the copy for this issue has been easily reproducible , with only a very limited amount of trimming. The Treasurer and I are both delighted! Thank you.

# TI USERS GROUP UK MEMBERSHIP NEWS

By Alasdair Bryce

Since the last issue we have been joined by 3 more new members. They are Andrew Stockley, John Capstick and Jan Lis who hails from Czechoslovakia. Stephen Shaw has also become an honorary life member of TIUG(UK). Stephen was a founder member of the group and has been a great servant to the TI community and I'm sure that you would all want to join me in thanking him for his invaluable contribution to the group.

It is once again my sad duty to report the death of a member of the group. David Jones from Cardigan passed away in July and will be sadly missed.

In response to your requests for area lists to be printed in TI\*MES I have at last got a full list of group members together for publication in this issue. You will see for yourselves that our membership is spread over the length and breadth of the country although some areas are better represented than others. I have only included in the lists the names and addresses of those who ticked the relevant "yes" box on their membership forms. Those who did not wish their names etc. to be published are grouped together under the category of "others" at the foot of each area list.

I hope that this may encourage some of you to get together and organise meetings or TI workshops at local level. If any of you do get a local group set up perhaps you could drop me line and let me know how you are getting on.

Those of you who are looking to buy back issues of TI\*MES may be interested to know that I now have a number of copies of issues 32 and 33. These can be purchased on the usual terms at £1 each plus p&p. If anyone is trying to get hold of another issue not listed below I will do my best to get a copy it for you so let me know what you need.

The full list of available issues is now as follows - 4,8,9,10,11, 13,14,16,18,19,20,22,24,25,27,29,32,33,34,35,36 and 37.

## TI USERS GROUP MEMBERS

### SCOTLAND

ALASDAIR BRYCE	51 DUMBUIE AVENUE, SILVERTON, DUMBARTON, SCOTLAND, G82 2JH	03	65	3
ROY DOUGLASS	BRIDGES' COTTAGE, AROS, ISLE OF MULL, PA72 6JP	06	30	62
PETER JACKSON	61 CRUMMOCK GARDENS, BEITH, Ayrshire, KA15 2HF	05	53	7
ROBERT.R. WILSON	24 MUIREND STREET, KILBIRNIE, Ayrshire, KA25 7DG	KI	.68	43
+ 1 OTHER				

### NORTH EAST

KEITH BULLOCK	8 ST MARGARETS GROVE, THORNABY, STOCKTON-D-TEES CLEVELAND, TS17 8JR	06	67	10
DAVID DUNCAN	15 INGLEWOOD CLOSE, DARLINGTON, CO. DURHAM, DL1 2TX	03	48	89
PHILIP TROTTER	80 MARTONBURN ROAD, GROVEHILL, MIDDLESBROUGH, CLEVELAND, TS4 2TH	06	81	56
+ 2 OTHERS				

### NORTH WEST

ROGER G NICHOLL	"SUNNYHURST", FOXFIELD, BROUGHTON-IN-FURNESS, CUMBRIA, LA20 6BX	02	71	91
+ 1 OTHER				

### YORKSHIRE & HUMBERSIDE

DEREK R HAYWARD	2 DORE HALL CROFT, DORE, SHEFFIELD, YORKSHIRE, S17 3HD	07	36	88
PETER J HUTCHISON	6 MOORLANDS VIEW, FREE SCHOOL LANE, SAVILE PARK, HALIFAX, WEST YORKSHIRE, HX1 2XQ	04	35	57
+ 5 OTHERS				

## MANCHESTER & MERSEYSIDE

PAUL BAKER	149 WIGAN ROAD, STANDISH, WIGAN, LANCASHIRE, WN6 0AG	0257 4 571
CHRISTINE & ROSS BENNETT	20 OAK AVENUE, ROMILEY, STOCKPORT, CHESHIRE, SK6 4DN	061 43 7298
ALAN BRAY	4 TYRRELL ROAD, REDDISH, STOCKPORT, CHESHIRE, SK5 6UZ	061 44 6748
WARREN BRAY	THE RING O' BELLS,, 153 MARGARET STREET,, ASHTON-UNDER-LYNE, LANCASHIRE, OL6 7TS	061 33 0203
EDDY J CARTER	74 CRAIG ROAD, HEATON MERSEY, STOCKPORT, SK4 2AP	061 43 4270
ROBERT G CHRISTIAN	27 HOWEY RISE, FRODSHAM, CHESHIRE, WA6 6DN	0928 3 23
JOHN S DUNNING	8 SHERBROOKE AVENUE, UPPERMILL, OLDHAM, LANCASHIRE, OL3 6EF	0457 8 056
DAVID McKIE	16 FINCHAM ROAD, KNOTTY ASH, LIVERPOOL, MERSEYSIDE, L14 9NM	051 22 1471
BILL MORAN	30 ELSIE STREET, FARNWORTH, BOLTON, LANCASHIRE, BL4 9HT	0204 7 78
ALAN B RUTHERFORD	13 THE CIRCUIT, WILMSLOW, CHESHIRE, SK9 6DA	0625 5 642
CLIVE & AUDREY SCALLY	4 WOODBANK PARK, SUMMERFIELDS, DIXON, WIRRAL, L43 9WN	EX DIR
STEPHEN SHAW	10 ALSTONE ROAD, STOCKPORT, CHESHIRE, SK4 5AH	061 43 6097
THOMAS N SHAW	15 TOWNFIELD AVENUE, ASHTON-IN-MAKERFIELD, WIGAN, LANCS, WN4 9PD	0942 7 560
TOM SOUTHWELL	12 CHEQUER LANE, UPHOLLAND, LANCASHIRE, WN8 0DE	0695 6 006
ANDREW STOCKLEY	40 OLDFIELD ROAD, HESWALL, L60 6SF	051 34 9339

## NORTH EAST MIDLANDS

JIM BALLINGER	5 OFFERTON AVENUE, DERBY, DE3 8DU	0332 77 512
IAN A HEWITT	26 OUTWOODS DRIVE, LOUGHBOROUGH, LEICESTERSHIRE, LE11 3LT	0509 21 259
GEORGE MICHEL	34 PIPPIN HILL, DENBY VILLAGE, RIPLEY, DE5 8PD	0332 8E 367
VICTORIA MORRIS	LONGLEY, 19 KITLINGS LANE,, WALTON ON THE HILL,, STAFFORD, ST17 0LE	660008
GARY SMITH	55 BOUNDARY ROAD, NEWARK, NOTTINGHAMSHIRE, NG24 4AJ	0636 7C 767
TREVOR STEVENS	249 SOUTHWELL ROAD EAST, RAINWORTH, NOTTINGHAMSHIRE, NG21 0BN	0623 75 377
RICHARD TWYNING	24 PEEL ROAD,, MANSFIELD,, NOTTINGHAMSHIRE., NG19 6HB	0623 27 70
SAM WARDLE	83 NORBETT ROAD, ARNOLD, NOTTINGHAM, NG5 8EA	0602 20 336
+ 4 OTHERS		

## WALES & THE MARCHES

COLIN ASTBURY	16 EATON AVENUE, HANDBRIDGE, CHESTER, CHESHIRE, CH4 7HB	0244 6 802
MIKE GODDARD	"SARNIA", CEMETERY ROAD, RHOSLLANERCHRUGOG, WREXHAM, CLWYD, LL14 2BY	0978 8 547
DAVID ELLIS GRIFFITHS	2 HEOL-Y-TYLA,, DYFFRYN RHONDDA, P. TALBOT WEST GLAMORGAN., SA13 3EY	0639 8 771
ANDREW J LARSEN	121 MAES TY CANOL, BAGLAN, PORT TALBOT, WEST GLAMORGAN, SA12 BUR	0639 8 628
HENRY McCOOK	COLLEGE HOUSE, PENRHOS COLLEGE, LLANNERCH ROAD EAST, COLWYN BAY CLWYD, LL28 4DA	0492 5 669
WILLIAM B VOWLES	35 HALTON ROAD, UPTON BY CHESTER, CHESHIRE, CH2 1SN	0244 3 895
+ 3 OTHERS		

## BIRMINGHAM & N.W. MIDLANDS

ROBERT J BATES	34 BROWNS LANE, TAMWORTH, STAFFORDSHIRE, B79 8TF	0827 6 91
DAVID H CAINE	19 CLAUGHTON AVENUE, CREWE, CHESHIRE, CW2 6EZ	0270 51 529
PASCAL G HARRIS	6 ALBANY ROAD, HARBORNE, BIRMINGHAM, B17 9JX	021 42 4316
EDWARD H SHAW	CROW HOLT FARM, BASFORD, LEEK, STAFFORDSHIRE, ST13 7DU	0538 31 382
ERIC J WILLIAMS	211 MANOR WAY, CREWE, CHESHIRE, CW2 6PH	0270 68 55
+ 1 OTHER		

## EASTERN COUNTIES

WALTER J ALLUM	8 NEWBRIDGE ROAD, TIPTREE, COLCHESTER, ESSEX, CO5 0HS	0621 81 530
CHRISTOPHER T SNELL	6 BELMONDE DRIVE, SPRINGFIELD, CHELMSFORD, ESSEX, CM1 5TY	0245 46 312
JOHN STOCKS	11 STONEHILL ROAD, ROXWELL, CHELMSFORD, ESSEX, CM1 4PF	0245 24 137
+ 4 OTHERS		

## SOUTH EAST MIDLANDS

MARTIN LEONARD ARCHER	3 TELFORD WAY, BLAKELANDS, MILTON KEYNES, MK14 5LB	0908 61 182
PETER G O BROOKS	96 BANBURY ROAD, OXFORD, OX2 6JT	0865 51 122
SIMON DORRICOIT	27 LONDON ROAD, BEDFORD, MK42 0PB	0234 27 171
DAVID G HEWITT	1 STADHAMPTON LANE, DRAYTON ST LEONARD, WALLINGFORD, OXFORDSHIRE, OX10 7AR	0865 89 166
IAN D JAMES	36 HARRISON CRESCENT, BEDWORTH, WARWICKSHIRE, CV12 8SL	0203 31 147
FRANCESCO L LAMA	14 GRANVILLE COURT, CHENEY LANE, OXFORD, OX3 0HE	0865 72 182
+ 2 OTHERS		

## SOUTH WEST MIDLANDS

BARRIE CLARK	SHANTAY,53 CLEEVE DRIVE, CLEEVE, AVON, BS19 4NP	0934 83: 55
KENNETH FRANCIS HUGHES	220 BROADLAND DRIVE,LAWRENCE WESTON, BRISTOL, AVON, BS11 0PN	0272 82: 90
DAVID NORMAN POLLICOTT	"ROSCOE",CARLTON STREET, CHELTENHAM, GLOUCESTER, GL52 6AQ	0242 58: 57
GRAHAM J STEWARD	40 CROWN LEA AVENUE, MALVERN; WORCESTERSHIRE, WR14 2DP	
+ 3 OTHERS		

## CHILTERNS

STANLEY J PHILLIPS	56 WINDMILL AVENUE, MARSHALSWICK, ST ALBANS, HERTFORDSHIRE, AL4 9TF	0727 86: 25
CHRISTOPHER ROWSON	5 HILLCREST ROAD, SHENLEY, RADLETT, HERTFORDSHIRE, WD7 9DB	0923 854 67
PAUL B SAUNDERS	24 TYLER ROAD, HAZLEMERE, HIGH WYCOMBE, BUCKINGHAMSHIRE, HP15 7NS	0494 81: 72
NICHOLAS P SUMMERS	36 DRAYTON ROAD, TILEHURST, READING, BERKSHIRE, RG3 2PH	0734 59: 69
GERALD J WHITE	11 CLARENDON ROAD, HIGH TOWN, LUTON, BEDFORDSHIRE, LU2 7PD	
ANDREW R WYARD	11 ROEBUCK ESTATE, BINFIELD, BERKSHIRE, RG12 5DG	0344 42: 90
+ 7 OTHERS		

## NORTH THAMES

ALAN V BAILEY	14 SHELLY GROVE, LOUGHTON, ESSEX, IG10 1BY	081 508 053
DAVID R CHAMBERS	134 FOREST LANE, FOREST GATE, LONDON, E7 9BB	081 555 900
IAN P KILGOUR	3 CARLTON COURT,65 STUDLEY GRANGE ROAD, HANWELL, LONDON, W7 2LU	081 840 666
DAVID.J. McCANN	24 THE GRANGE,GRANGE ROAD, CHISWICK, LONDON, W4 4DE	08199481 1
WILLIAM R RICHARDSON	5 NELSON GARDENS, BETHNAL GREEN, LONDON, E2 7AA	071 729 87
PETER J WALKER	24 BACONS DRIVE, CUFFLEY, POTTERS BAR, HERTFORDSHIRE, EN6 4DU	0707 873 8
+ 2 OTHERS		

## SOUTH THAMES

JOHN BINGHAM	MOORCROFT,SANDOWN AVENUE, ESHER, SURREY, KT10 9NT	0372 467 8
JOHN C HARRIS	2A LITTLE BIRCHES, SIDCUP, KENT, DA15 7LW	081 300 76
+2 OTHERS		

## THE SOUTH EAST

DARYL MUNCEY	36 REDFORD AVENUE, HORSHAM, WEST SUSSEX, RH12 2HJ	0403 218 7
JOHN SEAGER	"IŞIAN",WOODCHURCH ROAD, SHADOXHURST, ASHFORD, KENT, TN26 1LF	023373 2 3
LANGDON H SHAYLOR	21 THE CRESCENT, OLD TOWN, EASTBOURNE, SUSSEX, BN20 8PU	0323 370
SUMAELA YASSIN	FLAT ONE,8 BURLINGTON STREET, BRIGHTON, EAST SUSSEX, BN2 1AU	

## CENTRAL SOUTHERN ENGLAND

GORDON F A GILES	101 FOREST HILLS DR.TOWNHILL PARK, SOUTHAMPTON, HAMPSHIRE, SO2 2FZ	0703 558 2
LESLIE KINGSTON	"LYNTHORNE",GUILDFORD ROAD, NORMANDY, SURREY, GU3 2AR	0483 234 1
TERENCE J LEACH	145A WHITE HART LANE, PORTCHESTER, HAMPSHIRE, PO16 9AY	0705 386 5
JOHN MURPHY	65 HONEYSUCKLE LANE, CREEKMOOR, POOLE, DORSET, BH17 7YY	0202 603 3
VICTOR REINAR	14 HOYLAKE CLOSE, ROWNER, GOSPORT, HAMPSHIRE, PO13 0EX	0329 221 7
TREVOR TABERNER	51 CANFORD VIEW DRIVE, COLEHILL, WIMBORNE, DORSET, BH21 2UW	0202 880 8
+ 3 OTHERS		

## THE SOUTH WEST

JOHN CHARLES CAPSTICK	28 VIRGINIA ORCHARD, RUISHTON, TAUNTON,SOMERSET, TA3 5LP	0823 443 12
MICHAEL CURTIS	21 TRELISKE ROAD,ROSELAND GARDENS, REDRUTH, CORNWALL, TR15 10E	0209 219 11
+ 2 OTHERS		

## NORTHERN IRELAND

ROBERT W BEATTIE	2 BEECHILL AVENUE, BELFAST, NORTHERN IRELAND, BT8 4NS	0232 701 12
+ 1 OTHER		

FROM THE CHAIRMANS CHAIR  
T.Stevens 1992

Well here we are again once more with another news letter. As you will have read in the last issue, the AGM had some interesting things at it. Mike GODDARD had at the show some old and still boxed ROMOX cartridges. These are of no use today as they were supposed to be reprogramable. The idea was you took the cartridge down to your local computer dealer and selected a game, utility or whatever and had it loaded into it. ROMOX had the same cartridges for the Commodore 64 and the Apple II. However it did not take off. Well you might ask, what has this useless piece of junk got a use for? Well this little item can add as much as 8k extra ram to your little computer. How is this done? Well what we have is a board which has all the tracks used. Some cartridges only have the tracks which it requires to use, the others are left out. The Editor assembler is a prime example. Take it out and take a look, there are only tracks on one side of the board. Having all the lines allows us to access extra memory from the point >6000. This is where all your cartridges reside and run from in their own right. So if we have a cartridge with memory on board say 8k we can put our own program in there and run some pretty fancy stuff. Also if you have programs like Sound FX it will look at the address >6000 and take advantage of the extra memory. To be able to use the extra memory we have to have an operating program which will allow access. So what we do is make it an Editor Assembler with memory. So if we run the Editor Assembler then we can use the extra Ram. This cartridge how ever will not run without a full system as you really require disk access to load the program. Mike Goddard still has some of these available. He is selling them I believe for about £3.00 plus P&P. So be quick else the limited stocks will not last long and that will be the end of them. I do intend to uprate the memory by piggy backing the IC RAM chip with the same type and after I've had time to work out the address lines I will let you all know how to do it then we will have 16K!!!

So here are the plans, which I must give credit to Mark Playle for the idea and testing.

How To Make an 8K Super Module

**Requires**

1. A ROMOX Module.
2. An Editor/Assembler Module, or an E/A GROM chip (Available still from TI in the USA. Price about £4.00 + TAXES)
3. A 100 Ohm 0.25 Watt resistor.
4. A 16 Way D.I.L. socket.
5. A 28 Way D.I.L. socket.
6. An 8K \* 8 RAM chip (5565).
7. 16 short lengths of wire.
8. A soldering iron.
9. Some solder.
10. Copper braiding or solder sucker.
11. A pair of wire cutters/strippers or knife.

### Step 1

Open the ROMOX module. This is done by squeezing the back and front together. The top should pop off.

### Step 2

Take out the P.C.B. board. You should now have something like Figure 1. Desolder the EPROM chip, using the copper braiding or solder.

### Step 3

Replace the EPROM chip with a 28 way I.C. Socket.

N.B.: the end of the socket with a notch or marked pin 1 points towards the two diodes. You should now have something like Figure 2.

### Step 4

Cut the track marked in Figure 3 and solder a 100 Ohm resistor across the gap.

### Step 5

**N.B. THIS BIT IS TRICKY.** Sixteen holes have to be drilled into the board for the 16 way D.I.L. socket. **CARE MUST BE TAKEN** not to drill any of the tracks on the board.

Mark the bottom of the D.I.L. socket with an inkpad (or whatever) and place it as in Figure 4/Figure 5. Hold the board up to a strong light and **MAKE SURE** that none of the tracks on the bottom of the board will be cut when you drill the holes. **This is very important!!!**

Drill the holes just the right size to accept the socket legs.

### Step 6

The more time you spend on this step, the more chance your Super Module has of working the first time.

Place the D.I.L. socket in the board with the notch or pin 1 facing the edge connector. Glue it in place if you like.

Now the 16 pins on the socket have to be connected to the edge connector. N.B. All the connections are on the underside of the board.

Solder a wire onto each leg and run it to the edge connector as in Fig 6 and Fig 7.. i.e. Pin 1 on the socket to Pin 3 on the edge connector, Pin 10 on the socket to Pin 21 on the edge connector.

**TAKE YOUR TIME. DO NOT RUSH!**

### Step 7

Place your RAM chip into its socket. **MAKE SURE** pin 1 or the notch on the chip points towards the two diodes. **IF IT DOESN'T. \*BANG\*.**

### Step 8

Open your Editor/Assembler Module by undoing the screw on the bottom and unclipping the sides.

Remove the board and desolder the GROM chip. Place the chip in the 16 way socket on your ROMOX board. **NOTE PIN 1** or the **NOTCH** should face the edge connector. **IF NOT. \*BANG\*.**

### Step 9

**MAKE SURE YOU HAVE FOLLOWED STEPS 1-8 CORRECTLY.**

### Step 10

Put the ROMOX board back in its case and slot it in your console. It

should act as an Editor/Assembler. If not, switch off the console, remove the module, and make sure you followed steps 1-8 correctly. Look for bad joints and solder bridges.

The 8K of RAM can be used by quite a lot of software. i.e. 1) Funnelweb, 2) Some versions of Quickcopy, 3) Super Forth for the Super Space Module, and 4) Batch-It!!

In fact, you should be able to use any software for the 8K Super Space Module by DataBiotics.

The RAM in your Super Space Module occupies address >6000 to >7FFF. You can treat this as normal C.P.U. addressable memory, for data or executable instructions. N.B.: to load a program into this space you must use an AORG >6000 instruction in your assembly program.

I hope that the above is clear enough for you to follow, and PLEASE READ at least twice so as not to make any mistakes.

### THE DRAWINGS

Figure 1.

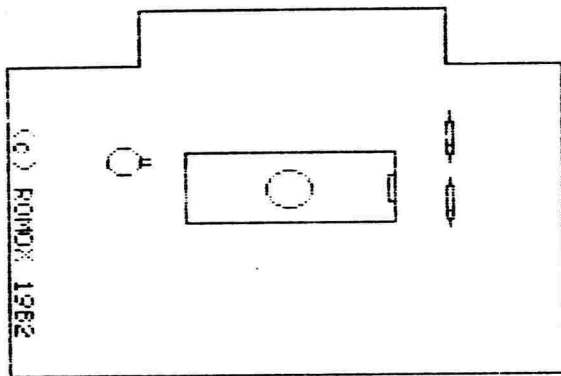


Figure 2.

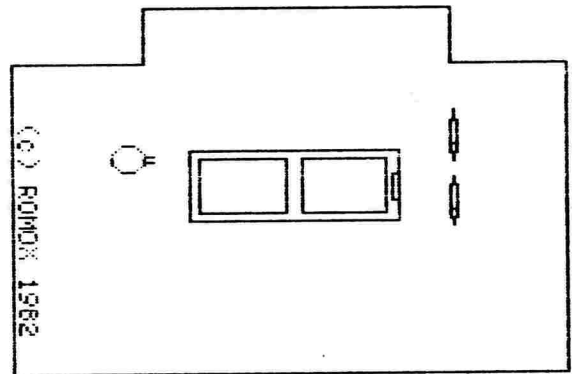


Figure 3.

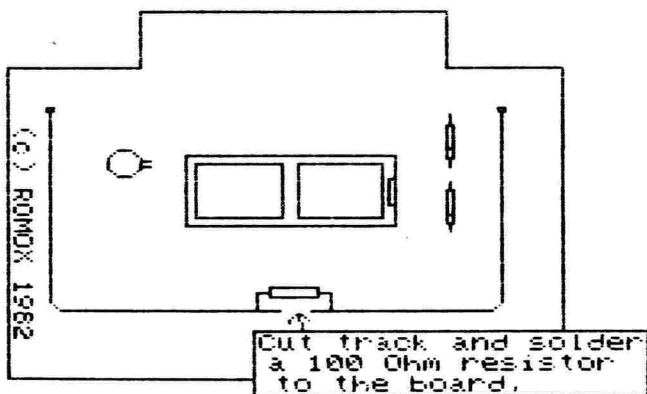
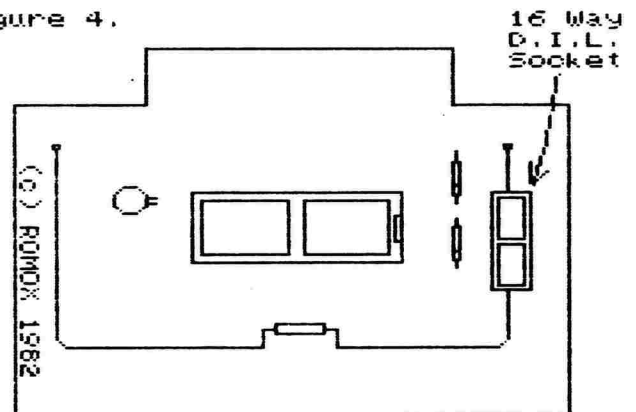


Figure 4.





+ = Hole to Drill SEE TEXT

Figure 5.

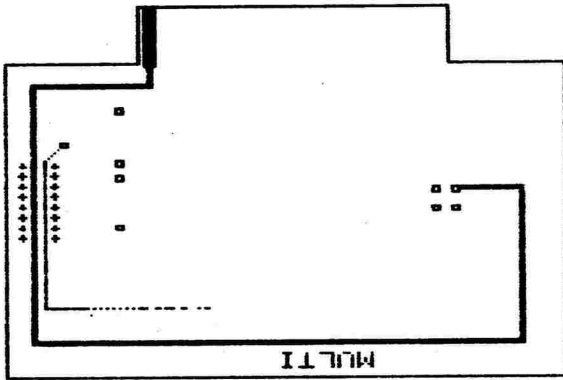


Figure 6.

GROM  
to  
Edge Connector

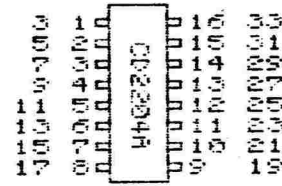
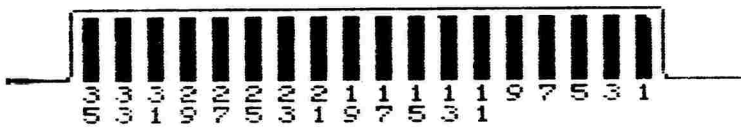
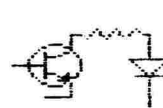


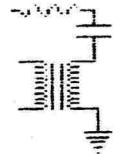
Figure 7.  
Edge Connector  
of ROMOX underside



TI USER GROUP



1992



T. STEVENS

Well that's it for this month I hope you don't get to many headaches with this little lot. Just for the record the drawings were done on Joy Paint Pal, the 'User Group' and printing on TIARTIST Plus.

Next quarter I will be announcing the winner of the COMPETITION for the one liner I set last Quarter. If you missed it then see issue 37 page 3. So there is still time to win some software that will work on a unexpanded Extended Basic System.



# NEWS AND REVIEWS

From the man with

## THE BLUES.



BY

Richard  
Twynning



Dear TI'ers,

Well, I hope you like the new title. I copied the picture by tracing one of my Blues Brothers posters, and a Blues Brothers T-shirt onto a sheet of acetate. I then sellotaped the sheet over the monitor and traced it onto the screen using GRAPHX!

For those who are musically minded, you will realize that the notes represent the start of an actual piece of music called, "The Ninety Nine Year Blues", by Julius Daniels & Torrence, 1927! It came from a book of music called "Roots Of The Blues Vol.2" available from Southern Music Publishing Co. Ltd. 78 Newman St. London W1.

Hopefully this will be the first of many articles and I'll try not to miss an issue!! I must apologize first for not having put the minutes of the meeting in the last newsletter as I didn't have time to replay the tape and do a report.

Well, here goes:

Item 1 on the agenda were the apologies from the people who could not attend the A.G.M. These were Jim Ballinger who had just been in hospital, and also Trevor Taberner. I hope you are both feeling much better.

Item 2 were the minutes from the last meeting.

Item 3 were the reports from the officers of the group.

It has been a relatively quiet year in the TI community and several people have given up their positions. These were Peter Walker who gave up as membership secretary. Jim Ballinger who gave up as General Secretary and Ed Shaw gave up as module librarian.

The Vice Chairman (Mark Wills) was absent even though he said he would attend, and I noticed that he had no articles in the last newsletter. Are you O.K. Mark???? You missed the GIF image demo on the Amiga and the 4A. I've also scanned you the book from the War of the Worlds for the TI demo you were planning at Maidenhead.

The first officers report was from our publicity officer Phil Trotter who is continuing to place adverts in micromart, and as far as I know we are still listed in the user group list in Personal Computer World, even though I couldn't find the list at all in May, June, or July issues.

The next report was from the Treasurer, Alan Rutherford. The membership is fairly steady which means there's a steady income into the group's funds. The magazine has been slightly larger lately, but the funds will support it.

Another cost to the group is the hire of halls for meetings etc., but the cost for last year's AGM was low due to the charge of an entrance fee.

There is no reason at the moment to raise the subscription fee.

