COLOR COMPUTER NEWS

$2.95

March 1982
Issue No. 7

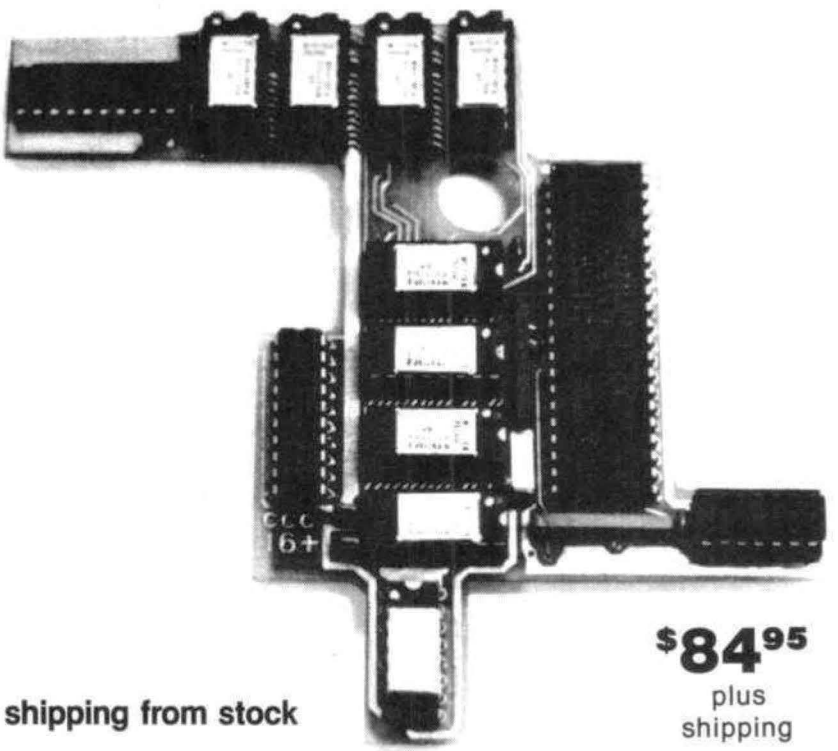
REMarkable Software
P.O. BOX 1192
MUSKEGON, MI 49443

BULK RATE
U.S. POSTAGE
PAID
MUSKEGON, MI
PERMIT #220

16 PLUS

MEMORY EXPANSION FROM 16K TO 32K FOR THE

COLOR COMPUTER



shipping from stock

$84.95
plus
shipping

- No soldering! Easy to plug in!
- Fits under RF shield — complete instructions included
- Allows video display (graphics) to reside anywhere in the 32K memory
- Call or write for our complete catalog of Color Computer products!

COMPUTERWARE®

Box 668 • Encinitas, CA 92024
(714) 436-3512

6809 Specialists

Computerware is a trademark of Computerware.

This month's Editorial comes from Bob Lentz of The Micro Works. The only addition I would make to Bob's Editorial is to inform everyone that the "Rumors" we publish are verified by at least three sources before we print them, one of those sources being Radio Shack themselves. I'm certain we'll "blow it" someday but to date all of the rumors we've printed have later proven to be totally accurate. Bill Sias.

After reading the "Rumors, Rumors..." article in the December issue of Radio Shack's "Micro computer News", I felt that it would be of some interest if a few points about the Color Computer were brought up by one of the "various sources" to which the Product Line Manager refers.

Here at The Micro Works we have learned a lot about the Color Computer while writing programs and designing hardware for firstclass support of this wonderful machine, and have been in a position to publish information about it which is of interest to users. We can all agree with the Shack's article in that idle rumors and incorrect "insiders' information" is of no benefit to anybody, but it's time to turn the tables and clarify some of the misrepresentations made by the Shackfolk themselves.

For example, it is NOT true that you need the version 1.1 ROM in order that the machine "recognize the full 32K of memory"; it is only necessary in order to recognize the different kind of chips which Radio Shack supplies in its 32K upgrade. The old ROM works fine with 32K kits such as those sold by The Micro Works and others. So does the new ROM, of course; just leave the chip-type jumpers set for 16K chips. True, their upgrade is, as they say, "neat", and is useful for those who cringe at the sight of a soldering iron; but then, at $149.00, it ought to be.

The statement is made that outsiders' knowledge of the Color Computer stops at the chip level. Never underestimate experts; we'll find out anything of interest or use. For example, did you know that on the earlier Color Computers, the bonding wires were left off pins 33, 36 and 38 of the CPU die? These CPUs were therefore not properly called 6809E's and were not labelled as such. (Later machines do have these pins connected and the chips are 6809E's.) Why this is, I don't know. I do know that no one should come out with any sort of kit that involves the use of the LIC, BUSY or AVMA signals on the CPU.

There is a lot of interest in the difference between the 1.0 and 1.1 ROMs. Radio Shack refuses to sell the 1.1 ROM to people who want to upgrade, but surely they can see that with all the 8-bit graphics printers around, a built-in 8-bit printer driver is a very useful thing. And they can surely expect interest in the ROM, when their notes say, "You can now have a full 255 characters in data files without having to worry about losing any information." The rest of us should just worry? Actually, I've never lost any data and I expect that the only effect of the bug at location $A440 (changed from $08 to $03) is that no end of file was written on files which contained no data at all.

For the disassembly minded, the complete list of bytes which differ between 1.0 and 1.1 is as follows:

A01C, A024, A027-A04B, A04D-A04E, A050-A0C7, A102-A104, A155, A1C2, A1C7-A23A, A23D-A26D, A2C5-A2E7, A2E9-A2FA, A440, A6EB.

By the way, someone said that the reset vector at BFFE had been changed to $A227; if so, they've got a different 1.1 than I have. But what would a reset routine be doing lurking around the middle of the keyboard scanner, anyway?

Speaking of new ROMs, has anyone noticed the bug in the disk system which allows graphics pages to be allocated at odd-page boundaries where they cannot be displayed? Or the one which crashes a BASIC program from indescriminate use of the FILES statement? It has already been pointed out that the PCLEAR statement can crash a program if not used very carefully. It would be nice if there were some new versions of the ROMs at some time in the future, especially if Radio Shack were to do something interesting with the unused 2K at the end of the disk ROM.

I hope that these notes are of some help in clarifying these matters. Those who are interested in purchasing the workings of the Color Computer may wish to purchase a good cross-referencing disassembler such as the one sold by The Micro Works. Meanwhile, don't believe everything you hear!

Robert A. Lentz
President
The Micro Works
PO Box 1110
Del Mar, CA 92014
(714) 942-2400

THE MICRO WORKS

EXCITING NEWS FOR COLOR COMPUTER USERS

FLEX, OS-9 and the Radio Shack Disk System ALL on the SAME Color Computer

Would you believe that you can run FLEX, OS-9 and Radio Shack disk software on the same Color Computer, and all you have to do is change the disk? That's right, just change the disk. If you have a 32K Color Computer with the Radio Shack disk system, all you need to do is make a trivial modification to access the hidden 32K, as described in the Feb. issue of COLOR COMPUTER NEWS and the March issue of '68' Micro. You can get FLEX from us right now. OS-9 will be ready by summer. Please note that this will only work with the Radio Shack disk system and 32K/64K memory chips that RS calls 32K. Maybe they put 64K's in yours, too. If you don't have a copy of the article, send a SASE and we'll send it to you.

Using this system to run FLEX and OS-9 has many advantages. First, it gives you 48K from zero right up to FLEX. This means that *ALL* FLEX compatible software will run with *NO MODIFICATIONS* and *NO PATCHES!* There are no memory conflicts because we moved the screen up above FLEX which leaves the lower 48K free for user programs.

What you end up with is 48K for user programs, 8K for FLEX and another 8K above FLEX for the screens and stuff. We are working on a multi screen format so you can page backward to see what scrolled by and a Hi-Res screen that will enable us to have 24 lines by 42 character display. That's better than an Apple!

We also implemented a full function keyboard, with a control key and escape key. All ASCII codes can now be generated from the Color Computer keyboard!

We also added some bells and whistles to Radio Shack's Disk system when you're running FLEX or OS-9. We are supporting single or double sided, single or double density, 35, 40 and 80 track drives. If you use double sided drives, the maximum is three drives because we use the drive 3 select for side select. When you are running the Radio Shack disk, it will work with the double sided drives but it will only use one side and only 35 tracks. Using 80 track drives is okay, but will not be compatible with standard Radio Shack software. You can also set each drive's stepping rate and drive type. (SS or DS - SD or DD)

In case you don't understand how this works, I'll give you a brief explanation. The Color Computer was designed so that the roms in the system could be turned off under software control. In a normal Color Computer this would only make it go away. However, if you put a program in memory to do something first (like boot in FLEX or OS-9), when you turn off the roms, you will have a full *64K RAM* System with which to run your program (FLEX or OS-9). When the roms are turned off, it is as if you had removed them from the computer. They are gone!

Now, we need the other half of the 64K ram chips to work, and this seems to be the case most of the time, as the article states. Of course, you could also put 64K chips in.

We decided that this was the best way to run FLEX and OS-9 on the Color Computer because it does remove the roms from the memory map and leaves the full 48K for user programs. If you just put in memory for FLEX and use the Basic hooks for I/O, all you have is a little over 30K for user programs. In addition, very few FLEX programs will run without being modified and some won't run very well, if at all (our DATAMAN + for example). Let me state it again. ALL FLEX COMPATIBLE PROGRAMS WILL RUN WITHOUT MODIFICATION!!! and the same goes for OS-9!

It is also the ONLY way OS-9 will run because 30K is just not enough.

Some neat utilities are included.

MOVEROM moves Color Basic from ROM to RAM. Because it's moved to RAM you can not only access it from FLEX, you can run it and even change it!! You can load Color Comuter cassette software and save it to FLEX disk. Single Drive Copy, Format and Setup commands are also included.

If you don't have a Color Computer, we can sell you one complete with 64K ram, 24K rom, Single RS disk drive and FLEX for only $1,490, set up and ready to go.

FLEX with Edit, Asmb and installation disk is $199.

FRANK HOGG LABORATORY, INC.

130 MIDTOWN PLAZA • SYRACUSE NEW YORK 13210 • (315)474-7856

Dear Bill,

In the January Issue of CCN, did you leave out a subroutine in the Othello Program on page 42? There is some GOSUB 5000 lines in the program. I can't get it to run right and it looks like a nice program.

Your truly,
George P. Barber Jr
2410 New Haven Ave
Fort Wayne, IN 46803

The GOSUB 5000s in lines 217 and 317 were part of a debugging routine that Tony used during developement. The variable listing routine was removed but the lines calling it were not. You can safely remove lines 216, 217, 316 and 317, there are no other problems in that program.

Gentlemen,

My wife and I bought our 32K machine with line VII printer and disk drive around last Thanksgiving. We have both been spectacularly disappointed in the Radio Shack software available. The total absence of programs on disk make my disk drive totally useless. The Personal Finance cartridge doesn't have a print feature. This Scripsit package is alright for things like this, but has some features I'm not crazy about and I'm stuck with cassettes again.

Neither of us is computer crazy. We could care less about the guts of this thing or the technical tricks that could be employed. What we do want is the software packages that would make our household finances, listings of things for file, school papers, and personal letters fun.

W.R. Taylor
398 Bigstone Dr
Xenia, OH 45385

We're looking at some applications software now for possible inclusion in a future issue.

Dear Sir,

I have entered "Sample Program" from the July/Aug issue, "Nam Primes" from Sept/Oct issue and "Tapetype" from the Nov/Dec issue of CCN.

I ran into some trouble with "Nam Primes" and TapeType". Both had some address and op-codes missing

EXAMPLE "NAME PRIMES" Sept/Oct Page 44

ADDRESS	OP-CODES
1BC2	64
1BC3	00 0A
1BC5	00 01
1BC7	00 00

10000, 1000, 100, 1,0. program did not work without these

EXAMPLE "TAPETYPE" Nov/Dec Page 50

ADDRESS	OP-CODES	
639	52	
63A	2A	
64B	20	
64C	20	
67F	41	
680	4C	ECT.....

Banner type information

Without these address and op-codes the programs did not run. I was able to figure out all the op-codes I needed but it did take a long time. If I had an ASSEMBLER this would not have been a problem.

My reason for writing was to let you know that some important information was missing from your listings.

I hope you will continue to put more machine language in CCN. Your "Basic Monitor" works great.

I liked C.J. Roslund "Variable Lister" program in the Jan. issue of CCN. But I made so many typing errors and got so many ?SN ERROR's that I made the following changes.

I typed in Line numbers at the end of my program like this:

```
60000 FOR I=PEEK (27)*256+PEEK (28) TO PEEK (29)*256+ PEEK (30)-5 STEP 7
60010 PRINT CHR$ (PEEK(I);CHR$ (PEEK(I+1) AND 127);
60020 IF PEEK (I+1)>127 THEN PRINT"$";
60030 PRINT " ";:NEXT I
```

Note: use any line # ;you want, just put it at end of your program.

Note: Only few changes from original program.

Now run program long enough to define variables. Press "BREAK" and enter GOTO 60000 (or what ever line # you use) you will get a print out of all variables. Ignore the "I" in list, that in our program.

The nice thing about doing it this way is you only have to type it in once. You can keep going back to it as often as you need as you write your program.
Thank You.
Michael B. Kromeke
6308 Harper Dr. N.E.
Albuquerque, NM 87109

*Typing in machine language programs without an assembler is extremely difficult and very error prone. We will be running an article about how to do it shortly.

Dear Bill,

Like Tom Mix (CCN Nov/Dec 1981), I became entranced with the problem of solving the "BLACK SANCTUM" Adventure. I, too, was developing anxiety symptoms and just had to get done with it. Not being patient or very fastidious, I turned to dishonest methods.

First I hauled out my CBUG monitor from Micro Works. I inspected the code in ASCII, starting with P 0600, where it loads. Liking what I saw, I hitched up a printer and listed all memory bytes having an "A" to "Z" upper case value (ASCII Codes $40 to $50, and $20, the space character). Some junk came out, but I did get the complete dictionary of all verbs and nouns, and the responses of the program. With the dictionary in hand the job became much easier. I know what to expect in those long corridors.

While I was at it, I found the addresses for fixing the incorrect spelling. (These are the actual program addresses, not the offset addresses of the BASIC program listed after the end of this letter.)

```
POKE &H2510,&H41   CHANGE TO "A"
POKE &H2519,&H45   CHANGE TO "E"
POKE &H2684,&H45
POKE &H3509,&H45
```

Now we can say that we have been to the real ST. SEBASTIAN MONASTERY! To save a corrected version, prepare a tape for recording and:

```
CSAVEM "SANCTUM",&H0600,&H3D40,&H0600
```

This is the usual machine language save: Name, Starting Address, Final Address, and Future Loading Address.

In closing I must say that I and members of my family enjoyed "BLACK SANCTUM" immensely, each in his own way. Very young visitors even felt that is was spooky. Come to think of it, I feel an eerie sensation.....

Invoca..Ca..Ca.....
Arnold H. Kahn
2706 Roos Road
Chevy Chase, MD 20815

Here is a short BASIC program for examining text in machine language program. For a ML program as long as "SANCTUM" you'll need 32K or memory. But for shorter programs it will work readily.

```
PCLEAR 1
CLOADM "SANCTUM",&HA00 OR ANY NAME, OFFSET LOAD
CLEAR 5,H0FFF MAKE ROOM FOR BASIC
05 REM PROG LOADS AT $600 + $A00 = $1000
10 FOR I=&H1000 TO &H4FFF  OR ANY RANGE
20 A=PEEK(I)
30 IF A=&H20 THEN PRINT CHR$(A);
40 IF (A)&H40)AND(A<&H5A) THEN PRINT CHR$(A);
50 NEXT I
```

Who says cheating isn't fun?

Dear Bill,

Sorry, no printer yet. I'm pleased with your magazine and glad to see it expanding. I would like some info on techniques such as CSAVE Insurance, PClearing 0, screen dump to printer, draw alphabet, appending program, POKEs for HI SPEED or Cassette motor, etc. I saw in Jan. 82 issue that you are using K6AEP RTTYCW Software. I am interested in information on how to add a log book and auto-increment to my K6AEP RTTYCW program. I have never been able to make the screen pointer program step right (pg. 43-47, Nov/Dec 1981). I didn't see any corrections in the next issue. I have tried changing 270 GOTO 170 to 270 GOTO 230. HELP!
Con W5BWF

* Adding features to Clay's CWRTTY package is not as simple as it may have sounded. It required disassembling the entire program and removing parts to make room for the changes. Since it's not an easy task and since it would require violating Clay's copyright to show how to do it I probably will never show it, perhaps the best solution is to write to Clay about it and if he receives enough mail he may be convinced to add those features.