Next was the Editors report from Alan Bailey which concluded that there is a lack of material for publication in the magazine and he's scraping the barrel for articles. I have also heard a contradicting report that there's enough to keep it going for a considerable time yet so I'm unsure of exactly what to say on the subject.

The subject of the quality of copy was also discussed, and the problems that were encountered in the Spring issue of TI\*MES where whole sections of text had to be gone over by hand. Up to the date of the AGM Alan had only received one submitted article for the Summer issue of TI\*MES.

Fortunately though, this enabled everyone who had not yet submitted an article to improve their copy, and you will notice that Alan made a comment on the much improved copy submitted for the summer edition.

Next was the hardware report from Mike Goddard.

It has also been a quiet year as far as hardware is concerned, but at the AGM as you will have read in Trevor Stevens' article, there was a member from the Dutch TI User Group who was selling plans for a new 80-column card. You will have also read that it may be available in kit form through our group.

I will discuss this again later on in the article.

The next, and most worrying report was from Stephen Shaw on the state of the disk library. Everyone, including myself, is guilty of not using the library, and it is becoming very difficult for Stephen to purchase new software due to a severe lack of funds. By the time you are reading this I will have hopefully ordered another couple of disks, and I may have even entered a couple of my own disks into the library. If I do, I will include it in my Winter article (Oh God!! Who mentioned Winter!!).

Stephen went on to say that the shareware concept is failing, and programmers are trying to publish programs themselves. One such program is YAPP by Alexander Hulpke which I don't think ever made an appearance in our library. I obtained my early copies from Gordon Pitt who was in contact with Alexander, but he hadn't heard anything from him for a long time. I discovered myself that it had been published through Asgard Software when I tried ordering Press and Batch-It.

In order to allow the support of the disk library, it was suggested that a donation could be made when renewing membership. Those who have recently renewed their membership will have noticed an addition to the renewal form that reminds the member that it is possible to make an additional donation to the group to support the disk library, and to allow the purchase of new software.

Next came a report from Nicky Goddard on the cassette library which is much the same as the disk library. There was one customer who had stopped ordering. However, Trevor Stevens has informed me that he's been contacted by a cassette user who's got 32K and is looking for some software.

Edward Shaws final module library report closed the reports from the officers. The state of the module library isn't quite as bad as the other two libraries, but there has been less activity this year.

Receipts were £495.50 and costs were £375.50

Next on the A.G.M. agenda was the matter of B.A.C.C. membership. This is the British Association of Computer Clubs and provides insurance on hardware which covers it at the show, and in transit, and also covers public liability. It was decided that membership should be applied for, and this was left for the treasurer to attend to.

The next item to be discussed referred to an extra meeting that was held to add extra points to the constitution. These points were concerned with certain comments made in TI\*MES articles.

In future, any complaint will need the support of at least two people on the committee, and the committee will not consider any complaint valid if it concerns any item of software. However, it is not the intention of the committee to stop free speech.

As you will have read in Alan Bailey's Editorial comment in the summer issue of TI\*MES, it was decided to give the editor a full expansion system to allow the easy modification of submitted articles. If an editor was ever elected who already owned an expanded system, then this expansion system would go to another member who needed it for whatever reason.

Alan mentioned also that he wanted the pinouts of the Miniwriter III+ module and I suppose he's talking about one of the word processing modules for the console only which has its own built in printer port. I can remember Parco selling one of those, so I hope someone can help Alan with the information.

Well, that's it for the A.G.M. report. It's now time for some news and reviews, so Let's Boogie.

Myarc produced a leaflet on the GENEVE which listed the dealers who stocked the machine. Gary Smith and I decided to each write to half of the companies listed. Gary received a

reply from Disk Only Software who had FORTRAN 99/9640 for sale.

Gary had also written to Beery Miller however, and Beery was offering a discount on FORTRAN and so he ordered it from him, along with the latest version of Windows 9640.

I produced my letters by using the mailing list option of the text formatter and printed the letter at the top of the sheet, and printed the address at the bottom of the sheet. I then folded the sheet and sellotaped up the sides. It saved alot on envelopes.

I printed the addresses in enlarged mode which I initialized by using Control U, and then generating the necessary control codes by using SHIFT and FCTN key combinations. I also recently used this ability to change colour in the middle of a piece of text to highlight certain areas. Gary has a friend who owns the 9-pin version of the LC-200 colour printer, and he also owns an IBM. He can't use any feature similar to CTRL/U on his current word processor, and so he can't change font or colour. The word processor he is using is Word Perfect which costs around £300!!!! It just shows what an overpriced rip off most I.B.M. software is.

Well, anyway, back to the subject!!!

I have had two replies from the list so far. The first reply was from Great Lakes Software. They no longer offer any hardware, but do have some new software titles.

Their range of titles includes the following:

Certificate 99 This is now up to version 2. \$9.99 + \$1.00 Shipping.

They also have two companion disks for certificate 99.

Joy Paint 99 Probably the most famous of all Great Lakes titles which offers a large drawing area that can be windowed. The last time I looked at this program however, it would not run on the GENEVE because they were making direct 4A calls instead of following TI's specifications. I will have to check up if this problem has been sorted out, and if not, see if there's any chance of them doing a minor re-write, because it's an excellent package. \$24.95 + \$1.00 Shipping.

Joy Paint 99 also apparently has its very own user group, where users can exchange ideas and clip art!!

Joy Paint's Pal which is the companion program for Joy Paint. This allows the transfer of Artist \_P or 25 sector Draw 'n' Plot files, to and from Joy Paint format. \$9.99 + \$1.00 Shipping

Clip Art Many more graphic objects for use with Joy Paint. All clip art disks cost \$9.99 + \$1.00 Shipping.

Extended Business Graphs II Bar Graphs, Pie charts, and Linear charts. \$9.99 + \$1.00 Shipping.

Banner 99 Create huge banners in minutes with ANY PRINTER. Generates smooth letters, not just block characters. \$9.95 + \$1.00 Shipping.

Great Lakes Software were really very helpful, and have even offered to do a discount of 50% for our group for any five items or more ordered.

The second item to reach me has made it just in time to be included in this article and was from TENEX Computer Express. I'm afraid that this contains bad news, but this is outweighed by the good news. The bad news is that TENEX stopped dealing with TI products about two years ago, but the good news is that they provided me with the names, and U.S. phone numbers of four new companies I've never even heard of, so perhaps our American members can tell us a bit more about these.

The companies are:

Joy Electronics 1-800 527-7438

Braatz 1-414 731-3478

Competition Computers 1-414 672-1600

Northwest Family 1-206 336-3586

Still on the subject of publicity, recently, on Radio Nottingham someone came on and asked if anyone could help him find a users manual for a 4A. I phoned and got his number, and he came around to look at the computer. When he arrived he said someone else had phoned who had owned a 4A for years, but had lost contact with the group. Later, the second person phoned, and he also came and saw the GENEVE. Unfortunately, he is a bit busy at the moment to join the group but I've still got his number, and he lives in Mansfield, and he's lent me the books, Best of 99'er, Programs for the TI by Steve Davis, and Smart Programming Guide for Sprites.

I've also managed to contact someone else via VAX mail who did have a TI a few years ago, but he sold it, and now wants to buy another. I sent him some info, but unfortunately haven't heard anything from him yet, unless he's contacted Alasdair and joined without letting me know. I will have to phone the VAX and check my mail to see if he has left me anything.

Well, I've got that much information to deal with, that I don't really know where to start with it. I will try and cover it bit by bit, and hope that I don't leave anything out.

In my previous article, I mentioned that I had converted a PC sound file over to Sound FX, which was the 1940 recording of The Midnight Special. I've been experimenting with it, and have got it to a listenable, but still distorted form. The main problem with this is that the dood who did the digitizing, saved the files out as .VOC, and not .SND which is the one that Sound FX is designed to use. I asked him if he could convert them into .SND files, but something was screwing up on his PC (what's new!) and he couldn't do it.

But, I've got a friend who's just come home from Leeds Polytechnic, and he's just bought a PC with a Sound Blaster card, and his dad's got a 386 PC that's got an internal 2400 bps FAX modem card. I can take whatever tapes I want digitized around to his place, and do the digitizing, and when I get back, I can

just phone him and download the files. He was going to bring his PC around here and do it, but, IBM's only have a 15 way D plug on their serial ports, so I'd need to make a new cable!!!

This morning (using my new toy!) I have also transferred another sound file. This is HAL 9000.

"Good Afternoon, Gentlemen. I am a HAL 9000 Computer. I first became operational at the H A L Labs, in Irvana, Illinois, on the 12th of January 1992. My instructor was Mr. Langley, and he taught me to sing a song. If you like, I could sing it for you. It's called Daisy." And then the file runs out!!!

I'll try and get this re-digitized and hopefully get the actual song as well. I am hoping for the same quality of the files that we have already got, but I think that they might have been created on an Amiga or a Mac, and not a PC.

Well, I've got you all wondering what my new toy is now!!! In fact, it's an Amiga 600, which is now being used as a transfer device, and is connected to port 2 of my Cray 2.

The computer industry is really in a sad state at the moment. Trevor Stevens has had an Amiga 500 Plus for a while, and I tried to get one of those first. I tried around the usual computer shops first, with no luck, and then went to the Mansfield branch of Dixons. The assistant said they had got one 500 Plus left, but it was Ex-demonstration. I decided to have it, but when I got it home my problems started. My Workbench, and LEMMING's disks were missing, so I phoned, and my dad went back down to get the disks while I put a plug on and tested it out. It was totally dead. I phoned, and said it was useless, but told them I would test the power supply with a voltmeter. I found that nothing at all came out of the power supply. They said they'd got one there that they would let my dad bring back. I tried it and it came on. I decided I'd try out Lemmings first and turned on the telly, and put the disk in the drive. For a split second as the title screen came up, it looked like a picture of a hand holding a disk, which meant that this was an Amiga 500, and not a 500 Plus. Lemmings loaded O.K., so I thought that before I went any further, I'd make sure that I had got a Plus, and had not been ripped off with a 500. I pressed the eject button on the drive, and nothing happened!!! The damn drive had eaten my Lemmings disk!!!! I phoned up straight away and said I'd take the whole pile of crap back, and have a 600 instead!!!!

After a week of using the Amiga I noticed some very strange problems. Some of the disks that Trevor copied for me would not work on mine. And some of the disks I copied on mine would not work on mine, but would work on Trevor's!!! Then one of my friends from Poly said he had just got an Amiga 600, so he came round for a mass copying session. We copied about four disks, but they wouldn't work when we tried them. We went straight round to his place, and put them in his Amiga, and they ran first time! Trevor said it's my Denise chip that's dodgy. Mark Wills told me that the Amiga has a sad encoding method, and it looks like the Amiga could encode a disk when it copied it out, but would not decode it again. But if that's true, how come the originals worked!!! I know exactly what was wrong with it. Commodore!!! It's a crime that they can still even be selling the Commodore 64.

It looks like it was a good idea to get a 3 year warranty. I don't think it can be relied on as much as the 4A or the GENEVE.

I took it back to Dixons and they replaced it.

Trevor is also on his second 500 Plus because that had a few weird problems.

Very Poor!!!! I had none of that trouble with my 4A, and that's been demonstrating things for 10 years!!! Doing simple jobs are a nightmare on the Amiga. If you use the Amiga shell and attempt using DOS commands then you have to waste more time swapping disks. If you type dir, then it asks you to replace the Workbench disk so that it can load the dir program!!! The same goes for copy, delete etc. The Amiga has got impressive sound and slightly flash graphics, but I can't imagine how anybody could rely on it as a serious computer. The keyboard doesn't feel half as good as the 4A and they only supply a programming language called ARex, but they don't give you a manual!

Computing really has progressed a long way!

Trevor did get Amiga BASIC, but only because he asked for it! And apparently, Commodore, don't even repair Amiga's themselves, so that's why I've opted for a 3 year warranty!

The next strange piece of news came from Trevor Stevens who said that the Amiga Telecoms program JR-Comm listed Paul Charlton in its credits. I've not managed to find this yet, but for those who don't know, Paul Charlton is the man who wrote FastTerm, Myarc DOS!, and Picture Transfer, etc.

Having Amiga's available to the group means that we're now in the main stream at last, and in the couple of weeks that I've had it I've been transferring files over to the GENEVE at 9600 bps and have generated over fifteen disks of GIF images, which include articles from Trevor Stevens' old issues of 99'er magazine. These include pictures of Bill Cosby advertising the 4A, and also pictures from the June 1983 Chicago Consumer Electronics Show. There is also a silly product called the joystick stabilizer which was a bit of wood which fitted over the joystick and held it down! If there's enough room, then you will see a picture of this produced with YAPP.

As I mentioned in the last article, Oliver Timm from Germany said he would send me the information on how to modify the hard disk controller, and also Myarc Disk Manager 5. He did send me a disk, but I couldn't find any information on the modifications, so I don't know if it's actually on the disk or not, since the disk is not readable. I will have to put my 4A back on line so that I can check if the disk is in 80 tracks or not, since I've not tested 80 tracks out on the GENEVE yet, since I am unsure if this version of MDOS (0.97H) will support floppies from the hard disk controller. The Wrexham workshop will give me the excuse to take the computer apart so that I can put the 4A back on-line to see if the disk does also contain the information on upgrading the controller. I have also written to Barry Boone, however, but have still not heard anything, so I might try and obtain a phone number for him, and beg him for some information.

Still on the subject of the hard disk controller, Gary Smith has received the complete data manual from Beery Miller, for the chip that Louigi has used in the Hard and Floppy Disk Controller. I was amazed when Gary told me what the chip would do. It will support either 4 hard disks, or 3 hard disks, and a



tape streamer, which is what Louigi opted for. It will also support 65536 bytes, yes that's 64K, PER SECTOR!!!!!!! And that's not bad for a drive which might have half a million sectors! Do that on an Amiga! If you're still not impressed, then how does 16Megabytes of disk cache sound!!!!

Gary has now started work placement at National Semiconductor in Swindon, and therefore has the opportunity to bump into the Marlborough man (alias Richard Sierakowski).

If Richard has a look through the data manual (which actually includes a complete circuit diagram of a suggested controller), then he will probably be able to work out what modifications are necessary to turn the Write Precompensation line, into another head select line.

In my last article you may remember that I mentioned that Gary Smith was thinking about designing a new Hard Disk Controller around the same chip as Louigi. The article said it would lack support for floppies, but the news from Gary is that it will handle floppies, and will do any format you could ever want, even allowing IBM disks to be accessed.

Staying on the subject of hardware projects, and returning to the 80-column card that Trevor Stevens and Derek Haywood are building. The card fits into the expansion box and requires a couple of wires that extend all the way to the VDP socket in the console.

This allows the card to detect when video calls are being made. Somehow, the actual video commands are directed through the P.E.B. and to the C.R.U. address at which the card is located.

The actual analog R.G.B. video output comes from the back of the card. The sound, however still needs to come from the normal monitor socket on the back of the console. The only line that TI provided for sound in the box is Audio In, which allows the input of sound from a card such as the Video Controller Card. This allowed the input of signals from a Philips Laser Disk Player, which then had TI graphics super-imposed over the top using the TMS9918, or 9929's built-in Gen-Lock feature. That's over half a decade before the Amiga was even thought of!

The new 80-column card will support either a YAMAHA 9938, or 9958. The 9938 is the one which is used in the GENEVE, but I would recommend anyone thinking of building the card to go for the 9958. Louigi used the 9938 in the GENEVE because it's got hardware mouse and light pen support built in, but this is not really needed on the 4A since you will still need a mouse, and an Asgard mouse is currently available for \$39.95. The Asgard mouse is a serial device that connects through any RS232 port, and will therefore also run on the GENEVE. This is what Beery Miller's done, because he's replaced the 9938 on his GENEVE with a 9958.

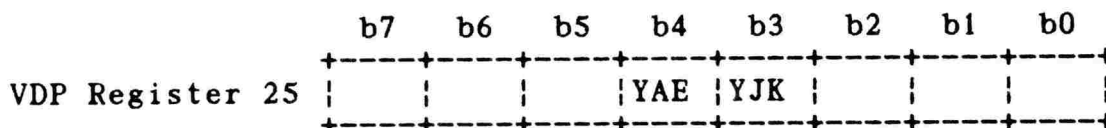
The reason the 9958 lacks mouse and light pen support is because it's been removed, along with the NTSC composite output, to make way for YJK Display. What's that I hear you cry. Well, this is where I return to the Amiga. Some people may know that the Amiga has a video mode called HAM (Hold And Modify) which allows the simultaneous display of 4096 colours.

HAM works by starting with an initial pixel colour for the top left of the screen. The next two bits of the screen memory are then tested. If they are 00, then the next pixel will retain the colour of the pixel just displayed, and the system will check the next two bits. If they are 01, then the colour of the pixel will be the same as the current one, but with the Red content of the colour modified by the value in the next four bits. If they are 10, then the Green content is modified. If they are 11, then the blue content is modified. Using this method, an image with much more colour content can be displayed, but the use of RAM is greatly minimized.

YJK Display on the 9958, or V9990 works in a similar way, but YJK allows 19268 colours to be displayed at a resolution of 512 \* 424, and that definitely beats the Amiga!

I've not managed to sort out the information on the YJK display yet, but here it is anyway. This is taken from the YAMAHA V9958 MSX-Video Technical Data Book. Yamaha catalog number: 249958Y

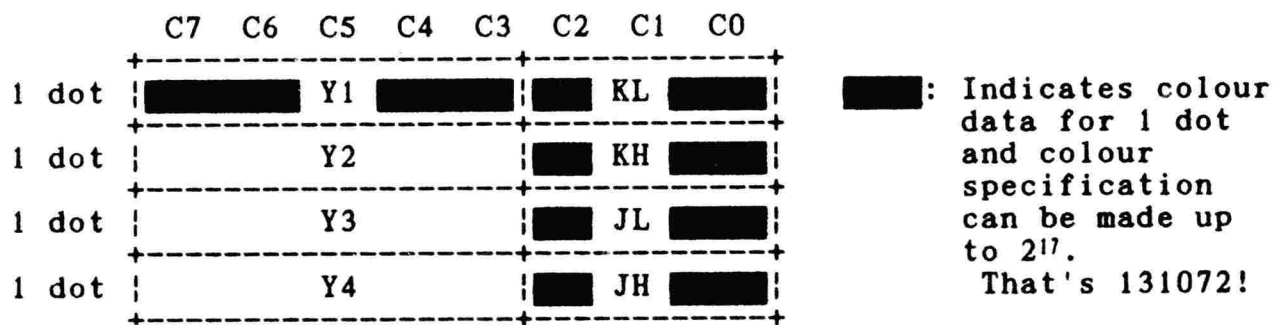
Page 8, Section 5-1-4 YJK-Type Data Display Function.



YJK 0: Handles the data on VRAM as RGB type data. (Initial value)  
 (Example: G7 mode = 3,3 and 2 bits each)  
 Displayed colours of the sprite are the same as conventional type.

YJK 1: Handles the data on VRAM as YJK type data, converts them to RGB signals (5 bits each) and outputs them through RGB terminals as analog signals.  
 The colour palette is used to display colors of the sprite in G7 mode.

YAE 0: Without attributes



YJK type data is categorized based on the data on 4 continuous dots as follows.

Y1 . KL . KH . JL . JH : colour data for the 1st dot  
 Y2 . KL . KH . JL . JH : colour data for the 2nd dot

Y3 . KL . KH . JL . JH : colour data for the 3rd dot  
 Y4 . KL . KH . JL . JH : colour data for the 4th dot

YAE 1: With attributes

	C7	C6	C5	C4	C3	C2	C1	C0	
1 dot	■		Y1	■	A	■	KL	■	A : Attribute
1 dot	Y2			A	■	KH	■		
1 dot	Y3			A	■	JL	■		
1 dot	Y4			A	■	JH	■		

When A= 0

Just like when YAE="0", ■ indicates color data for 1 dot and colour specification can be made up to 2<sup>16</sup>. ( The "A" bit is ignored.)

When A= 1

Y1, Y2, Y3, and Y4 become colour codes respectively and they are output as RGB signals through the colour palette. (16 colours).  
 The KL, KH, JL and JH data are ignored then.

\* Combination of YJK and YAE data

YJK	YAE	
0	0	Via the conventional colour palette
	1	Via the conventional colour palette
1	0	Via the YJK to RGB conversion table
	1	A=0 : Via the YJK to RGB conversion table A=1 : Via the colour palette

\* The formulas for YJK to RGB conversion are as follows:

1) From YJK to RGB

$$R = Y + J$$

$$G = Y + K$$

$$B = \frac{5}{4}Y - \frac{J}{2} - \frac{K}{4}$$

## 2) From RGB to YJK

$$Y = \frac{B}{2} + \frac{R}{4} + \frac{G}{8}$$

$$J = R - Y$$

$$K = G - Y$$

You will also notice from the above data, that the 9958 will support standard colour data, and YJK data on the same screen, which HAM on the Amiga definitely will not do!!! Because of the way it is encoded, HAM, and YJK data takes slightly longer to display, which is why there aren't any HAM games on the Amiga, and even the update on some art packages is painful, and they aren't half as good as having 256 colours on YAPP! The advantage of the 9958 in being able to mix YJK, and standard colour data is that you could have part of the screen providing a large number of colours, such as the scenery in a game, which would be in YJK Display, and the other areas of the screen where faster access is required could be produced by using normal colour palette data.

Well, I hope this has been informative for those who are building an 80-column card, or has inspired others to possibly build a card. Don't forget, Gary and I would still like to hear from anybody who would be interested in a super-hi-res card, possibly built around the Texas Instruments TMS34020, or the even cheaper TMS34010. Please note, that this is intended purely for experimental work such as experimentation with Fractals, or for a CAD system that I'm considering, or for a new breed of games that leave the Amiga standing. Our card would not be accessible as a normal 80-column card to 80-column software such as YAPP, or 80-col Funnelweb or Multiplan etc.

It would be totally separate, and would work on a TI or a GENEVE with or without 80-columns, and it would even work on a console with a small adapter which would fit into the side of the console. The card would then fit into the adapter and would be usable from TI-BASIC with CALL's that we would provide on the EPROM such as VMODE, POINT, LINE, FILL, MOVE, COPY, PALETTE etc.

As with the 80-column card, it would require an RGB monitor, but if possible we would design it so that you would only require one monitor. The output from the 80-column system or GENEVE would be fed into the 34010, or 34020, and the 340's graphics would be super-imposed over the 9938, or 9958's graphics. The combined signal would then be re-encoded as Analog RGB by the 340, and would pass to the monitor.

A few people may know that I've got a Philips CM8833 monitor for my GENEVE, and it's damn good. It's got a composite video input which is connected to the video in the lounge!

(I've got a 38ft cable under the floor all the way to my room!)

It's also got a standard SCART input which allows Analog RGB input, which is what I'm using for the GENEVE. There's a couple of buttons on the front which allows switching between the video and the GENEVE, and the changing from colour to Green Screen. This is damn good for word processing. I change over

to green and turn the brightness right down. It helps to save the screen when typing very long documents such as the odd 230 page manual etc.!!!

Gary's got a CM8833 II, which is updated, but it uses a 9-pin D connector for Analog RGB, which I suppose is a bit easier to wire up than a scart plug, if a bit non-standard, but the thing I don't like about it is that the on/off switch has been moved around the back. Mine's on the bottom right at the front.

The 8833 II however, has got a trendy curved casing that matches the contour of the tube. I can recommend either of the 8833's for either GENEVE or 80-column card. I've never had a single problem with it, and they're probably the best resolution you can get for the price.

Well, back to the subject!!

The reason I'm so interested in the new 80-column card is that I've had a V9958 and 128K VDP RAM sat on my shelf for the past year, waiting for something to be done with it. I don't want to replace my 9938 in the GENEVE like Beery Miller, because that would make all the normal GENEVE software incompatible because it uses the Myarc Mouse which is directly fed into the 9938. I am hoping to slightly modify the 80-column card so that it will work independently and will ignore the 9938 commands. I will then be able to use it as a separate video system which will allow me to hopefully create a super-enhanced art package, possibly in FORTRAN 4A which would allow the loading and saving of Amiga HAM files, plus the more Highly defined YJK format!

If I've managed to sort something out, I will report on it in the next issue.

Well, what's next in the article!

I suppose, the next part is both news, and a review, and for my own package! If you dig out TI\*MES Winter 1991/92 and turn to page 15, you will find a very early screen dump of my Extended BASIC window type program called WORKSPACE. I can now announce that this is now 99.9% complete, I've only got to test my pull down menu Task Definition Language interpreter, and then that should be about it.

In figure 1 you can see a screen dump of the initial screen. The top row of icons are from left to right: Calculator, Disk catalog, Delete File (Trash), Activate printer, File Finder, Clock On/Off, Alter Clock Size, Return to XB, Text Editor, On Line Help On/Off, Screen Colour Modification, Previous Sub-directory, Next Subdirectory, Load disk manager, Dump current screen to a MYART file, Display information about author, and system, disable use of ICONDATA file, Copy File, Return to Title Screen, Enter new path from keyboard, Display graph showing free storage on all devices. I think the icon on the right hand side speaks for itself!

If you select a standard device from the icons at the bottom of the screen, you will get a window that shows the data contained in that device. Figure 2 shows the window of my Horizon RAM disk. The system displays files as sheets of paper, directories as folders, and EMULATE files as a picture of a disk.

You can see an example of EMULATE files displayed in figure 4. For this type of directory, the system makes a guess

of what type a file may be. Any Display/Variable files will be displayed on the screen, any TI-Artist images will be displayed, and any XB programs in INT/VAR 254 format will be recognized. At this time, I'm currently working on a routine that will recognize Low, and High-res MYART images.

The other kind of directory, is one that contains an ICONDATA file. If this is found on a directory, then all of the information for that directory's icons will come from the ICONDATA file. This means that any files that are contained in that directory, but are not included in the ICONDATA file will remain hidden from the user, but you will have noticed that there is an icon on the top row that allows the disablement of the ICONDATA file, and the system will read the files from the actual directory, and make a guess at the file formats. The advantages with using ICONDATA files are that they are much more friendly, because you can have real pictures instead of just dull symbols, and the information on the exact type of each file is contained in the ICONDATA file. This means that after a file is selected, everything is totally automated, and after a short delay, the requested program appears on your screen, whether its an XB program, or an Option 5 file, or an XB Batch File. Using the alternative method, if you select a PROGRAM file, you have to then tell the system whether the file needs an XB loader, or an EA loader.

An example of an ICONDATA file directory is shown in figure 3. As you can see, this is a little more advanced than the picture I included in my Winter article.

The feature I'm most proud of is the calculator. You can see this in figure 5. You may have already got my other versions of this already. My first calculator which was inspired by the ones found on systems such as the Macintosh, and GEM, first appeared in PARCO's 99/4A magazine Volume 2, Issue 2 in 1985. This was a very poor attempt at the calculators I'd seen previously. All I did was have a load of sprites slapped on the screen at magnification 1, and have an arrow which was moved with the joystick to select each button. I was just relieved to get the operation of the calculator working, and didn't spend any time on actually getting the thing to look like a calculator.

I wasn't satisfied with it, so I set about writing version 2 which did look a bit more like a calculator. This version appeared on an issue of 4FRONT (New Day Computing) (either issue 3 or 4). It was still quite messy inside however, and had a couple of problems, so I've done a total re-write, and I'm now up to version 3. The first point of my re-write was to tart up the display even more and make it displayable on the XHI screen, and the second, and probably most important was to save on memory. The first version that appeared in PARCO magazine took up page 28, and over half of page 29 (How's that for a filing system! Almost 8 years ago and I can still find it! That proves my room doesn't need tidying!) I think this time though, I've totally out-done myself as you can see in figure 6. This shows a complete listing of version 3. This not only includes the internal crap of the calculator, but also the displaying of the graphics as well! The CALL DRAW routine is used to display the calculator, and is the same routine that displays the device

icons, the FWEB icon, and symbols for warning messages etc. The actual character definitions in the CALL DRAW routine are done by calling the FONT routine which also loads all the different styles of text. If you're wondering, the smallest font is actually the 6 \* 4 font from Missing Link!, and the main font that you can see on the title bar is actually ripped off from the Apple Mac! The part that I'm most pleased with is how well the actual LCD display turned out. None of the calculators I've seen on machines such as the Mac, and Amiga, or on MessySoft Windows have true digit displays. They just use standard computer text! And they don't have half the number of functions as mine.

The calculator took me two days to re-write from start to finish, and you'd think that if a company wanted a calculator to look more familiar to the user they'd spend some time on designing a simulated LCD font. You'd think that will all the money available to companies such as Apple, Microsoft, and Commodore, they could employ some decent people to add a decent selection of commands. I wouldn't mind a job if anyone out there needs a bit of quality putting back into computing.

There might even be a Calculator 4, but Sigourney Weaver definitely won't be in it. (Sorry, couldn't resist the topical joke!) This will be totally scientific and will include every function I can implement.

The thing that pleases me is that the best on-screen calculator is now available, and it's running on the best system available. Texas Instruments. Shows what you can do with a decent BASIC interpreter. I think our systems are much more user friendly than the MAC. There's nothing good about the average IBM today. We've had our TI's that long, that we can almost operate them by telepathy. I could produce diagrams on Picasso more efficiently, and quicker than the average PC owner on my course at Poly. The problem today is that half of them haven't grown up with this brain extension we call a 4A, and their heart really isn't in it.

They constantly read the usual b\*\*\*\*\*t adverts from all these two-bit money grabbing companies who's managing directors can only spell PROFIT and IBM, and they swap software at college, and they end up with so many programs that all do the same thing, and they don't have a clue which one they should really be using. By the time they've worked out all the pull down menu's and control Z, & Y's etc. I'm probably printing out a couple of things with Picasso!!!

By the way, I've got my exam results, and I've passed everything, with an A in Computer Systems, PASCAL Programming, and Software Engineering. The turd that said 16-bit came out in 1982 gave me a D for his subject, but that might be because on my exam paper, I said that the IBM PC was the worst designed machine in history. TI Rules O.K.

It appears that I've got carried away again!

If anyone is so desperate that they want to upgrade to something a bit faster than a 4A and can't find a GENEVE, then you could possibly upgrade to a Sun Microsystems workstation.

If you use a current Sun Workstation today, you will find that it will support Workspaces, like our good old 9900, but Sun never quite worked out how to handle parameter passing without overlapping the workspaces. That's where the TI connection comes

in. A couple of issues ago of Personal Computer World, there was a report that TI were helping Sun to sort out their problems and were working on a machine codenamed "Tsunami".

Last week, I got the September issue of Personal Computer World and read the following information: (on page 460)

#### SPARCstation Ten

Sun has finally pulled the wraps off the SPARCstation Ten, fooling everyone who thought it was going to be the SPARCstation Three. The machines are impressive to say the least, mainly due to their expandability.

To best see this, think of the SPARCstation Ten as a chassis, with two MBUS slots and four SBUS slots, all the I/O, and space for 512Mb of RAM (using Sun's fancy new 144-bit wide 64Mb SIMM). Into the MBUS slots you can put processors. The bottom-end machine comes with one TI SUPERsparc, Aka Viking at 36MHz, going up to an announced machine with four SUPERsparc chips at 40 or 50MHz. The Viking chip is a super-scalar version of the SPARC: it attempts to run more than one instruction per cycle, currently up to three per cycle. With 36K of on-chip cache it's a big chip. TI also has a 1Mb external cache for it which is available on some of the new SPARCstation Tens.

Word is that Sun is working with TI on a high-powered low-end machine for next year, called Tsunami, which is aimed at giving the PC market a shake-up.

The target is said to be a sub-\$5000 machine ready to run, unlike the overtly crippled sub-\$5000 units that have been announced in a blaze of glory by some workstation manufacturers and which, upon examination, lack minor things like a monitor or a keyboard or a hard disk or memory!

Well, what's the next subject! I recently attended the Wrexham workshop which was not very well supported. There were only four of us! Mike Goddard, Derek Haywood, Mark Wills, and Me! Mark had not gotten around to re-subscribing yet, so he didn't have Summer 92 TI\*MES, and so the only reason he knew it was happening was because I sent him a letter telling him that I was doing a demo of my windowing system. Despite the low turn-out, it was an excellent day, and we talked about the number 42, life, the universe, 80-column card's etc. Mike had some well hard demo's running on the 4A, with some screen scrolling that easily matched the speed of the Amiga. I obtained a copy, but haven't had time to try it on the GENEVE yet. Derek brought his copies of the 80-column card circuit, and we discussed that. Unfortunately, Trevor Stevens couldn't get hold of the actual boards in time for the show, so we will have to wait a little while longer before we see one in action. I showed Mark my windowing program, and he's offered to help me with tarting a few bits of it up with machine code. He showed me his program, but he said there was a bug in the joystick routine and it crashed the GENEVE.

Mark and I also managed to obtain the Infocom adventure "Hitch Hiker's Guide to the Galaxy" (yes! that's where the number 42 first appeared!)



After returning from Wrexham, I stuck the Cray 1 back on-line and accessed the Myarc DMS disk. I've now managed to copy the disk manager over to a standard DSSD disk, but I can confirm that the GENEVE will not recognize the drive, which is my single-sided 40 Track TI drive in my box. It's connected to the Hard and Floppy Disk Controller, and is set as drive 4, which means that I accessed it as drive 8 on my 4A, because the Hard On controller is located at CRU 1000. I tried cataloging drive 8 with the GENEVE, but just got scrambled characters appearing for disk name etc. and nothing else appeared. It could possibly be because 0.97H MDOS which I am currently using is expecting the controller to be at 1100 for floppy support. I've had no problem with the hard disk control with the card at 1000, and I'm continuing to use my good old 1983 TI controller for floppies. I'd like to see my Amiga last that long! Amiga's just aren't built to military standards I'm afraid!

By reading the disk, I discovered that Oliver had not sent any information on the modifications, so I will have to try and write to him again, or contact Barry Boone.

I can also report that I've now also ordered from the disk library, and have also submitted some disks.

I've sent Myarc Disk Manager 5 Version 1.3 to be included in the disk library. This will allow Hard Disk Controller owners to gain access to the latest version of probably the best disk manager available for the 4A. It will only run with the Hard Disk Controller on-line, so there are no copyright problems to worry about. I've also sent a disk of Blues which contains most of my Blues pictures, and includes pictures of John Lee Hooker, Robert Johnson, Leadbelly, Muddy Waters etc. It also contains D/V 80 files that have information on JLH, Robert Johnson, and Leadbelly. It also contains the digitized sound file of Leadbelly's June 15th 1940 recording of Midnight Special. It needs slowing down to around 60 or 70 on a GENEVE to make it listenable, but the GENEVE needs to be slowed down to speed 18 to be the same speed as speed 0 on the 4A, so you will have to play with it until it sounds O.K.

I've also included a disk of MYART images which contains some files which are screen dumps of my windowing program, and also my 3D picture of an office which shows a chair, shelves, a cupboard (including a plant). There's also a disk of GIF's which are a few of the images I scanned at Poly using the 24-bit scanner. The quality is not quite up to scratch, because the Amiga won't currently save out GIF's in more than 32 colours!

I suppose Stephen Shaw will do report on them in the Winter issue of TI\*MES and give you their disk numbers to allow you to order them. 4A owners will either need a GIF loader for the 80-Column card, or GIFmania. For GENEVE owners I've included Paul Charlton's public domain demo GIF loader called GIF2-EXE, which loads from MDOS.

In my last article I mentioned that I had got some programming projects to sort out. Unfortunately, we've not got After Hours to re-compile yet, but Mark gave me a copy of RAG software's Macro Assembler at Wrexham, so I will try it out on that next. When I tried compiling it with Funnelweb, or Editor/Assembler cartridge, it gave me error codes that I've not been able to find, and DSR errors, and it also screwed up all of

the character definitions. Can anyone suggest what might be causing it!

Most of my time has been spent completing my windowing system, but I did spend a short time working on some routines for my CAD system. My 128-column word processor is also fully de-bugged and allows the full editing of a D/F 128 file on disk. All I need to do now is add a menu to it, and a loading facility which will load your D/V files into the D/F 80, or D/F 128 file. The system uses the Dis/Fix files as memory, which means you are only limited by your disk space. I will then need a little routine to re-save the Dis/Fix files back as Dis/Var. It will even include an outline editing facility, support for control U functions on Icons to save messing about with the TI-Writer manual. Incidentally, the control U facility is already working in the version that I've completed. Hopefully, colour will also be supportable. I will also include an 80-column editor that has simultaneous outline editing, so you can have little windows opened that contain information you wish to include in whatever text you are typing in the main window.

After recently seeing an high-res plotting program in TI\*MES after I announced mine, I've decided that the market's already swamped, and I'm running out of time to finish the other projects. I think the main project at the moment will have to be an icon editor for the windowing system. By the way, each icon that you can see in the windowing system is 32 pixels down, by 64 pixels across. I've also discovered how to save Amiga icons as IFF brushes. This will allow me to do an IFF to GIF conversion on every Amiga icon I can find, and then transfer them all over to the Cray. I will then compile a load of them into TI-Artist pictures and stick them in the library as clip-art! Chew on that Commodore!

Another job that I've got to do is modify the address decoding on my CorComp RS232, Horizon RAMdisk, and Rave Speech Adapter. This is because I've received my Dynamical Systems Research (Ron Walters) MEMEX half Meg memory expansion. All TI's cards are fully decoded, which means they don't clash with the 512K RAM that can be addressed in the expansion box.

However, CorComp, Horizon and Rave, didn't follow the guidelines set down by the almighty creators, and they didn't add the extra address decoding. Therefore I have had to move a DIL switch on the 512K card to disable one of the banks that might clash. Also, Ron Walters has added a damn clever bit of hardware and software to the card which causes the card to disconnect areas of its RAM from the bus if it detects another peripheral accessing that area which might clash.

I can remember Leo Hughes having some problems with his Myarc 512K card, and I think that it could have been a problem of clashing addresses, because the Myarc 512K card, originally designed for the 4A doesn't include the auto-clash-detect!

If you have got a box full of decent TI cards, however, then you should have no trouble.

I will let you know how I get on with the soldering, which will hopefully be in time for Winter TI\*MES!

Also next issue will be a review of Batch-It, and a list of the commands available in my Task Definition Language, which allows the writing of code for Pull Down Menu's. There will also be a Myart image loader to type in which will run with Missing Link, and allow standard 4A owners to view Myart files

before deciding whether its worth printing them with HARDCOPY.

Trevor Stevens and I are also planning a new game for the 4A which will be written in FORTRAN-4A, and will be a compiled Option 5. Watch this space for Lemmings4A!

What's the difference between a TI Lemming, and a normal Lemming? A normal Lemming commits suicide by jumping off of a cliff. A TI Lemming does it by pulling out of the market when it's still got a world beating product!!

Well, that's it for this article. Thank you for reading!

I'LL BE BACK.

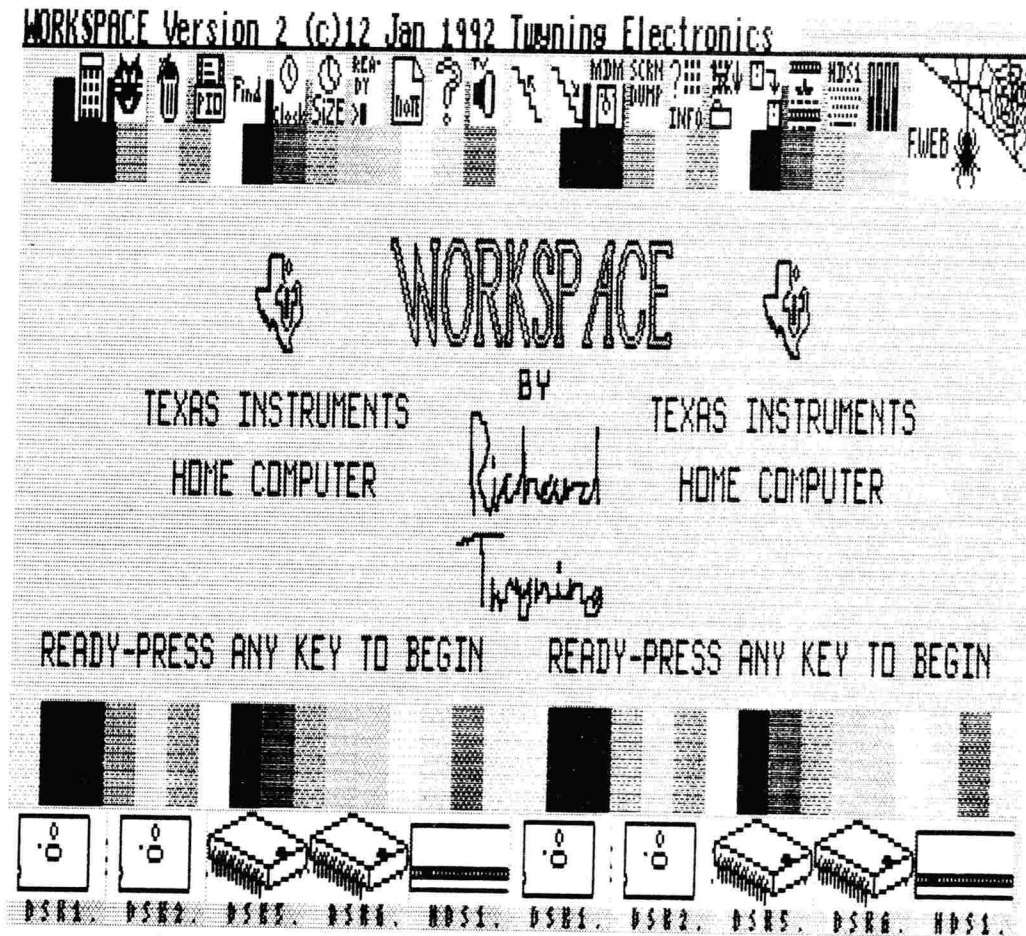


Figure 1.

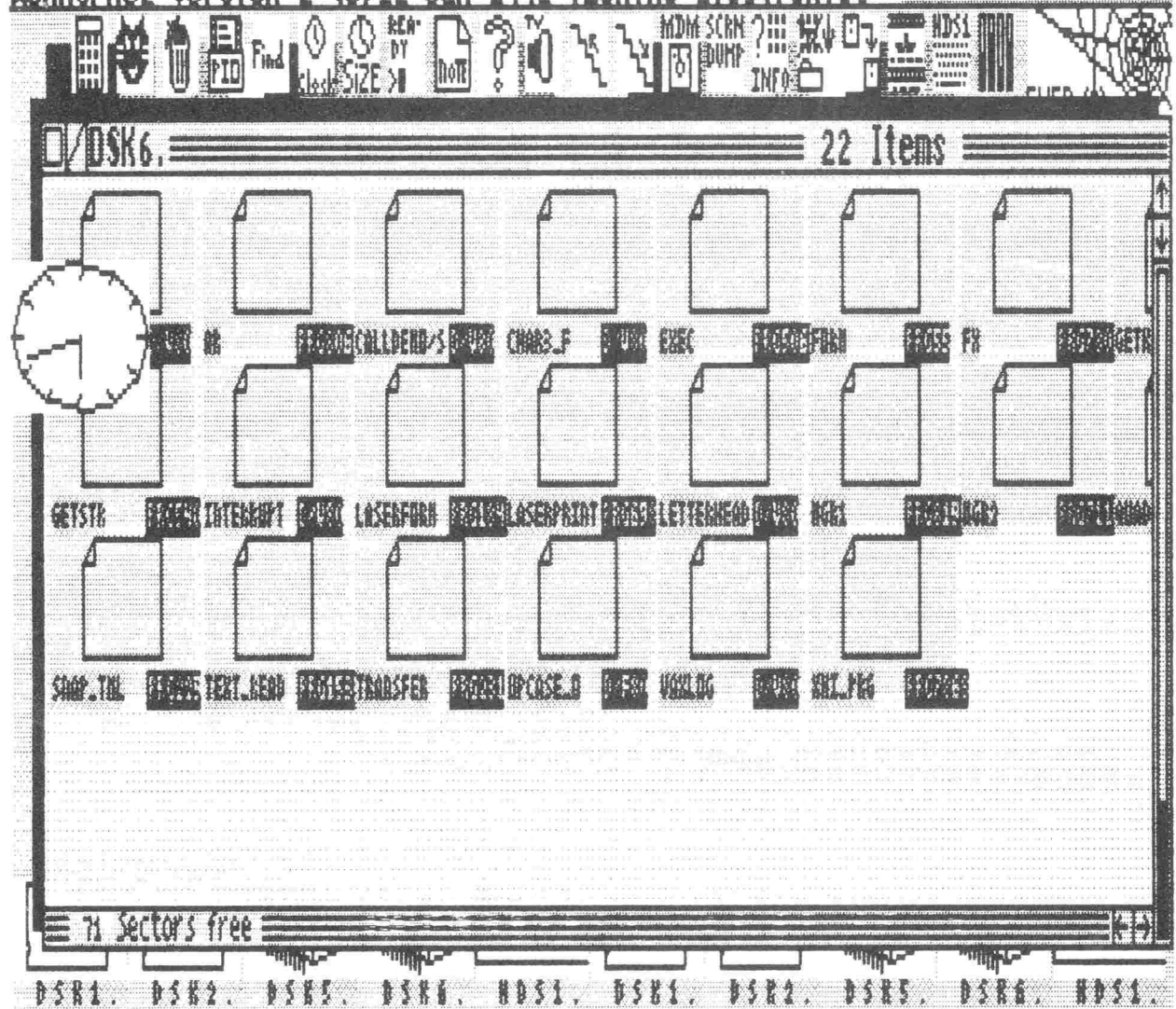


Figure 2.

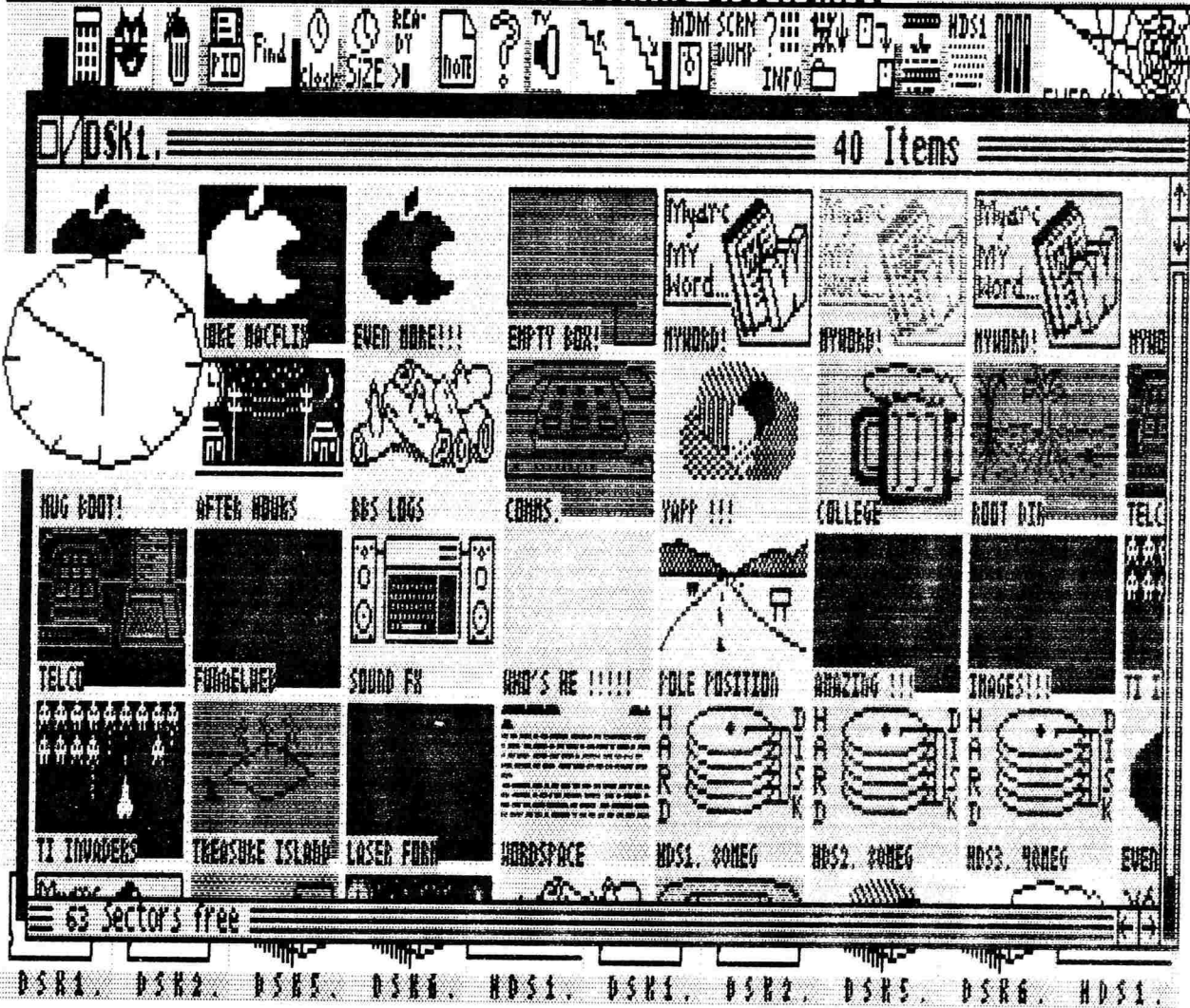


Figure 3.

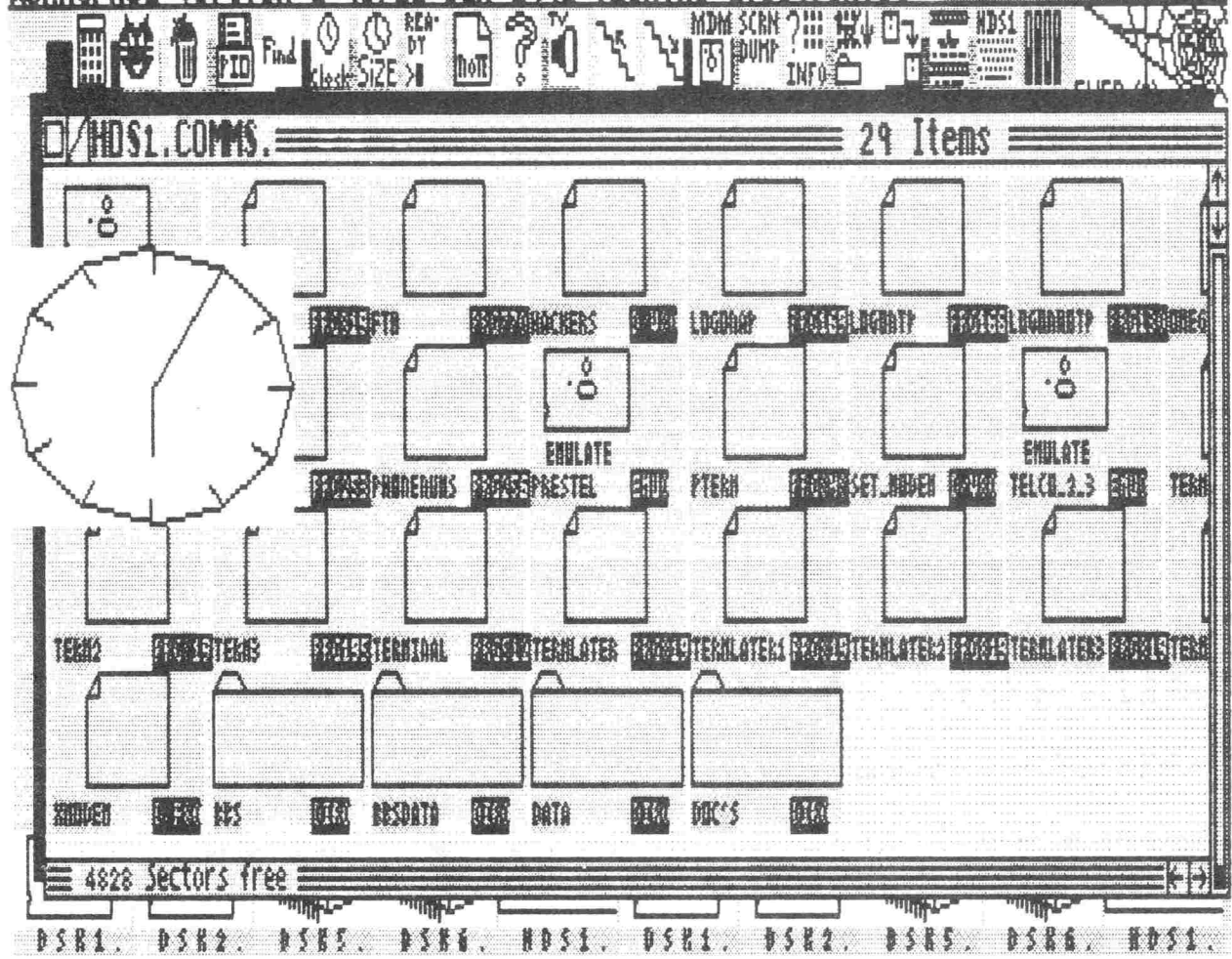


Figure 4.