Dear Mr. Sias,

I write a color computer column for our local club newsletter and I would like to know your feelings on the use of material from your magazine in our newsletter. The newsletter is a non commercial venture limited to club members only. The club is for users of all TRS-80 machines, of which the color gadget is a small but growing part. I myself have the color computer with 32K, disc, ext. basic, line Printer Vii, modem I, etc.

In conclusion, I repeat that I am happy to find out that your publication is alive and well and I do look forward to further issues and I hope to hear from you soon.

Thank You,
William Bruff
215-47 Deerfield Dr.
Nepean, Ont. K2G 3R7

* Club newsletters are allowed to reprint from CCN if the following notice accompanies the reprinted information and a copy of the newsletter containing the reprinted material is forward to us.

Reprinted from Color Computer News
copyright (c) 1982
REMarkable Software
P.O. Box 1192
Muskegon, MI 49443

Dear Sir,

I recently purchased an MX-80 EPSON Printer. The Printer works fine with my TRS 80 16K Extended Basic Color Computer except for GRAPHICS.

The MX 80 USER'S MANUAL indicates that outputting ASCII codes 160/224 to the printer will cause the printer to print it's built in graphics characters. However my printer prints only alphanumerics with these codes. I devised a test program to output the graphics to the printer and the monitor screen. The screen properly displays the right graphics characters. The printer prints alphanumerics as its response.

Perhaps you can shed some light on my problem.

A reply at your earliest convenience would be greatly appreciated.

Sincerely,
Peter J. Paulos
16579 Creekside Drive
Sonora, CA. 95370

The MX-80 has two modes of operation; the TRS-80 mode and normal. Try reversing the switch 2-4 or try sending codes up to 255.

Dear Sir;

Thank you for bringing out a Magazine in support of the most unadvertised gem on the microcomputer market.

I have a Color 32K, with Extended Basic of which I am very proud. I only wish more people realized the capabilities of the TRS 80C.

By working together maybe us Color Lovers can get rid of the "play thing, Game" stigma.

If we support it with good software, it will survive!!

Thank You,
Joseph W. Gehrke
101 A Hanna NBU 120C
Ft. Huachuca, AZ 85613

I agree and also think that a lot of the stigma is disappearing with the introduction of things like FLEX and OS-9. Now it seems to be getting the reputation of a 6809 software developement system. Alot of folks are using the Color Computer to examine the 6809 and looking for larger systems to move up to.

Dear Sir;

I have enjoyed reading the CCN. I especially enjoy articles and programs on machine language that are useful, such as "TapeType". I would like to call your attention to the January issue of '80 Microcomputing Magazine, page 8. You probably have seen it already. I think such articles concerning rumors that can't be verified are irresposible!!!! Either way it can only hurt the Color Computer. People will be afraid to buy one not knowing if the rumor is true and companys may not want to invest time and money for software if the rumor is true. I think the Color Computer has received some unfair publicity. Those of us who know the Color Computer are aware of its capabilities. The way Radio Shack promotes it is part of the problem. I would like everyone who has a Color Computer to write to Radio Shack should the rumor prove to be true, or to '80 Microcomputing if it is not, and tell Mr. Green how unprofessional such rumor reporting is!!

Thank You,
Verne R. Winter, Jr.

502 Davis Ave.
Des Moines, Iowa 50315

* This is just one of several similar letters about Wayne Greene's editorial. I think if he'd look around he'd see that the reason he isn't getting many articles about the Color Computer is because we're getting the "good stuff".

Dear Sirs:

A few days ago I recieved my first issue of Color Computer News. I was/am very impressed and very pleased. I couldn't believe how many specific needs of mine were met in one issue. To say that I am happy I subscribed would be an understatement. (I've already recommended your magazine to two friends that have TRS 80C).

I do have some questions though. I enjoyed the January so much that I would like to purchase ALL the back issues--can I do that and if so - how much? Next question I have found many articles that seemd to be exactly aimed at my level of programming. But not too much on some of the simpler programming techniques that we all just seem to pick up. Would you be interested in a program and article teaching some of the simpler techniques. An article expecially for newcomers. Things that either are not found or are unclear in the COLOR BASIC manuals. If so let me know. Also let me know if you require any specific format. I am extremly happy with the CCN. Hope to hear from you soon.

Sincerely,

Rich Petty
4551 Hercules Dr
Salt Lake City, Utah 84120

I'm always interested in "Beginner's Articles", we really haven't received many of them. However, we will still remain a "technical" magazine and support machine language programmers. I haven't pushed back issues because they are in such short supply. You can get them for $3.25 each which includes first class postage. Issue 5 is gone and the reprints of issues 1 and 3 are almost gone also.

GAME PROGRAMMING WITH EXTENDED BASIC

by Mark Barnes

Here I sit with my computer over heating from some nasty number crunching, check book balancing, payroll figuring, and general record keeping. My mind is shot and I need a break. Time to sit back, relax and play a game. I've always enjoyed inventing and playing games of all sorts and naturally when I got my Color Computer a year ago I wanted to create some video games. I had the Extended Basic put in and I thought I was ready to go, but...I ran into a few problems.

My first problem was the quick exhaustion of memory when using the 'GET' and 'PUT' statements. The 'GET' function allows you to store a picture into a two dimensional array, while the 'PUT' function allows you to reproduce this picture anywhere on the screen. This works great, but, a two dimensional array takes up a lot of memory and by the time I saved all my pictures I barely had any room for my program.

This problem was solved when I found out that a small one dimensional array could be used instead of the two dimensional array. If a one dimensional array is dimensioned using the 'DIM' statement and the size figued from the equation shown, the 'GET' and 'PUT' statements will work as if a two dimensional array was used. This is quite a shock since your Extended Basic manual just tells you to use two dimensional arrays. As you can see, this technique can actually save thousands of bytes of memory in a graphic program.

ARRAY SIZE=(LENGTH*WIDTH)/N
(ROUNDED DOWN)
WHERE N=40 FOR PMODE 3&4
N=80 FOR PMODE 1&2
N=160 FOR PMODE 0

The second problem I encountered was the fact that my figures moved in a choppy, flashing manner. I wanted them to move smoothly like in the arcades, this flashing could drive you nuts while playing the game. I know that this was due to the fact that one picture was being erased and then another was put onto the screen, but there had to be some way to do this faster so that it wouldn't be seen.

There is a way. The pages of memory come in handy for this purpose. By flipping back and forth between sets of pages, one can be drawing and erasing on one page while another page is being shown. Then by flipping pages (by use of the PMODE and SCREEN commands) animation is produced without any flashing. Notice, however, that different PMODES use different amounts of pages to produce a picture and all these pages must be copied and changed onto the other set of pages (i.e. in my example program I used PMODE 1 and copied pages 1 and 2 to 3 and 4 and vice versa). This flipping back and forth can be confusing at first, but using it can add a lot of class to your graphics program.

My last problem was speed. I wanted to duplicate some of my favorite arcade games, but I found out that most of the challenge was in their speed. Since I don't have a background in machine language programming, I thought I would have to be content to play games rather than program my own. At this point, however, I realized I would just have to be a little more creative and make games that were challenging, but didn't require great speed. That is what I have done in the program following this article. It is called "SPACE PATROL" and it uses the techniques described in this article. The game has good graphics, fair speed, and it's a challenge even though only one alien appears at a time. The program is long, but I think you'll find it's worth the time it takes to type it. As your game programming gets better you will find that there is no greater reward than to show a new game to a friend and after an hour of play have them say, 'Let me play just one more game!'

```
5 CLS: PRINT"  ******SPACE PATROL******": PRINT
10 PRINT"THE OBJECT OF THIS GAME IS TO SCORE AS MANY POINTS AS POS-  SIBLE BY SHOOTING THE DE-  SCENDING ALIENS BEFORE THEY  REACH THE EARTH. THE CLOSER  THEY COME TO THE EARTH, THE HIGHER THEIR VALUE. YOUR"
20 PRINT"SCORE IS SEEN IN THE UPPER RIGHT CORNER.": PRINT: INPUT"<PRESS ENTER>"; O1
30 CLS: PRINT"HAZARDS: ": PRINT: PRINT"NOT ONLY WILL YOU BE SHOOTING  AT THE ALIEN, BUT HE WILL BE  SHOOTING BACK(WITH GREATER  ACCURACY AS THE GAME GOES ON)!": PRINT"OTHER HAZARDS INCLUDE THE RE-  MAINS OF THE ALIEN ONCE HE HAS"
40 PRINT"BEEN SHOT,  AS WELL AS THE SPACE MINES WHICH HE MAY LAY IN YOUR  PATH. THERE ARE TIMES,  HOWEVER  WHEN AN ALIEN CAN BE HELPFUL  BY PICKING UP THE REMAINS,  OR  MINES."
```

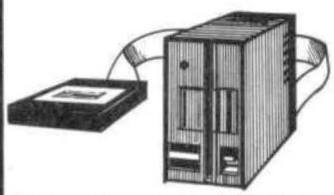
```
50 PRINT: INPUT"<PRESS ENTER>"; O1
60 CLS: PRINT"GAME ENDS WHEN: ": PRINT:
PRINT"1) 3 ALIENS LAND ON EARTH      2) 3
SHIPS HAVE BEEN SHOT      3) YOU HIT A
SPACE MINE"
65 PRINT: PRINT"THE NUMBER OF SHIPS LEFT
IS    SEEN IN THE UPPER LEFT CORNER
BEFORE THE ACTION BEGINS. THE   NUMBER
OF ALIENS THAT HAVE     LANDED APPEARS
IN THE LOWER    RIGHT CORNER."
70 PRINT: INPUT"<PRESS ENTER>"; O1
80 CLS: PRINT"HELPFULL HINTS: ": PRINT:
PRINT"THE RIGHT JOYSTICK IS USED TO
CONTROL THE SPACESHIP. TO BEGIN THE
ACTION,  PUSH THE JOYSTICK"
90 PRINT"FORWARD. DURING ACTION,  YOUR
SHIP WILL REMAIN STATIONARY IF  YOUR
JOYSTICK IS IN THE MIDDLE."
100 PRINT"TO FIRE YOUR LASER,  PRESS AND
HOLD THE BUTTON UNTIL THE LASER IS
FIRED(TIME YOUR SHOTS)."
110 PRINT: INPUT"<PRESS ENTER>"; O1
120 CLS: PRINT: PRINT"LASTLY,  YOU WILL
FIND THAT     AFTER FIVE ALIENS HAVE BEEN
SHOT(& 5 MORE FOR EACH AD-      DITIONAL
1000 PTS.),  YOU CAN"
130 PRINT"REMOVE ONE(BY SHOOTING IT)
GHOSTLY REMAIN FOR EVERY ALIEN  SHOT."
140 PRINT: PRINT: PRINT"       GOOD LUCK!"
150 PRINT: PRINT"SET RECORDER TO PLAY,
AND PRESS <ENTER> TO LOAD IN PROGRAM."
160 INPUTO1
170 CLS: PRINT"LOADING": CLOAD"PATROL"

1 '   MARK BARNES
2 '   805 DWYER APT. E
3 '   ARLINGTON HEIGHTS,  IL  60005
8 CLEAR50: FOR O1=1 TO 50: CLS(RND(8)):
PRINT@169, "SPACE PATROL"; : SOUND O1*3, 1:
NEXT: PRINT@465, "BY MARK BARNES"; : FOR
O1=1 TO 900: NEXT
9 PMODE1, 3: PCLS
10 DIMA(6), B(6), G(6), C(6), M(6), N0(1), N1(1),
N2(1), N3(1), N4(1), N5(1), N6(1), N7(1), N8(1),
N9(1), NN(1)
12 CLS: PRINT@165, "PREPARE FOR BATTLE!":
SCREEN0, 1
15 SC=0: U=1: V=3: YG=168: Q=1: TM=200: L=3:
C1=0: AE=0
20 PMODE1, 1: COLOR5, 7: PCLS
25 GET(10, 10)-(31, 30), B, G
29 'DRAW ALIEN
30 CIRCLE(20, 17), 10, 8, .71, .5, 1
40 DRAW"C8BM10, 16D9R21U9C5"
50 PAINT(20, 20), 8, 8
60 CIRCLE(16, 16), 3, 5: PAINT(16, 16), 5, 5
70 CIRCLE(24, 16), 3, 5: PAINT(24, 16), 5, 5
80 LINE(15, 26)-(15, 30), PSET
90 LINE(11, 29)-(15, 30), PSET, BF
100 LINE(25, 26)-(25, 30), PSET
110 LINE(26, 29)-(28, 30), PSET, BF
120 PSET(16, 16, 6): PSET(24, 16, 6)
130 DRAW"C5BM13, 23R15C5"
140 GET(10, 10)-(31, 30), A, G: PCLS
159 'DRAW SHIP
160 CIRCLE(24, 20), 3, 8, 1, .75, .25
170 DRAW"C8BM24, 17D7"
180 PAINT(26, 20), 8, 8
190 LINE(18, 15)-(21, 18), PSET
200 LINE(18, 25)-(21, 22), PSET
230 LINE(14, 13)-(19, 13), PSET, BF
240 LINE(14, 26)-(19, 27), PSET, BF: DRAW"C5"
260 LINE(12, 19)-(23, 21), PSET, BF
270 LINE(28, 21)-(28, 21), PSET
280 GET(10, 10)-(31, 30), G, G: PCLS
284 'DRAW GHOST
285 CIRCLE(20, 17), 10, 5, .71, .5, 1
286 DRAW"C5BM10, 16D9R21U9"
287 PAINT(20, 20), 5, 5
288 LINE(15, 26)-(15, 30), PSET
289 LINE(11, 29)-(15, 30), PSET, BF
290 LINE(25, 26)-(25, 30), PSET
291 LINE(26, 29)-(28, 30), PSET, BF
292 PSET(16, 16, 6): PSET(24, 16, 6)
293 DRAW"C7BM13, 23R15C5"
310 GET(10, 10)-(31, 30), C, G: PCLS
319 'DRAW MINES
320 DRAW"C6": LINE(14, 14)-(26, 26), PSET
340 LINE(20, 12)-(20, 28), PSET
350 LINE(26, 14)-(14, 26), PSET
360 LINE(28, 20)-(12, 20), PSET: DRAW"C5"
380 GET(10, 10)-(31, 30), M, G
390 COLOR8, 5: PCLS
399 'DRAW NUMBERS
400 LINE(17, 10)-(17, 20), PSET: GET(10, 10)-(20,
20), N1, G
410 LINE(12, 10)-(17, 10), PSET: GET(10, 10)-(20,
20), N7, G
420 LINE(12, 14)-(17, 14), PSET: LINE(12, 20)-(17,
20), PSET: GET(10, 10)-(20, 20), N3, G
430 LINE(12, 10)-(12, 14), PSET: GET(10, 10)-(20,
20), N9, G
440 LINE(12, 14)-(12, 20), PSET: GET(10, 10)-(20,
20), N8, G
450 LINE(17, 11)-(17, 13), PRESET: LINE(12,
10)-(17, 10), PSET: GET(10, 10)-(20, 20), N6, G
```

```
460 LINE(12, 13)-(12, 19), PRESET: LINE(12,
10)-(12, 14), PSET: GET(10, 10)-(20, 20), N5, G
470 PCLS: LINE(12, 10)-(17, 20), PSET, B: GET(10,
10)-(20, 20), N0, G
480 LINE(12, 14)-(17, 14), PSET: LINE(12, 11)-(12,
13), PRESET: LINE(12, 10)-(17, 10), PSET
490 LINE(17, 15)-(17, 19), PRESET: GET(10,
10)-(20, 20), N2, G
500 PCLS: LINE(12, 14)-(17, 14), PSET: GET(10,
10)-(20, 20), NN, G
510 LINE(12, 10)-(12, 14), PSET: LINE(17, 10)-(17,
20), PSET
520 GET(10, 10)-(20, 20), N4, G: COLOR5, 7: PCLS
524 'SET UP SCREEN TO BEGIN GAME
525 DRAW"C8": LINE(0, 190)-(255, 191), PSET,
BF: DRAW"C5": SCREEN1, 1
530 PUT(2, 168)-(23, 188), G, PSET
540 GOSUB1000: GOSUB6000: GOSUB7000
560 J=JOYSTK(0): IF JOYSTK(1)>5 THEN560
564 'ALIEN MOVEMENT
565 PUT(0, 0)-(21, 20), B, PSET
570 R=RND(10): RS=(RND(9)-1)*21: X=R*22+1
580 FORY=0 TO 168 STEP21
590 PUT(X, Y)-(X+21, Y+20), A, PSET: SCREEN1,
1: COLOR5, 7
591 IF PEEK(65280)=126 OR PEEK(65280)=254
THEN PP=1
592 IF PP=1 THEN GOSUB3000
595 ON Q GOSUB 4000, 5000
600 PCOPY U TO V: PCOPY U+1 TO V+1: W=V:
V=U: U=W
610 PMODE1, U
615 GOSUB2000
620 R=(RND(3)-2)*22: X=X+R
630 IF X<23 THEN X=23
635 IF X>221 THEN X=221
640 NEXT Y
650 AE=AE+1: GOSUB 6000: IF AE<3 THEN 570
660 GOSUB1000
670 PCLS7: SCREEN1, 1: GOSUB1000:
GOSUB6000: GOSUB7000
680 FOR X=221 TO 45 STEP-22
690 PUT(X, 168)-(X+21, 188), A, PSET: SCREEN1,
1
695 SOUND X, 1
700 PCOPY U TO V: PCOPY U+1 TO V+1: W=V:
V=U: U=W
710 PMODE1, U: PUT(X, 168)-(X+21, 188), B,
PSET
720 NEXT
725 PMODE1, V: SCREEN1, 1
730 FOR Y=184 TO 22 STEP-4
740 LINE(68, 188)-(68, Y), PSET: NEXT
750 FOR X=68 TO 200
760 LINE(X, 22)-(X, 110), PSET: NEXT
770 DRAW"C5; BM78, 37; R19; D2; B; D8; N; L9;
D9; L19; U19; B; R30; R19; D10; N; D9; L18; N;
D9; U10; B; R30; N; F10; N; D19; B; R19; N; G10;
D19; B; R11; N; R19; U9; N; R9; U10; R19"
780 DRAW"BM78, 76; R19; D19; L19; U19B; R30;
D10; F9; R1; E9; U10; B; R11; N; R19; D9; N; R9;
D10; R19; B; R11; U19; R16; F3; D6; G3; L5; N; F8;
L11"
790 PLAY"V20; T15; L1; O4; C; L2; O3; G; P5; G;
L1; A; G; P1; B; O4; C"
800 FOR T=1 TO 500: NEXT
820 CLS: PRINT@106, "YOUR SCORE":
PRINT@137, "**********": PRINT@172, SC
840 PRINT@321, "PRESS YOUR JOYSTICK
BUTTON TO  START NEXT GAME"
850 SCREEN0, 1
860 IF PEEK(65280)=127 OR PEEK(65280)=255
THEN 860
870 CLEAR50: GOTO10
1000 SC$=STR$(SC): LINE(243, 0)-(255, 191),
PSET, BF: GOSUB6000
1010 FOR T=1 TO LEN(SC$)
1020 J$=MID$(SC$, T, 1): IFJ$=" "THEN 1050
1030 IF J$="-"THEN D=10 ELSE D=VAL(J$)
1040 J1=T*14: J2=J1+10: ON D+1 GOSUB 1070,
1072, 1074, 1076, 1078, 1080, 1082, 1084, 1086,
1088, 1090
1050 NEXT
1060 RETURN
1070 PUT(244, J1)-(254, J2), N0, PSET: RETURN
1072 PUT(244, J1)-(254, J2), N1, PSET: RETURN
1074 PUT(244, J1)-(254, J2), N2, PSET: RETURN
1076 PUT(244, J1)-(254, J2), N3, PSET: RETURN
1078 PUT(244, J1)-(254, J2), N4, PSET: RETURN
1080 PUT(244, J1)-(254, J2), N5, PSET: RETURN
1082 PUT(244, J1)-(254, J2), N6, PSET: RETURN
1084 PUT(244, J1)-(254, J2), N7, PSET: RETURN
1086 PUT(244, J1)-(254, J2), N8, PSET: RETURN
1088 PUT(244, J1)-(254, J2), N9, PSET: RETURN
1090 PUT(244, J1)-(254, J2), NN, PSET: RETURN
1999 'SHIP'S MOVEMENT SUB
2000 FOR SM=1 TO 2
2005 PUT(1, YG)-(22, YG+20), B, PSET
2010 J=JOYSTK(0): J=JOYSTK(1)
2020 IF J<6 THEN YG=YG-21: IF YG<0 THEN
YG=0
2030 IF J>58 THEN YG=YG+21: IF YG>168 THEN
YG=168
2040 PUT(1, YG)-(22, YG+20), G, PSET: SCREEN1,
1
```