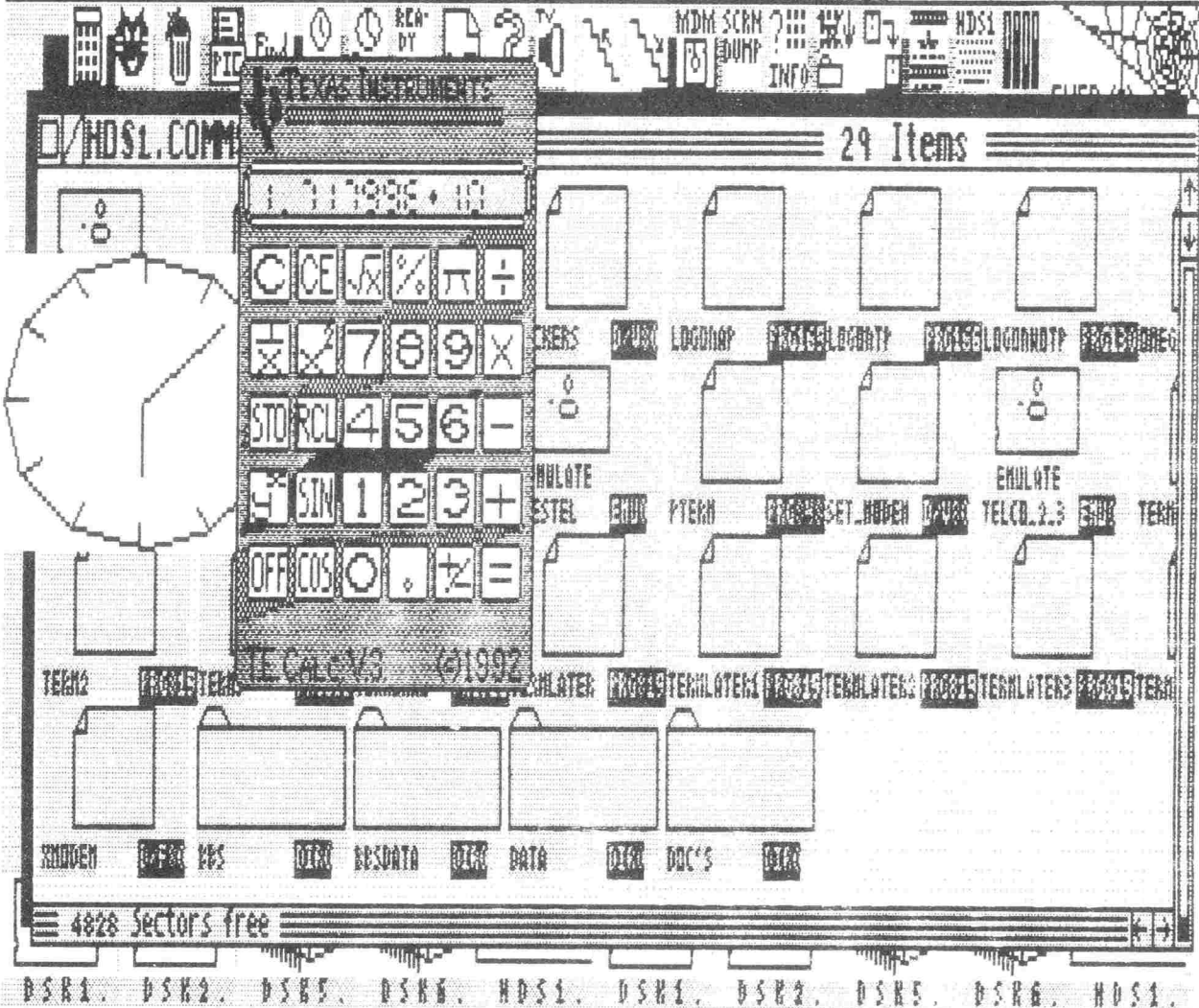


Figure 5. The clock is updated after each button press of the calculator.

```

599 CALL OH(OLH,C):: D=0 :: A$,B$="0" :: CALL V2 :: FOR CR=2
0 TO 116 STEP 32 :: FOR CC=100 TO 164 STEP 64
600 CALL DRAW(CR,CC,STR$(CR)&STR$(CC),2,16):: NEXT CC :: NEX
T CR :: CALL LINK("FILL",21,105,11):: CALL LINK("FILL",94,14
4,11):: CALL FONT("LCD",9,40,68)
602 CALL VP(44,105,51,222):: IF LEN(A$)>1 THEN IF ASC(A$)=48
THEN A$=SEG$(A$,2,30):: IF ASC(A$)=46 THEN A$="0"&A$
603 CALL LINK("FILSCR",15):: CALL PR(A$,44,107,2,15):: CALL
CURSOR(CR,CC,B,CS):: GOSUB 710 :: IF A$="" THEN A$="0"
609 IF A=1 THEN A$,B$="0" :: D=0
610 IF A=2 THEN A$="0" ELSE IF A=3 THEN IF A$<>"" THEN A$=ST
R$(SQR(VAL(A$)))
613 IF A=4 THEN GOSUB 700 :: D=1 ELSE IF A=5 THEN A$=STR$(PI
)ELSE IF A=6 THEN GOSUB 700 :: D=2 ELSE IF A=7 THEN A$=STR$(
1/VAL(A$))
614 IF A=8 THEN A$=STR$(VAL(A$)^2)ELSE IF A>8 AND A<12 THEN
A$=A$&STR$(A-2)ELSE IF A=12 THEN GOSUB 700 :: D=3
615 IF A=13 THEN CALL CURSOR(CR,CC,B,CS):: GOSUB 710 :: P$(A
)=A$ :: GOTO 602 ELSE IF A=14 THEN CALL CURSOR(CR,CC,B,CS)::
GOSUB 710 :: A$=P$(A):: GOTO 602
616 IF A>14 AND A<18 THEN A$=A$&STR$(A-11)ELSE IF A=18 THEN
GOSUB 700 :: D=4 ELSE IF A=19 THEN GOSUB 700 :: D=5 ELSE IF
A=20 THEN A$=STR$(SIN(VAL(A$)))
617 IF A>20 AND A<24 THEN A$=A$&STR$(A-20)ELSE IF A=24 THEN
GOSUB 700 :: D=6 ELSE IF A=25 THEN CALL LD :: GOTO 300
618 IF A=26 THEN A$=STR$(COS(VAL(A$)))ELSE IF A=27 THEN A$=A
$&"0" ELSE IF A=28 THEN IF POS(A$,".",1)=0 THEN A$=A$&"."
619 IF A=29 THEN IF SGN(VAL(A$))=-1 THEN A$=SEG$(A$,2,20)ELS
E A$="-"&A$
635 IF A<30 THEN 602 ELSE IF A$="" OR B$="" OR D=0 THEN 602
636 A=VAL(A$):: B=VAL(B$):: IF D=1 THEN A=A/100*B ELSE IF D=
2 THEN A=B/A ELSE IF D=3 THEN A=A*B ELSE IF D=4 THEN A=B-A E
LSE IF D=5 THEN A=B^A ELSE IF D=6 THEN A=B+A
637 A$=STR$(A):: GOTO 602
700 B$=A$ :: A$="0" :: RETURN
710 CR=INT((CR-58)/15):: CC=INT((CC-105)/20):: A=(CR*6)+CC+1
:: RETURN

```

Figure 6.

30

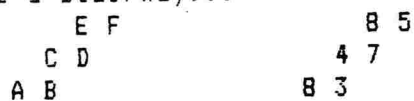
2022 note- a word of explanation- this code is referred to ten pages back as being "the full code" for the calculator shown on screen in Figure 5 (previous page). It is however only a small code fragment with no SUB OH or SUB V etc etc etc. ss



PUZZLE TIME...

New Scientist magazine has a long running puzzle item each week, which is often suitable for computer solution. The following is Enigma No 669 (like I said, long running!) and is set by Keith Austin:

Imagine a stairway...



Shown above using letters for reference and numbers which we shall actually be using!

If we start with the two digits A B as 8 and 3:

Position C is the unit portion of the number we get by multiplying A & B - so, A \* B is 8 \* 3 or 24, so C is 4.

Position D is the unit portion of the number we get by adding B and C together- so B + C is 3+4, or 7.

And so on and so on and so on.

Puzzle: What is the largest number of steps you can have in a stairway without repeating a step (stop when a number repeats (eg AB) and exclude the repetition in your count).

I wrote a little program for this and got the answer pretty quickly, so to make it more interesting, give not only the number of steps, but the two digits that start the stairway. If more than one pair of digits leads to that number of steps, quote all relevant pairs of digits!

While I have a program to do this, if you want a solution PRINTED then send in YOUR solutions and programs and we shall see what the fastest way to tackle it is!

=====

ATTRACTORS...

This is a fractal program which works on the basis of attraction, which we have covered before - the pattern is drawn in an apparently haphazard manner (watch it form) but after many dots have been plotted, there IS a pattern, as the plotted dots are "attracted" to the final pattern.

The program is short, but as many dots are required, it has been left as a closed loop. Using The Missing Link, it is possible to dump to printer from the running program, and if you BREAK, you can save it in Artist format from command mode.

Other pixel-based languages can of course be used, but in the absense of interrupt driven dumps and the ability to retain a graphic after BREAK, you may need to insert a key test routine.

For those mathematically inclined the routine is based on the formula:

$$f(z,g)=(au+bv+g)z+cz^{(-m-1)}$$

In the program on the next page, the variable M represents the number of sides in the final outer envelope, with a minimum of 3. 6 works well, but not 4 or 8. Odd values seem to be all OK - as the value gets higher so the shape tends to a circle.

Variables A and C always have the same sign (- or +) which always differs from the sign of variable L. The pattern can zoom off to infinity (easily tested) or very quickly settle into a small stable area or two stable areas, quickly apparant watching the screen.

```

100 READ A,B,C,M,L,SC,X0
110 DATA 1,0,0.5,3,-1.804,80,80
120 REM STRANGE ATTRACTORS
130 ! functions with symmetric attractors
140 ! uwe quasthoff, leipzig
150 ! fractal report #21 June 1992
160 ! for ti99/4a + TML by
170 ! stephen shaw uk 6/92
180 !
190 X,Y=0.01
200 U=X*X+Y*Y
210 XX=1
220 YY=0
230 FOR I=1 TO M-1
240 XXN=XX*X-YY*Y
250 YY=XX*Y+X*YY
260 XX=XXN
270 NEXT I
280 V=XX*X-YY*Y
290 X=(A*U+B*V+L)*X+C*XX
300 Y=(A*U+B*V+L)*Y-C*YY
310 CALL LINK("PIXEL",X0+SC*X,94+SC*Y)
320 GOTO 200
330 ! CYCLES TIL YOU STOP IT
340 !
350 ! FRACTAL REPORT is just 10 pounds per year
360 ! reeves telecomms labs ltd
370 ! west towan house
380 ! porthtowan, truro,
390 ! cornwall, uk, tr4 8ax
400 !
410 END

```

Required: Extended Basic,  
32k ram, The Missing  
Link.  
Or adjust for other  
graphics programs.  
2022 ss

In order to investigate other variable values, I have amended the program as below, to produce random numbers within a range which TEND to produce results...

```

10 RANDOMIZE
20 CALL LINK("CLEAR")
30 M=INT(RND*7+3) :: IF M=4 OR M=8 THEN 30
...
100 A=INT(RND*6-RND*6)
101 C=RND*1.1*SGN(A)
102 IF ABS(C)<0.3 THEN 101
110 B=2*RND :: L=1.2+RND*1.6 :: L=L*-SGN(A)
115 SC=95 :: X0=96
....
205 IF U>1E12 THEN RUN
...

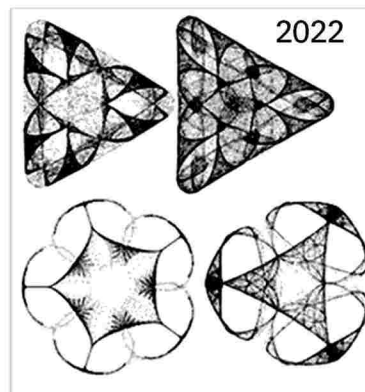
```

Other DATA values which are nice are:

```

DATA 2,-0.2,1.0,3,-1.75,140,80
DATA -1,0.1,0.8,3,1.520,80,110
DATA -2,0.0,-.5,5,2.600,75,105
DATA -1,0.1,-.8,5,1.300,100,100
DATA 5,2.00,1.0,6,-2.70,120,100
DATA 5,2.00,1.0,6,-2.585,130,100
DATA 1,0.04,0.1,7,-2.065,80,100
DATA 4,2.00,1.0,9,-2.60,100,100
DATA -4,1.32,-.51,9,2.64,95,95

```



There are some very fine details to be found by taking a closer view (the same as using a higher definition) but this takes a great deal longer- if the image was plotted to 1200x1200 pixels, you would need to plot maybe 500,000 dots to get the detail!

=====

LIMA.

## LIMA 1992

On May 16th 1992, the UK User Group held its AGM in Derby. On the same day, ... the USA, the tiny LIMA group put on its huge annual show, attended by all the activists in the TI world.

I have obtained almost 16 hours of video recording made on this ONE day, and would be happy to share them with you.

The tapes supplied are of course NTSC standard and comprise three E120 tapes recorded at EP (Extended Play - that's 3 times ). If you have a recorder that can handle them, rental is five pounds per week per tape, with a fifteen pound deposit per tape.

These tapes are an amateur recording, much copied, at very slow speed, on an inferior tv system to ours! A significant amount of wow is apparent, and some portions suffer from external noise or poor presentation standards.

In some cases the presentations have no visual content and audio copies are perfectly OK.

Here is the sequence of presentations and notes on the recordings available...

8:00 - 9:00AM Room 101. Speaker Ken Gladyszewski- "Do it yourself products for the TI, including analog to digital conversion". Good sound, strong visual content, demo of robot etc. More of a practical demo than a "how to" but hardware hackers may find it of interest.

8:30 - 9:30AM Room 150. Speaker Charles Good- "A preview of FUNNELWEB v5 with a completely rewritten editor"- this is the 80 column version, which is very different to the 40 column version! Strong visual content and reasonable sound. It is hardly going to be viewable on a NTSC set let alone after conversion! Audio tape can be made for you though.

9:00 - 10:00AM Room 101. Don O'Neil - 4A Memex, an SCSI interface, Digiport, the Accelerator. Hardware projects discussed at some length, together with news for developers: don't use wire-wrap. Three layouts of the Accelerator failed, although PERFECT as the wire connecting two chips was 3cm long, and the maximum allowed was a path of just 1cm! Also wire wrap appears to be intrinsically too NOISY for these new sensitive chips.

Interesting talk, very suitable for audio only. Also on VHS video. NOTE: Don makes a statement regarding SFX which is in agreement with the issued docs, but those are very very WRONG. SFX is much better than the docs suggest, and you CAN speed up files using a supplied conversion program - the docs incorrectly refer to slowing it down.

9:30 - 10:30 Room 150. Speaker Joe Ross- "Applications for C-SHELL 99"- reviewed as a commercial program in MICROpendium, April 1990, a GUI program with an emphasis (not exclusive) on serving c99 programs and programmers.

10:00-11:00AM Room 101. Speaker Dolores Werths- "Programming music for the MIDI Interface; new music from Harrison Software"- not very suitable, only a PC was used, not hooked up to the VCR, and Lori talked OVER the music that was playing. Add the wow and this is not a good thing to watch. It is available as a poor VHS video though.

10:30 - 11:30AM Room 150. Speaker Bill Nelson- "Comrodine Products"-scheduled but not on the Lima videos. Possibly replaced with Ken Gilliland-see below.

11:00 - Noon Room 101. Speaker Mike Maksimik- "The latest MIDI hardware and software from Crystal Software". Available on audiotape or available on VHS video.

11:30AM - 12:30PM Room 150. Speaker Gary Bowser- "Hardware and software from O.P.A."

NOON - 1:00PM Room 101. THE MULTI USER GROUP "CONFERENCE", a meeting of user group officers to discuss our common problems and solutions.

12:30 - 1:30PM Room 150. Speaker Bruce Harrison- "The latest non-music offerings from Harrison Software"- available on VHS videotape. Not really suitable for audiotape.

1:00 - 2:00PM Room 101. Speakers Eunice Spooner and Meaghan Good- "Teaching TI LOGO to a first grade student: a real lesson taught to a real first grader"

1:30 - 2:30PM Room 150. Speakers Mike Sealy and Mickey Schmitt- "The latest software from MS Express"

2:00 - 3:00PM Room 101. Speaker Beery Miller- "Software for the Geneve from 9640 news" - or how not to make a presentation. Available on a poor quality VHS tape, but I found it incomprehensible. NO visual element so can be put onto audio tape.

2:30 - 3:00PM Room 150. Speaker Jim Peterson- "A Programmable Calculator"- an even better how not to make a presentation. Absolutely no sound! Not at all suitable for anything (somebody neglected to make sure Jim switched the mike on!).

3:00 - 4:00PM Room 150 Speaker Bud Mills- "Products of Bud Mills Services"

3:00 - 4:00PM Room 101 Speaker Karl Romstedt- "A new character editor"- not supplied on the video tape from Lima.

4:00 - 4:30PM Room 150 Speaker Lee Bendick- "A demonstration of the TI-99/8 and its unique set of peripherals". Not a brilliant presentation but of distinct archival value. Available on VHS videotape. Not suitable for audio tape. Recorded at his home.

4:30 - 5:30PM Room 150 Speaker Barry Traver- "Porting MS-DOS text files and software over to the 99/4A"- very suitable for audio tape.

Unknown! Ken Gilliland of Notung Software demonstrating some of his software. Available on VHS videotape.

Unknown! "A demonstration of Rich XB-6K and the Gramulator"

THROUGHOUT- videocam shots of the user group and dealer areas and short interviews with each stall holder. Sound is not so good- lots of background sound- but relatively interesting.

Drop me a line if anything interests you, be it NTSC video, PAL video, or audio copies. Rental cost of videos is five pounds per tape per week, with a fifteen pounds deposit. PAL tapes are E195 at standard play.

The VHS tapes I have are as follows:

1. O'Neil, Bendick, Gilliland, Harrison, part Maksimik
2. Bowser of OPA, Gladyszewski, and Spooner (80 mins).
3. Rich XB, Hall interviews (67 min), Bud Mills, MS Xpress.
4. POOR quality transfer of Miller, Werths, Rich-XB and part hall tour.

Or send a C90 cassette for a recording of Barry Traver and Beery Mills, plus one pound which includes return postage. [eof]

Catalogue received:

"Strange Attractions - the worlds only chaos shop"

A collection of fractal/chaos related items, especially posters, also a couple of videotapes, some books, badges, mugs, T-Shirts, post cards.

Anyone in the London area may find a visit interesting- the shop is said to be painted externally with a colourful fractal design.

Full colour mail order catalogue is free on request.

204, Kensington Park Road, London W11 1NR. Nearest tube is Ladbroke Grove (about 300 yards). Closed Mondays. Open 10am-6pm.

=====

MANCALA:

The superb never released module is available from the library and includes some docs by myself. Here is a discovery, which is to be credited to my 8 year old son George Shaw: Mancala has a HELP mode, very like the old Chess module.

Whether you play two players or against the computer, when it is your turn, and you are stuck, press AID, then select the level of aid you require (from dumb to genius, 5 levels). The computer will then move the oversize cursor to the pit it suggests you empty. To select the pit just press fire or Q. If another move is immediately possible, you can again select AID or play on your own.

You are not obligated to play the suggested move!

This program was evidently ready for realease, as it is in a very well finished state. Highly recommended!

=====

TI WRITER TIP:

Should you ever wish to load or save a text file with a lower case filename, you may have noticed that TI WRITER and its clones will only allow the use of upper case characters in the command line. If you really want lower case, then use Funlweb, which allows you to convert upper case to lower case by passing over the upper case letters pressing CTRL and the semi colon key together (that's a ";"- just to the left of ENTER).

=====

Here is another listing, keyed in by my 8 year old son, so no excuses about it being too long for you...

```

100 RANDOMIZE
110 CALL CLEAR
120 TED=5+INT(10*RND)
130 DIFF=2+INT(8*RND)
140 DISPLAY AT(3,10):"AGE PUZZLE"
150 DISPLAY AT(7,1):"Dave is";DIFF;" years older than Ted"
160 DISPLAY AT(9,2):"The sum of their ages is";
170 DISPLAY AT(11,5):DIFF+(2*TED)
180 DISPLAY AT(13,1):"How old is Ted?"
190 INPUT "Age?":ANSWER
200 IF ANSWER=TED THEN 260
210 DISPLAY AT(17,10):"Sorry, that's wrong"
220 DISPLAY AT(18,5):"The correct answer is"
230 DISPLAY AT(19,12):TED
240 FOR I=1 TO 2000 :: NEXT I
250 GOTO 110
260 DISPLAY AT(23,1):"CORRECT! Well done!"
270 FOR N=1 TO 5 :: CALL SOUND(-200,110+N*5,8):: NEXT N
280 FOR T=1 TO 200 :: NEXT T
290 GOTO 110
300 END

```

age  
puzzle

Program credited to David and Michael Curl for Spectrum computer.

[from The Book of Classic Puzzles and Word Games by Gyles Brandreth, Chancellor Press, 1992. ISBN 1 85152 114 3] [eof]

M  
A  
N  
C  
A  
L  
A

TI  
Writer

DATE 890305

by WESLEY R. RICHARDSON  
NORTHCOAST 99ERS - CLEVELAND, OH

There are many conventions used to represent the date, but I have found one to be more useful for computer applications. The format which I use is YYMMDD, such as 890305 for March 5, 1989. Below are some of the ways to represent this date:

March 5, 1989  
Mar. 5, 1989  
5 Mar 89  
3-5-89  
3/5/89  
890305

The March 5, 1989 format has the least possible confusion over meaning, but is also the least efficient for use by a computer. The length of the notation depends upon the specific date, for example December 31, 1989 takes more characters than March 5, 1989. With the format 890305, there are always six characters for the date.

The input can be read as either a numeric field or as an alphanumeric (string) field. When sorted, the dates will be in proper date sequence. If you sort a format such as 3-5-89, then the month and day sort properly, but there is a problem with 12-22-88 coming after 3-5-89 instead of before. Some programs get around this by changing the representation of the date internally for calculations, but then presenting the date in a format such as 5 MAR 89 for the user.

I am sure you will continue to use the date format which is most convenient for your particular application, but now you have at least one more format to consider, and one which find useful for computer applications.

=====

#### ALGORITHM DESIGN

-----

Garry J Christensen  
Brisbane User Group, Australia

ALGORITHM - a systematic procedure for solving a problem or accomplishing some end.

Programming some tasks on the computer is easy. Many short programmes can be written quickly and with little fore-thought. This is not always the case. The more complicated the problem, the more involved the solution. There comes a stage when the programmer must sit down and work out both how he will approach the problem and how the programme will work.

An ALGORITHM is usually the result of this work. The algorithm is probably the most important part of any programme yet it is so often ignored.

Once the algorithm has been written, the programme is much easier to write and frequently will have fewer bugs. That does not mean that algorithms only apply to computing. An algorithm can apply to any situation where a result is required. Consider the algorithm for finding the partner of your dreams using a dating service: [see next page]

Send money.  
Wait for reply from dating service.  
Contact date.  
Go out with selected partner.  
Repeat until suitable partner is found.

This process is far from perfect as you may go broke before getting married but the algorithm still exists.

The first part of algorithm design is the analysis of the problem. In simple cases, the answer is obvious but sometimes it may require some thought to determine the solution. Consider also that the solution may not be unique. There may be other ways of doing the same thing. Before even touching the keyboard, try to determine that the method you have chosen is the best.

The second part is to determine the steps that will lead to the solution. This part involves a method called top-down development or step wise refinement.

Start with the steps written in very broad terms (generally using English) then refine each step into a series of more detailed steps. Each of these steps can then be further refined, and so on until the steps are close enough to the programming language to allow easy conversion. Don't start to write programme code in your algorithm until the very last step.

Let's consider the algorithm necessary to instruct a robot to make a cup of coffee. The solution is simple so lets define it in broad terms.

- 1 Boil the water.
- 2 Put coffee in the cup.
- 3 Put water in the cup.

These steps are obviously too complex for a robot that has no experience in the human world so they need to be defined further.

- 1 Boil the water.
  - 1.1 Fill the kettle.
  - 1.2 Plug in the power.
  - 1.3 Turn on the power.
  - 1.4 Wait until the water has boiled.
  - 1.5 Turn off the power.

Refine the steps still further.

- 1 Boil the water.
  - 1.1 Fill the kettle.
    - 1.1.1 Place kettle under tap.
    - 1.1.2 Turn on the tap.
    - 1.1.3 Wait until the kettle is full.
    - 1.1.4 Turn off the tap.

Continue through all the steps using this method until a full list of detailed instructions is produced.

As you can see, this is an easier way to end up with detailed instructions than simply starting at the top and working straight through. This method of refinement can continue until the instructions are something that the robot can perform. These are called primitive operations. In computer programming, the primitive operations are the statements that make up the language.

There are two types of traps that I should point out at this stage. The first is called the "first catch your rabbit bug". We have so far assumed that there is a kettle for the robot to use and that there is a tap available. Nowhere have we told the robot to "find a kettle" or "go to the tap".

The second trap is called the "initial state of things bug". What will the robot do if the water is already turned on. When told to "turn on the tap" it may screw it out of the wall.

The secret to writing an algorithm is to keep the following in mind:

1. Start at high levels (use English to express very broad steps).
2. Concentrate on refining one step at a time.
3. Don't start writing programme code until the steps are approaching the level of the programming statements.

Lets now work through a couple of examples.

I will use the selection of all the prime numbers between 1 and 100 as an example. (A prime number cannot be evenly divided by any number other than 1 and itself, eg 7). Problem: How do I determine whether a number is prime.

The simplest answer is to divide it by all the numbers less than the number in question and see if any produce a whole result. That is a valid method but not very efficient.

If the number is 50, dividing it by any number greater than 25 cannot produce a whole result ( $50/26 = 1.\text{something}$ ). If the number is called N, we need only divide by all the numbers up to  $N/2$ .

To test all the numbers from 1 to 100, we need to test them all in this method. The algorithm development follows:

- 1 Set N to 1
- 2 Is it prime?
- 3 If so then print it.
- 4 Add 1 to N
- 5 Go to step 2 if N is not greater than 100.

Steps 1, 4, and 5 are very close to a basic statement so the next step of refinement will convert them to code. The others need a little work.

- 1 FOR N=1 TO 100
- 2 Is N prime?
  - 2.1 Set D to 2
  - 2.2 Is N/D a whole number?
  - 2.3 If so, go to step 4 (not prime).
  - 2.4 If not, add 1 to D.
  - 2.5 If D is less than half of N, then go to step 2.2.
- 3 PRINT N (this step is jumped over if not prime).
- 4 NEXT N

Further refinement is not needed.

- 1 FOR N=1 TO 100
- 2 Is N prime?
  - 2.1 FOR D=2 TO N/2.
  - 2.2 IF N/D=INT(N/D) THEN step 4
  - 2.4 NEXT D
- 3 PRINT N
- 4 NEXT N



The programme:

```
100 FOR N=1 TO 100
110 FOR D=2 TO N/2
120 IF N/D=INT(N/D) THEN 150
130 NEXT D
140 PRINT N
150 NEXT N
```

While this algorithm demonstrates the best way to build up an algorithm, it falls short in one point. The approach to the problem is not necessarily the best. Stop reading here and try to find a more efficient way to find prime numbers.

Did you think of one? Follow the next algorithm through. You will notice that I have used indenting to separate the levels of detail. If you want only a brief outline, read the parts of the algorithm that are not indented. The further indented, the more detailed.

REM This method sets up an array of flags for each number between 1 and 100.

- 1 Initialize the array
  - 1.1 Dimension array
  - 1.2 Set array pointer to 1
  - 1.3 Set array element to 0
  - 1.4 Increment the pointer
  - 1.5 Repeat until pointer is greater than 100
- 2 Remove the non-prime numbers
  - 2.1 Set N to 2
  - 2.2 Remove all multiples of N up to 100
    - 2.2.1 Set pointer to value of N.
    - 2.2.2 Check that this value has not been removed by a previous pass
    - 2.2.3 If it has then go to step 2.4
    - 2.2.4 Add N to value of pointer.
    - 2.2.5 Set array element to 1.
    - 2.2.6 Go to 2.2.4 and repeat until pointer is greater than 100.
  - 2.3 Increment N.
  - 2.4 Repeat until N is greater than 50 (100/2)
- 3 Print the remaining numbers.
  - 3.1 Set pointer to 1
  - 3.2 Does the flag indicate that the number is prime?
  - 3.3 If not then go to step 3.5
  - 3.4 Print the value of the pointer.
  - 3.5 Increment the pointer.
  - 3.6 Repeat until end of array.

The programme will look like this.

```
100 DIM FLAGS(100)
110 FOR I=1 TO 100
120 FLAGS(I)=0
130 NEXT I
140 FOR N=2 TO 50
150 IF FLAGS(N)=1 THEN 190
160 FOR I=2*N TO 100 STEP N
170 FLAGS(I)=1
180 NEXT I
190 NEXT N
200 FOR I=1 TO 100
210 IF FLAGS(I)=1 THEN 240
220 PRINT I
230 NEXT I
```

As you can see, the algorithm is easy to follow and you will have to believe me when I say that it made the programming much easier. I tried it first without an algorithm and the result was both longer and less efficient.

It seems that the computer science proverb is correct: 'The sooner you start coding, the longer it will take to write a programme that is correct'.

To give you a bit of practice at algorithm design, I have included several problems. Write algorithms for the following problems. One way to check the answer is to continue on and write the programme. You might try to write the programme first, then solve it using step-wise refinement. Compare the answers. Are they the same?