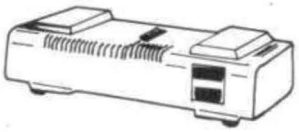
```
2050 PCOPY U TO V: PCOPY U+1 TO V+1: W=V:
V=U: U=W
2060 PMODE1, U: NEXT
2064 'PLACING OF MINES
2065 R1=RND(15): IF R1=3 THEN PUT(X,
Y)-(X+21, Y+20), M, PSET: GOTO 2080
2070 PUT(X, Y)-(X+21, Y+20), B, PSET
2080 RETURN
2999 'SHIP'S SHOOTING SUB
3000 PP=0: COLOR5, 7: FOR XS=33 TO 231
STEP22
3010 P=PPOINT(XS, YG+10): ON P-4 GOTO 3030,
3050, 3020, 3080
3020 NEXT XS
3022 LINE(24, YG+10)-(231, YG+10), PSET:
PLAY"T255; V20; O4; 12; 10; 8; 6; ; 11; 9; 7"
3024 LINE(24, YG+10)-(231, YG+10), PRESET:
RETURN
3030 LINE(24, YG+10)-(XS-2, YG+10), PSET:
PLAY"T255; V20; O4; 12; 10; 8; 6; 11; 9; 7"
3040 LINE(24, YG+10)-(XS-12, YG+10), PRESET:
IF C1<(INT(SC/1000)*5)+5 THEN RETURN
3042 PUT(XS-10, YG)-(XS+11, YG+20), B, PSET:
C1=C1-1: RETURN
3050 LINE(24, YG+10)-(XS, YG+10), PSET:
PLAY"T255; V20; O4; 12; 10; 8; 6; 11; 9; 7"
3060 LINE(24, YG+10)-(XS, YG+10), PRESET
3065 SCREEN1, 0: PLAY"T50; V31; O1; 12; 11; 10;
9; 8; 7; 6; 5; 4; 3; 2; L1; 1; 1; 1; 1; 1"
3070 FOR T=1 TO 100 STEP 2
3072 CIRCLE(XS, YG+10), T, 5
3074 NEXT
3076 GOTO670
3080 LINE(24, YG+10)-(XS, YG+10), PSET:
PLAY"T255; V20; O4; 12; 10; 8; 6; 11; 9; 7"
3082 LINE(24, YG+10)-(XS, YG+10), PRESET:
SC=SC+Y
3084 PUT(X, Y)-(X+21, Y+20), C, PSET: C1=C1+1:
SOUND100, 1
3086 GOSUB 1000: GOTO570
3999 'ALIEN'S TRACKING & SHOOTING SUBS
4000 IF RS<>Y THEN RETURN
4010 YS=YG: Q=2: TIMER=0: RETURN
5000 IF TIMER<TM THEN RETURN
5010 PCOPY U TO V: PCOPY U+1 TO V+1:
PMODE1, V: SCREEN1, 1: COLOR5, 7
5020 LINE(X+10, Y+13)-(10, YS+10), PSET:
PLAY"T255; V20; O2; 12; 9; 6; 3; 12; 9; 6; 3"
5030 Q=1: PMODE1, U: SCREEN1, 1: COLOR5, 7:
TM=TM-5: IF TM<20 THEN TM=20
5035 IF YS<>YG THEN RETURN
5040 FOR T=1 TO 5: PUT(1, YG)-(22, YG+20), G,
PSET
5043 PLAY"T255; V31; O1; 6; 5; 4": PUT(1,
YG)-(22, YG+20), B, PSET: NEXT
5050 L=L-1: GOSUB7000: IF L<=0 THEN GOTO660
5060 YG=168: PUT(X, Y)-(X+21, Y+20), B, PSET:
GOTO530
5999 'NUM. OF ALIEN LANDED
6000 COLOR5, 7: LINE(243, 174)-(255, 191), PSET,
BF: IF AE=0 THEN PUT(243, 175)-(253, 185), N0,
PSET
6010 IF AE=1 THEN PUT(243, 175)-(253, 185), N1,
PSET
6020 IF AE=2 THEN PUT(243, 175)-(253, 185), N2,
PSET
6030 IF AE=3 THEN PUT(243, 175)-(253, 185), N3,
PSET
6040 RETURN
6999 'NUM. OF SHIPS LEFT
7000 COLOR 5, 7: LINE(0, 0)-(14, 14), PSET, BF
7010 IF L=0 THEN PUT(2, 2)-(12, 12), N0, PSET
7020 IF L=1 THEN PUT(2, 2)-(12, 12), N1, PSET
7030 IF L=2 THEN PUT(2, 2)-(12, 12), N2, PSET
7040 IF L=3 THEN PUT(2, 2)-(12, 12), N3, PSET
7050 RETURN
```

From Computer Plus to YOU . . .

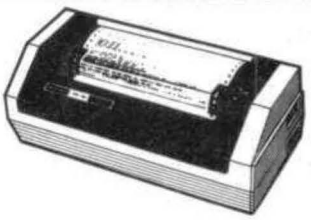
PLUS *after* PLUS *after* PLUS



Color Computer Disc Drive
$470



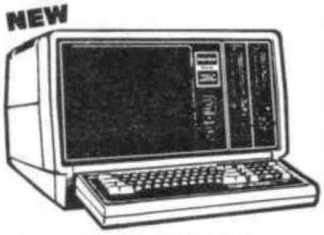
32K Exp. Int. $334



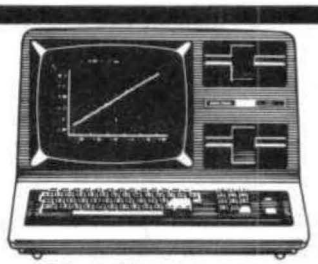
Line Printer VII $315

Color Computer 4K $310
w/16K Ext. Basic $459
w/32K Ext. Basic $525

NEW



Model 16 128K
1 Drive $4299



Model III 16K $839
Model III 48K
2 Disc & RS232C $2059

BUY DIRECT *Here are just a few of our fine offers . . . call TOLL FREE for full information.*

COMPUTERS	
Model II 64K	$3300
Model III 4K LEV I	599
MODEL III 16K	839
MODEL III 32K	945.50
*MODEL III 32K	881.50
MODEL III 48K	1052
*MODEL III 48K	924
Model III 48K 2 Disc & RS232 c	2059
Color Computer 4K	310
Color Computer 16K	416.50
*Color Computer 16K	352.50
Color Computer 16K w/extended basic	459
Color Computer 32K w/extended basic	525
Color Computer Drive 0	470
Pocket Computer	189
VIDEOTEX	310

*Computer Plus New Equipment, with NEC RAM installed.
180 Day Computer Plus Warranty.

PERIPHERALS	
Expansion Interface OK	$249
Expansion Interface 16K	355.50
*Expansion Interface 16K	291.50
Expansion Interface 32K	462
*Expansion Interface 32K	334
16K RAM N.E.C. 200 N.S. chips	25
MODEMS	
Lynx Direct Connect MI/MIII Auto Ans./Dial	235
Telephone Interface II	169
R.S. Modem I D.C.	130
R.S. Modem II D.C.	210
PRINTERS	
Daisy Wheel II	1695
Epson MX80	479
Epson MX80 FT	589
Epson MX100	759
Line Printer VII	315
Line Printer VIII	620
Line Printer V	1610
Microline 80	345
Microline 82A	479
Microline 83A	745
Microline 84 Parrabel	1090
Pocket Computer Printer	130
DISK DRIVES	
R.S. Model III 1ST-Drive	712
Tandon 40 Track MI	329
R.S. 1 Drive Exp MII	999
R.S. 2 Drive Exp MII	1518
R.S. 3 Drive Exp MII	2040
SOFTWARE	
R.S. Software 10% off list Newdos 80 MIII	149
ST80III	149
ETC.	
Verbatum 5" Double Density	32
Verbatum 8" Data Life	49.95
Ctr-80A recorder	52
C. C Joysticks	22

We have the lowest possible Fully Warranteed Prices AND a full complement of Radio Shack Software.

call TOLL FREE **1-800-343-8124**

computer plus

245A Great Road
Littleton, MA 01460
617-486-3193

Write for your free catalog

DEALER INQUIRIES ARE INVITED

Prices subject to change without notice.
Not responsible for typographical errors.

TRS-80 is a registered trademark of Tandy Corp.

MasterCard VISA

Good-by DRAW and PSET Good-by LINE and COLOR
(ENTER) MOTION PICTURE PROGRAMING (MPP)
by Superior Graphic Software

An exciting new concept for producing hi-resolution graphic art in motion has been aptly named Motion Picture Programing (MPP) by its creator, Arnold Pouch who hides out in the hill country of North Carolina with his small but powerful TRS-80C. It has been reported that "MPP may well be the best thing to come out of the Mountains since Moon-shine". We believe it!

MPP will enable all you CC users to produce superior action graphics with an ease of programing you have never before enjoyed. MPP does away with reserving 6- 8- 10 thousand blank memory positions where old fashioned programs laboriously DRAW their pictures after loading the program. Seldom, if ever, does MPP use in its programs the difficult to handle and memory consuming commands DRAW, LINE, COLOR, PCLS, PSET, CIRCLE, PAINT, ect.. Doing away with these cumbersum, mechanical and difficult to handle blind commands, returns graphic programing to what it should have always been - artistic, creative, and most of all fun. All this and even more, MPP tapes load and auto-start with just one (ENTER).

MPP is simple, fast and easy to use but rather difficult to explain. Superior Graphic Software is therefore providing Color Computer News with all the necessary details to produce MPP.

Four essential TOOLS are needed to produce MPP; The MPP 3 step programing methods (TOOL 4) are provided by Superior Graphic Software; A good DRAWER program (TOOL 3) such as the one available from Chromasette Magazine's August 1981 Issue. If you do not already have that DRAWER program, single tape cassettes are available at $5.00 per back issue (see Chromasette advertisement), or it is available as part of the complete MPP Tutorial Package from Superior Graphic Software (see advertisement). Instructions to convert Chromasette's DRAWER to an MPP DRAWER (16K, 32K, 32K DOS) are listed elsewhere in this issue of Color Computer News.

TOOL 1 and TOOL 2 cannot be provided for you and you cannot purchase them. They are, your own good graphic ideas and your personal artistic ability to carry them out. We will, however, in the following months provide you with a fairly simple but complete hi-res 256x191 action graphic layout which you will be able to copy using your MPP DRAWER, but that's down the road a way. If you're ready to go back to school, let's get started learning MPP.

A Short Course In Motion Programing (MPP)
INTRODUCTION:

You haven't even started yet and already I am going to give you some home work to do. Read the full DRAWER instructions contained in a seperate special program in Chromasette's August 1981 Issue named "Drawer Instructions". Please familiarize yourselves with them as much as time will allow. Later, if you forget these instructions, the MPP DRAWER has an "H" (Help!) key which will display a two page menu type summary of all the MPP DRAWER key functions, a list of the current settings of all your graphic commands, a display of the last command you gave and the exact X and Y (0-255 and 0-191) location of your drawing curser (your drawing tool).

To help simplify these MPP instructions we will assume that you are drawing in PMODE 3,1 on a 4 page MPP picture with 2 extra pages of Motion Modifiers (more about MMs later). Your 4+2 pages = 6 pages of (PCLEAR 6). All referances to higher page numbers, higher PCLEARs, higher memory addresses for 32K Extended and 32K DOS will be omitted. Them that has 'em - knows how to use 'm (I hope).

Before our first class let's go over the MPP schematic diagram and flow chart which we will be following. It is layed out in three sections. A MPP Memory Map is in the center. On the left side of the Memory Map are the three steps necessary to produce MPP and the four TOOLs required. On the right side of the Memory Map are the I/O routines which will allow you to stop or start your MPP drawing or programing any time you want (after all Rome and superior graphics programs weren't built in a day).

The connecting lines between your flow charts and the memory map have been removed for clairity's sake. They were cris-crossing all over the place. Instead we have given the exact memory addresses for START and END above each I/O box. Follow that? Good! Then let's get started. The Motion Picture Programing School is now in session.

STEP I - DRAW:

1. Place your MPP DRAWER (TOOL 3) in your tape drive and PCLEAR 6; CLOAD (ENTER) it, RUN (ENTER) it.

2. What you see is blank memory (pages 1 thru 4, 1536 thru 7679). It's a mess. Press the K (KLEAR) key and you will see a klear green screen. Now press P (PAGE) key and then 3. You have switched your TV viewing to pages 3 thru 6 (4608 thru 10751). Pages 5 and 6 are still a mess so press the K (KLEAR) key again. Ok, all is klear so press P and 1 to go back to pages 1 thru 4.

3. Well, don't just sit there - DRAW - something! Don't use the JOYSTK option. They are to difficult to control for the detailed superior graphics I want you to produce. Use the ARROW keys (they will

single step or repeat move if held down). Draw in hi-res PMODE 3 or 4, anything lower is a waste of time. What's that? You forgot the drawing commands! This teacher is not going to give them to you. Press the H (HELP!) key and review them.

4. I still don't see any picture - so NOW is the time to use TOOL 1 and TOOL 2. Draw anything simple or complex, single-color or multi-color, large or small I don't care. Try a round faced clock. Ah, that's better. Keep up the good work. When you are finished you can go on to instruction #5.

5. Let's now draw what we called Motion Modifiers, MM. MMs are simply repetitive drawings of a certain section of your picture where you want action to take place. The repetitive drawings are each a little different from the previous one and show a "time/motion" relationship like 10 seperate pictures of a clock hand moving from 15 minutes past the hour to 25 minutes past the hour. Motion Modifier MM #1 shows the clock hand at 16 after; MM #2 shows the clock hand at 17 minutes after; etc., etc., until MM #10 shows the clock hand at 25 minutes after.

6. Sounds like a lot of tedious drawing work? It is, but your MPP DRAWER will do most of it for you automatically! We don't want the MMs to be visable in our picture (pages 1 thru 4) so press P and 3. Now press I (IMAGE COPY) and follow the menued instructions to copy and move as many MMs as you think you will need for smooth animation. When you P (PUT) your MMs on pages 5 thru 6, set them in neat rows so later your program will be able to GET them with just one GET command inside a FOR - Next loop.

7. Ten pictures of a section of a clock all showing 15 minutes after the hour. Move your cursor in there and change each picture to the correct time - 16 after to 25 after. If you have a complex picture you can use as many MMs as you want 'til you fill up pages 5 and 6.

8. Tired? Want to stop now? Just one more thing before we finish with our MPP picture and our MMs. Use your H command and your cursor to find the exact X-Y coordinates for all your GET MMs and their X-Y PUT coordinates in your MPP picture. Write them down neatly as follows;

MM NAME GET (X,Y)-(X,Y) P number (PMODE)
MPP NAME PUT (X,Y)-(X,Y) P number (PMODE)
etc.

9. We can stop the processes for STEP I at any point by pressing the > key and following the menu instructions. The MPP DRAWER program will automatically produce an MPP PICTURE TAPE. These tapes can be produced any time and in any quantity. Note the direction of the MPP I/O keys. > is pointing out. < is pointing in. Easy to remember without having to call for H (HELP!).

10. Test yourself now and it will make you feel more secure about shutting down your computer for the day. Draw a horizontal line all the way thru your picture. Messed it up? Not for long. Press < and follow the menu instruction. The MPP DRAWER will automatically load you MPP PICTURE TAPE back into memory. You can watch it happen. See the line you drew thru your picture disappearing . Actually your whole memory is being replaced with the contents of your MPP TAPE. Want to check that out? Press K and get a klearscreen. Now press < and watch your whole picture redevelop.

STEP II - PROGRAM:

1. You MUST have produced a MPP PICTURE TAPE in STEP I before you can start STEP II. This BASIC programing and debugging process will mess up your original picture as it GETs and PUTs your MMs. You may want to reload your original picture several times during the programing and in STEP III you must have a clean original picture to produce your final finished MPP tape.

2. Before you start your programing you may want to type in these little utilities;

```
9998 GOTO 9998 'This will hold the program you are)
9999 END 'writing on the screen.)
10000 PCLEAR6:PMODE3,1: SCREEN 1,0: CLOADM
'RUN10000 (ENTER) will visably load your MPP
PICTURE tape any time you
10001 GOTO 10001 'want a fresh picture.
20000 PMODE3,1:SCREEN1,0: GOTO 20000 'RUN20000
(ENTER) will display your picture any time.
30000 PMODE 3,3: SCREEN1,0: GOTO 30000
'RUN30000 (ENTER) will display your MMs any time
you want.
```

3. Start ALL of your MPP BASIC programs with the following programs:

```
1 '      MPP BY (YOUR NAME)
2 '       SPRING OF 1982
3 '
4 'POKE 27,XXX: POKE 28,XXX: POKE 29,XXX:
POKE30,XXX: POKE31,XXX: POKE 32,XXX
5 ' PMODE3,1: SCREEN1,0
```

You can RUN (ENTER) the above and see your MPP PICTURE again while you are programing and debugging. Your programs start at line 6.

4. Use normal BASIC language and techniques BUT what a blessing! No long DRAW strings to fumble with and try to set blindly. No more worrying and

fumbling with LINEs. No more time consuming figuring where PSETS and PRESETS should go when you can't see 'm. No more COLOR, PCLS, SCREEN or program stopping PAINTs. What a blessing - your graphic work was all completed in STEP I. Your program only needs GETs and PUTs and you already have them all written down, with their exact addresses. So have at it. Write your program. Good Luck!

5. Here are a few helpful BASIC programing hints;

a. Try using only one DIMensioned array for all your GETs and PUTs. Just use it over and over again.

b. Write all your GETs and PUTs inside FOR - NEXT loops to step off the motion.

c. Write your MMs at maximum speed and slow them down later to the visual effect you want with time delay loops, PLAY commands, even GET/PUT loops within GET/PUT loops.

d. Make your GET/PUTs as small as possible and still do the job.

e. If you PUT garbage, remember GET and PUT must be the exact same size. The array doesn't matter as long as it's big enough for your largest GET/PUT.

f. DIMension your arrays big enough while programing then EDIT them down to exact size when you're finished.

g. Keep your PMODE page starts set correctly. Changing them will not effect your picture, only SCREEN can do that.

h. I could go on and on with BASIC tips but this is a lesson on MPP. If you want more and in more detail let Bill Sias know.

6. If your programing is finished or you just want to take a break for a day or so, you can CSAVE"MPPBASIC" (ENTER) any time you want to stop. You will now have two tapes saved; MPP PICTURE and MPP BASIC.

7. You can restart from this point at any time by reloading the two tapes as follows;
MPP PICTURE tape - PCLEAR6; CLOADM (ENTER)
MPP BASIC tape - - CLOAD (ENTER)
You are all set again to either continue programing or, if all your work is finished, go on to STEP III.

STEP III - COMBINE;

1. When your MPP PICTURE tape and MPP BASIC tape are completely to your satisfaction, load them both into memory (as above, STEP II, 7). DO NOT RUN OR EXEC THE PROGRAM.

2. This is TOOL 4 - - This is MPP programing.

a. Your MPP PICTURE resides in memory from 1536 to 10751.

b. Your MPP BASIC resides in memory STARTING at 10752 (because we PCLEAR 6 before it was loaded). The END of your program is at ?????. That is what we have to find out.

3. Follow these steps carefully!
Type ?PEEK(27) the answer will be your A value.
Type ?PEEK(28) the answer will be your B value.
Type ?HEX$(A) the answer will be your A$ value.
Type ?HEX$(B) the answer will be your B$ value.
Type ?&HAB the answer will be your AB value.
You now have everything you will need to finish your MPP.

4. EDIT4 Remove the '(REM) and change the POKES to; POKE27,A; POKE28,B; POKE29,A; POKE30,B; POKE31,A; POKE32,B

5. Type CSAVEM "MPPNAME", 1536, AB, 44661
Ready your cassette with fresh tape and set it to record your

FINISHED MPP TAPE!!

press (ENTER)

CONGRADULATIONS YOU HAVE GRADUATED!!
You are a Motion Picture Programer. Superior Graphic Software authorizes you to place the letters MPP after your name.
GRADUATION EXERCISE;

Place your finished MPP tape in the cassette. Power up. Type PCLEAR6; CLOADM; EXEC (ENTER). Your MPP tape will load and auto-start, so sit back, relax and behold your MPP graphics in motion.
POST GRADUATE COURSE;
PCLEAR6 sets the start address (10752) for your program into BASIC scratch pad memory.
CLOADM sets the MPP picture and program in memory from 1536 to AB.
EXEC executes at 44661 which tells the machine language CLOADM to GOTO the BASIC RUN command.

Program step 1 pokes the program END A B into the BASIC scratch pad memory
Program step 2 sets up your programs variables, strings and arrays starting at memory location AB+1
Program steps 3 and 4 GET your MMs from graphic pages 5 and 6 and PUT them on graphic pages 1 thru 4 where you can see them in motion.

It was a pleasure doing this article on the conversion of Chromasette's excellent DRAWER program (Chromasette Magazine, August 1981 Issue)

for usage as a Motion Picture Programing, MPP DRAWER, because while doing it I finally realized that our CC has grown up. It is necessary to provide you with three sets of conversion instructions!

I - 16K EXTENDED BASIC
II - 32K EXTENDED BASIC
AND III - 32K DISK OPERATING SYSTEM BASIC (DOS)

I can remember just a short 18 months ago we Old Time (Ha! 1 1/2 years old time) CC buffs were concerning ourselves with whether we had 4K or 16K plain BASIC - look at us now, 32K DOS, with 1 to 4 disks, providing on line memory up to almost 3/4 MEGABITES - wow - what next?

For you expanding CC users here are the conversions - I, II, III!

I - 16K EXTENDED BASIC
EDIT 10 Change CLEAR 10 to CLEAR 20
EDIT 14 Remove PCLS! (in fact never ever use PCLS You'll lose your MPP)
EDIT 16 Remove FOR - - - - NEXT!
EDIT 20 Insert after A$=INKEY$! IF A$="(" THEN 250 ELSE IF A$=")" THEN 280 ELSE
EDIT 228 Delete PRINT "TAPE"! PRINT" SAVE OR LOAD"!

Insert PRINT "MPP TAPE INPUT/OUTPUT"! PRINT" INPUT =("! PRINT" OUTPUT =)"!

ADD 250 CLS! PRINT! PRINT"TO LOAD MPP PICTURE TAPES"! PRINT! PRINT "READY YOUR MPP TAPE IN CASSETTE"! PRINT! INPUT "PRESS (ENTER) WHEN READY";ZZ
ADD 252 SCREEN1, SC! CLOADM! GOTO20
ADD 280 CLS! PRINT! PRINT"TO SAVE MPP PICTURES TO TAPE"! PRINT! PRINT"READY YOUR CASSETTE WITH FRESH TAPE AND PRESS (RECORD/PLAY), DON'T FORGET TO CHECK THE VOLUME"
ADD 282 PRINT! PRINT "SAVE HOW MANY GRAPHIC PAGES?
4 - PAGES?
5 - PAGES?
6 - PAGES";ZZ
ADD 284 PRINT! INPUT "TYPE 8-CHR$ NAME OF MPP PICTURE ";ZZ$
ADD 286 CLS! PRINT@236, "STAND-BY"
ADD 288 CSAVEM ZZ$, 1536, 1536+(1536*ZZ)-1, 44553
ADD 290 SCREEN1, SC! GOTO20
Delete DEL 156-168 DEL 170-178

II - 32K EXTENDED BASIC
EDIT 10 CHANGE PCLEAR6 to PCLEAR8
EDIT 10 REMOVE CLEAR 20
EDIT 12 CHANGE MP=6 to MP=8
EDIT 226 CHANGE BOTH 1-6 to 1-8
EDIT 282 CHANGE and ADD 6 - PAGES?
7 - PAGES?
8 - PAGES";ZZ

III - 32K DISK OPERATING SYSTEM BASIC (DOS)
EDIT 16 CHANGE 1536 to PEEK(188)*256
EDIT 252 CHANGE CLOADM to LOADM
EDIT 288 CHANGE CSAVEM ZZ$, 1536, 1536 + (1536*ZZ) -1, 44553 TO SAVEM ZZ$, PEEK (188)*256, PEEK (188)* 256+ (1536*ZZ) -1, 44553

DOS MPP DRAWER program whips your MPP pictures on and off the screen in less than TWO SECONDS! You can reference dozens of MPP pictures and copy any part or all of them in your current MPP picture project in less time than it took to read this sentence. Now you CC buffs should all run out and buy 32K DOS - don't we all wish it!

Continued on page 83

TELEWRITER

Provides your COLOR COMPUTER with:

REAL LOWER CASE CHARACTERS ■ A POWERFUL TEXT FORMATTER
51 COLUMN × 24 LINE SCREEN DISPLAY ■ SPECIAL DRIVER FOR EPSON MX-80
ADVANCED CASSETTE HANDLING FEATURES ■ A SOPHISTICATED FULL-SCREEN TEXT EDITOR

and requires absolutely no hardware modifications

TELEWRITER

Telewriter is a powerful word processor designed specifically for the Color Computer. It can handle almost any serious writing job and it is extremely easy to use. It has all the advanced features you need to create, edit, store, format and print any kind of text. With Telewriter you can quickly produce perfect, finished copy for letters, reports, term papers, articles, technical documentation, stories, novels, screenplays, newsletters. It is also a flexible and efficient way to take notes or organize ideas and plans.

51 × 24 DISPLAY

The Color Computer is an incredibly powerful and versatile computer, but for text editing it has some major drawbacks. The small 32 character by 16 line screen format shows you too little of the text and, combined with its lack of lower case letters, bears little resemblance to the way text really looks on the page. Reverse video in place of lower case just adds confusion.

Telewriter eliminates these shortcomings with **no hardware modifications required**. By using software alone, Telewriter creates a new character set that has **real lower case letters**, and puts 24 lines of 51 characters on the screen. That's more on-screen characters than Apple II, Atari or TRS-80 Model III. That's more than double the Color Computer's standard display.

FULL SCREEN EDITOR

The Telewriter editor is designed for maximum ease of use. The commands are single key (or single key plus control key), fast, and easy to remember. There is no need to switch between insert modes and delete modes and cursor movement modes. You simply type. What you type is inserted into the text at the cursor, on the screen. What you see on the screen is always the current state of your text. You can move quickly through the text with one key cursor movement in all 4 directions, or press the shift key simultaneously for fast, auto-repeat. You can jump to the top or bottom of the text, the beginning or end of a line, move forward or backward a page at a time, or scroll quickly up or down. When you type past the end of a line, the wordwrap feature moves you cleanly to the next.

You can copy, move or delete any size block of text, search repeatedly for any pattern of characters, then instantly delete it or replace it with another. Telewriter gives you a tab key, tells you how much space you have left in memory, and warns you when the buffer is full.

FORMAT FEATURES

When it comes time to print out the finished manuscript, Telewriter lets you specify: left, right, top, and bottom margins; line spacing and lines per page. These parameters can be set before printing or they can be dynamically modified during printing with simple format codes in the text.

Telewriter will automatically number pages (if you want) and automatically center lines. It can chain print any number of text files from cassette without user intervention. You can tell it to start a new page anywhere in the text, pause at the bottom of the page, and set the Baud rate to any value (so you can run your printer at top speed).

You can print all or any part of the text buffer, abort the printing at any point, and there is a "Typewriter" feature which allows you to type straight to your printer. Because Telewriter lets you output numeric control codes directly (either from the menu or during printing), it works with any printer. There's even a special driver for the Epson MX-80 that lets you simply select any of its 12 fonts and do underlining with a single underline character.

CASSETTE HANDLER

Telewriter makes cassette as simple to use as possible. It will search in the forward direction til it finds the first valid file, so there's no need to keep retyping a load command when you are lost in your tape. You can save all or any part of the text buffer, and you can append pre-existing files to what you have in the buffer already. You can abort an append or filesearch without harming the program or the text in the buffer.

Telewriter will maintain compatibility with popular Color Computer disk systems, but, since it makes using cassette almost painless, you can still have a powerful word processor without the major additional cost of a disk.

AVAILABLE NOW

Telewriter turns your Color Computer into the lowest cost hi-power word processor in the world today. It runs in 16K or 32K (32K recommended) and is so simple you can be writing with it almost immediately. It comes with complete documentation and is fully supported by Cognitec. Telewriter costs $49.95 (California residents add 6% tax). To order or request more information write:

Cognitec
704 Nob Ave.
Del Mar, Ca. 92014

Or call (714) 755-1258 (weekdays, Saturdays, and early morning). We will gladly answer your questions.

MORE ON 64K

By Frank Hogg

Last month we showed you how to access the other 32K in your Color Computer bringing it up to the full 64K. This month we're going to discuss some additional uses for that memory, plus a program to copy the ROM Basic into RAM and run it there; but first, a word about last months program to test the new memory.

The program tests memory and reports any errors. It will report what it wrote to memory and what it found there. The bits in error refer to the memory chips as follows; Bit 0 is U20, 1 is U21, 2 is U22, 3 is U23, 4 is U24, 5 is U25 6 is U26 and 7 is U27. If the program wrote FF and read FE the error is bit 0 and the bad memory chip is U20.

On to more interesting stuff. Now that you have 64K, just what do you do with it? Well... you could buy FLEX and/or OS-9 from the good folks at FRANK HOGG LABORATORY, Inc. (a little plug) That would sure open up a lot of very powerful software to you.

But how about copying Basic from ROM to RAM! The accompanying program does just that. Once you have Basic in RAM you can change it, and if you run the program you will see that we do just that to demonstrate that capability. This means that besides changing Basic for fun, you could also fix the bugs, enhance and enlarge it -- the sky is the limit. What other computer is there that allows you to change what is in the ROM?