1. Write an algorithm for placing a phone call.
2. Include in the algorithm for question 1 the possibility of busy signal, no dial tone and no answer.
3. Write an algorithm to test if a string is a palindrome. (A palindrome is a word or sentence that reads the same forward as backward, ignoring the spaces. eg RADAR or A MAN A PLAN A CANAL PANAMA)
4. Devise an algorithm for a programme that will ask for numbers to be input until the value 0 is entered then print the largest number that was input.
5. Write an algorithm for a programme that will ask for a input of a single positive number then print the number with the order of the digits reversed. eg. 13542 results in 24531.

=====  
For those of you who don't subscribe to that superb, excellent TI only magazine MICROpendium, you won't have seen their reports on Myarc, reported to be no longer trading. It is reported that a third party has taken on board hardware maintenance; that a group of users have purchased software development rights; that Lou will write programs; and that Lou reports selling less than a thousand units in all.  
=====  
stephen shaw. 1992.  
=====

"EASY SORT"

ASSEMBLY LANGUAGE SORTING OF XB DATA STATEMENTS

reviewed by Charles Good

Lima Ohio User Group

Many of us have our favorite extended basic name/address program, or household inventory program, or other "list of things" program. There are lots of these floating around in the TI community and the nice thing about them is that, since they are in XB, the inexperienced user can modify them to the user's particular requirements. These programs either load in a separate data file to sort and display or they contain their data in XB data statements. Many "only know how to program in XB" users use data statements in such software.

The advantage of using data statements is that the data is loaded into the computer just as fast as the controlling program. There is no need to spend time loading the program and then loading a data file. One disadvantage of data statements is the haphazard unsorted order that the data is usually entered into the program. New data, irrespective of alphabetical or numerical order, is usually added after all the existing data statements. Reading all this data into memory and sorting it in XB takes a lot of time.

Bruce Harrison to the rescue! Bruce sent me an evaluation copy of a new commercial offering which sorts data in data statements FAST. No assembly language knowledge is needed, just use the SKELETON program or add the appropriate CALL LINK to your own software. SKELETON is very flexible. Many different kinds of data can be stored with a neat menu popping up at the start of SKELETON showing what the data is (video tapes, names and addresses, household inventory, etc). DATA statements can be part of the program, MERGED into the program, or loaded in from a separate disk file.

Bruce's disk includes a demo name/address program that contains the addresses of many prominent personalities in the TI community. The demo program itself is REALLY NEAT with quick sorting by first or last name, street address, city, state, or zip code. The list of TI personality's addresses is in and of itself useful to owners of TI home computers.

Below are excerpts from the letter from Bruce that accompanied my evaluation copy of EASY DATA:

"Here's our latest commercial offering, a little goodie we call EASY DATA. It's for unskilled XB programmers, so they can do great things with DATA statements and a little help from Harrison's Assembly routines.

"The package sells for \$6.00 including S&H for US and Canada customers. Its main ingredient is a routine called MSORT, which is supplied buried in the XB program SKELETON. That XB program also contains a slightly upgraded version of our Menu Driver (QMENU), so the unskilled programmer can make nice looking menus from simple DATA statements.

"MSORT is a magical routine in many ways. It performs sorts by any two fields in the data, as designated in the LINK statement. The kind of sort performed (Numeric or String) is determined by the kind of variable that the field being sorted reports into. It does not waste time on the secondary sort unless the primary sort is a tie.

"Three demos are provided. The one called BIGDEMO sorts 55 records of six fields each on two criteria in a bit over three seconds. [Charles Good note: BIGDEMO is the name/address demo mentioned by me above.]... The time required to sort is essentially insensitive to the original ordering of the data. Lists that are almost in order take no less time than lists that are completely reversed."

EASY DATA is NOW available by mail for \$6 plus overseas post

Harrison Software  
5705 40th Place  
Hyattsville MD 20781

OR FROM YOUR DISK LIBRARY- BY ARRANGEMENT- at an all-in price of FOUR POUNDS. \*this is a commercial program and should not be passed around please!\*

~~~~~  
LETTER FROM AUSTRALIA No. 1  
-----

Tony McGovern Mar/17/92

A while back I made the suggestion to Charlie Good that instead of writing letters in which a lot of the material could well be of general interest as newsletter articles, that I might as well do the relevant parts as a disk file under the name "Letter from Australia". No sooner had I made this rash suggestion than Charlie accepted it before I could change my mind, and had the idea that it could be a bit like Alastair Cooke's Letter from America which is broadcast on the BBC world service. That was indeed the model I had filched for the name - I regularly listen to the program on ABC radio here most Sunday evenings at 7.15, but I don't think it is going to be in the same journalistic league. I always enjoy the mixture of whimsy and often very penetrating comment on the state of the US of A.  
First for the whimsy.

The question arose here a little while back as to whether security guards would be more effective if they had a solid classical education, particularly in ancient Greek. The incident that set this off wasn't really anything to do with George Bush's flying visit here on the way to Japan, but it did follow on in a sort of way. Now George's table manners in Australia were impeccable, and our newly minted Prime Minister made a truly awful speech, but that wasn't it either. It was in fact a Trojan Horse that caused the problem. A what ? Well, Australian farmers are quite unhappy about the way the US is soaking its own taxpayers to subsidize wheat sales around the world to do battle with the EEC who do even more of it. They don't really care about anyone else's taxes, but they do care about their markets, and were demonstrating to make their point to Mr Bush. They rolled out a metal horse made by a former political candidate for campaigning in a previous election. The body of this thing is a metal tank like the ones that get buried in the driveways of gas stations, so you have an idea of the scale of the thing.

After the official visit the metal horse was hauled back to Sydney on a very large truck. In the meanwhile one of the banks, they are no different here, had decided to foreclose on the business of the ex-political candidate. The bank locked everyone out of the factory and posted security guards around it with instructions to let no one in.

This was all publicized in the newspapers. A day or two later the owner rolled up in the huge truck and asked the guards if he could park the Trojan Horse in the factory yard - according to reports in literally those words. Of course once it was inside, out piled 30 or 40 of the factory people and retook possession from the bank. The rest of Australia was still laughing a day later, and wondering about security men who had never heard of a Trojan horse. There can't be anyone with an interest in computers who does not have some idea.

~~~~~  
25 Mar 92 12:11:46  
SOUPED UP SUPERCARTS  
By: Andy Frueh, Lima UG

It all started with the original TI Cartridges. The cartridge ("cart") port of the TI is one of its advantages in both ease-of-use and young education. This port (called the GROM port) allows users to insert cartridges into its slot. This offers instant access to the programs contained in it. I mentioned that this port is an advantage. It is also now unique to the TI. Most other computers do not offer such a port and those that do not offer a whole lot of software. Older computers usually have such a port. The main reason for this is that home computers appeared shortly after the home video games (Atari, VCS, ect.) came out. Computer manufacturers feared that many people would be afraid of disks or cassettes, so they enabled their machines to use software in cart format, just like the popular home video game machines. A good marketing move. In addition, children could use them with considerable ease.

The quality of TI made cartridges grew from awful to excellent. The chief reason for this is the 99/4 was not as capable as the 99/4A. If you look at some later carts (1983), you will see that the manual may say, "for the 99/4A Home Computer only." This means they will NOT work on the 99/4. Most of these are games, such as Parsec and Star Trek. I am not sure if it was the graphics or the speech that would cause problems on the 99/4. Could somebody please submit a letter or an article describing the internal differences between these sister computers? Compare an early module or cart such as the 1980 Hunt the Wumpus to the 1983 graphic adventure Return to Pirate's Isle. The features and graphics of "Return" is significantly greater.

But what does all of this have to do with Supercarts? Well, just as TI tried to improve the quality of their cartridges, many users were "playing" with the cartridge and the GROM port it plugs into.

When most people think of Supercarts, they are thinking of cartridge-like hardware that plugs into the GROM port and have battery backed memory. Most Supercarts can save Assembly programs or even other cartridges and store them in this memory. However, when I use the term Supercart, I intend to discuss almost EVERY device available that can be plugged into this port.

Let's start from the beginning. One of the first Supercarts ever made was marketed by DataBioTics. This was the Superspace cartridge. This included an Editor/Assembler, it could save Assembly programs in its 8K memory and could save any cartridge that didn't use TI's GROM (such as AtariSoft and Funware). What is a GROM? Well it stands for Graphics Read Only Memory. It is completely separate from RAM and ROM, which I assume most users are at least slightly familiar with. These are unique to TI. Three inside the console control the Operating System and BASIC language. Up to five addition GROMs (each with 6,144 bytes of ROM, I believe) can be added via the GROM port, adding 30K to the computer's ROM memory.

DataBioTics later marketed an upgrade to their Superspace. Superspace II had all the features of Superspace as well as letting users use 8K of memory as extra RAM for Assembly or TI BASIC programs. It also had 32K of battery backed memory, instead of only 8K. One thing I should note. Both Superspaces contained an Editor/Assembler. They did NOT include the manuals or support file disks. This may have been due to copyright restrictions. I am not sure if the operation of this Ed/Assm is identical to TI's or not.

The next device I would like to talk about was the GRAMCracker. And no, I won't discuss other GRAM devices, such as the P-GRAM or GRAMulator. Not to take anything away from these two. All three devices perform similar functions. However, since the focus of the article is on devices that actually plug INTO the GROM port, I will only discuss the GRAMCracker.

I believe this came out sometime in 1986. At least that's the earliest ad I've seen for it. It was marketed by MG (originally Miller's Graphics). The main thing it could do was have another cartridge plugged into, then save the contents of that to a disk, RAMdisk or cassette file. You could also MODIFY your cartridges. A manual describing typical customizing jobs was included. It also allows you to modify the TI operating system. For example, you could load in a new character set with true lowercase letters. Such changes would always be in effect, as long as the GRAMCracker was installed. It had 56K of memory, and could handle up to 80K. Obviously, this module was advanced over the Superspace.

What about prices? These are ORIGINAL prices, and I'm sure these devices could be found at significantly lower rates. But here's how they used to compare.

Superspace.....\$39.95 (This was a sale price. I can't find it's original)  
Superspace II.....\$89.95 (\$69.95 on sale)  
GRAMCracker.....\$174.95

The price of the Superspace dropped sharply after Superspace II was introduced. All three prices were around the same time, 1986-1987.

Those prices seem a little, well, pricey. And a lot of other users felt the same way. For that reason, pre-programmed and a few user-programmable "multi-carts" appeared. These were cartridges that contained many modules in one. I assume the idea for this started with the "Widget" or Navarone's module expander, which could hold three cartridges. The disadvantage was the size. This expander is fairly large.

One of the first multi-carts was the MultiMod by the late John Guion. His product was a plug-in modification to people who owned the Super Extended BASIC cartridge. This module was manufactured by Triton, which recently became TM Direct Marketing. I am not sure if TM offers this product. MG developed this cartridge for Triton. Super XB adds 33 new commands and 6 altered ones. Most of these are in the forms of CALLs. The Fall '89 catalog for Triton listed this module with a price of \$49.95, including a graphics utility, Draw N' Plot. The MultiMod gave users Super XB, the Editor/Assembler, Disk Manager III, and TI-Writer in one module. The Multi-Mod was offered in kit form for \$22.95 including manual and all support files for E/A and TI-Writer. I am not sure how well this sold, but I assume reasonably well. It gave users essentially the same thing as Funnelweb, without Funnelweb's enhancements and user expandability. The Multi-Mod's one major advantage was cartridge speed. Of course, the support files still had to be loaded from disk.

There is another similar product on the market, only this one is user-alterable. WAS Controllers offers a device which allows users to install 5 other modules in addition to Extended BASIC. For example, you could have XB, E/A, TI-Writer, Multiplan, Personal Record Keeping, and Tunnels of Doom in one cartridge. This is also sold as a kit, for \$25. This includes the module, which has a modified case so all the other module's chips will fit. The flyer I have states that "only modules with 16 pin GROMs can be used with the Extended BASIC Expander." The GROMs are a normal looking computer chip. What that means is that you have to count the pins coming from the GROM chip. Only modules having 16 pins will work.

OPA is selling a cartridge device with amazing features. It is called the Pop-Cart and looks about like any other TI cartridge. This device contains 256K chunks of RAM and GRAM. You tell OPA what modules or Assembly programs you want on the Pop-Cart, and they burn it into the module. This uses no batteries, but it does mean that users can NOT program it themselves. Supposedly, if you put XB and the TE2 cartridge, the text-to-speech features of TE2 carry over into XB. This could have significant advantages, since many modules (especially some of TI's older ones) "add" commands to TI BASIC.

What about weaknesses? Well this is the one aspect of TI cartridges that is rather disappointing. The connection between the cart and the GROM port can get very weak. This can cause some modules, especially Extended BASIC, to "lock up" The computer refuses to respond, and you usually have to turn it off.

There you have it. I believe that this is basically the principal "Supercarts" available. These devices prove the power and versatility of TI's GROM port. Many modern machines lack such a port. The users of these big machines claim that such a port is "infantile" and has no usefull purpose. Bull! I feel this article PROVES how useful this port can be!

[sjs update: The older ti99/4 did not support bit map graphics. Ed/As manual states also "DXOP only useful on the TI99/4A (p233)". "gp11nk data >004A 99/4A only -p254" "Address of user interrupt routine-99/4A only-p406"]  
~~~~~

HARRISON SOFTWARE'S "CODE BREAKERS"  
reviewed by Charles Good  
Lima Ohio User Group

Do you like solving CRYPTOGRAMS such as those published in newspapers or "mixed puzzle" magazines? These are phrases encoded in a monoalphabetic substitution. Each instance of a single letter in the phrase is represented by a "code" letter that is different from the real letter. The same real letter is always represented by the same code letter in the encryption. Harrison software's CODE BREAKERS gives cryptogram fans the chance to solve such puzzles on a 99/4A with the aid of assembly speed an an excellent screen display.

Code Breakers randomly selects a phrase from a file of available phrases and displays the encoded phrase on screen. Coding is based on a "keyword", also selected randomly. The encoded screen display is in either of two formats:

1- NORMAL, as such puzzles are usually published in newspapers or puzzle magaines with letters grouped into words with all the original punctuation. In this case two and three letter groups are a great help in solving the puzzle, allowing the solver to decode words such as "the", "as", "it", "it's", etc., and to use the letters in these decoded words to partially decode other words in the encoded phrase.

2- PROFESSIONAL. The encoded phrase is displayed as groups of five letters, with the last 5 letter group padded with extra letters if needed. The solver has no initial clues about the length of individual words or where one word ends and the next word begins. This format is really hard!

When solving puzzles you use the arrow keys to move the cursor over a code letter and then press the keyboard key of what you believe the real letter represented by the code letter is. The letter under the cursor AND all similar letters in the coded phrase instantly (at assembly speed) are changed to your guess. Different foreground and background colors make it obvious to the viewer which letters are in still displayed in the original code and which have been guessed (maybe correctly, maybe not) by the solver. When the solver thinks the code is correctly translated, or gives up, a press of the <enter> key reveals the correct solution and the keyword used to generate the encrypted phrase.

There are three possible sources of phrases for encryption. CODE BREAKERS comes with a large disk file of phrases that the software will randomly select from for encryption. Users can also create their own files of phrases, up to 124 characters in each phrase. There is also a "two player game" option that does not use an existing file of phrases. One player types in an uncoded phrase while the other player looks away. The computer then displays an encrypted version of this phrase on screen and the second player try to solve the puzzle.

My testing panel of two cryptogram puzzle fans (my wife and one of my children) gives high marks to CODE BREAKERS. The ability to quickly try different letter substitutions and see their effect on screen speeds the solutions of the problems. Lots of wear and tear on pencil erasers, and the resulting mess, are also eliminated. The only complaint my testing panel had was that they were confined to a chair sitting in front of the computer.

=====

PROGRAMMING MUSIC THE EASY WAY  
PART 1  
by Jim Peterson

Personally, I am about like the country fiddler who admitted that he could read music a little, but not enough to hurt his playing. I know just a little about reading music but that has been all I needed to know to program more than 50 songs. And, if you have ever heard my Tigercub Country or Tigercub Gospel disks, you will know that I have programmed those songs in a wide variety of styles.

Now, I have put together a few little routines to enable anyone to program music on the TI-99/4A very easily, and in many ways. You DON'T need to know how to program and you DON'T need to know how to read music!

First, key in this one-liner and save it as DSK1.SCALE,MERGE

```
100 DIM N(36):: F=110 :: FOR
  J=1 TO 36 :: N(J)=INT(F*1.0
59463094^(J-1)+.5):: NEXT J
:: N(0)=40000
```



Next, NEW to clear memory and then key in this music program, which we will use as an example to experiment with.

```
110 T=2 :: A=13 :: GOSUB 100
0 :: T=1 :: A=18 :: GOSUB 10
00 :: GOSUB 1000 :: T=3 :: G
OSUB 1000
120 T=1 :: A=20 :: GOSUB 100
0 :: A=22 :: GOSUB 1000 :: A
=23 :: GOSUB 1000 :: T=2 ::
A=27 :: GOSUB 1000 :: T=4 ::
A=25 :: GOSUB 1000
130 T=1 :: A=30 :: GOSUB 100
0 :: A=29 :: GOSUB 1000 :: T
=5 :: A=27 :: GOSUB 1000
140 T=1 :: A=25 :: GOSUB 100
0 :: A=27 :: GOSUB 1000 :: A
=25 :: GOSUB 1000 :: A=22 ::
GOSUB 1000 :: T=5 :: A=25 ::
: GOSUB 1000 :: T=2 :: GOSUB
1000
150 T=1 :: A=27 :: GOSUB 100
0 :: GOSUB 1000 :: T=3 :: GO
SUB 1000 :: T=1 :: A=22 :: G
OSUB 1000
160 A=25 :: GOSUB 1000 :: A=
22 :: GOSUB 1000 :: T=2 :: A
=20 :: GOSUB 1000 :: T=4 ::
A=18 :: GOSUB 1000
170 T=1 :: GOSUB 1000 :: A=2
0 :: GOSUB 1000 :: T=5 :: A=
22 :: GOSUB 1000 :: T=1 :: A
=18 :: GOSUB 1000
180 A=22 :: GOSUB 1000 :: A=
27 :: GOSUB 1000 :: T=6 :: A
=25 :: GOSUB 1000 :: T=1 ::
A=18 :: GOSUB 1000 :: A=20 ::
: GOSUB 1000
190 T=6 :: A=22 :: GOSUB 100
0 :: T=2 :: A=18 :: GOSUB 10
00 :: A=20 :: GOSUB 1000 ::
T=4 :: A=18 :: GOSUB 1000 ::
```

Save that by SAVE DSK1.SHEN just so you don't lose it, but keep it in memory, and enter MERGE DSK1.SCALE to get that one-liner back in.

The music you just keyed in is in one voice without harmony. Let's see what you can do with just one voice. Put in a line 105 D=200 and another line -

```
1000 CALL SOUND(T*D,N(A),0)
:: RETURN
```

Enter RUN, wait a second, and listen. If you didn't make any mistakes in keying in the music, you should hear a fairly pleasant single-note rendition of a beautiful old folk song.

Maybe you would prefer a higher key? Here's the neat part about starting with that formula in line 100 - besides the fact that it lets you key in frequencies in shorthand. To change key, just change that 110 in line 100 to a higher frequency number. They are listed in the "blue book" that came with your computer, but if you lost it they go upward 110, 117, 123, 131, 139, 147, 156, 165, 175, 185, 196, 208, 220.

You can also lower the key, providing you do not cause the lowest note in your music to go below frequency 110. In the piece you keyed in, the lowest note number used was 13 so you could go down 12 steps. The frequencies are not in the book, but they go 110, 104, 98, 92, 87, 82, 78, 73, 69, 65.

Want the music faster or slower? Just change the 200 in line 105.

Now let's see what else we can do with single-note music. Try this -

```
1000 CALL SOUND(T*D,N(A),0,N
(A)*1.01,0):: RETURN
```

Has a richer sound, doesn't it? How about this?

```
1000 CALL SOUND(T*D,N(A),0,N
(A)/2,0):: RETURN
```

```

Or combine the two -
1000 CALL SOUND(T*D,N(A),0,N
(A)*1.01,0,N(A)/2,0):: RETUR
N

```

Multiplying a note by 1.01 in another voice will always give a more resonant sound, and dividing a note by two (providing its note number is not less than 13) will always be in harmony - so will multiplying by two, or by four.

How about some real deep down bass music? The TI's tone generators can only go down to frequency 110, but the noise generator can be tuned far below that. The timber of the sound is different and doesn't blend too well with the tones, so use it with caution - but it's great for a tuba solo. Try this -

```

1000 CALL SOUND(T*D,N(0),30,
N(0),30,N(A)*3.75,30,-4,0)::
RETURN

```

Want to go deeper? Try changing the 3.75 to 1.875 - too deep to even be musical, isn't it? Maybe you could improve it by raising the frequency in line 100.

Try changing the 3.75 to 7.5 - not bad, is it? So try doubling it again to 15 - oops! When you go that high you get some very sour notes!

So, go back to 7.5 and change one of those N(0) to N(A) and change the 30 following it to 0. Pretty good, so try also changing the other N(0) to N(A)\*1.01 and the 30 after it to 0.

If any of those effects sound like something you might want to try in a piece of music someday, clear the memory with NEW, key it in and save it with SAVE DSK....,MERGE using a different filename for each one. Then, after you have keyed in some music, you can very quickly merge in different routines and try them. You will find that different ones go better with different songs.

The routines we have been trying all play music with a very strong beat. For a smoother effect, try this -

```

1000 FOR J=1 TO T :: CALL SO
SOUND(-2999,N(A),0):: GOSUB
1100 :: NEXT J :: RETURN
1100 FOR D=1 TO 99 :: NEXT D
:: RETURN

```

You will notice one thing right away; with this method, a series of the same note gets run together into one long note. Later we will look at ways to get around that.

To change the tempo of the music, just change the value of 99 in line 1100. Try this method in combination with the effects we tried previously.

Here's another one that gives a very nice effect -

```

1000 FOR J=1 TO T :: CALL SO
UND(-999,N(A),0):: GOSUB 110
0 :: CALL SOUND(-999,N(A)*1.
01,0):: GOSUB 1100 :: NEXT J
:: RETURN
1100 FOR D=1 TO 8 :: NEXT D
:: RETURN

```