How about changing the display that Color Basic keeps in low memory and put it in the almost 8K above the disk Basic (&HE000 to &HFFE0). That would open up the lower 32K for program storage.

Or, how about changing the disk software to allow double sided 40 track drives. Maybe modify the copy utility in ROM to a single disk copy like the backup command. With a RAM system, anything is possible given enough time and desire.

You probably know by now that much of the cassette software won't work with the RS Disk system because it uses low memory and many of these programs use the same memory. As you can imagine, this has caused a lot of headaches for the sellers of cassette software. With the ability to move Basic to RAM, all you have to do is move just the lower 16K (the non-disk ROM) to RAM with a copy of the cassette software and then just jump to that Basic. The system will run as a non disk system, but when you exit, you will go back to the ROM and have the disk. I'll let you figure out how to do it. The accompanying program should give you a good start.

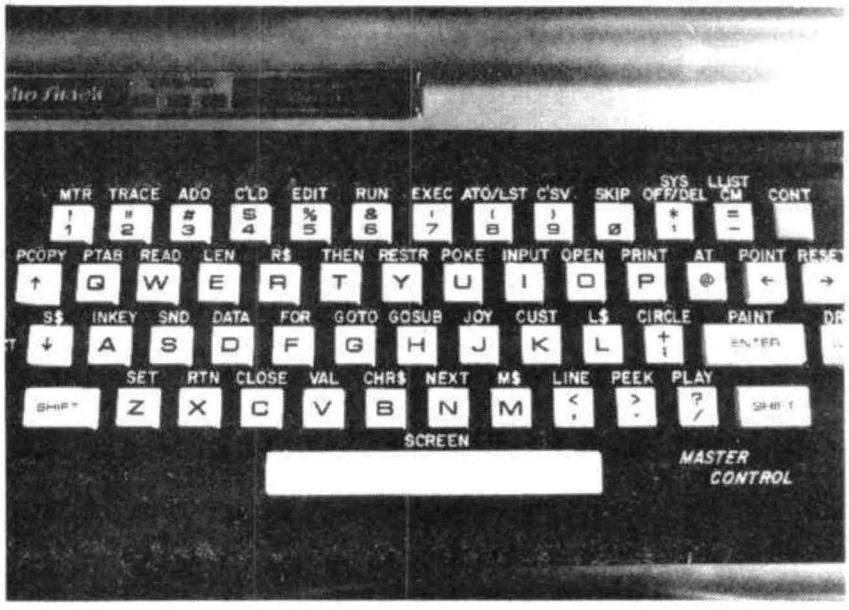
```
1000 ' THIS PROGRAM MOVES
1010 ' RADIO SHACK COLOR BASIC,
1020 ' EXTENDED COLOR BASIC, AND
1030 ' DISK EXTENDED COLOR BASIC
1040 ' FROM ROM TO RAM IN A
1050 ' MODIFIED 32K TRS80
1060 ' COLOR COMPUTER. WHEN THE
1070 ' PROGRAM TERMINATES, BASIC
1080 ' WILL BE RUNNING IN RAM
1090 ' AND THE ROM WILL BE
1100 ' DISABLED.
1110 '
1120 ' REQUIRES EXTENDED COLOR
1130 ' BASIC VERSION 1.0.
1140 '
1150 ' REQUIRES ABOUT 20 SECONDS
1160 ' TO SET UP THE MACHINE
1170 ' LANGUAGE PROGRAM
1180 ' BEFORE RUNNING.
1190 '
1200 ' RESERVE RAM
1210 '
1220 CLEAR 256,&H7EFF
1230 '
1240 ' SET UP MACHINE LANGUAGE
1250 ' PROGRAM
1260 '
1270 GOSUB 2140 : SA=H
1280 GOSUB 2140 : EA=H
1290 GOSUB 2140 : EP=H
1300 '
1310 ' SA=STARTING ADDRESS
1320 ' EA=END ADDRESS
1330 ' EP=ENTRY POINT
1340 '
1350 FOR A=SA TO EA
1360 GOSUB 2140
1370 POKE A,H
1380 NEXT A
1390 '
1400 DEFUSR0=EP
1410 '
1420 ' RUN THE PROGRAM
1430 '
1440 X=USR0(0)
1450 '
1460 PRINT "BASIC IS NOW IN RAM"
1470 PRINT "ROMS ARE DISABLED."
1480 PRINT
1490 '
1500 FOR I=1 TO 1000
1510 NEXT I
```

```
1520 '
1530 ' CHANGE SPELLING OF PRINT
1540 '
1550 POKE &HAA79,ASC("W")
1560 POKE &HAA7A,ASC("R")
1570 POKE &HAA7B,ASC("I")
1580 POKE &HAA7C,ASC("T")
1590 POKE &HAA7D,ASC("E")+&H80
1600 '
1610 PRINT " THE SPELLING OF"
1620 PRINT " 'PRINT' HAS NOW"
1630 PRINT " BEEN CHANGED. "
1640 PRINT
1650 PRINT " LIST 1610-1670 TO"
1660 PRINT " SEE FOR YOURSELF:"
1670 PRINT
1680 '
1690 FOR I=1 TO 1000
1700 NEXT I
1710 '
1720 LIST 1610-1670
1730 '
1740 END
1750 '
1760 ' MACHINE LANGUAGE PROGRAM
1770 '
1780 ' SA, EA, EP
1790 '
1800 DATA 7F00, 7F19, 7F00
1810 '
1820 ' THE MACHINE LANGUAGE
1830 ' PROGRAM TO MOVE BASIC TO
1840 ' RAM IS AS FOLLOWS:
1850 '
1860 '
1870 ' EP    ORCC #$50 DIS. INTS.
1880 '       LDX #$8000 1ST ADDR.
1890 ' LOOP  LDA ,X
1900 '       STA $FFDF MAP TYPE 1
1910 '       STA ,X+ IN RAM!
1920 '       STA $FFDE MAP TYPE 0
1930 '       CMPX #$FF00 LAST +1
1940 '       BNE LOOP
1950 '       STA $FFDF MAP TYPE 1
1960 '       ANDCC #$AF ENBL INTS
1970 '       RTS
1980 '
1990 ' ACTUAL CODE
2000 '
2010 DATA 1A,50
2020 DATA 8E,80,00
2030 DATA A6,84
2040 DATA B7,FF,DF
2050 DATA A7,80
2060 DATA B7,FF,DE
2070 DATA 8C,FF,00
2080 DATA 26,F1
2090 DATA B7,FF,DF
2100 DATA 1C,AF
2110 DATA 39
2120 '
2130 '
2140 ' READ A HEX NUMBER TO H
2150 '
2160 READ A$
2170 LZ=LEN(A$)
2180 H=0
2190 IF LZ<=0 THEN RETURN
2200 C$=LEFT$(A$,1)
2210 FOR I=0 TO 15
2220 IF I<>0 THEN 2240
2230 IF C$="0" THEN 2270
2240 IF C$=HEX$(I) THEN 2270
2250 NEXT I
2260 RETURN
2270 H=H*16+I : LZ=LZ-1
2280 A$=RIGHT$(A$,LZ)
2290 GOTO 2200
```

For Your Color Computer

MASTER CONTROL

Copyright ©1981 Soft Sector Marketing, Inc. – Written by A. Schwartz



Requries 16-32K

1. 50 preprogrammed command keys. Standard and Extended command.
2. Direct control of motor, trace, and audio from keyboard.
3. Automatic line numbering.
4. Programmable Custom Key.
5. Direct Run Button.
6. Keyboard overlay for easy program use.
7. Easy entry of entire commands into computer.

Load Master Control into your machine then either type in a BASIC program or load one in from tape to edit. Cuts programming time by 50% or more **$24.95**

For The Radio Shack
Color Computer*

COLOR BONANZA

Some 4K, Some 16K, Some 32K

Written by E.R.I.

50 PROGRAMS
In One Package

Distributed by:

SSM SOFT SECTOR MARKETING, INCORPORATED

*The Color Computer is a product of Radio Shack, division of the Tandy Corporation

Tape 1		
Keys	4k	NE
Bagels	4k	NE
Find	16k	NE
Darts	4k	NE
Motor	4k	NE
Bomber	4k	NE
Football	16k	EXT
Kapow	4k	NE
Dodge	4k	NE
Tape 2		
Bounce	16k	EXT
Tank	32k	EXT
One Arm	4k	NE
Chute	16k	EXT
Where is it	16k	EXT
Lunar Lander	16k	EXT
Stock Market	4k	NE
Tape 3		
Multiply	16k	EXT
Divide	16k	EXT
Add Sub	16k	EXT
Simple Simon	4k	NE
Hangman	16k	NE
Beast	16k	NE
Count Down	4k	NE
Acey	16k	NE
Genie	16k	NE
Protect	16k	EXT
Tape 4		
Graphics	16k	EXT
Songs	16k	EXT
Joy	16k	EXT
Mortgage	16k	EXT
Checkbook	16k	EXT
Draw 1	16k	EXT
Morris	16k	EXT
Sound	16k	EXT
Tape 5		
Ram	16k	MA
Trace	16k	MA
MMaster	16k	MA
Demo	16k	NE
Disassembler	16k	NE
Basbug	16k	NE
Ohmlaw	4k	NE
Convert	4k	NE
Drawer 2	32k	EXT
Degrees	4k	NE
Tape 6		
Hurdler	32k	EXT
Entrap	16k	EXT
Search	16k	NE
Flip Flop*	16k	EXT
Lost Atom	16k	EXT
Attack	16k	EXT
Cartel*	32k	EXT

List **$49.95**
6 Cassettes – Disk version available soon.

MA = Machine language
NE = Non Extended Basic
EXT = Extended Basic

C.O.D. – certified check, M.O. or cash only. Most orders shipped next day. All orders must have shipping included. Please add 2% or $2.50, whichever is higher for shipping. Michigan residents, please add 4% tax. Add extra $1.50 for C.O.D. Personal checks take 3 weeks to clear. Out of the country orders add $10.00 extra shipping.
*TRS-80 is a product of Radio Shack, div. of the Tandy Corp.

SSM SOFT SECTOR MARKETING, INCORPORATED
6250 Middlebelt • Garden City, Michigan 48135
Order Line **800-521-6504**
Michigan Orders & Questions **313-425-4020**

The American Express® Card. Don't leave home without it.™

MasterCard VISA

The Quality Continues . . .

REVIEW--"CAVE HUNTER"
by Ron Beatty

Your long journey initiated by your lust for gold has brought you to an eerie cavern. You peer into its interior only to see ominous darkness. An imagination is something you wish you did not prossess at this time. You know you must proceed with the task at hand but your mind envisions all forms of terror lurking just inside the caves' entrance. You have come seeking gold and you are known as the CAVE HUNTER!

Well maybe the description is a little unrealistic but for a gamer such as I, it's easy to place yourself in this situation. The game is call "CAVE HUNTER" and the people that created it are at MARK DATA PRODUCTS.

The game begins with the credits and a chorus of My Darlin' Clementine. This is followed by a well done display of flashing colors and the title with a sound effect that sound like...well, like a gun shot. Next, you find yourself at the entrance of the cave (top of the screen) with a maze of passages you can take to get to the bottom of the cave. At the bottom is the gold you seek. To be precise, there are four deposits of GOLD. The challenge is to traverse through the maze to the bottom, scoop up a deposit of gold and carry it back to the cave entrance. There you can leave the gold on the ground and go back in for the next deposit. Yes, you guessed it, you can only carry one deposit of gold at a time. You begin to accomplish this task by pressing the right joystick button. This will start the action. The CAVE HUNTERS' movements are also controlled by the right joystick.

This all seems relatively simple until you find out that you are not alone in the cave. There are three rather unfriendly cave creatures that will attempt to stop you "DEAD" in your tracks. Don't worry though, you have three hunters at the beginning of the game.

You begin play a little "green". Please excuse me but that is the color you are when you begin. When you are green, you are vulnerable to the cratures. To help you cope with this predicament, there are four red "POWER DEPOSITS". If you reach one of these deposits, you will be charged with power (you are now orange) and will be able to destroy the creatures. Be careful because your power only lasts for a short period of time.

If you manage to retrieve all of the gold you are rewarded with a display that is similar to the one produced at the beginning of the game. One of the features I like about this game is that you are also rewarded with a free game which means you have three hunters again. Some games are not quite as kind as this and I usually need all the help I can get.

As long as you continue to retrieve all of the gold deposits the play will automatically accelerate to keep you challenged. The scoring is as follows:

1. For each gold piece retrieved....200 Points
2. For each creature destroyed.....150 Points
3. All gold pieces retrieved.....FREE GAME

I might add that with the free game your score will continue to accumulate until you lose all three of your HUNTERS. At that point, it is time for the next player. My normal game is around 4,000 to 6,000 points. My highest was 13,750.

In the description of the game there was a statement made about "DOUBLE POINTS" occasionally flashing on the right side of the screen. While this occurs, you will receive double points (300) for each creature destroyed. Up to this time, I have not seen the flashing "DOUBLE POINTS". This could be due to the fact that I have either not lasted long enough or that I have been to busy trying to accomplish my goal.

Your current score is displayed at the top left of the screen. The highest score received will be displayed directly under the current score. This gives you a second goal to shoot for. Which is, trying to beat other HUNTERS' scores.

When I first played the game I was disappointed because I thought that it was too similar to a well known popular arcade game. Since then, my disappointment has disappeared. In fact it has turned into downright enjoyment.

There are two points that I would like to comment on. One is that if you are trapped at the entrance of the cave there is usually no escape. You can even get caught outside of the cave. Many times you have no choice. I would have liked a better chance to get back inside the cave. The second point is a tactical maneuver. You can allow yourself to get caught with a gold piece just outside of the creatures home which is in the center of the cavern. The gold deposit will remain wherever you're caught. It will also block the creatures from getting out of their home. Keep two things in mind here. This requires that you sacrifice one of your hunters and you must remember that there are two exits from the

creatures home. So far, I have not been very successful with this strategy.

I have two other MARK DATA PRODUCTS games. The frustrating COLOR BERSERK and the adventure game THE BLACK SANCTUM. I would like to mention here that I have been a total failure with COLOR BERSERK. The highest score reached is only 3500. THE BLACK SANCTUM is my first adventure. If this is what adventuring is all about, then this will definitely not be my last.

I believe that CAVE HUNTER is the type of products that I have come to expect from MARK DATA PRODUCTS. Challenging, frustrating and well done.

My thanks to Ron Krebs for the many hours, days, months, probably years of fine entertainment that my family will have.

NUCLEAR ATTACK
by Timo Talasmaa

This program is a game patterned after the popular arcade game, "Missile Command". It uses about 4K of memory and has 112 lines. It is in extended basic and uses pmode 4.

The object of the game is to protect your six cities from the incoming missiles. You will have a cannon to shoot-em-up (use the right joystick). The joystick controls a flashing dot on the screen. Position this dot on top (or close enough) of the falling bomb. Then use the fire-button.

I wanted to keep the listing as short as possible so there are only two rays coming in at a time. One could easily add one or two extra rays. The bombs will get faster and faster as the game proceeds. The game is over when all your cities are gone.

Important variables:

A$,B$ for DRAWing the cloud of explosion
J time delay for firing, keeps you from firing repeatedly. When J is greater than 0 fire-button is disabled
Q the rate of fall of the bomb (2.5-7.5)
PL$ sound of explosion
M,M1 # of the city the bomb is aimed at
L,L1 x-coordinate of city #M or M1
X(n) when X(n)=1, city #n is gone
UF when UF=1 the cannon is gone and can't be fired
R,R1 the x-coordinate of the place the bomb is coming from
H,K the coordinates of the cannon's target
NU # of cities destroyed

```
1 ' NUCLEAR ATTACK
2 ' BY TIMO TALASMAA-JAN. 1982
3 ' 879 QUINTINIA DR.
4 ' SUNNYVALE, CA 94086
5 ' FOR CCN READERS
10 PMODE 0, 1: PCLEAR 4: : PMODE 4, 1:
PCLS(5): CLEAR 1000
20 DIM C(20, 10): J=5: Q=2.5: Q$=CHR$(128):
GOTO 140
30 PCLS5: A$="U16 L4 H2 U1 H1 U2 E2 R3 U1 R3
U1 R5 F1 R3 F1 R2 D3 F1 D1 L1 D3 L1 D1 L4 D15
BL2 NU10 L2 NU11": B$="BR2 U18 BU2 L2 U1 L2
U2 BR4 F1 R1 BU1 BR2 L3 D1 L2 D1 BD3 BL2 D1
L2"
40 LINE(0, 176)-(255, 176), PRESET
50 PL$="T255; V31; O1; 1; 10; 1; 3; 4; 2; 5; 5; 6; 3;
5; 6; 4; 9; 8; V25; 7; 8; 1; 2; 2; 6; 9; 6; 7; 5; 1; 2; 1;
6; 6; 9; 3; 4; 8; 9; 6; 5; 4; 8; 5; 1; 9; 3; 5; 5; V20; 4;
6; 7; 5; 5; 8; 9; 1; 4; 5; 4; 3; 4; 3; 7; 5; 3; 4; 9; 1; 2;
3; 4; 9; 2; 3; V15; ; 2; 3; 4; 5; 9; 4; 1; 3; 2; 9; 1; 4;
1; 5; 9; 2; 1; 2; 9; 1; 2; V10; 1; 1; 2; 1; 9; 4"
60 DRAW"BM125, 175; C0; E2U2D2F2"
70 DRAW"BM10, 175; C0; U1 R2 U2 R2 D1 R2 U5
R3 D6 R1 U3 R1 D3 R2 U5 R2 D4 R2 D2":
PAINT(17, 172), 0, 0
80 GET(9, 167)-(28, 175), C, G: COLOR 5, 0: LINE
(10, 150) - (50, 175), PSET, BF: COLOR 0, 5
90 FOR X=1 TO 6: READ A
100 PUT (A, 167) - (A+19, 175), C, PSET
110 NEXT X
120 DATA 9, 44, 80, 152, 188, 224
130 COLOR 0, 5: SCREEN 1, 1: GOSUB 160: GOTO
170
140 CLS0: PRINT@168, "nuclear"; Q$; "attack"; :
PRINT @258, "press"; Q$; "right"; Q$; "joystick";
Q$; "button"; : PRINT@331, "to"; Q$; "begin";
150 IF PEEK(65280)=254 OR PEEK(65280)=126
THEN 30 ELSE GOTO 150
160 PLAY"T40O2; 1; 2; P2; 3; 4; P2; 5; 6; P2; 7; 8;
P2; 9; 10; P2; 11; 12; P2; O3; 1; 2; P2; 3; 4; P2; 5;
6; P2; 7; 8; P2; 9; 11; P2; O2; 4; 5; 4; 5; 4; 8; 9; P2;
8; 9; P2; 8; 9; P2; 8; 9; P2; 8; 9; P2; 8; 9; P2; 8; 9":
RETURN
170 M=RND(7): ON M GOTO 180, 190, 200, 210,
220, 230, 240
180 L=17: GOTO 250
190 L=53: GOTO 250
200 L=89: GOTO 250
210 L=127: GOTO 250
220 L=161: GOTO 250
230 L=197: GOTO 250
240 L=233
250 IF X(M)=1 THEN 170
260 R=RND(240)+12: X=R-L
270 IF UF=1 THEN Q=7.5
280 IF Q>7.5 THEN Q=7.5
290 Q=Q+.003: Y=Y+Q: LINE(R,
0)-(R-INT(Y*X/176), Y), PSET
300 IF Y<167 THEN 370 ELSE GOSUB 330
310 IF L=127 THEN UF=1: GOTO 560
320 GOTO 560
330 FOR P=-6 TO 6
340 LINE (ABS(R+P), 0)-(ABS(R+P-(Y*X/173)), Y),
PRESET
350 NEXT P
360 Y=0: RETURN
370 'AIMING
380 H=JOYSTK(0): H=4.05*H
```

```
390 K=JOYSTK(1): K=2.38*K
400 IF PPOINT(H, K)=0 THEN PP=1 ELSE PP=0
410 PSET(H, K)
420 IF UF=0 AND J<0 THEN 430 ELSE 490
430 E=PEEK(65280): IF E=126 OR E=254 THEN
E=1 ELSE E=0
440 IF E=0 THEN 490 ELSE LINE(127, 170)-(H, K),
PSET: J=5: FOR F=1 TO 4: CIRCLE(H, K), F, 0:
NEXT F
450 LINE(ABS(H-4), ABS(K-4))-(H+4, K+4),
PRESET, BF
460 IF H>R-INT(Y*X/173)-4 AND
H<R-INT(Y*X/173)+4 AND K>Y-4 AND K<Y+4
THEN 520
470 IF H>R1-INT(Y1*X1/173)-4 AND
H<R1-INT(Y1*X1/173)+4 AND K>Y1-4 AND
K<Y1+4 THEN 940
480 LINE(127, 170)-(H, K), PRESET
490 J=J-1: IF PP=0 THEN PRESET(H, K)
500 IF P2=0 THEN 740
510 GOTO 850
520 LINE(127, 170)-(H, K), PRESET: Q=Q+.1:
GOSUB 330
530 X(M)=0
540 FOR DL=1 TO 100: NEXT DL: LINE(H-5,
K-5)-(H+5, K+5), PRESET, BF
550 GOTO 170
560 L$=STR$(L-3)
570 IF X(M)=0 AND L<>127 THEN NU=NU+1
580 DRAW"BM"+L$+", 175"+A$+"BM"+L$+",
175"+B$
590 PLAY PL$: X(M)=1
600 FOR DL=1 TO 900: NEXT DL
610 LINE(L-12, 175)-(L+15, 147), PRESET, BF
620 IF NU=6 THEN 640
630 GOTO 170
640 U$="R8 NR8 D18 BR20 U9 NU9 R20 NU9 D9
BR25 L17 U18 R17 BD9 L17"
650 D1$="L23 U31 R23 BD14 L23"
660 D2$="U31 D3 F24 D4 U31 BR8 D30"
670 H$="U25 R4 D25 L4"
680 DRAW"BM91, 44"+U$
690 DRAW"BM107, 106"+D1$
700 DRAW"BM115, 106"+D2$
710 DRAW"BM171, 100"+H$
720 CIRCLE(147, 90), 16, 0, 1, 0, .25: CIRCLE(147,
90), 16, 0, 1, .75: CIRCLE(173, 105), 2, 0
730 GOTO 730
740 P2=1: M1=RND(7): ON M1 GOTO 750, 760, 770,
780, 790, 800, 810
750 L1=17: GOTO 820
760 L1=53: GOTO 820
770 L1=89: GOTO 820
780 L1=127: GOTO 820
790 L1=161: GOTO 820
800 L1=197: GOTO 820
810 L1=233
820 IF X(M1)=1 THEN 740
830 R1=RND(240)+12: X1=R1-L1
840 IF Q>5 THEN Q=5
850 Q=Q+.001: Y1=Y1+Q: LINE(R1,
0)-(R1-(Y1*X1/173), Y1), PSET
860 IF Y1<167 THEN 930 ELSE GOSUB 890
870 IF L1=127 THEN UF=1
880 GOTO 990
890 FOR P1=-6 TO 6
900 LINE(ABS(R1+P1),
0)-(ABS(R1+P1-(Y1*X1/173)), Y1-4), PRESET
910 NEXT P1
920 Y1=0: RETURN
930 GOTO 290
940 LINE(127, 170)-(H, K), PRESET: Q=Q+.1:
GOSUB 890
950 X(M1)=0
960 LINE(H-5, K-5)-(H+5, K+5), PRESET, BF
970 R1=RND(256)-1
980 GOTO 740
990 L1$=STR$(L1-3)
1000 IF X(M1)=0 AND L1<>127 THEN NU=NU+1
1010 DRAW"BM"+L1$+", 175"+A$+"BM"+L1$+",
175"+B$
1020 PLAY PL$: X(M1)=1
1030 FOR DL=1 TO 900: NEXT DL
1040 LINE(L1-12, 175)-(L1+15, 147), PRESET, BF
1050 IF NU=6 THEN 640
1060 GOTO 740
```


```
 147), PRESET, BF
1050 IF NU=6 THEN 640
1060 GOTO 740
```

IN-MEMORY DISK FILE SORT

by Michael Bartmon

After a long wait, I finally received my Radio Shack Disk drive. I immediately began to work on all the ideas that had been kicking around in my head. And, as with any new system, I quickly discovered the need for some utility programs that are not supplied.

I found myself in need of a means to sort various disk files. Rather than write a separate routine for each file, I decided to create a single routine that could be used to sort any fixed length disk file. Being impatient and with files in great need of sorting, I could not wait to develope a full blown machine-language sort. The following BASIC code is my interim (if slow) solution. I hope that others will also find it useful.

In order to have a sort routine that can be called from any other program and work on any file, there has to be a means to pass instructions to it. The information needed by this in-memory disk sort is: 1) name of file to be sorted, 2) record length of file to be sorted, 3) which part of the file record is to be used to determine the final sorted order, 4) whether to sort in ascending or descending order, and 5) the name of the file which is to contain the sorted records. This is accomplished by using a character string (sort parameter string) containing the necessary information in the following format:

position	description
1-12	input file name
13-24	output file name
25-27	record length of input file
28-30	starting position of sort key
31-33	length of sort key
34	sort order(A=ascending,D=descending)

Using P$ as our sort parameter string, then if:

```
P$="SORTIN /DATSORTOUT /DAT032010005A"
```

the instructions being passed to the sort routine would be:

Sort file, "SORTIN /DAT" (note the necessary spaces between the file name and extension. Name must be 8 characters long, left justified. Extension is required).

Place sorted records in file, "SORTOUT /DAT" (again, note the space).

SORTIN /DAT record length is 32 bytes.

The portion of the record to use in sorting begins at position 10 and is 5 bytes long.

The 'A' on the right of the string tells the sort routine to do the sorting in ascending order.

A more sophisticated sort utility could sort on more than one portion (or field) of a record as well as perform other, useful functions (i.e. record selection, alternate orders for specific ASCII codes), but I deliberately kept this one simple.

Now that we know what form the sort parameter string will take, we can look at the sort program itself (listing 1). The first line (#30) parses the parameter string, P$, into separate variables: F$=input file, G$=output file, RL=record length, KP=key position, KL=key length, O$=sort order. How the parameter string gets to the sort program will be covered later.

The input file is opened at line 60 and the number of records it contains is determined by: LF=LOF(1). The FIELD statement at 70 lets the program read an entire record into one string variable. Line 80 dimensions the array into which the entire file will be read. This is done on lines 90-110. The actual sort routine is contained on lines 130-200. It is a simple bubble sort and can be replaced by other, compatible, sort routines if desired. Note line 160: it is here that the value of O$ is used to determine the sort order. The output file is opened on line 210 and the sorted records written to it in 211-240.

Note that one very important statement does not appear in listing 1. It is necessary to CLEAR enough string space in memory to accomodate all the records of the input file. How many records that can be and, therefore, how much string space needs to be cleared is of course determined by the amount of memory available. PCLEAR 1:CLEAR 9000 works for a 16K machine. PCLEAR 1:CLEAR 25000 would work for 32K. This means that for an input file record length of 32 bytes, appoximately 280 records will fit in 16K and 780 records in 32K. If you have a larger file to sort then the sort program would have to be modified to sort one part of the file at a time to different output files and then all the output files merged into one final file. Or run the unmodified sort once for each sort key.

But how does that CLEAR statement get into the sort program? That depends on the method chosen to pass the parameter string. THere are several ways to pass it.

METHOD 1: Forget about passing the string. Change line 30 so that the separate variables are directly assigned (i.e. 30 F$="SORTIN/DAT": etc.) or use PRINT and INPUT staements (i.e. PRINT"INPUT FILE :";:INPUT F$). You would also add the following statement to the sort program: 10 PCLEAR: CLEAR 9000 (or 25000)

METHOD 2: Pass the parameter string from another program. You would have to add the following statements to the calling program at its exit point using appropriate line numbers:
PCLEAR: CLEAR 9000 (or 25000)
OPEN "D",1, "SORTSPEC",64
RUN"SORT",R

where "SORTSPEC" is the name of the file containing the parameter string and "SORT" is the name of the sort program.

The sort program would have the following lines added to it:
20 GET #1:INPUT P$
50 CLOSE #1

The "R" specification at the end of the RUN statement keeps file #1 open between programs.

METHOD 3: Pass the parameter string from an intermediate program.

the calling program would chain to an intermediate program with the statement:
RUN"SORTPASS"
where "SORTPASS" is the name of the intermediate program.

The intermediate program is of the form:
10 GOTO 20
11 OPEN"D",1,"SORTSPEC",64
20 MERGE"SORT",R

The sort program would be changed as for METHOD 2. It must be saved in ASCII format so that it can be merged (SAVE"SORT",A).

Method 1 is the simplest, but does not allow for "dynamic" sorting of files. The sort program must be loaded and run from the keyboard each time it is needed.

Method 2 allows us to sort files without having to re-specify the parameters each time. The cost is one specification file for each file to be sorted.

Method 3 costs more; an additional program is needed for each file to be sorted as well as a specification file. However, because of the merge and run performed by the intermediate program, it allows one added feature. Any valid BASIC statements with HIGHER line numbers than any statements in the sort program will be executed when the sort is completed. This could include merges or chains to other programs.

Merge and run works like this: The intermediate program is in memory and operating. Line 10 transfers control around line 11 to the MERGE statement on 20. The sort program also contains lines 10 and 20. Those lines in the intermediate program are replaced by lines 10 and 20 from the sort program. When the merge is completed, the new program (SORT plus line 11) is executed (because of the "R" specification on the MERGE statement). Since there is no line 11 in the sort program, that line remains in memory and becomes part of the SORT program. Likewise, any lines from the intermediate program with numbers not found in the sort program will also become part of the sort program.

Listing 2 is a program for the creation and maintenance of sort parameter files.

Some additional comments:

1. The input and output files can either be the same file or different files.

2. Numeric fields created by MKN$ can be sorted as if they were character fields.

3. If you specify a different file for output, it would be best if that file were a new one. At the least, be certain that the output file is the same size or SMALLER than the input file to avoid ending up with unwanted records.

```
30 F$=MID$(P$, 1, 12): G$=MID$(P$, 13, 12):
RL=VAL(MID$(P$, 25, 3)): KP=VAL(MID$(P$, 28,
3)): KL=VAL(MID$(P$, 31, 3)): O$=MID$(P$, 34, 1)
40 CLS: PRINT "SORTING <"F$"> FILE"
60 OPEN"D", 1, F$, RL: LF=LOF(1)
70 FIELD #1, RL AS Z$
80 DIM A$(LF)
90 FOR R=1 TO LF
100 GET #1, R: A$(R)=Z$
110 NEXT R
120 CLOSE #1
```

```
130 N=R-1
140 C=0: N=N-1
150 FOR I=1 TO N
160 IF O$="A" THEN 180
170 IF MID$(A$(I+1), KP, KL)<=MID$(A$(I), KP,
KL) THEN 200 ELSE GOTO 190
180 IF MID$(A$(I), KP, KL)<=MID$(A$(I+1), KP,
KL) THEN 200
190 X$=A$(I): A$(I)=A$(I+1): A$(I+1)=X$: C=1
200 NEXT I: IF C=1 THEN 140
210 OPEN "D", 1, G$, RL
211 FIELD #1, RL AS Z$
220 FOR R=1 TO LF
230 LSET Z$=A$(R): PUT #1, R
240 NEXT R
250 CLOSE #1
```

```
10 CLS
20 P$=STRING$(60, " ")
30 PRINT"  * * SORT SPECIFICATIONS * *":
PRINT
40 PRINT"SORT SPEC FILE NAME: ";:
LINEINPUT F$: IF F$="END" THEN END
50 OPEN"D", 1, F$, 64
60 IF LOF(1)=0 THEN 90
70 GET #1: INPUT #1, P$
80 S$=MID$(P$, 1, 12): G$=MID$(P$, 13, 12):
RL$=MID$(P$, 25, 3): KP$=MID$(P$, 28, 3):
KL$=MID$(P$, 31, 3): O$=MID$(P$, 34, 1)
90 CLOSE
100 PRINT @128, "  INPUT FILE: "S$
110 PRINT @160, "  OUTPUT FILE: "G$
120 PRINT @192, "RECORD LENGTH: "RL$
130 PRINT @224, " KEY POSITION: "KP$
140 PRINT @256, "  KEY LENGTH: "KL$
150 PRINT @288, "      ORDER: "O$
160 PRINT @448, "CORRECT (Y/N/C): ";:
LINEINPUT Y$
170 IF Y$="C" THEN 10
180 IF Y$="Y" THEN 400
190 IF Y$<>"N" THEN 160
200 PRINT @143, "";: LINEINPUT X$
210 IF X$<>"" THEN S$=X$
220 PRINT @143, S$
230 PRINT @175, "";: LINE INPUT X$
240 IF X$<>"" THEN G$=X$
250 PRINT @175, G$
260 PRINT @207, "";: LINEINPUT X$
270 IF X$<>"" THEN RL$=X$
280 PRINT @207, RL$
290 PRINT @239, "";: LINEINPUT X$
300 IF X$<>"" THEN KP$=X$
310 PRINT @239, KP$
320 PRINT @271, "";: LINEINPUT X$
330 IF X$<>"" THEN KL$=X$
340 PRINT @271, KL$
350 PRINT @303, "";: LINEINPUT X$
360 IF X$<>"" THEN O$=X$
370 IF O$<>"A" AND O$<>"D" THEN 350
380 PRINT @303, O$
390 GOTO 160
400 '
410 P$=STRING$(60, " ")
420 MID$(P$, 1, 12)=S$: MID$(P$, 13, 12)=G$:
MID$(P$, 25, 3)=RL$: MID$(P$, 28, 3)=KP$:
MID$(P$, 31, 3)=KL$: MID$(P$, 34, 1)=O$
430 OPEN"D", 1, F$, 64
440 WRITE #1, P$: PUT #1, 1: CLOSE
  ";:LINEINPUT X$
```

Sun Mik Software

3424 College N.E.
Grand Rapids, MI 49505

JUST RELEASED:

★COLOR COMPUTER DANCING DEVIL★

Watch him tap dance to music. Program your own routines. Youngsters and adults alike love him. Machine language. A Demon of a deal!