```

Or for a more mournful sound -
1000 FOR J=1 TO T*4 :: CALL
SOUND(-999,N(A),0):: CALL SO
UND(-999,N(A)*1.01,0):: NEXT
J :: RETURN

```

You can control the tempo by changing the value of 4, but not as precisely as with the previous method, and it does not work well with bass notes. Try changing the 1.01 to 1.02 - also try erasing the \*1.01 and change the following 0 to 8, for a mandolin effect.

Those are just a few of the effects you can create with just a single-note melody - experiment and see what else you can discover.

So, just imagine what you will be able to do using all three voices - coming up in part 2 of this article!

=====

TI WRITER (AND CLONES). ABBREVIATED INFORMATION:

#### 1. EDITOR COMMAND MODE.

E (Edit) returns from Command Mode to Edit Mode.

T (Tabs) allows tab setting using L=left I=indent R=right.

RE (Recover Edit) might recover all but the first line of text that was purged.

Q (Quit) gives the options:-

S (Save) to select the Savefile option.

P (Purge) to clear the text buffer.

E (Exit) to return to title screen.

#### FILES.

F (Files) to display the files options.

LF (LoadFile) to load all or part of a file from disk.

(a) DSK1.NAME loads all of that file.

(b) 16 108 DSK1.NAME loads lines 16 through 108 of the file.

(c) 10 DSK1.NAME merges all of the disk file after line 10 of the text buffer.

(d) E DSK1.NAME merges all of the disk file after the end of the text buffer.

(e) 0 DSK1.NAME merges all the disk file before the first line of the text buffer.

SF (SaveFile) to save the text buffer to disk.

(a) DSK1.NAME saves all the file to disk.

(b) 1 108 DSK1.NAME saves lines 1 through 108 to disk.

(c) 26 E DSK1.NAME saves lines 26 to the end of the text buffer to disk.

NOTE If tab settings are not required to be saved, use the command PF (PrintFile) instead of SF.

PF (PrintFile) to print all or part of the text.

(a) PIO prints all of the file to PIO.

(b) 1 50 PIO prints lines 1 through 50.

(c) 48 E PIO prints lines 48 through to the end of the text.

(d) L PIO prints line numbers as well as the first 74 characters of each line.

(e) F PIO prints in fixed 80 format.

DF (DeleteFile) to delete a file stored on disk.

P (Purge) to empty text buffer in the computer.

SD (ShowDirectory) to display a disk directory. [more->]

## LINES.

L (Lines) displays the Lines options.

M (Move) to move lines in the Text buffer.

- (a) 8 10 19 moves lines 8 through 10 to a position after line 19.
- (b) 86 E 7 moves lines 86 to the end and inserts after line 7.
- (c) 12 12 0 moves line 12 to the beginning of the text.

C (Copy) to copy lines and place in another location.

- (a) (b) (c) as for Move above.

D (Delete lines) to delete lines.

- (a) 11 12 deletes lines 11 and 12.
- (b) 127 E deletes lines 127 to the end of text.

S (Show line) to show a particular line at the top of the screen.

## SEARCH.

SH (Search) allows for string searches.

FS (FindString) to find the first occurrence after the cursor position of a character, word or phrase. To place the cursor at the beginning of text, type @FCTN/9 S <ENTER> 0 <ENTER> CTRL L.

- (a) /hello there/ places the line containing "hello there" at the top of the screen.
- (b) 5 20 /goodbye/ searches through columns 5 to 20 for the word.

RS (ReplaceString) to change a character, word or phrase to another.

- (a) /center/centre/ will correct the spelling of the word.

When the original word is found, options are given to type

- Y to replace this one and find the next.
- N to ignore this one and find the next.
- A to replace all occurrences of the word.
- S to stop and exit to the Edit Mode.

WARNING:- Before using any Search commands turn word-wrap off and use the hollow cursor otherwise the text will be reformatted using the current tab settings.

## COMMANDS FOR USE BY THE TEXT FORMATTER.

### 1. TEXT DIMENSION COMMANDS.

#### LINE MANIPULATION.

- .FI (Fill) puts as many words as possible on each line to fill within the left and right margin limits.
- .NF (NoFill) cancels the Fill and prints the document exactly as it appears on the screen.
- .AD (Adjust) spreads the spacing between words so that the text is printed to reach the right margin. Not applicable to lines ending with a CR.
- .NA (NoAdjust) cancels the Adjust command.

#### LEFT MARGIN.

- .LM 6 sets the left margin at column 6.
- .LM +10 adjusts that margin 10 columns more than the previous setting.
- .LM -4 adjusts that margin to be 4 less than previously.

#### RIGHT MARGIN.

- .RM 68
- .RM +10 Similar comments to Left Margin.
- .RM -4

#### INDENT.

.IN 8 indents the first line after a CR to column 8.(absolute value).  
.IN +10 indents 10 columns more than the .LM setting. (relative).  
.IN -4 changes indent to 4 columns less than the .LM setting. (relative).

#### LINE SPACING.

.LS 2 causes the printout to have double line spacing.

#### PAGE LENGTH.

.PL 50 prints 50 lines then starts a new printed page.  
.PL +5 adjusts page length 5 more relative to that before.  
.PL -5 adjusts page length 5 less than previously.

#### BEGIN PAGE

.BP forces a new page break. The current .PL value is then restored.

### 2. INTERNAL FORMAT COMMANDS.

=====

#### CENTRE TEXT.

.CE centres the next line with equal left and right margins.  
.CE 3 centres the next three lines.

#### SPACE

.SP leaves one blank line on the printed page.  
.SP 10 leaves 10 blank lines.

### 3. HIGHLIGHTING AND SPECIAL EFFECTS COMMANDS

=====

The symbols used are @, and ^ . They are not printed when encountered by the text formatter. If one of these symbols needs to be intentionally printed, just type two symbols the same side by side.

& (Underscore). The text following the as far as the next space is underlined.

@ (Overstrike). The text following the as far as the next space is overprinted a number of times to emphasize it.

^ (Required space). Ties words together so that the formatter will treat the word group as one entity for underscoring or overstriking.

.TL (Transliterate). Uses the first ASCII character as a kind of variable to represent the ASCII character/s following it.

e.g. .TL 35:95 prints a \_ (95) wherever a #(35) appears in the text.  
or .TL 95:27:83:1 would send the printer code for Subscript print whenever a \_ (95) appears in the text.

COMMENT. Like REM in BASIC, a comment can be entered in the text.

.CO File Update 2 February 1988. will be ignored by the formatter.

### 4. PAGE IDENTIFICATION COMMANDS.

=====

#### HEADER.

.HE^^^^^^^^^^^^^^^^^^^^DOCUMENT NAME^^^^^^^^^^%

Causes text to be printed as a header to each printed page.

The ^ symbols space the heading over on the page, the heading is printed and the % symbol, if used, causes the page number to be printed consecutively on each page.

.HE (followed by a full line of spaces) cancels the printed header.

FOOTER.

.FO^^^^^^^^^^User's Guide^^^^^^^^^^Page %

Causes text to be printed as a footer to each printed page.  
See notes on header above.

PAGE NUMBER RESET

.PA 3 resets to a value of 3 the automatic page numbering by the %  
command in the .HE or .FO commands.

.PA +2 increments the current page number by 2.

.PA -1 reduces the page number by a value of 1.

5. FILE MANAGEMENT COMMANDS. The include file command inserted in a  
file to be printed causes another file to be called and printed  
as well. The .IF command can be placed at the beginning of any  
line in a file. The following might be entered at the end of a  
file saved to disk as TEXT/1.

.IF DSK2.TEXT/2

.IF DSK2.TEXT/3

.IF DSK.RECORDS.TEXT/4

At the end of printing the file, TEXT/1, the .IF will cause the  
other files to be printed too, first TEXT/2, then TEXT/3, then the  
disk controller will do a search of all disk drives to find the  
disk named RECORDS and on it a file called TEXT/4 which is then  
printed (if found).

## DISK FIX

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

When you have a disk with several files that you have been working on and you do a catalog and it comes up DISKETTE IS BLANK, or DISK NOT INITIALIZED, it can be very frustrating. There are times when the sectors used and available get changed to values like 2389 free and 7887 used, but you know you have a single sided, single density (SSSD) disk drive, with a maximum of 360 sectors. It is also possible to have a disk which will not catalog, yet when Extended BASIC is selected, the disk will run the LOAD program and continue without a problem. These have happened to me and I am sure it has happened to others, so I thought I would document a way which may recover your disk for you.

The items which you will need are:

- your blown disk,
- two blank disks,
- Disk Manager 1000 v3.5,
- Disko or Disk Patch,
- and a sector or track copier program, or the equivalent of any of the above. I will use the Funnelweb v4.10 DISK-PATCH for the sector editor.

Programs which combine sector editor and disk manager are Funlweb's DISK REVIEW and also DISK UTILITIES (DSKU).

- 1) The first step is to initialize a disk in the format which you believe the blown disk was, for example SSSD. For the disk name, use the name that you want on the blown disk after it is restored.
- 2) Using the sector copier or track copier, make a copy of the blown disk. If you get a read error in sector 0, just tell the program to ignore the error. If you are unable to copy the disk with the copier programs which you have available, you may still continue the following steps with the original disk, but be advised that you may lose everything on the disk. The FORTH COPY command is a powerful tool for copying odd disks as it is track based and ignores the usual disk directories.
- 3) Load DISK-PATCH or DISKO and then insert the back-up copy of the blown disk in drive 1. Select option 1 for disk sector editor. Then disk 1, and sector 0. The screen should come up with the data from sector 0. Pressing FCTN 2 will change the screen to ASCII and pressing FCTN 1 will change it to HEX. In ASCII, the first ten characters will be the disk name. In HEX, at byte 12h (h=HEXADECIMAL) will be 01 for single sided and 02 for double sided. At byte 13h, will be 01 for single density and 02 for double density.
- 4) Press FCTN 4 to go to sector 001h. You should find groups of four digits of HEX numbers such as 0002 0003 0009 0015 and so on. These indicate where the file names and file maps may be found. Write down each of these numbers in the order which they are found when read from left to right and top to bottom on the screen. Note also if the first number is 0000, then the disk will catalog as being blank and no file names will appear.

5) Press FCTN 4 to go to sector 002h. In the first ten ASCII characters you will find a file name. Write this down next to the appropriate four digit number you had in step 4). Do this for each of the numbers from step 4). If there were several files on the disk, you may need to press FCTN 9 and then option 1 again to go directly to the location. While in sector edit mode, pressing FCTN 6 will take you to the next lower numbered sector.

6) You now should have a table similar to the one below with the file name and location of each file on the disk.

|      |           |      |            |
|------|-----------|------|------------|
| 0000 | A-SECTOR2 | 000D | PACMAO     |
| 0003 | CENTIPEDE | 0005 | PINBALL    |
| 0009 | DEFENDER  | 0006 | PINBALM    |
| 000A | KONG      | 0007 | POLE/POS   |
| 000B | KONH      | 0008 | POLE/POT   |
| 0004 | LOAD      | 000E | TI/INVADER |
| 000C | PACMAN    | 000F | TI/INVADES |

7) Note in the case that we did find a 0000 but a file was there, as in this case file A-SECTOR2 directory was located at sector 002h, then use the sector editor to view sector 001h. Move the cursor to the first 0000 in HEX and change it to read 0002. Then press CTRL W to write the sector back to the disk, and answer Y to the question RE-WRITE SECTOR?

8) Remove the copy of the blown disk and insert the formatted blank disk in drive 1. Select the sector editor, giving drive 1 and sector 0. After the sector comes up, remove the blank disk and insert the blown disk copy in drive 1. Press CTRL W to rewrite the sector.

9) Load Disk Manager 1000 version 3.5 (DM1000), and then put the blown copy disk back in drive 1. Select option 1, File Utilities. Then select option 2 for Recover file. Give the drive as 1. Enter the first file name on you list and press enter. The program will say SEARCHING DISK, then RE-BUILDING LOST FILE, then FILE RECOVERED. Press enter and then 2 for Recover file. Repeat these steps until all of the files are recovered.

10) Press 1 for Copy/Move/Delete... and give the disk number as 1. Your disk files should now be restored. If the disk free and used does not match up with the sum of the file sizes plus 2 sectors, then go to step 11), otherwise you are done.

11) Do this step only if the disk free is not correct. Place a D in the left column to delete all of the files and a U in the right column to unprotect all of the files. DM1000 will unprotect and then delete all of the files. At this point a catalog should show free 358, used 2 for a SSSD disk. Go back to the recover file section of step 9) and recover each file again.

One other piece of advise, if you have a disk with a bad directory, do not write any files to the disk until you have a chance to fix the directory. If you write a new file, then you are taking the chance that part of another file will be over-written. This can happen because sector 0 may show that a location is free, when in fact it has part of a file in it.



The Art of Assembly - Part 1  
By Bruce Harrison  
Copyright 1991, Harrison Software

Frustrated with Extended Basic? Tired of waiting for C? Fed up with Forth? P. O.'d at Pascal? The answer to your problems is the "Native Language" of your computer's heart, Assembly Language.

Many programmers today shun this language as being unnecessary, antiquated, and obsolete. We who do our programming in Assembly believe that it's the most valuable of all computer languages. There are three things that make Assembly worth while: (1) It maximizes the speed of execution of any operation we're trying to perform; (2) It can minimize the memory required to perform any given tasks; and, (3) Through Assembly we gain access to all the facilities and capabilities the computer has to offer. No other language can make those three things true at the same time.

From the programmer's viewpoint, there are two major drawbacks to Assembly: (1) It is very labor-intensive. A simple "Accept At" function may require two pages of source code to implement; (2) It requires a much more intimate knowledge of what really goes on in the computer. Such knowledge takes lots of study, and much trial and error plodding to acquire.

This series of articles is based upon years of experience, much of it painful, in exploring the capabilities and limitations of the TI-99/4A through Assembly programming. It is not designed as a beginner's course. For that, we recommend Ralph Molesworth's excellent book Introduction to Assembly Language Programming on the TI-99/4A from Steve Davis publishing.

In this first installment, we'll cover some general topics as background for the programmer who's ready to move beyond the beginner stage, but is not quite sure how to proceed. We'll cover the topics of Structure and Memory mapping. This will be very general coverage, just to give you the "feel" of thinking through your programming efforts. In later installments, we'll get into the more detailed aspects so you can become comfortable in programming with Assembly.

Structure is your servant! We say that deliberately. For many programmers the relationship becomes the wrong way around, as they slavishly "structure" far beyond any logical reason or necessity. Structure in your programming effort should help you to keep your efforts organized and focused, and in some cases will help minimize the memory required to hold your programs and data. It must not be allowed to become an end unto itself.

Perhaps a small example will help illustrate my point. In a book on PC Assembly language, the author put together a whole book of subroutines which, for the most part, could be lifted directly and used in PC programs. In some instances, however, he went overboard with structure. He gave a subroutine to place a single character on the screen. To use the subroutine, one would place the desired character's ASCII value in a register, then call the subroutine to display that character. He presented another subroutine to place a space on the screen. That subroutine simply placed ASCII 32 in the register, then called the "display a character" subroutine.

What's wrong with that process is mainly that there's twice as much "overhead" in both time and memory usage to print a space that way. The main program could put any character, including a space, in the register, then call the "display character" subroutine, rather than involve two levels of subroutine to perform the same function.

That kind of thinking is rampant in the PC Community, and is one of the reasons PC owners need MEGABYTES of memory to run commercial software packages. On the TI, with its limited memory capacity, we can't afford that kind of thinking. Again, structure is useful only so long as it serves the programmer.

I'll cite just one other example of structure gone amok, from a TI Basic program I once examined. (I won't name the program or the author.) This program used a menu selection to execute its functions. Each function was organized as a subroutine. Not one of those subroutines was called from more than one place in the main program. A simple ON-GOTO to branch directly to the desired section of the code would have done nicely, with a GOTO at the end of each function to return to the menu. In later installments of this series we'll show an efficient and effective way to perform branching from a menu-select situation, using an Assembly version of the ON-GOTO function.

So how does one sensibly apply structure without going overboard? There are two approaches which we use here at Harrison in combination. They're called Top Down and Bottom Up. From the Top Down, we recommend that some kind of overall flow chart be constructed early in the "thinking" stage of the program. For many programmers, it will help to actually draw a chart of the flow through the program's major functions. In some cases, a physical chart won't be required, but there should be at least a mental image of what the major functions are and how they should relate to one another. On occasion in my programming experience, I've ignored my own advice on this matter, and in all such cases have gone through endless agonizing revisions and re-writings of code because I omitted that first step. Once the major functions are identified, the Top Down approach proceeds to break those into smaller and smaller subdivisions of what needs to be done. From this a pattern will emerge, showing that many places in the main stream program will need the same primitive operations performed. This is where the idea of subroutines becomes a powerful tool, and it's also where the Bottom-Up idea can be useful.

In Bottom-Up programming, we start with simple functions, such as getting keystrokes from the keyboard, or placing characters on the screen, then build a program structure to optimize the use of these "primitive" tools.

Good programs need the influence of both these approaches at the same time.

Once the overall structure is broken down a couple of levels, we should have a clear view of what kinds of subroutines we'll need, and how to use them in building upward to bigger structures like menu drivers, input screens, and so on. Experienced Assembly programmers usually have a stable of existing subroutines developed as part of other efforts, so they can use those, usually with minor modifications, in the new program. In future articles, we'll present actual source code for subroutines we've found useful.

Memory is your Master! Now let's move on to the subject of Memory. There isn't much, so we must be careful how we use it. That starts with a knowledge of what we can use. There are two major blocks of memory available to the Assembly programmer. In Low Memory, from >2000 thru >3FFF, there are about 6K bytes that we can safely use, reserving the space at the beginning for the E/A utilities, and space at the end for the REF/DEF table.

In High Memory, there is lots of space, about 24K bytes from >A000 through >FFE6. In a normal Option 3 E/A program, only this 24K byte section will be open for your use as program storage. There are ways to make effective use of the low memory part as well as the high memory part, but these require techniques such as AORG, which we're not ready to cover just yet. Just to give you a hint, virtually every program we write here at Harrison involves use of AORG to give us maximum use of the available memory.

One frequently overlooked memory resource is the memory associated with the Video Display Processor, also known as VDP RAM. This can't be used directly for executable code, but can be used for a kind of "auxilliary" data storage. In most modes of VDP operation, there are about 10K bytes of VDP RAM that can be safely used to stash data.

In this series of articles, we'll show many techniques for saving memory in performing various functions.

In our next article, we'll start from the bottom up with some primitive subroutines that we've used. Along with that, we'll show the techniques for minimizing use of memory and maximizing speed of execution. When the series is done, we'll offer the whole series on disk as D/V 80 files to make them easier to access.

~~~~~

#### DISK LIBRARY REPORT:

The disk library will copy disks for a copying fee of one pound per disk, plus a one pound per package handling charge (any number of disks). If you do not wish to send blank disks, there is also a charge of one pound for disks supplied by the library.

Please make cheques payable to S Shaw.  
a complete library catalogue please send four disks and return postage. The catalogue is in text files (Display Variable 80) to be read with TI Writer etc. There are also a couple of simple utilities to display the catalogue contents to screen (Program file "LOAD" in the first two disks).

#### NEW DISKS received since June 1992:

>CARD FILE EDITOR... subject to disk space have a card index with up to 26 cards for each of up to 26 headings (say A/Z!). Uses DV80 files.

>DISK MANAGER 1000 V 5 by Caron, Romans, Mathis. Tidies up Vn 3.5. Please specify version number!!!!

>EXTENDED BUSINESS GRAPHS formerly from Great Lakes Software. You key in data and it plots a graph. Nothing too sophisticated.

>EZ KEYS. Up to 55 keys to generate macros of over 500 characters, including all the control keys. Interrupt driven so can be used while programming or running an XB program.

>SPEECH SET. A collection of demo and utility programs for speech with XB,

>VIDEOTAPES of the Lima 1992 conference are available on hire for five pounds per tape per week, deposit 15 pounds per tape.

>AUDIO CASSETTES available to buy of TI music and Barry Traver speaking at Lima. Details in library catalogue.

>RAG WRITER is now into Version 5.0 - this is Art Green's XB loading version of TI Writer, and if you only want a word processe it is a good alternative to the complexities of Funlweb. If you use the Formatter a lot, it is the best. New commands include the ability to view a second file, pass file names, and load a directory INTO your text file!!

[---> continued--->]

[disk library continued]

>MONTE CARLO Vn 4.3, over 240 sectors of XB game! (well, 32 sectors are machine code), this is the Monte Carlo roulette wheel with authentic play and bets. By Bob Guellnitz.

>REMIND ME Vn 1.2 by John Johnson, as reviewed in TI\*MES issue #31. A personal calendar program. Nicely done, especially the search routine.

>PEB for short- in full- Peripheral Expansion System Theory of Operation & Technical Training Manual. Phew! THREE DISKS of text. This is the DESIGNERS book and full of lots of interesting data, references to the 99/4B and 99/4X and much more. Covers the PE Box, the disk controller, 32k ram, Pascal card and RS232 card. Not such a dry style as it might sound!

>MINIBASE2 by Clulow and Romer. For Editor Assembler OR mini memory module only. Does not work from XB. Routines to allow you in a TI Basic program to operate a small database, based on a maximum of 300 records. As many fixed length fields as you want in each record, but the maximum characters is 80 per record. Move data between disk/memory and program/memory. Supports a 2 level ascending sort.

>Multiplan 4.03 - minor upgrade from 4.00, fixes minor non fatal bugs with Transfer Load command. MULTIPLAN MODULE REQUIRED.

>UTIL33: A loader for Myarc Extended Basic, requires a Myarc ram card plus ram at >6000 and an Editor Assembler prom. (eg Supercart). A useful backup to the no-longer made Myarc module. Requires Myarc XB on disk also (ask!). PLATO interpreter- requires ram at >6000 and ed/as prom. Also requires PLATO data disks- anyone have any??? Plus SPLITJOIN, a utility to split huge DV80 files OR make a huge DV80 file, just by writing file descriptors- the text on disk does not move!

>OFFICIAL SCOTT ADAMS ADVENTURE HINT BOOK, text on disk. Unlike the book the hints are NOT coded but appear in plain English. Covers all official SA releases.

>UTIL 32. BACKSIDE - formats the back of a disk ONLY, making your SS disks DS WITHOUT interfering with existing contents!!! and MANDELZOOM by Thomas Kneisal, 1988, a really fast Mandelbrot plotter which also uses TI Artist format picture saves.

>SIGNS: A series of disks which have ready made signs to print out- the format is TI Artist picture, can be printed with JBM103, The Missing Link, MAX-RLE, TI Artist etc. Mostly of a humorous nature.  
SIGNS 1A = 14 signs incl Grant Me Patience Lord....Hurry.  
SIGNS 1B = 10 signs (some more than one file) incl Forget It, the Answer is NO.  
SIGNS 2 = 14 signs incl Dont you dare put another thing on this desk!  
SIGNS 3A = 14 signs incl Life is like a piano...  
SIGNS 3B = 14 signs incl Do you know what you are doing?

>GEORGE 9. Pyramid Puzzler (multiplication from Scott Foresman) and Soundtrack Trolley (previously for MBX). And a filler, Honey Hunt.

>BORDERS (TWO DISKS) -for TI Artist, font files, using letters WERSDZXC to form the portions of a frame- as seen in the Signs disks above! 47 different borders.

>FONTS (TWO DISKS)- for TI Artist. 10 fonts plus another 25 borders.

58 [end of new disks. oldies follow]

[disk library-oldies]

Here are some educational items available from the disk library:

GEORGE'S-1 : ALIEN ADDITION, ALLIGATOR MIX , DEMOLITION DIVISION, DRAGON MIX , METEOR MULTIPLY, MIND CHALLENGERS 1 , MINUS MISSION .

GEORGE'S-5 : FRACTIONS 2 , FROG JUMP, NUMBER BOWLING.

GEORGES 6: Scott Foresman Division. Milliken Division. Number Readiness.

ED1-TIB : HAPPYMATH ; MATHFLASH ; QWERTY ; SUMSAREFUN ; TAKEAWAY ; TIMETUTOR

~~~~~

TEXT ON DISK: (Dis/Var 80 files)

====

>GETTING STARTED WITH THE TI99/4A by Stephen Shaw. The text of the book! on THREE disks.

> ASSEMBLY LANGUAGE TUTORIAL by Mack McCormick, regular MICROpendium author. TWO DISKS with examples. From novice up.

>GPL MANUAL. Original TI info on GPL, edited down a little by Art Green. FOUR DISKS. There are some GPL devices around, and you may find it interesting anyway!

>MULTIPLAN TUTORIAL (TWO DISKS) taken from an excellent SAMS book, by Herbert Schlesinger, who has made those little alterations the TI system requires. A first class resource for all TI Multiplan users. I would consider this your second reference after the TI Manual, and it is an excellent tutorial. Highly recommended.

THE BIBLE:

>The COMPLETE King James BIBLE on disk. Lots of disks. Each chapter in its own file. Each book archived into one file. Fully archived, occupies 13 double sided disks. Some books are over 400 sectors archived! If you dont want the lot in DS archived format, write and ask! Also included are utilities to: Print a verse at a time; to search for words in single verses- eg list every verse with ANGEL in; OR list every verse which has both FISHES and either MANY or FULL. Nifty! Plus for TE2 a program to read the verses to you.

>TI MANUAL 1. Text from TI, 1980/Vn 2 1983: Functional specs for the Basic user of the disk system, GPL interface specs for the disk system and most interesting notes on hardware debugging the speech synth. A must for hardware hackers and, as official documents, an interesting and readable read.

>TI MANUAL 2. More text from TI, this is for machine code programmers, being an official software specification for the disk peripheral.

>Timeline by Bill Gaskill- a specialist database listing the events important to the TI99/4A from 1979 to 1989 with UK additions by S Shaw. A sort of potted history, month by month.

>TI WRITER TUTORIAL. Articles by S Shaw, Dick Altman and Tom Kennedy! Largely replaces the TI Writer manual - and even if you have the manual you may find the restatement useful.

>CURRENT LIBRARY LISTING: RECORDED FREE, JUST SEND FOUR BLANK DISKS AND RETURN POST AND PACKING.

## TIPS FROM THE TIGERCUB

### NO #4

Tigercub Software  
156 Collingwood Ave.  
Columbus, OH 43213  
(614)235-3545

Want to fix your program so no one can run it without your permission? If you are in protected Extended Basic, it's easy, of course, Just:  
100 INPUT "PASTWORD? ":P\$  
110 IF P\$<>"POPSICLE" THEN 100

And if they don't type in "popsicle", or whatever password you choose, the program won't run. But if the program is in Basic, or unprotected, that method won't work with anyone who knows anything about computers - they will simply list the program and find the password. So, let's get a little more tricky. Hold down the CTRL key while you are typing POPSICLE (or whatever word you want) in line 110.

The line will then read 110 if PS<>" Then 100 (or, you may get some random bits of graphics instead of blanks) - but you can run the program by holding down the CTRL key while you enter your password.

This is still far from foolproof, because anybody can simply delete those two lines. So:

```
120 FOR J=1 TO LEN(P$)
130 CODE=CODE+ASC(SEG$(P$,J)
)
140 NEXT J
141 PRINT CODE
142 BREAK
```

Now run the program, type in your password with the CTRL key depressed. The number that prints out will be the sum of the ASCII codes of all the letters of the password as typed with the CTRL key down.

Delete lines

141 and 142, then find the line in your program where a value is assigned to a variable name- preferably a very essential variable with a value that is not readily deducted from the program logic. Suppose the line is 1210 Y=64.

If you used POPSICLE as your password, the value of CODE is 607. 64 multiplied by 607 is 38848, so let's change line 1210 to read Y=38848/CODE. I'm sure that the whiz kids can figure out a way to break that, but it won't be easy!

One of the newsletters published a routine to line up the decimals vertically in a column of figures. The algorithm was 6 lines long, took 760 bytes, and would not handle mixed positive and negative numbers. Another published routine was shorter but could not handle mixed whole numbers and decimals. I thought I could do better than that, and came up with this 88-byte one-liner which columnizes both positive and negative numbers:

```
PRINT TAB(C-LEN(STR$(INT(ABS(X))))-(X<1)+(X<0)+((X<0)*(X>-1)));X . C is the column selected for the decimal, an X is the number to be columnized.
```

I was rather proud of myself, until I received the 4A FORUM of the Central Iowa 99/4A User's Group, in which John "Jeb" Hamilton had written this little jewel of a one-liner which does the same thing in only 64 bytes - PRINT TAB(C-POS(STR\$(X)&".",",",1)-(X<0));X

There are many ways to skin the cat!!

## TIPS FROM THE TIGERCUB NO #5

100 REM - TIGERCUB WIPE FROM

```

OUTSIDE IN
110 CALL HCHAR(1,1,90,768)
120 FOR R=1 TO 12
130 CALL HCHAR(R,R,32,34-(R*
2))
140 CALL HCHAR(25-R,R,32,34-
(R*2))
150 CALL VCHAR(R+1,R,32,26-(
R*2))
160 CALL VCHAR(R+1,33-R,32,2
6-(R*2))
170 NEXT R

```

```

100 REM TIGERCUB 3-D SPRITE
DEMO
110 CALL CLEAR
120 CH$=RPT$("FF",32)
130 ($;"FFB18181818181818181
8181818181818181818181818181
0101010101010101010101010101
AR(40.X$)
140 CALL CHAR(36,CH$)
150 FOR X=2 TO 26 STEP 2
160 CALL SPRITE(#X,36,X/2+1+
ABS(X>13),X*6,40+X*6):: CALL
MAGNIFY(4):: NEXT X
170 A=8 :: B=48 :: CALL SPRI
TE(#1,40,2,A,B):: CALL MOTIO
N(#1,0,15):: FOR D=1 TO 350
:: NEXT D :: CALL POSITION(#
1,A,B)
180 FOR X=3 TO 23 STEP 2 ::
CALL SPRITE(#X,40,2,A,B):: C
ALL DELSPRITE(#X-2):: CALL M
OTION(#X,0,-15):: FOR D=1 TO
400-4*X :: NEXT D
190 CALL MOTION(#X,15,0):: F
OR D=1 TO 39 :: NEXT D :: CA
LL MOTION(#X,0,15):: FOR D=1
TO 400 :: NEXT D :: CALL POS
ITION(#X,A,B)
200 NEXT X :: CALL DELSPRITE
(#23):: GOTO 170

```

#### TIPS FOR BEGINNERS

If your white text on a black screen looks blurry, try grey on black.

If your colored graphics on a black screen look pale or colorless, try this:

```

100 CALL CLEAR
110 CALL COLOR(1,2,2)
120 CALL SCREEN(16)
130 CALL VCHAR(1,31,1,96)

```

Now put on your colored graphics and see the difference! The same trick will give you that professional looking bordered screen for

your text:

```

100 CALL CLEAR
110 CALL SCREEN(5)
120 CALL VCHAR(1,31,1,96)
130 FOR SET=1 TO 12
140 CALL COLOR(SET,2,16)
150 NEXT SET

```

Now put your text on the screen, with a blank in the 1st and 28th spaces of each line. This border is not affected by scrolling but will be erased by CALL CLEAR so clear the screen with CALL VCHAR(1,3,32,672)

This routine will print key input on the screen in position corresponding to its keyboard position:

```

100 CALL CLEAR
110 KEY$="1234567890=QWERTYU
IOP/ASDFGHJKL;"&CHR$(13)&"ZX
CVBNM,."
120 CALL KEY(3,K,ST)
130 IF ST=0 THEN 120
140 X=POS(KEY$,CHR$(K),1)
150 Y=ABS(X>11)+ABS(X>22)+AB
S(X>33)+1
160 R=Y*3
170 C=((X-ABS(Y>1))*(Y-1)*11)
*2)+4+Y
180 CALL HCHAR(R,C,K)
190 GOTO 120

```

Does your sorting routine allow you to add another name without resorting the whole list?

100 REM - SHORT SHOE HORN by Jim Peterson, Tigercub Software

```

110 REM - This routine will
insert a name into its prop-
er place in a presorted list
of N number of M$(N) names.
120 REM - Takes 12 to 14
seconds for a list of 500
130 INPUT "ADD NAME":ADD$
140 J=0
150 J=J+1
160 IF J>N THEN 230
170 IF ADD$>M$(J) THEN 150
180 FOR L=N+1 TO J STEP -1
190 M$(L)=M$(L-1)
200 NEXT L
210 M$(J)=ADD$
220 GOTO 240
230 M$(N+1)=ADD$
240 N=N+1

```

If you have the memory

expansion and are sorting long lists, you might find this faster:

```
100 REM - LONG SHOEHORN by
Jim Peterson
110 INPUT "ADD NAME":ADD$
120 LEAST=0
130 MOST=N
140 IF ADD$>N$(1) THEN 170
150 D=0
160 GOTO 280
170 IF ADD$<M$(N) THEN 200
180 M$(N+1)=ADD$
190 GOTO 320
200 D=INT((MCST-LEAST)/2)
210 IF (ADD$>M$(D))*(ADD$<M$(D+1)) THEN 280
220 IF ADD$>M$(D) THEN 250
230 MOST=D
240 GOTO 200
250 LEAST=D
260 D=D+INT((MOST-LEAST)/2)
270 GOTO 210
280 FOR J=N+1 TO D+2 STEP -1
290 M$(J)=M$(J-!)
300 NEXT J
310 M$(D+1)=ADD$
320 N=N+1
```

Pat Saturn, Editor of the Spirit of 99 newsletter of the Central Ohio Ninety-Niners, gave me this trick for making the computer go absolutely crazy. Type 100 REM and then hold down the CTRL and U while the computer runs through 4 lines and stops; or in Extended Basic, type 100 ! and do the same for 5 lines. Enter. Type 100 again, press FCTN and down-arrow. When the computer finishes printing RANDOMIZE, start pushing any keys and see what happens.

#### MORE TIPS FOR BEGINNERS:

To get the computer to read the CALL KEY input as upper case letters, even if the Alpha Lock is up, just use key-unit 3 :CALL KEY(3,KST)

To get the computer to hold 24 lines of text on the screen without jumping the first line off the top -

just put a semi-colon after the 24th line.

You don't have to RESTORE anything with the RESTORE statement. In other words, you don't have to read a DATA statement before you can RESTORE it. You can write your program to optionally or randomly RESTORE any one of your DATA statements and thus to begin reading DATA from any one of the DATA statements.

If you have the Extended Basic module, why not leave it plugged in and select the Extended Basic option even when you are programming in Basic? This will allow you to type 5 lines on a line number, which will still run in Basic (unless you put too many short items in a DATA statement), and the Extended Basic option will accept input of your program lines much faster, especially when the program gets long.

It also accepts changes and deletions much more quickly, and is especially useful when you want to delete a large number of lines. It will quickly tell you how much memory you have left with the SIZE command (but you'll have more in Basic) and will bring your rejected input back to the screen for correction, with FCTN B. It will also run your program if you stay away from character sets 15 and 16, and watch those double colons.

#### TIPS FROM THE TIGERCUB NO #6

```
100 CALL CLEAR
110 PRINT " THE TIGERCUB'S
CHRISTMAS": :TAB(11);"ORNAME
NT": : "by Jim Peterson": :
120 PRINT " Use arrow keys a
nd W,R,Z": "and C keys to mov
```



```

e cursor,": "Then press any O
THER key to"
130 PRINT "print design, or
Enter to": "skip. Use both sh
ifted and": "unshifted letter
s and punct-"
140 PRINT "uation, also type
any letter": "with CTRL held
down."
150 PRINT " Use FCTN 1 to ch
ange pat-": "tern of last cha
racter prin-": "ted, or FCTN
2 to change its"
160 PRINT "color, or FCTN 3
to change": "all the colors,
or FCTN 7": "to slowly change
all the"
170 PRINT "patterns - but st
ay away": "from FCTN 4 and FC
nt = !": " Press any
key"
180 CALL KEY(0,K,ST)
190 IF ST=0 THEN 180
200 DIM A$(16)
210 CALL CLEAR
220 FOR SET=1 TO 16
230 CALL COLOR(SET,5,16)
240 NEXT SET
250 CALL VCHAR(1,31,1,96)
260 CALL SCREEN(2)
270 FOR CH=33 TO 159
280 PRINT CHR$(CH); " ";
290 Z=Z+1
300 IF Z/14<>INT(Z/14) THEN 3
20
310 PRINT : :
320 NEXT CH
330 GOSUB 630
340 GOSUB 940
350 CALL VCHAR(1,3,32,672)
360 CALL VCHAR(1,29,1,192)
370 R=6
380 C=11
390 CALL KEY(3,K,ST)
400 IF ST=0 THEN 390
410 IF K<8 THEN 820
420 ON POS("WERDCXZS",CHR$(K
),1)+1 GOTO 390,490,470,460,
440,430,530,520,500
430 R=R+ABS(R<12)
440 C=C+ABS(C<16)
450 GOTO 540
460 C=C+ABS(C<16)
470 R=R-ABS(R>1)
480 GOTO 540
490 R=R-ABS(R>1)
500 C=C-ABS(C>5)
510 GOTO 540
520 C=C-ABS(C>5)
530 R=R+ABS(R<12)
540 CALL HCHAR(R,C,42)
550 CALL KEY(5,CH,ST)
560 IF ST=0 THEN 550
570 IF (CH=87)+(CH=69)+(CH=8
2)+(CH=83)+(CH=68)+(CH=90)+(
CH=88)+(CH=67) THEN 550
580 CALL HCHAR(R,C,CH)
590 CALL HCHAR(25-R,C,CH)
600 CALL HCHAR(R,33-C,CH)
610 CALL HCHAR(25-R,33-C,CH)
620 GOTO 390
630 DATA 00,18,24,30,42,5A,6
6,7E,81,99,A5,BD,C3,DB,E7,FF
640 FOR J=1 TO 16
650 READ A$(J)
660 NEXT J
670 FOR CH=33 TO 159
680 GOSUB 700
690 GOTO 800
700 FOR L=1 TO 4
710 RANDOMIZE
720 X=INT(16*RND+1)
730 B$=B$&A$(X)
740 C$=A$(X)&C$
750 NEXT L
760 CALL CHAR(CH,B$&C$)
770 B$=NUL$
780 C$=NUL$
790 RETURN
800 NEXT CH
810 RETURN
820 ON K+1 GOTO 390,830,390,
850,830,390,390,820
830 GOSUB 670
840 GOTO 390
850 IF (CH<33)+(CH>159) THEN
390
860 GOSUB 700
870 GOTO 390
880 SET=INT(CH/8)-3
890 IF SET<2 THEN 390
900 GOSUB 980
910 GOTO 390
920 GOSUB 940
930 GOTO 390
940 FOR SET=2 TO 16
950 RANDOMIZE
960 GOSUB 980
970 GOTO 1030
980 X=INT(15*RND+2)
990 Y=INT(15*RND+2)
1000 IF Y=X THEN 990
1010 CALL COLOR(SET,X,Y)
1020 RETURN
1030 NEXT SET
1040 RETURN

```

This routine contains many ideas which may be educational to beginning program-

ming. Lines 110-170 will show how to neatly format the text on the screen. Lines 180-190 hold the text on the screen until any key is pressed; this is much better than a delay loop, because different people read at much different speeds.

Lines 220-260 create a white screen with a black border, which makes colored graphics stand out much more brightly. After a CALL CLEAR, the screen is actually filled, except for a strip at top and bottom, with 24 rows of 32 columns of the blank space-bar character.

The normal screen color is light green (4) in Basic or cyan (8) in Extended Basic, and the normal color of all the character sets is black (2) on transparent (1); The blank character has no foreground color and the transparent background color let's the screen show through.

Line 230 changes all the sets to dark blue on white, so the blank space character becomes white. The strip at the top and bottom of the screen is filled with the edge character, ASCII 31 which is transparent. Line 250 prints two columns of the edge character at the right edge, then wraps around to print two more columns at the left edge. The black screen color, called in Line 260, can now show through the transparent edge character on all four sides.

Lines 270-280 put all the printable characters on the screen, to give you something to watch while the program is being initialized. The " " in Line 280 puts a blank space after each character, and the semi-

colon keeps the print on the same line until it is full. Since 14 characters can be double-spaced on a 28-column print line, a print row is skipped whenever the value of Z becomes divisible by 14.

Now the program goes to the routine in Lines 630-810 to redefine every keyboard character to a symmetrical ornamental pattern. The DATA statement in Line 630 contains 16 pairs of hexadecimal numbers which are mirror images of each other; Look at the chart on page 109 of your "Beginner's Basic" to see what I mean. Lines 640-660 read these into 16 subscripts of A\$; thus the dimension statement in Line 200 was necessary.

Next, Lines 670--800 randomly combine these pairs to form symmetrical characters. Lines 680, 690 and 790 are only so that we can later re-enter the middle of this routine. Lines 700-750 loop 4 times to pick a random value of 1 to 16 for X, to randomly select those mirrored pairs.

B\$ is built up to 4 pairs of these numbers, forming the upper half of a character which will have left/right symmetry. At the same time, C\$ is built up of these same 4 pairs in reverse.

Finally, Line 760 combines B\$ and C\$ to redefine a character which is also symmetrical top/bottom. Since B\$ and C\$ were built up by adding onto themselves Lines 770-780 must cancel them out before they are used again in the next loop of CH. NUL\$ is not a magic word, it is just a string variable name which is never given a value, there-

fore represents an erased value.

The program returns to Line 340 and then to 940, where each character set from 2 to 16 is given a different color combination selected randomly. Line 1000 insures that the foreground and background color will not be the same. Again, Lines 960-970 and 1020 are just so that we can later re-enter this routine in the middle.

Since a CALL CLEAR would erase our screen borders of ASCII 1, the screen is cleared by the CALL VCHAR wipe in Line 350, and Line 360 then widens this transparent border to leave only a 24x24 square.

Lines 390-540 are a very compact and efficient routine to move the cursor with the arrow keys and diagonal keys, without going beyond predetermined borders. We don't have room to explain it here, so we will give a more complete version in a future Tips.

OUT OF MEMORY,

HAPPY HACKIN'  
Jim Peterson  
TIPS FROM THE TIGERCUB

#27  
Copyright 1985

TIGERCUB SOFTWARE  
156 Collingwood Ave.  
Columbus, OH 43213

Now, here is the new, and final, version of the Tigercub Menu Loader. (the following is actually the much later and really last version 10. - Ed.)

100 !by A. Kludge/M. Gordon/  
T. Boisseau/J. Peterson/G. S  
teffen/etc. Version #10, 9/86  
110 CALL PEEK(8198,A):: IF A  
<>170 THEN CALL INIT  
120 OPTION BASE 1 :: DIM PG\$

```
(127),V(127,3),LL$(400):: DN
$="1" :: CALL LOAD(-31806,16
):: ON ERROR 130 :: GOTD 160
130 DISPLAY AT(12,9)ERASE AL
L:"I/O ERROR" :: RUN 100
140 @,@$,@@, A,A$,B,B$,C,CF$,
D$,DN$,D1$,D2$,E,F,FLAG,I,J,
K,KD,KK,LL$(),LX,M,M$,N$,N,D
FLAG,P,PFLAG,PG$(),PP,PP$,Q$
,S,ST,T$(),TT,VT,V(,),W$,X,X
X,Y,K2,S2
150 CALL LINK :: CALL PEEK :
: CALL KEY :: CALL SCREEN ::
CALL COLOR :: CALL CLEAR ::
CALL VCHAR :: CALL SOUND ::
CALL CHAR :: !@P-
160 CALL CLEAR :: CALL LOAD(
8196,63,248):: CALL LOAD(163
76,67,85,82,83,79,82,48,8)
170 CALL LOAD(12288,129,195,
126,165,129,153,102,60)
180 CALL LOAD(12296,2,0,3,24
0,2,1,48,0,2,2,0,8,4,32,32,3
6,4,91):: CALL LINK("CURSOR"
)
190 CALL CLEAR :: FOR S=1 TO
14 :: CALL COLOR(S,7,16)::
NEXT S :: CALL COLOR(0,2,16)
:: CALL CHAR(48,"003A444C546
444BB")
200 T$(1)="d/f" :: T$(2)="d/
v" :: T$(3)="i/f" :: T$(4)="
i/v" :: T$(5)="pro" :: ON WA
RNING NEXT
210 CALL SCREEN(5):: CALL VC
HAR(1,31,1,96):: DISPLAY AT(
1,4):"TIGERCUB MENU LOADER"
220 DISPLAY AT(12,6):"DRIVE?
"&DN$ :: ACCEPT AT(12,13)SI
ZE(-1)VALIDATE("1234"):DN$ :
: D$="DSK"&DN$&". "
230 OPEN #1:D$,INPUT ,RELATI
VE,INTERNAL :: INPUT #1:N$,A
,J,K :: DISPLAY AT(1,2)SIZE(
27):SEG$(D$,1,4)&" - Disknam
e= "&N$;
240 DISPLAY AT(2,2):"Availab
le=";K;"Used=";J-K:" Prog Fi
lename Size Type": "---- ---
-----" :: I,V
T=0 :: TT=J-K
250 FOR X=1 TO 127 :: IF X/2
0<>INT(X/20)THEN 280
260 DISPLAY AT(24,1):"Choice
? Enter for more 0" :: ACCEP
T AT(24,24)VALIDATE(DIGIT)SI
ZE(-3):K :: IF K=0 THEN 270
:: IF K>0 AND K<N+1 THEN 500
ELSE 260
270 X=1
```

```

280 N=N+1 :: INPUT #1:PG$(N)
,V(N,1),V(N,3),V(N,2):: B$=S
TR$(V(N,2)):: IF B$="0" THEN
B$=""
290 IF LEN(PG$(N))=0 THEN 33
0
300 DISPLAY AT(X+4,1):USING
"### ##### ### ## ###
#":N,PG$(N),V(N,3),T$(ABS(V(
N,1))),B$,CHR$(ABS(V(N,1)<0)
*57+32):: VT=VT+V(N,3)
310 CALL KEY(0,KK,ST):: IF S
T=0 THEN 320 :: FLAG=1 :: GO
TO 330
320 NEXT X
330 DISPLAY AT(X+4,1):USING
"### Quit":N :: DISPLAY AT(
X+5,1):USING "### Delete":N
+1
340 IF VT=TT OR FLAG=1 THEN
360
350 FOR @=1 TO 10 :: DISPLAY
AT(3,25):" " :: DISPLAY AT(
3,25):VT :: CALL SOUND(-99,1
10,0,-4,0):: NEXT @
360 IF FLAG=1 THEN 370 :: DI
SPLAY AT(X+4,13):USING "###
Print":N+2 :: DISPLAY AT(X+
5,13):USING "### Rescan":N+
3
370 DISPLAY AT(X+6,5):"Choic
e?" :: ACCEPT AT(X+6,16)SIZE
(-3)VALIDATE(DIGIT):K
380 IF FLAG=1 THEN 400
390 IF K=N+2 THEN 780 ELSE I
F K=N+3 THEN CLOSE #1 :: N=0
:: GOTO 190
400 IF K<>N AND K<>N+1 THEN
490
410 IF K=N THEN CALL CLEAR :
: CLOSE #1 :: END
420 DISPLAY AT(X+5,12)SIZE(1
2):" #?" :: ACCEPT AT(X+5,15
)SIZE(3)VALIDATE(DIGIT):KD :
: IF KD<1 OR KD>N THEN 420
430 IF V(KD,1)>0 THEN 450
440 FOR J=1 TO 10 :: DISPLAY
AT(11,1):" ":" PROTECTED -
CANNOT DELETE":" " :: DISPL
AY AT(12,1):" " :: NEXT J ::
GOTO 470
450 DISPLAY AT(X+6,1)SIZE(27
)BEEP:" Verify - Delete ";PG
$(KD);"?" :: DISPLAY AT(X+6,
28)SIZE(1):"Y" :: ACCEPT AT(
X+6,28)SIZE(-1)VALIDATE("YN"
):Q$ :: IF Q$<>"Y" THEN 470
460 DELETE D$&PG$(KD)
470 CLOSE #1
480 CALL VCHAR(1,3,32,672)::

```

```

N=0 :: X=0 :: FLAG=0 :: GOT
O 210
490 IF K<1 OR K>127 OR LEN(P
G$(K))=0 THEN 330
500 A=ABS(V(K,1)):: B=V(K,2)
:: IF A=5 OR A=4 AND V(K,2)=
254 THEN 860
510 DISPLAY AT(12,1)ERASE AL
L:"Print to ? S": "(P)rinte
r?": "(S)creen?": "(D)isk?" ::
ACCEPT AT(12,12)SIZE(-1)VAL
IDATE("PSD"):Q$ :: IF Q$="S"
THEN PP=0 :: GOTO 540
520 IF Q$="P" THEN 530 :: DI
SPLAY AT(12,1)ERASE ALL:"Fil
ename? DSK" :: ACCEPT AT(12,
14):CF$ :: CF$="DSK"&CF$ ::
C=(SEG$(D$,4,1)=SEG$(CF$,4,1
)):: GOTO 540
530 CALL CLEAR :: PP=3 :: GO
SUB 840
540 CALL CLEAR :: CALL SCREE
N(16)!
550 @$=D$&PG$(K):: IF A=1 TH
EN OPEN #2:@$,INPUT ,FIXED E
LSE IF A=2 THEN OPEN #2:@$,I
NPUT ELSE IF A=3 THEN OPEN #
2:@$,INPUT ,INTERNAL,FIXED E
LSE OPEN #2:@$,INPUT ,INTERN
AL
560 IF Q$="D" THEN 570 :: ON
A GOTO 630,630,680,680
570 XX=XX+1 :: IF A<3 THEN L
INPUT #2:LL$(XX)ELSE INPUT #
2:LL$(XX)
580 LX=LX+LEN(LL$(XX)):: IF
EOF(2)<>1 AND LX<8000 AND XX
<401 THEN GOTO 570 :: ELSE I
F C=-1 THEN D1$="master" ::
D2$="copy" :: GOSUB 770
590 IF OFLAG=1 THEN GOTO 610
ELSE OFLAG=1
600 IF A=1 THEN OPEN #4:CF$,
OUTPUT,FIXED B ELSE IF A=2 T
HEN OPEN #4:CF$,OUTPUT,VARIA
BLE B ELSE IF A=3 THEN OPEN
#4:CF$,OUTPUT,INTERNAL,FIXED
B ELSE OPEN #4:CF$,OUTPUT,I
NTERNAL,VARIABLE B
610 FOR J=1 TO XX :: PRINT #
4:LL$(J):: NEXT J :: XX,LX=0
:: IF EOF(2)THEN CLOSE #4 :
: C,OFLAG=0 :: GOTO 660
620 IF C=-1 THEN D1$="copy"
:: D2$="master" :: GOSUB 770
:: GOTO 570 ELSE GOTO 570
630 LINPUT #2:W$ :: PRINT #P
P:W$ :: IF EOF(2)THEN 660
640 CALL KEY(0,K,S):: IF S=0
THEN 630

```

```

650 CALL KEY(0,K2,S2):: IF S
2<1 THEN 650 ELSE 630
660 CLOSE #1 :: CLOSE #2 ::
PRINT " >>>press any key<<
<"
670 CALL KEY(0,K,ST):: IF ST
<1 THEN 670 ELSE 480
680 IF EOF(2)=1 THEN 660 ::
INPUT #2:M$ :: IF LEN(M$)=8
THEN 700
690 PRINT #PP:M$ :: GOTO 750
700 FOR Y=1 TO 8 :: @@=ASC(S
EG$(M$,Y,1)):: IF @@<32 OR @
@>127 THEN 720
710 NEXT Y :: GOTO 690
720 F=1 :: E=ASC(SEG$(M$,1,1
)):: M=ASC(SEG$(M$,2,1)):: I
F E=0 AND M=0 THEN GOTO 740
ELSE IF E>128 AND M>128 THEN
F=-1 :: E=255-E :: M=256-M
730 FOR I=1 TO 6 :: M=M+(ASC
(SEG$(M$,I+2,1)))/100^I :: N
EXT I :: M=M*F*100^(E-64)
740 PRINT #PP:M
750 CALL KEY(0,K,S):: IF S=0
THEN 680
760 CALL KEY(0,K2,S2):: IF S
2<1 THEN 760 ELSE 680
770 DISPLAY AT(12,1)ERASE AL
L:"Remove ";D1$;" disk": : "I
nsert ";D2$;" disk": : "Press
ENTER" :: CALL KEY(0,K,ST):
: IF ST=0 THEN 770 ELSE CALL
CLEAR :: RETURN
780 GOSUB 840 :: PRINT #3:SE
G$(D$,1,4)&" - Diskname= "&N
$
790 PRINT #3:RPT$("*",28):"A
vailable=";358-VT;"Used=";VT
:RPT$("~",28)
800 PRINT #3:"FILENAME SIZE
TYPE":RPT$("_",28)
810 FOR P=1 TO N-1 :: PRINT
#3:PG$(P);TAB(15);V(P,3);TAB
(20);T$(ABS(V(P,1)));TAB(25)
;V(P,2);TAB(31);CHR$(89*ABS(
V(P,1)<0)):: NEXT P
820 DISPLAY AT(12,3)ERASE AL
L:"(P) to print again": " (R
) to rescan": " (Q) to quit"
830 ACCEPT AT(15,4)VALIDATE(
"PQR")SIZE(-1)BEEP:Q$ :: IF
Q$="P" THEN 780 :: CLOSE #1
:: N=0 :: IF Q$="R" THEN 190
ELSE END
840 IF PFLAG=1 THEN RETURN E
LSE PFLAG=1 :: DISPLAY AT(24
,1):"PRINTER NAME? PID" :: A
CCEPT AT(24,15)SIZE(-14):PP$
850 OPEN #3:PP$,VARIABLE 132

```

```

:: PRINT #3:CHR$(27);"B";CH
R$(2);CHR$(27);"M";CHR$(10);
CHR$(27);"N";CHR$(6):: RETUR
N
860 CLOSE #1 :: IF SEG$(PG$(
K),LEN(PG$(K)),1)="*" THEN D
ISPLAY AT(12,1)ERASE ALL:"RE
TURN TO BASIC AND LOAD BY":
TYPING OLD ";D$&PG$(K):: STO
P
870 CALL PEEK(-31952,A,B)::
CALL PEEK(A*256+B-65534,A,B)
:: C=A*256+B-65534 :: A$=D$&
PG$(K):: CALL LOAD(C,LEN(A$
))
880 FOR I=1 TO LEN(A$):: CAL
L LOAD(C+I,ASC(SEG$(A$,I,1)
)):: NEXT I :: CALL LOAD(C+I,
0)
890 CALL VCHAR(1,3,32,672)::
CALL SCREEN(8):: FOR S=0 TO
14 :: CALL COLOR(S,2,1):: N
EXT S :: DISPLAY AT(12,2):"L
OADING ";A$
900 RUN "DSKX.1234567890"

```

This version turns off the Quit key, restarts itself rather than crashing on an I/O error, and has pre-scan for faster startup. It displays disk name, sectors available and sectors presumably used - it also totals up actual sectors used and sounds a warning if any sectors are not accounted for.

It lists up to 127 programs and files by number, filename, number of sectors, program or file type, file record length, and write-protection. It will stop for menu selection on any keypress or at the end of each screen, continuing on Enter. It will load and run any program that can run from Extended Basic, displaying its filename while loading.

If the filename ends in an asterisk, it will warn you to return to Basic. It will delete any unprotected program or file, after first requiring verification by

filename, and will inform you if the file is protected. It will read any readable file, including internal numeric, and list it to screen or printer.

It will dump a catalog of the disk to your printer, & it will offer the option of quitting or rescanning the disk or another disk. And it's free, I do not even want a freeware donation.

(the following is an improved version of the program originally published here. - Ed.)

```

100 CALL CLEAR :: RANDOMIZE
:: DISPLAY AT(3,4):"TIGERCUB
MATH PUZZLE"
105 DISPLAY AT(6,1):"Level 1
, 2, 3 or 4?" :: ACCEPT AT(6
,21)VALIDATE("1234"):L$ :: L
=VAL(L$)
106 IF L<3 THEN M$="Insert +
, -, or * (multiply)" ELSE M
$="Insert +, -, * (multiply)
or / (divide)"
110 DISPLAY AT(5,1):M$;" bet
ween the digits":" to equal
the total": "Type Q to give
up"
130 DISPLAY AT(12,1):" " ::
T,X=INT(9*RND+1):: M$=STR$(X
):: Z$=M$&" "
140 FOR J=1 TO 4 :: Y(J)=INT
(9*RND+1):: @=3+ABS(L>2):: Z
=INT(@*RND+1):: ON Z GOSUB 2
40,250,260,270 :: Z$=Z$&STR$
(Y(J))&" " :: NEXT J
150 IF L/2<>INT(L/2)AND T<>I
NT(T)THEN 130 :: Z$=Z$&" "&S
TR$(T)
160 DISPLAY AT(12,1):Z$ :: D
ISPLAY AT(18,1):" " :: DISPL
AY AT(20,1):" " :: DISPLAY A
T(22,1):" "
170 P=2 :: FOR J=1 TO 4 :: A
CCEPT AT(12,P)VALIDATE("Q+-*
/")SIZE(1):S$
180 IF S$="Q" THEN 200 ELSE
IF S$="+" THEN X=X+Y(J)ELSE
IF S$="-" THEN X=X-Y(J)ELSE
IF S$="*" THEN X=X*Y(J)ELSE
X=X/Y(J)
190 P=P+2 :: NEXT J :: IF X=
T THEN 230 :: DISPLAY AT(18,

```

```

1):"WRONG!"
200 DISPLAY AT(20,1):"ANSWER
IS ";M$
210 DISPLAY AT(22,1):"PRESS
ANY KEY"
220 CALL KEY(0,K,ST):: IF ST
<1 THEN 220 :: GOTO 130
230 DISPLAY AT(18,1):"RIGHT!
" :: GOTO 210
240 M$=M$&"+"&STR$(Y(J)):: T
=T+Y(J):: RETURN
250 M$=M$&"-"&STR$(Y(J)):: T
=T-Y(J):: RETURN
260 M$=M$&"*"&STR$(Y(J)):: T
=T*Y(J):: RETURN
270 M$=M$&"/"&STR$(Y(J)):: T
=T/Y(J):: RETURN

```

TIPS FROM THE TIGERCUB  
#28  
Copyright 1985

While programming the file reader in that menu loader, I ran into a peculiarity of the TI-99/4A that surprised most of the expert programmers whom I called for help. When you "read blind" you must read everything as a string, because attempting to read a string as numeric will crash the program. This is no problem with DISPLAY files - but when I tried it with INTERNAL files, I got the strangest garbage!

My solution (not quite fool-proof) was to identify a record as numeric if it was 8 bytes long and contained an ASCII out of printable range, and then RESTORE the file, read back to that point and re-read it as numeric. Not very efficient!

The following routine will save a numeric input in an internal file, read it back out as a string, show you the way it was saved, and then attempt to translate it back to numeric. It works for positive and negative integers or non-integers of not less than -99, but not

for less than that.