$14.95

MOON LANDER

2 Programs - A real treat!

Train on "Moon Lander" then move up to "Lander II". An outstanding flying experience. Visit the moon on your CC. Ext. Basic. Reviewers say most realistic on the market today! **$15.95**

ML RABBIT

No serious programmer can afford to pass this up!

Make copies of any machine or basic program effortlessly. Even copies programs that automatically execute. Completely automatic. (Caution-Intended to make back-up tapes only).

only $14.95

EDUCATIONAL PROGRAMS

(Now in use in several schools)

(16K Ext. Basic)

ABC 123

★**Math Drill** - Designed for teaching addition, subtraction, multiplication and division to the student of any age.

★**Spelling Test** - Hear your CC talk to you! An outstanding learning tool. Students love to learn with this program. For any age.

★**Word Drill** - Complete package designed by a teacher as a vocabulary teaching aid.

Recommended by teachers as an outstanding learning aid.

$19.95 each
All 3 for $49.97

CASINO - 3 game pak. 3 Vegas games on one tape. 5 Card Draw ★ Slots ★ Acey/Ducey.

Only $14.95

Also: CW Software for Color Computer

$19.95

CONNECT FOUR

Challenging game played either with two players or against the computer. 4K

$12.95

CCN-SAMPLER SERIES

$7.95 ea.

Add $1.00 Postage & Handling
Michigan residents add 4% Sales Tax.

★*LOOKING FOR NEW SOFTWARE*★
—TOP ROYALTIES PAID—

PHONE (616) 364-4791 (Evenings)

The following is a list of comments which could be added to a disassembly listing of the Color Computer ROM. If you don't have a disassembler yet, you will still be interested in the discussion at the end; but

Variables, areas, and routines -

Addr	Comments
0041	4 BYTES USED BY TOKENIZE
00A6	NEXT CHARACTER POINTER
0120	TOKEN TABLE DIRECTORY
02DC	CONSOLE I/O BUFFER
AA29	FUNTION ADDRESSES
AA66	COMMAND KEYWORD TABLE
AB1A	FUNCTION KEYWORD TABLE
AB67	COMMAND ADDRESSES
B7C2	UNTOKENIZE
B7E6	UNTOKENIZE ONE TOKEN
B821	TOKENIZE
B892	TOKENIZE ONE WORD

Line-by-line comments -

Addr	Comments
B7C2	RAM HOOK, NOT USED
B7C5	SKIP LINK AND LINE NUMBER
B7C7	GET ADDRESS OF I/O BUFFER
B7CB	GET A TOKEN
B7CD	IF ZERO, WE'RE DONE
B7CF	IF MINUS, MUST BE TOKEN
B7D1	IS IT A COLON?
B7D3	IF NOT, GO STORE
B7D5	GET NEXT TOKEN
B7D7	IS IT AN "ELSE"?
B7D9	FORGET THE COLON THEN
B7DB	IS IT A QUOTE?
B7DD	FORGET THE COLON THEN
B7DF	SKIP THE NEXT TWO BYTES
B7E0	GET "!" IF DON'T RECOGNIZE
B7E2	GO STORE CHARACTER
B7E4	GO GET NEXT TOKEN
B7E6	UNTOKENIZE TABLE MINUS 10
B7E9	IS IT A FUNCTION?
B7EB	SKIP NEXT TWO IF NOT
B7ED	GET THE REAL TOKEN
B7EF	GET TOKEN SECTION OF TABLE
B7F1	CLEAR HIGH BIT
B7F3	POINT U AT FIRST (OR NEXT)

if you're serious about the Color Computer, get a yourself a disassembler! It's a fun way to find out what's going on and to learn 6809 code at the same time.

The subject this time is the tokenize / untokenize routines.

B7F5	AT END OF TABLE?
B7F7	IF SO GO TYPE "!"
B7F9	MAX TOKEN THIS ENTRY
B7FB	IF MORE, GO TO NEXT ENTRY
B7FD	RESTORE TOKEN VALUE
B7FF	ADDRESS OF KEYWORD TABLE
B801	COUNT DOWN TOKEN
B802	IF THERE, GO PRINT
B804	FIND END OF THIS KEYWORD
B806	LOOP UNTIL FOUND
B808	GO COUNT DOWN TOKEN
B80A	GET CHARACTER OF KEYWORD
B80C	STORE IT TO PRINT BUFFER
B80E	AT END OF WORD?
B810	LOOP FOR WHOLE WORD
B812	GO GET NEXT TOKEN
B814	PAST END OF BUFFER?
B818	IF SO FORGET IT
B81A	CLEAR HIGH BIT
B81C	STORE INTO PRINT BUFFER
B81E	FLAG END-OF-STRING
B820	RETURN
B821	RAM HOOK
B824	GET ADDRESS OF TEXT
B826	GET ADDRESS TO PUT TOKENS
B829	INDICATE NOT VARIABLE YET
B82B	NOT DATA STATEMENT YET
B82D	GET A CHARACTER
B82F	IF END OF LINE, GO STORE
B831	ARE WE IN A VARIABLE?
B833	IF NOT, GO INVESTIGATE
B835	CHECK FOR ALPHA
B838	IF SO, GO STORE
B83A	NUMERIC?
B83C	IF NOT, GO CLEAR VAR FLAG
B83E	GREATER THAN 9?
B840	IF NOT, GO STORE
B842	CLEAR VARIABLE FLAG
B844	SPACE?
B846	IF SO, GO STORE
B848	SAVE IN CASE IT'S A QUOTE
B84A	IS IT A QUOTE?
B84C	IF SO, GO TO VERBATIM LOOP
B84E	ARE WE IN A DATA STATMENT?
B850	IF NOT, GO TRY TO TOKENIZE

```
B852  STORE CHARACTER TO BUFFER
B854  IF END, CLEAR AND LEAVE
B856  COLON? (END OF LINE)
B858  GO CLEAR VAR. DATA FLAGS
B85A  GO GET NEXT CHARACTER
B85C  ANOTHER CLEAR BYTE
B85E  AND ANOTHER
B860  ADDRESS OF END
B862  MINUS START: GIVES LENGTH
B865  POINT TO START
B868  STORE TO GETCHAR POINTER
B86A  RETURN
B86B  IS IT "?" (PRINT)
B86D  IF NOT SKIP
B86F  GET PRINT TOKEN
B871  GO STORE
B873  IS IT QUOTE (REMARK)
B875  IF NOT GO CHECK NUMBERS
B877  GET COLON AND REM TOKEN
B87A  STORE THEM BOTH
B87C  DON'T END ON END QUOTE
B87E  GET A CHARACTER
B880  IF END OF LINE, GO STORE
B882  IS IT TERMINATOR?
B884  IF SO GO STORE
B886  STORE IT
B888  LOOP (VERBATIM LOOP)
B88A  IS IT NUMERIC
B88C  IF LOWER, TRY TO TOKENIZE
B88E  HIGHER THAN NUMBERS,:,;
B890  IF NOT GO STORE
B892  BACK UP TO START OF WORD
B894  SAVE GET & STORE POINTERS
B896  CLEAR FUNCTION FLAG
B898  TABLE MINUS 10
B89B  INIT TOKEN COUNTER
B89D  TO NEXT (OR FIRST) ENTRY
B89F  GET NUMBER OF TOKENS
B8A1  NO MORE?  GO TRY FUNCTIONS
B8A3  GET ADDRESS OF KEYWORDS
B8A6  GET ADDRESS OF SOURCE
B8A8  GET LETTER FROM KEYWORD
B8AA  COMPARE WITH SOURCE
B8AC  LOOP AS LONG AS IT MATCHES
B8AE  JUST HIGH BIT IF MATCH
B8B0  IF NOT GO TRY NEXT KEYWORD
B8B2  REMOVE OLD SOURCE POINTER
B8B4  RESTORE STORE POINTER
B8B6  GET TOKEN COUNT; SET BIT 7
B8B8  GET FUNCTION FLAG
B8BA  IF FUNCTION, GO STORE BOTH
B8BC  IS IT "ELSE"?
B8BE  GO STORE TOKEN IF NOT
B8C0  GET AN EXTRA COLON
B8C2  STORE FF/: AND TOKEN
B8C4  DONE; GO GET MORE
B8C6  STORE THE TOKEN
B8C8  WAS IT "DATA"?
B8CA  IF NOT SKIP
B8CC  SET DATA FLAG
B8CE  IS IT "REM"?
B8D0  IF SO GO COPY VERBATIM
B8D2  DONE; GO GET MORE
B8D4  TABLE + 5 - 10 (FUNCTIONS)
B8D7  SET FUNCTION FLAG TO $FF
B8D9  GO TRY FUNCTIONS
B8DB  ALREADY DID, HUH?  RESTORE
B8DD  GET THE CHARACTER
B8DF  STORE IT
B8E1  WAS IT ALPHA?
B8E4  NO:  DONE; GO GET MORE
B8E6  SET VARIABLE FLAG
B8E8  DONE; GO GET MORE
B8EA  BUMP TOKEN COUNTER
B8EC  COUNT DOWN TOKENS IN TABLE
B8ED  IF DONE, GO TRY NEXT ENTRY
B8EF  BACK 1 (IN CASE PAST WORD)
B8F1  SKIP REST OF WORD
B8F3  UNTIL MINUS (END OF WORD)
B8F5  GO TRY NEXT KEYWORD
```

QUESTION: What is tokenizing?

Tokenizing is replacing Basic words (such as PRINT or REM) with a single byte code.

Why are Basic programs tokenized?

Not only does a tokenized program take up less memory, but it is easier and faster to run; therefore all Basic programs are tokenized as they are typed into the Color Computer.

How does it know what words to replace with tokens?

There is a list of keywords, starting at location $AA66. The words are listed in order of token value, and the last byte of each word has the high bit set to indicate the end of word. Each string of letters typed in is compared against each word in

COLOR COMPUTER SYSTEMS SOFTWARE

EDITOR/ASSEMBLER

The Micro Works Software Development System (SDS80C) is a complete 6809 editor, assembler and monitor package contained in one Color Computer program pack! Vastly superior to RAM-based assemblers/editors, the SDS80C is non-volatile, meaning that if your application program bombs, it can't destroy your editor/assembler. Plus it leaves almost all of 16K or 32K RAM free for **your** program. Since all three programs, editor, assembler and monitor are co-resident, we eliminate tedious program loading when going back and forth from editing to assembly and debugging!

The powerful screen-oriented Editor features finds, changes, moves, copys and much more. All keys have convenient auto repeat (typamatic), and since no line numbers are required, the full width of the screen may be used to generate well commented code.

The Assembler features **all** of the following: complete 6809 instruction set; complete 6800 set supported for cross-assembly; conditional assembly; local labels; assembly to cassette tape or to memory; listing to screen or printer; and mnemonic error codes instead of numbers.

The versatile ABUG monitor is a compact version of CBUG, tailored for debugging programs generated by the Assembler and Editor. It features examine/change of memory or registers, cassette load and save, breakpoints and more. **SDS80C Price: $89.95**

MODEM COMMUNICATIONS

Make your Color Computer an intelligent printing terminal with off-line storage! The Microtext module is just what you'll need for:

- —Talking to a timeshare system or information service
- —Printing out what is received as it is received
- —Saving received text to cassette tape
- —Re-displaying the received text even while on-line
- —Communications with other computers
- —Using your computer as a general-purpose 300-baud terminal
- —Downloading programs from other computers

The Microtext module is a program pack containing not only firmware but a second serial port so that both your printer and modem can be connected at the same time. Microtext can be configured for any serial printer that will work with the Color Computer, even if it requires line feeds! But even if you don't have a printer, you can keep a permanent copy of your data by storing to cassette tape. Also, any Radio Shack/Centronics-compatible parallel printer may be used by adding the Micro Works' PI80C parallel interface.

For those of you with special terminal applications, Microtext has selectable parity; it sends odd, even, mark or space. With mark parity (which is default) you can send to computers requiring either seven or eight bits. All 128 ASCII codes can be sent. Exchange programs with other Color Computer users! Basic programs may be downloaded from other computers or timesharing systems.

You'll find many uses for this versatile module! Available in ROMPACK, ready-to-use, for **$59.95.**

MACHINE LANGUAGE

MONITOR TAPE: A cassette tape which allows you to directly access memory, I/O and registers with a formatted hex display. Great for machine language programming, debugging and learning. It can also send/receive RS232 at up to 9600 baud, including host system download/upload. 19 commands in all. Relocatable and reentrant. **CBUG Tape Price: $29.95**

MONITOR ROM: The same program as above, supplied in 2716 EPROM. This allows you to use the entire RAM space. And you don't need to re-load the monitor each time you use it. The EPROM plugs into the Extended Basic ROM Socket or the Romless Pak I. **CBUG ROM Price: $39.95**

SOURCE GENERATOR: This package is a disassembler which runs on the color computer and generates your own source listing of the BASIC interpreter ROM. Also included is a documentation package which gives useful ROM entry points, complete memory map, I/O hardware details and more. A 16K system is required for the use of this cassette. **80C Disassembler Price: $49.95**

LEARN 6809!

6809 ASSEMBLY LANGUAGE PROGRAMMING, by Lance Leventhal, contains the most comprehensive reference material available for programming your Color Computer. **Price: $16.95**

HARDWARE

PARALLEL O!

USE A PARALLEL PRINTER with your Color Computer! Adaptor box plugs into the serial port and allows use of Centronics/Radio Shack-compatible printers with parallel interface. Assembled and tested. **PI80C Price: $69.96**

ROMLESS PAK I — is an empty program pack capable of holding two 2716 or 2732 EPROMS, allowing you up to 8K of program! The PC board inside comes with sockets installed, ready to go with the addition of your custom EPROMs. **Price: $24.95**

32K RAM!

MEMORY UPGRADE KITS: Consisting of 4116 200ns. integrated circuits, with instructions for installation. **4K-16K Kit Price: $39.95. 16K-32K Kit** (requires soldering experience) **Price: $39.95**

SPARE PARTS — SAMs, 6809Es, RAMs, PIAs. Call for prices.

GAMES

Pak Attack — Try your hand at this challenging game by Computerware, with fantastic graphics, sound and action! Cassette requires 16K. **Price: $24.95**

Star Blaster — Blast your way through an asteroid field in this action-packed Hi-Res graphics game! Available in ROMPACK; requires 16K. **Price: $39.95**

Berserk — Have fun zapping robots with this Hi-Res game by Mark Data Products. Cassette requires 16K. **Price: $24.95**

Adventure — *Black Sanctum* and *Calixto Island* by Mark Data Products. Each cassette requires 16K. **Price: $19.95** each.

GOOD STUFF!

THE MICRO WORKS

MasterCharge/Visa Accepted
California residents add 6% tax.

P.O. BOX 1110, DEL MAR, CA 92014 (714) 942-2400

this table. It is also compared against a similar table of function names (at $AB1A). It is also compared against similar tables for Extended Basic, and yet two more tables for Disk Basic.

Doesn't that take forever?

There can be a noticable pause, but it really doesn't matter as a program is being typed in or an Ascii tape is being read. The time is being invested now in order to speed up execution later.

How is tokenize called?

Tokenize is called with the address of the input line in locations $00A6 and $00A7. It returns with the tokenized line in the console I/O buffer starting at location $02DC. It is called from two places in the Basic program, once for direct statements and once for lines to be added to a program. In both cases the input line is starting at location $02DD, but there is no fear of the tokenized line running over the input line during conversion because the tokenized version can never be longer than the input at any point.

How is a program listed?

To list a program (or write an Ascii tape), each line must be untokenized. This involves replacing every token with it's equivalent keyword. It first is decided which list of keywords to use, then words are skipped (according to the value of the token) until the right one is found and listed.

With up to six lists of words, how is the right list found?

There is a table of keyword lists. It starts in location $0120, and has an entry for each table. Each entry takes five bytes. The Basic commands and the functions are the first two entries; Extended commands and functions are then added; Disk command and functions are added last.

What is the format of the table?

The first byte of each five-byte entry is the number of tokens. If this byte is zero, it means that there are no more entries. If non-zero, it gives the length of the keyword table. The next two bytes are the address of the keyword table. The last two are used by the execute routine to find where to go to run a token.

How is untokenize called?