```
100 INPUT X :: OPEN #1:"DSK1
.TEST",INTERNAL,OUTPUT :: PR
INT #1:X :: CLOSE #1
110 OPEN #1:"DSK1.TEST",INTE
RNAL,INPUT :: INPUT #1:A$ ::
PRINT A$ :: CLOSE #1
120 FOR J=1 TO 8 :: PRINT AS
C(SEG$(A$,J,1)):: NEXT J
130 FOR J=1 TO 8 :: A(J)=ASC
(SEG$(A$,J,1)):: NEXT J
140 X=A(1)-63 :: IF X<73 THE
N 150
142 X=192-A(1):: N$="-" :: F
OR J=2 TO X+1 :: N$=N$&STR$(
256-A(J)):: NEXT J :: GOTO 1
60
150 FOR J=2 TO X+1 :: N$=N$&
STR$(A(J)):: NEXT J
160 IF A(J)<>0 THEN N$=N$&".
"&STR$(A(J))
170 J=J+1 :: IF A(J)<>0 THEN
N$=N$&STR$(A(J)):: GOTO 170
180 N=VAL(N$):: N$="" :: PRI
NT N :: GOTO 100
```

So, here is another Tigercub Challenge! Can you fix it? Let's HEAR from you this time!

Another problem that I ran into was in recovering from an I/O error. When ON ERROR is used to prevent crashing on such an error, the file is "ajar" - you can't close it and you can't open it. My solution was to simply RUN the program again - and this will show you how the pre-scan speeds that up.

Since then, I have learned of three other ways. The method described in the Sydney (Australia) newsletter is a bit complicated, but Irwin Hott gave me a simple solution - just increment the file number! Works fine if you don't increment it into the number of another open file on the disk. Chuck Grimes gave me an even better way - open and close anything else, even "PIO"! Example -

```
100 ON ERROR 110 :: OPEN #1:
"DSK1.TEST",OUTPUT :: ON ERR
OR STOP :: PRINT "CONTINUE P
ROGRAM" :: END
110 OPEN #1:"PIO" :: CLOSE #
1 :: PRINT "I/O ERROR":"CHEC
K DISK AND DRIVE":"THEN PRES
S ANY KEY"
120 CALL KEY(0,K,S):: IF S=0
THEN 120 ELSE 100
(this method does not work -
see better method in Tips
No. 44. - Ed.)
```

There is a reason for that ON ERROR STOP, and it's why I don't use ON ERROR if I can avoid it. When an error occurs, the program goes to the line number specified by the last open ON ERROR statement, takes whatever action is directed by that line, and RETURNS as directed. If the error was not one that you expected to happen, the results can be very confusing!

For that reason, when you set out to modify a program, the first thing you should do is delete, temporarily, all the ON ERROR statements. The next thing you should do, if the program has a routine to turn off the pre-scan, is to disable that. Otherwise, you will be driven crazy by invalid SYNTAX ERROR messages and other strange happenings.

The third thing you should do is to make a list of all the lines that a GOTO or GOSUB goes to, so you don't delete or change them. And here is a program to do just that for you -

```
100 !GO-SEARCH by Jim Peters
on searches a MERGE format f
ile, finds all line numbers
containing a jump, sorts int
o "to" line number sequence,
110 !prints "to" line number
, statement (GO, GOTO or GOS
UB) and "from" line number
120 DIM C(200):: A=1 :: GO$(
```

```

1)="GO" :: GO$(2)="GOTO" ::
GO$(3)="GOSUB"
130 INPUT "FILENAME? DSK1.":
F$
140 OPEN #1:"DSK1."&F$,INPUT
,VARIABLE 163 :: OPEN #2:"P
IO"
150 LINPUT #1:A$
160 IF POS(A$,CHR$(133),1)=0
AND POS(A$,CHR$(134),1)=0 A
ND POS(A$,CHR$(135),1)=0 THE
N 210
170 LN=ASC(SEG$(A$,1,1))*256
+ASC(SEG$(A$,2,1)):: T=133 :
: P=1
180 G$=CHR$(T):: X=POS(A$,G$
,P):: IF X=0 THEN 200 :: LRE
F=ASC(SEG$(A$,X+2,1))*256+AS
C(SEG$(A$,X+3,1))!:: PRINT #
2:LN;G$(T-132);LREF :: P=X+
1 :: GOTO 180
190 C$=STR$(LREF)&". "&STR$(L
N)&STR$(T-132):: C(A)=VAL(C$
):: A=A+1 :: P=X+1 :: GOTO 1
80
200 IF G$=CHR$(135) THEN 210
:: T=T+1 :: P=1 :: GOTO 180
210 IF EOF(1) THEN CLOSE #1 :
: GOTO 220 :: ELSE 150
220 A=A-1 :: CALL LONGSHELLN
(A,C())
230 FOR J=1 TO A :: A$=STR$(
C(J)):: X=POS(A$,".",1):: Y=
VAL(SEG$(A$,LEN(A$),1)):: A$
=SEG$(A$,1,LEN(A$)-1)
240 PRINT #2:SEG$(A$,1,X-1);
TAB(7);G$(Y);" FROM ";TAB(2
1);SEG$(A$,X+1,LEN(A$)):: NE
XT J
250 SUB LONGSHELLN(N,NN())
260 D=N
270 D=INT(D/3)+1 :: FOR I=1
TO N-D :: IF NN(I)<=NN(I+D) T
HEN 300 :: T=NN(I+D):: J=I
280 NN(J+D)=NN(J):: J=J-D ::
IF J<1 THEN 290 :: IF T<NN(
J) THEN 280
290 NN(J+D)=T
300 NEXT I
310 IF D>1 THEN 270
320 SUBEND

```

According to the User's Reference Guide that came with your computer, if you open a file without specifying INPUT, OUTPUT, UPDATE or APPEND, the computer will assume the UPDATE mode as the default

and "UPDATE files may be both read and written." The usual processing is to read a record, change it in some way, and then write the altered record back out on the file." This is a very dangerous bit of misinformation! It is true only if you are using RELATIVE files with the REC clause. In any other case, the first record you write to the file will become the record FOLLOWING the last record you read, and it will also become the LAST record in the file - any records beyond that point will be lost!

The moral of the story - get in the habit of NEVER opening a file without specifying the mode. The only way to update a sequential file is to read it ALL into an array, update it, and then write it back to the file.

I reviewed hundreds of programs, in my PD library of about 2600, in order to select some of the best to fill up the collection disks. Often they needed only a few minor changes to greatly improve them.

One frequent flaw was in interpreting the status of CALL KEY. The User's Reference Guide says that a status variable of -1 means that "the same key was pressed during the performance of CALL KEY as was pressed during the previous performance." This is misleading. It actually means that the same key is STILL BEING pressed. Try this -

```

100 DISPLAY AT(12,1)ERASE AL
L:"TYPE YOUR NAME" :: R=14 :
: C=3
110 CALL KEY(0,K,S):: IF S=0
THEN 110 :: DISPLAY AT(R,C)
:CHR$(K):: C=C+1 :: GOTO 110

```



Difficult to type without unwanted repetition of letters? Now try changing the S=0 to S<1 !

IF S<1 (if S is less than 1) means that if no key is pressed (S=0) or if the same key is still being held down (S=-1) then CALL KEY again.

Another frequent flaw is INPUT "WANT TO PLAY AGAIN? " :Q\$ :: IF Q\$<>"Y" THEN END - or, more professionally programmed, IF SEG\$(Q\$,1,1)<>"Y" THEN...., which will accept either "Y" or "YES" as a reply. The problem is still that this question is often asked at the end of a joystick game, for which the Alpha Lock will be unlocked - and a response of a lower case "y" then terminates the program!

One solution is to precede the INPUT with a dummy CALL KEY(3,K,S), which will cause any subsequent upper case CALL KEY, INPUT, LINPUT or ACCEPT AT response to be read as lower case until you turn it off with CALL KEY(5,K,S).

Here's one that does nothing except look pretty.

```
100 DISPLAY AT(3,8)ERASE ALL
:"COLORSQUARES" :: DISPLAY A
T(8,1):"Select option 1, 2 o
r 3" ! by Jim Peterson, Tigercub Software
110 CALL KEY(0,K,ST):: IF ST
=0 OR K<49 OR K>51 THEN 110
:: ON K-48 GOTO 150,120,130
120 FOR CH=38 TO 142 STEP 8
:: CALL CHAR(CH,RPT$("A55A",
4)):: NEXT CH :: GOTO 150
130 FOR CH=38 TO 142 STEP 8
:: FOR L=1 TO 4 :: RANDOMIZE
:: X$=SEG$("0018243C425A667
EB199A5BDC3DBE7FF",INT(16*RN
D+1)*2-1,2)
140 B$=B$&X$ :: C$=X$&C$ ::
NEXT L :: CALL CHAR(CH,B$&C$
):: B$,C$=NUL$ :: NEXT CH
150 CALL CLEAR :: RANDOMIZE
:: FOR SET=0-(K>49)TO 14 ::
CALL COLOR(SET,SET+2+(K>49),
```

```
SET+2):: NEXT SET
160 Y=INT(4*RND+3):: R=INT(1
2*RND+1):: R2=25-R-Y :: C=IN
T(7*RND+7):: C2=32-C-Y :: IF
K=49 THEN X=INT(14*RND+1)*8
+22 ELSE X=INT(13*RND+1)*8+3
0
170 FOR T=R TO R+Y :: CALL H
CHAR(T,C,X,Y):: CALL HCHAR(T
,C2,X,Y):: NEXT T
180 FOR T=R2 TO R2+Y :: CALL
HCHAR(T,C,X,Y):: CALL HCHAR
(T,C2,X,Y):: NEXT T :: GOTO
160
```

```
=====
90 !THIS WON'T WORK, WILL IT
?
100 DISPLAY AT(9999,9999)ERA
SE ALL:SEG$("CAN'T DO THAT!"
,1,3)&SEG$("CAN'T DO THAT!",
6,8)
=====
```

If the Tigercub Math Puzzle in Tips #27 was a bit too tough, these changes will add a couple of easier levels.

```
105 DISPLAY AT(6,1):"Level 1
, 2, 3 or 4?" :: ACCEPT AT(6
,21)VALIDATE("1234"):L$ :: L
=VAL(L$)
106 IF L<3 THEN M$="Insert +
, -, or * (multiply)" ELSE M
$="Insert +, -, * (multiply)
or / (divide)"
110 DISPLAY AT(5,1):M$;" bet
ween the digits": " to equal
the total": "Type Q to give
up"
120 ! **DELETED LINE **
130 DISPLAY AT(12,1):" " ::
T,X=INT(9*RND+1):: M$=STR$(X
):: Z$=M$&" "
140 FOR J=1 TO 4 :: Y(J)=INT
(9*RND+1):: @=3+ABS(L>2):: Z
=INT(@*RND+1):: ON Z GOSUB 2
40,250,260,270 :: Z$=Z$&STR$
(Y(J))&" " :: NEXT J
150 IF L/2<>INT(L/2)AND T<>I
NT(T)THEN 130 :: Z$=Z$&" "&S
TR$(T)
```

MEMORY FULL

Jim Peterson

TIPS FROM THE TIGERCUB

TIPS FROM THE TIGERCUB

No. 66

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 each. I am out of printed documentation so it will be supplied on disk.

My TI-PD library now has well 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 and the latest supplement is available for \$1 which is deductible from the first order.

Re TI\*MES #37- page 43:

In Tips #65 I said that the TI could calculate to 14-digit accuracy, rather than the 8-digit accuracy of a PC. Actually the number in memory is calculated to 13- or 14-digit accuracy, depending on the number, but is rounded to 10 digits on the screen display, or shown in exponential notation if the number is extremely large or extremely small. If you want to see the complete number, this routine will show the normal screen display and the full number in memory. To see the complete range of numbers our little TI can handle, try inputting -  
-9.999999999999999E127 and  
-1.000000000000000E-128 and  
1.000000000000000E-128 and  
9.999999999999999E127.

```
100 OPEN #1:"DSK1.INT2",INTERNAL,RELATIVE,UPDATE
110 INPUT N
120 PRINT #1,REC 1:N
130 INPUT #1,REC 1:N$
140 X=ASC(SEG$(N$,1,1)): Y=ASC(SEG$(N$,2,1)): IF Y>99 THEN Y=256-Y :: N$=SEG$(N$,1,1)&CHR$(Y)&SEG$(N$,3,255)
150 FOR J=2 TO LEN(N$):: X$=STR$(ASC(SEG$(N$,J,1)))
160 IF ASC(SEG$(N$,J,1))<10 THEN X$="0"&X$
170 P$=P$&X$ :: NEXT J
180 IF X<63 THEN Y$="."&RPT$("00",63-X)&P$ :: GOTO 230
190 IF X>191 THEN Y$="."&RPT$("00",X-192)&P$ :: GOTO 230
200 IF X>185 THEN Y$=SEG$(P$,1,14-(X-185)*2)&"."&SEG$(P$,14-(X-185)*2+1,255):: GOTO 230
210 Y$=SEG$(P$,1,(X-63)*2)&"."&SEG$(P$, (X-63)*2+1,255)
220 IF ASC(Y$)=48 THEN Y$=SEG$(Y$,2,255)
230 IF N<0 THEN Y$="-"&Y$
240 PRINT TAB(2);N :: PRINT TAB(3);Y$ :: P$="" :: GOTO 10
```

But even the smart little TI has its limits. Try this-  
 $X=2/3-1/3-1/3 :: PRINT X$ .  
See the TI User's Reference Guide page III-13 for the explanation of all this.

Solving an equation such as  $X^X/X-X=A$  would be very difficult to solve by mathematical means, but our computer can find the answer quickly by systematic trial and error, to the 14-point limit of its accuracy.

```
100 DISPLAY AT(3,1)ERASE ALL
:"This program will solve even the most difficult equation with one variable."
110 DISPLAY AT(6,1):"Put your own equation in line 210, using A for the known value and X for the unknown value."
120 DISPLAY AT(24,6):"PRESS ANY KEY" :: DISPLAY AT(24,6):"press any key" :: CALL KEY
```

```

(0,K,S):: IF S=0 THEN 120
130 CALL CLEAR
140 DISPLAY AT(8,1):"KNOWN V
ALUE? " :: ACCEPT AT(8,14):C
150 X=1 :: DISPLAY AT(12,1):
""
160 GOSUB 210
170 IF A<C THEN Y=X :: X=X*2
:: GOSUB 210 :: GOTO 170 EL
SE 190
180 IF A>C THEN Y=X :: X=X/2
:: GOSUB 210 :: GOTO 180
190 Z=(ABS(X-Y))/2 :: Y=X ::
IF A<C THEN X=X+Z ELSE X=X-
Z
200 GOSUB 210 :: GOTO 190
210 A=X^X/X-X/2
220 IF A<C THEN DISPLAY AT(1
1,5):X ELSE IF A>C THEN DISP
LAY AT(13,5):X
230 IF A=C OR A=B THEN DISPL
AY AT(12,5)ERASE ALL:X :: GO
TO 140 ELSE B=A :: RETURN

```

In Recreational and Educational Computing, published 8 times a year at 909 Violet Terrace, Clarks Summit PA 18411, \$36 per year, I found a neat routine to find the greatest common divisor and least common multiple of any two numbers - so I converted it to TI Basic and modified to do the same with multiple numbers.

```

100 CALL CLEAR :: PRINT "PRO
GRAM TO FIND THE GREATESTCOM
MON DIVISOR AND LEAST COM
MON MULTIPLE OF ANY NUM
BER OF NUMBERS." ::
110 DIM N(100)
120 PRINT "INPUT ZERO WHEN F
INISHED" ::
130 T=T+1 :: INPUT "NUMBER "
&STR$(T)&"? ":N(T):: IF N(T)
=0 THEN 140 ELSE IF N(T)<>IN
T(N(T))OR N(T)<1 THEN T=T-1
:: GOTO 130 ELSE 130
140 AA=N(1):: GCD=N(2)
150 GOSUB 170 :: FOR J=3 TO
T-1 :: AA=N(J):: GCD=ABS(GCD
):: GOSUB 170 :: NEXT J
160 GOTO 210
170 R=AA-INT(AA/(GCD+ABS(GCD
=0)))*GCD
180 IF R<2 THEN GCD=R+GCD*(1
-R):: GCD=GCD*ABS(GCD>0)+ABS
(GCD=0):: GOTO 200

```

```

190 AA=GCD :: GCD=R :: GOTO
170
200 RETURN
210 PRINT "THE GREATEST COMM
ON DIVISOR OF YOUR";T-1;"NUM
BERS IS";GCD
220 L=N(1)*N(2)/GCD :: FOR J
=3 TO T-1
230 IF L/N(J)<>INT(L/N(J))TH
EN L=L*N(J)
240 NEXT J
250 LL=L/2 :: FOR J=1 TO T-1
:: IF LL/N(J)<>INT(LL/N(J))
THEN J=T-1 :: GOTO 270
260 NEXT J :: L=LL :: GOTO 2
50
270 PRINT "AND THE LOWEST CO
MMON MULTIPLE IS";L

```

Joy Warner called from the L.A. group, and mentioned that it would be nice to have a program to print out a page of math problems, and a page of answers. So here is one that will randomly create any number of either addition or subtraction problems, within any specified range of numbers, and output the desired number of copies to a printer in two columns of expanded print, numbered in sequence, plus a numbered answer sheet to make it easy for the teacher.

```

100 DISPLAY AT(1,4)ERASE ALL
:"MATH PROBLEM PRINTER" !by
Jim Peterson
110 DIM A(200),H(200),L(200)
:: OPEN #1:"PIO" :: PRINT #1
:CHR$(27)&"@"&CHR$(27)&"W"&C
HR$(1);
120 M$(1)="ADDITION" :: M$(2)
)="SUBTRACTION" :: D$(1)="+
" :: D$(2)="- " :: ON$=CHR$(
27)&"-"&CHR$(1):: OFF$=CHR$(
27)&"-"&CHR$(0)
130 DISPLAY AT(3,1):"Do you
want?":":":1. "&M$(1):"2. "&
M$(2):: ACCEPT AT(3,14)VALID
ATE("12")SIZE(1)BEEP:C
140 DISPLAY AT(8,1):"Range o
f numbers?":"From":"To" :: A
CCEPT AT(9,6)VALIDATE(DIGIT)
BEEP:LN :: ACCEPT AT(10,4)VA
LIDATE(DIGIT)BEEP:HN :: IF L
N>HN THEN 140 ELSE HN=HN-LN
150 DISPLAY AT(13,1):"How ma

```

```

ny problems?" :: ACCEPT AT(1
3,20)VALIDATE(DIGIT)BEEP:P
160 DISPLAY AT(15,1):"How ma
ny copies?" :: ACCEPT AT(15,
18)VALIDATE(DIGIT)BEEP:CC
170 FOR J=1 TO P :: GOSUB 29
0 :: H(J)=N1 :: L(J)=N2
180 IF C=1 THEN A(J)=H(J)+L(
J)ELSE A(J)=H(J)-L(J)
190 NEXT J
200 FOR J=1 TO CC :: GOSUB 3
10 :: FOR K=1 TO P STEP 2
210 T1$=STR$(K)&". "&STR$(
H(K)):: T2$=STR$(K+1)&".
"&STR$(H(K+1))
220 PRINT #1:TAB(15-LEN(T1$)
);T1$;TAB(35-LEN(T2$));T2$
230 T1$=D$(C)&STR$(L(K)):: T
2$=D$(C)&STR$(L(K+1))
240 PRINT #1:TAB(15-LEN(T1$)
);ON$&T1$&OFF$&RPT$(" ",20-L
EN(T2$))&ON$&T2$&OFF$
250 PRINT #1:"":":":":":":": IF
K/19=INT(K/19)THEN PRINT #1:
CHR$(12);
260 NEXT K :: PRINT #1:CHR$(
12);:: NEXT J
270 PRINT #1:TAB(16);"ANSWER
S":":":":":":
280 FOR J=1 TO P STEP 2 :: P
RINT #1:TAB(6);STR$(J)&". "
;A(J);TAB(26);STR$(J+1)&".
";A(J+1):: NEXT J :: STOP
290 RANDOMIZE :: N1=INT(RND*
HN+LN):: N2=INT(RND*HN+LN)::
IF N1=N2 THEN 290
300 IF C=2 AND N2>N1 THEN T=
N2 :: N2=N1 :: N1=T :: RETUR
N ELSE RETURN
310 PRINT #1:" " "&M$(C)
&" PROBLEM PRINTER":":":":":":
:":":":":":": RETURN

```

And this one will do the same with multiplication problems.

```

100 DISPLAY AT(1,4)ERASE ALL
:"MULTIPLICATION PROBLEMS":
PRINTER" !by Jim P
eterson
110 DIM A(200),H(200),L(200)
:: OPEN #1:"PIO" :: PRINT #1
:CHR$(27)&"@&CHR$(27)&"W"&C
HR$(1);
120 ON$=CHR$(27)&"-"&CHR$(1)
:: OFF$=CHR$(27)&"-"&CHR$(0)
130 DISPLAY AT(8,1):"Range o
f multiplicand?":"From":"To"
:: ACCEPT AT(9,6)VALIDATE(D

```

```

IGIT)BEEP:L1 :: ACCEPT AT(10
,4)VALIDATE(DIGIT)BEEP:H1 ::
IF L1>=H1 THEN 130 ELSE H1=
H1-L1
140 DISPLAY AT(12,1):"Range
of multiplier?":"From":"To"
:: ACCEPT AT(13,6)VALIDATE(D
IGIT)BEEP:L2 :: ACCEPT AT(14
,4)VALIDATE(DIGIT)BEEP:H2
150 IF L2>=H2 THEN 140 ELSE
R=LEN(STR$(H2)):: H2=H2-L2
160 DISPLAY AT(16,1):"How ma
ny problems?" :: ACCEPT AT(1
6,20)VALIDATE(DIGIT)BEEP:P
170 DISPLAY AT(18,1):"How ma
ny copies?" :: ACCEPT AT(18,
18)VALIDATE(DIGIT)BEEP:CC
180 FOR J=1 TO P :: GOSUB 31
0 :: H(J)=N1 :: L(J)=N2
190 A(J)=H(J)*L(J)
200 NEXT J
210 FOR J=1 TO CC :: GOSUB 3
20 :: FOR K=1 TO P STEP 2
220 T1$=STR$(K)&". "&STR$(
H(K)):: T2$=STR$(K+1)&".
"&STR$(H(K+1))
230 PRINT #1:TAB(15-LEN(T1$)
);T1$;TAB(35-LEN(T2$));T2$
240 T1$="X "&STR$(L(K)):: T2
$="X "&STR$(L(K+1))
250 PRINT #1:TAB(15-LEN(T1$)
);ON$&T1$&OFF$&RPT$(" ",20-L
EN(T2$))&ON$&T2$&OFF$
260 FOR S=1 TO R+3 :: PRINT
#1:"" :: NEXT S
270 LC=LC+5+R :: RC=LC+5+R :
: IF RC>=60 AND K<P THEN PRI
NT #1:CHR$(12):: LC=5
280 NEXT K :: PRINT #1:CHR$(
12);:: NEXT J
290 PRINT #1:TAB(16);"ANSWER
S":":":":":":
300 FOR J=1 TO P STEP 2 :: P
RINT #1:TAB(3);STR$(J)&". "
;A(J);TAB(23);STR$(J+1)&".
";A(J+1):: NEXT J :: PRINT #
1:CHR$(12):: STOP
310 RANDOMIZE :: N1=INT(RND*
H1+L1):: N2=INT(RND*H2+L2)::
RETURN
320 PRINT #1:" " MULTIP
LICATION PROBLEMS":":":":":":
" " :: LC=5 :: RETURN

```

And division -

```

100 DISPLAY AT(1,6)ERASE ALL
:"DIVISION PROBLEMS":
PRINTER" !by Jim Peterso
n

```

```

110 DIM A(200,2),H(200),L(200):: OPEN #1:"PIO" :: PRINT #1:CHR$(27)&"@"&CHR$(27)&"W" &CHR$(1);
120 DISPLAY AT(8,1):"Range of dividend?":"From":"To" :: ACCEPT AT(9,6)VALIDATE(DIGIT)BEEP:L1 :: ACCEPT AT(10,4)VALIDATE(DIGIT)BEEP:H1 :: IF L1>=H1 THEN 120
130 DISPLAY AT(12,1):"Range of divisor?":"From":"To" :: ACCEPT AT(13,6)VALIDATE(DIGIT)BEEP:L2 :: ACCEPT AT(14,4)VALIDATE(DIGIT)BEEP:H2
140 IF L2>=H2 THEN 130 ELSE R=LEN(STR$(INT(H1/H2)))*2 :: H2=H2-L2 :: H1=H1-L1
150 DISPLAY AT(16,1):"How many problems?" :: ACCEPT AT(16,20)VALIDATE(DIGIT)BEEP:P
160 DISPLAY AT(18,1):"How many copies?" :: ACCEPT AT(18,18)VALIDATE(DIGIT)BEEP:CC
170 FOR J=1 TO P :: GOSUB 310 :: H(J)=N1 :: L(J)=N2
180 A(J,1)=INT(H(J)/L(J)):: A(J,2)=H(J)-A(J,1)*L(J)
190 NEXT J
200 FOR J=1 TO CC :: GOSUB 320 :: FOR K=1 TO P STEP 2
210 LC=LC+1 :: T1$=STR$(K)&" "&RPT$(" ",LEN(STR$(L(K))))&RPT$("_",LEN(STR$(H(K))))
220 T2$=STR$(K+1)&". "&RPT$(" ",LEN(STR$(L(K+1))))&RPT$("_",LEN(STR$(H(K+1))))
230 PRINT #1:TAB(15-LEN(T1$));T1$;TAB(35-LEN(T2$));T2$
240 T1$=STR$(L(K))&"!"&STR$(H(K)):: T2$=STR$(L(K+1))&"!"&STR$(H(K+1))
250 LC=LC+1 :: PRINT #1:TAB(15-LEN(T1$));T1$;TAB(35-LEN(T2$));T2$
260 FOR S=1 TO R+3 :: LC=LC+1 :: PRINT #1:"" :: NEXT S
270 IF 66-LC<5+R AND K<P THEN PRINT #1:CHR$(12):: LC=5
280 NEXT K :: PRINT #1:CHR$(12):: NEXT J
290 PRINT #1:TAB(16);"ANSWER S":"";"
300 FOR J=1 TO P :: PRINT #1:TAB(3);STR$(J)&". ";A(J,1);"REMAINDER ";A(J,2):: NEXT J :: PRINT #1:CHR$(12):: STOP
310 RANDOMIZE :: N1=INT(RND*

```

```

H1+L1):: N2=INT(RND*H2+L2):: RETURN
320 PRINT #1:" DIVISION PROBLEMS":"";":: LC=5 :: RETURN

```

Bud Wright wrote this one for Irwin Hott, so he could listen to lower case text with the Speech Synthesizer. Imbed it with ALSAVE, access it with CALL LINK("CAPS",A\$) and it will instantly convert any lower case letters to upper case. I found it invaluable in writing keyword search programs.

```

* CAPS/S BY BUD WRIGHT
* VERSION 1.1 10/17/86
STRREF EQU >2014
STRASG EQU >2010
MREG BSS 32
STRBUF BYTE 255
BSS 255
DEF CAPS
CAPS LWPI MREG
CLR R0
LI R1,1
SETO @STRBUF
LI R2,STRBUF
BLWP @STRREF
MOVB @STRBUF,R2
SRL R2,8
JEQ CAPOUT
LI R1,STRBUF+1
CAPS2 MOVB *R1,R3
SRL R3,8
CI R3,96
JGT CAPS1
CAPS3 SWPB R3
MOVB R3,*R1+
DEC R2
JNE CAPS2
CLR R0
LI R1,1
LI R2,STRBUF
BLWP @STRASG
CAPOUT LWPI >83E0
B @>006A
CAPS1 CI R3,122
JGT CAPS3
AI R3,-32
JMP CAPS3
END

```

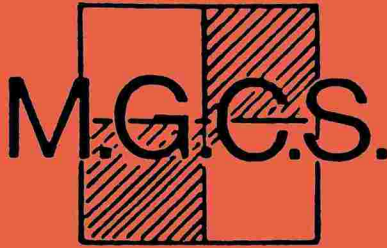
Memory full,

Jim Peterson

## BOOK REVIEW

Member Kenneth F Hughes of 220 Broadland Drive, Lawrence Weston, Bristol, Avon. reports on a book called "The Cuckoo's Egg" by Clifford Stoll published by Pan. "Clifford Stoll had just started as a systems manager running a central computer in a Berkeley lab. His job was sorting out the accounts for the month which were out by 75cents. From this insignificant beginning Stoll found himself plunged into an incredible true story. A search spanning continents and lasting many months, for a mysterious hacker using the Californian computer as an electronic way station to infiltrate dozens of other systems, including the most sensitive military, scientific, and intelligence networks in America. Clifford Stoll is an astronomer, computer security expert, and holds a Ph.D. from the University of Arizona awarded in 1980."

This page is blank



MIKE GODDARD COMPUTER SUPPORT

NEW AND USED COMPUTER EQUIPMENT BOUGHT AND SOLD

-----  
TI-99/4A SPECIALIST  
-----

SPARES

KITS

REPAIRS

COMPONENTS

BOOKS

PRINTER CABLES

RESISTORS

BATTERIES

DISK DRIVES

POWER SUPPLIES

POWER CABLES

PRINTERS

MONITORS

CAPACITORS

DISK DRIVE CABLES

RIBBON CABLE

FOR LATEST LIST CONTACT:

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

Tel: (0978)843547.

MBX: 022212529