Untokenize is called with the X register pointing at the tokenized line in memory. The resulting Ascii line is stored in the console I/O buffer starting at $02DD. Upon return the Y register is pointing at the zero byte which marks the end of the string. The routine is called twice, once from the LIST command, and once from Extended's EDIT command.

Tokenize and untokenize do a lot of processing regarding extra colons. Why is this?

Before every "ELSE" and every single quote (which means REM) the tokenizer puts in an extra colon. Because of this, a colon is not needed before these commands. The untokenizer looks for an ELSE or quote after each colon and, if it finds one, doesn't list the colon.

How does execute work?

There is a routine in Basic which executes tokens at the addresses given in the tables at $AB67 and $AA29. If the token is beyond the range of these tables, it jumps to a routine in Extended Basic. If that still doesn't recognize it, a routine is executed in Disk Basic. If it's still no good (or if Extended or Disk Basic isn't present), it jumps to Syntax Error.

Those bytes at location $0041. Haven't we seen them before?

Yes, those bytes are also used by the floating point routines. What's more, they are also used by the garbage collect routine. Beware of the reused variable!

MAKING EDUCATION MORE COLORFUL

by David Bodnar

It should be noted that this is not intended as a review of either of the cassette magazines Dave discusses below nor is this intended to be a comparision of the two cassette magazines, but a look at a few of the programs that both have published as they relate to education. A complete review of both of these cassette magazines is in the works for a future issue. Bill

Although my search for quality educational software for the COLOR COMPUTER has often been met with failure, I have turned up two publications that have the potential to supply us with good software at a reasonable price.

CHROMASETTE and TRC are cassette magazines that are exclusively for the COLOR COMPUTER. Although they cater to a very broad audience, they do occasionally come up with programs that can be used in educational context.

I have gone through the first two issues of TRC and the first six of CHROMASETTE and pulled from them the programs that I feel are educational in nature. What follows is a brief review of each program and suggestions that might make it more appropriate for use as an educational tool.

HOW FAR CHROMASETTE July 1981

This program computes the distance (in miles and kilometers) between any two points on the surface of the Earth. It also gives you the compass heading you would follow to reach the second point from the first. It contains an internal list of cities with their longitudes and latitudes. This list may be modified by changing the information in DATA lines.

Although the program would be of interest to any geography student or HAM radio operator, its main educational value may lie in motivating students to find the coordinates of various places around the world so that the computer can compute the distance between them.

The program is well written and seems to be fairly accurate. Its only weakness is one that is shared by many computer programs. The directions are minimal and younger children may need a lot of help to get the program running properly.

SPELLIT CHROMASETTE September 1981

SPELLIT is a program that gives you a spelling quiz using a format very similar to that utilized by many standardized achievement tests. Four words and the statement "NONE OF THESE" are displayed on the screen. You must decide which, if any, of the words is misspelled and respond with its number. If you correctly choose the misspelled word you are then asked to spell it correctly. If your choice is incorrect you are given a second chance after which you are told which word is misspelled and its correct spelling.

I have used this program in my classroom and have found it to be very valuable. All of the words that it uses are contained in DATA lines at the end of the program and I can put the current week's spelling list into the program in a short time.

I was initially disappointed with the program because you must include an incorrect spelling for each word. I wanted the computer to do that job for me!! It turns out to be better this way, however, because a random, computer chosen misspelling is likely to be obvious. By giving a person control over this a more subtile misspelling may be chosen.

UFO MATH CHROMASETTE November 1981

When I first loaded UFO MATH I looked forward to a program that would combine math practice and some flying saucer fun. That is what the title implies, but not what the program delivers.

When the program is run you are asked to choose practice in addition, subtraction, multiplication, or division. You must also choose a difficulty level ranging from 2 to 9. After these items are entered the high resolution graphics screen appears. A problem is written at the top of the screen and a UFO bounces around below. If you answer the problem correctly you are given the opportunity to shoot at the UFO from your joystick controlled spacecraft.

The program's main definciency is that the problems it generates are too simple. The difficulty level determines the maximum value of the members of the problem. For example, if you are doing subtraction at difficuty level 7 the most difficult problem that the program can generate is 7 minus a number less than or equal to 7. Multiplication does permit problems with two digit answers, but once you have entered the first digit, if you find you have made an error, it cannot be changed. (A poor educational practice.)

The division routine does not generate problems any more difficult than 9 divided by a number less that or equal to 9. If you are going to drill on mulitiplication facts up to 9 times 9 it seems logical that you would also include division facts up to 81 divided by 9.

Technically there are also some problems. The UFO is fun to shoot at, but if it is at the top of the screen and you make a direct hit on it nothing happens. The program also fails to clear the INKEY function before going to the high resolution screen so that if you accidentally press a number key while viewing the text screen, it will be taken as the first digit of your answer and this cannot be changed.

UFO MATH may be of limited use to those of you with young children who are learning the simplest of their facts, but even here its technical shortcomings may prove frustrating to the user.

MORSQUIZ CHROMASETTE November 1981

MORSQUIZ is designed to make learning Morse code as painless a task as possible. The instructions for the program are loaded first and they explain that you will be taught groups of related letters and numbers. After mastering each of these lessons, you will be tested on the entire alphabet then on words and finally on sentences.

When the program is run, a menu allows you to choose the level at which you enter the program. For beginners start with choice one, TEACH MORSE BY GROUPS. The first lesson in this section deals with "E,I,S,H, and 5". Each of these characters is made up of only dots. The next lesson is on "T,M,O, and 0", each of which is made up of only dashes. The rest of this lesson continues in a like manner.

The second menu item gives you a quiz on all of the Morse characters, the third gives practice on medium sized words and the last on short sentences.

Being unfamilar with Morse Code I was unable to give the program much of a workout beyond the first lesson. I did have a few concerns, however. When you are being given practice on short words the screen does not show what you have typed in. I thought this might be standard procedure in teaching Morse Code until I used the last lesson. It sounds out entire sentences then asks you to type in the correct words and it DOES show what you are typing. This seems to be an unnecessary inconsistency.

BASECONV CHROMASETTE December 1981

BASECONV works well in converting numbers in a base from 2 to 16 to another number in a base from 2 to 16. It is limited to working with numbers with 100 or less digits but this could be increased by modifying the dimension statement in line 10.

Its nicest feature is allowing you to chose the output format so that the numbers it calculates may be displayed in groups for easy reading. For example, if you choose an output format of 4 the base 2 equivalent of 500 (base 10) would be displayed as 1 1111 0100 rather than 111110100. This also makes it easier to see the relationship between base 2 and base 16 because each group of 4 base 2 digits converts to one base 16 digit.

WORLDMAP CHROMASETTE December 1981

Even though a world map or globe contains more detail and is easier to use, there is something about seeing a computer draw a map of the world in front of your eyes that is highly motivating.

This program will draw a Mercator projection (the most common flat map of the world), an orthographic projection (like looking at the globe from a spacecraft that is over the equator) or a polar orthographic projection (from the north or south pole).

It would fit in nicely if you were trying to teach about different map projections or other map skills. The orthographic projection is especially interesting because it allow you to choose the point on the equator that will beome the center of the map.

WORLDGEOG TRC MAGAZINE November 1981

WORLD GEOGRAPHY quiz is a weak attempt at a computer assisted instruction program. It uses a pleasant, letter by letter, scrolling technique to print questions about world geography at the top of the screen. Four numbered answers appear below. You are to choose the answer that you think is best. If you are incorrect a brief statement is printed that explains why you are wrong. You are then asked to answer the same question again. Here the first problem is encountered. The program allows you to give the same incorrect answer over and over, it should not. It also is riddled with spelling

errors (Robert Pearry discovered the North Pole??), puctuation, factual and logical errors.

It also seems to have been published before it was completed. On several questions there is no explanation provided for incorrect answers and a "@" is printed on the screen.

The program has some interesting routines in it, but its overall performance is inadequate.

ARRAYS TRC MAGAZINE November 1981

Each issue of TRC contains one or more tutorial programs. As the name implies this lesson is on arrays. It starts with a brief, and somewhat confusing discussion of single dimension arrays using a shopping list as a metaphor. It proceeds to an excellent description of how a two dimension array can be used to describe the positions on a chess board. The program causes the squares it is referring to to flash for easy identification.

The last portion of the program goes into great detail to describe how arrays are used in another of the November issue's programs, a peg jumping game called I.Q. TEST.

This tutorial would be a god review for a person with some knowledge of arrays, but a novice programmer would quickly be lost by the description of the I.Q. TEST program.

ALGEBRA TRC MAGAZINE December 1981

ALGEBRA is a program that could be a valuable tool for the student reviewing for a basic algebra test. It presents problems using four different, and increasingly difficult, formats. The easiest problems are similar to 7X=21. You must solve for X. The most difficult look like this:

2(X-3)+4(X-7)=-10

Technically the program works very well, but it suffers from some minor problems that make it difficult to use. If a problem is answered incorrectly you are asked to try again, but if your answer is still wrong you are given another problem without ever being told the correct answer.

Once you start the program you are locked in until you answer twenty consecutive problems correctly. It would be better if this were a variable that the user could select at the start of the program.

One final concern is that on some of the problems the INKEY function reads your answers without keying ENTER and on some ENTER must be keyed. I am sure this could confuse some users.

MODULAR ARITHMETIC and MAGIC SQUARES
TRC MAGAZINE
December 1981

MODULAR ARITHMETIC and MAGIC SQUARES is another of TRC's tutorial lessons. This one discusses how modular arithmetic is used in generating a puzzle called a magic square. A magic square is a pattern of numbers arranged on a square grid so that the sum of any horizontal, vertical or diagonal set of numbers is the same.

Modular arithemtic may be more familiar to you as CLOCK arithmetic. A clock is really a MOD 12 device. If it is 7 o'clock and you add 9 hours the answer is 4. That is the same as saying that the MOD 12 answer to 7+9 is 4. I will leave it to TRC's program to explain it in more detail.

After the concept of MODs is presented a very well documented magic square program is used to demonstrate their usefulness.

In conclusion I think that you will find both magazines a source of programs that can be used to expand the utility of the COLOR COMPUTER. CHROMASETTE has several years of experience with CLOAD magazine behind it and you can see this in the smooth operation of most of its programs. TRC still has some rough edges, but if it continues to improve as it did from the first issue to the second it will become a strong contender for our software dollar.

Both magazines offer back issues. CHROMASETTE at $5.00 each and TRC at $10.00 each. The addresses follow.

CHROMASETTE MAGAZINE
P.O. BOX 1087
SANTA BARBARA, CA
93102

TRC MAGAZINE
THE PROGRAMMER'S INSTITUTE
P.O. BOX 3191
CHAPEL HILL, NORTH CAROLINA
27514

ACTION GAMES

The fastest growing producer of computer games for your 6809 has the products you have waited for!!

NEW! ARCADE GAMES FOR THE COLOR COMPUTER

CAVE HUNTER

COLOR BERSERK

Fast paced action • Super Hi-Res Graphics
Dynamite sound effects • Runs in 16K of memory
These games will astonish you with their Detail and Quality.
They set a standard for others to follow.

— ADVENTURES —

Calixto Island • The Black Sanctum

Highly acclaimed by reviewers • Challenging situations
Fast, efficient machine language • Runs in 16K of memory
Save game in progress

Adventures on 5¼ TSC FLEX disc (specify 6800 or 6809)	ea. $24.95
Both adventures on single disc	$39.95
Adventures for color computer	ea. $19.95
Color Berserk for color computer	ea. $24.95
Cave Hunter for color computer	ea. $24.95

Shipped prepaid in continental U.S. California residents, please add 6% tax.

VISA

— MORE COMING SOON —

WE HONOR master charge THE INTERBANK CARD

MARK DATA PRODUCTS

23802 Barquilla, Mission Viejo, CA 92691 • (714) 768-1551

TRS 80 IS A TRADEMARK OF TANDY CORP.

Game: Moon Lander - cassette tape
Company: Tom Mix Software
Author: Tom Mix & Chris Latham
Type: Arcade
Requirements: 16K and joysticks
Price: $15.95
Info: Extended Basic, high Res graphics, color, sound

Lunar landers and computers just seem to go together. I remember the first one I played was on a D.E.C. mainframe with a printer, no graphics, no sound, just text; I was fascinated. So one of the first programs I wanted for my new C.C. was a lander. I fired off orders to several outfits and watched my mail box for several weeks with mounting excitement. Unfortunately the programs I received weren't worth the wait. Sure I got sound and lots of pretty colored dots but after a half an hour I had landed on the one position available; got a little tune for my effort and that was it. Not exactly what I thought the machine was capable of and certainly not enough for my hard earned bucks.

Well, apparently there are other lunartics around because Tom Mix Software has now supplied us with a really good lander program for the C.C., Moon Lander. Actually I should say programs because as a nice "moneys' worth" touch one side of the tape carries a simpler version of the game with different terrain and less difficult control features. This is an idea that might well be copied by other companys.

Being an experienced LEM pilot, I loaded the full featured game, chose the one pilot version, selected a five gravity level and immediately went into an uncontrolable tumble which ended in a ball of fire against the nearest mountain peak.

Next time I took it a little slower, finding to my delight that my lander rotated under commands from my joystick and fired its retros with the shoot button. A small flame flicked out, a satisfying display lacking in my old ship.

Getting things under control, I checked the graphically displayed instruments, a fuel gage and a horizontal velocity meter. Everything checked out so I looked for a landing spot. There were in sight five, ten and twenty point spots. Displaying the flashing intelligence LEM pilots are famous for I decided the five point pad looked the easiest and made for it. As I passed over the next peak, I got a second pleasant surprise, the display changed to a close-up of the landing pad (another feature sadly lacking in my other games.) I bungled this landing too but blamed it on the surprise factor.

I won't bore you with a complete log of my flights but after enduring various derisive comments on my piloting ability I thumped down finally and received a nice read out on velocities, fuel state and score. The score is based on landing pad chosen, fuel remaining and touch down velocities. This type of scoring adds playability to the game.

After playing (testing I told my wife) several hours I was still interested and having fun trying out various techniques for getting better scores. It's not easy, those joysticks are supersensitive and the lander configuration was sometimes a little hard to spot.

The one fault I found with the program centered around sound. Beeps are just not realistic enough for crashes and rocket blasts. Yes, I know there's no sound in space but there is in my living room. Future programmers take note, isn't there some way to get some blasts, white noise or something out of this machine?

This program is written in extended basic but this doesn't seem to be a handicap for this type of game. Other than you could pack more features in if it was written in machine language. In fact, I found it was rather fun to be able to change the text commands and mess around with the terrain.

All in all this lander is the best I've seen for the C.C. It compares well with its big brother arcade version and is as good or better than high priced landers on other machines. (A good example here is Vic Super Lander at $29.95) If you're like me, you'll be looking forward to a 32K lander with even more features but until then you won't be disappointed with MOON LANDER.

Jim Plank

80-U.S.

THE TRS·80 USERS JOURNAL

80-U.S. Journal is a monthly publication for the TRS-80 computer owner. The Journal covers Business, Scientific, Educational, and Recreational areas.

80-U.S. will keep you up to date on new products, software and hardware. Each issue will have listings of programs, reviews, tutorials. 80-U.S. is the complete "How to" Journal for the TRS-80!

If you haven't taken a look at 80-U.S., here is a no-risk opportunity to do it now. Become a trial subscriber now under the protection of a *full* money-back guarantee!

PLEASE enroll me as a trial subscriber to 80-U.S. and bill me just $16. I understand I reserve the right to cancel my subscription any time, for any reason, and receive a refund for the balance of my subscription. If I should decide to cancel after receiving my first issue I will simply mark my bill *cancel* and keep the first issue *FREE!*

Name _____

Address _____

City _____ State _____ Zip _____

☐ I prefer to enclose payment now.

Visa/MC # _____

Exp. Date _____

Canada & Mexico: $25 per year, other foreign subscriptions $30 surface mail, $72 per year via airmail.

80-U.S. Journal
3838 South Warner Street
Tacoma, Washington 98409
(206) 475-2219

Order # CCN

TRS-80 is a Registered Trademark of the Tandy Corp.

COLOR CODE PRACTICE

by Charles T. Wrye

Introduction

So you say you're trying to learn the code for your Ham licence? And you say you're tired of listening to the same tapes over and over? And you're not sure whether you are learning the code or just memorizing the tapes? Well take heart, Bucky, this program is for you!

This program is for the 4K Color Computer with Standard Color Basic. It will run on any other Color Computer though, including the 32K with Disk Extended Color Basic.

The computer will send you code for as long as you want it to. It will never repeat itself so you cannot anticipate the next character. The code sent can be as simple as you want or can contain all characters.

The operation of the program is very straightforward. You will first be asked for the speed you want. You must enter a number from one to five. One will give you a speed of about five words per minute, and five will give you about 13 WPM, with the rest in between.

Then you will be asked for the level of difficulty. You must enter a number between one and twenty. If you enter a one you will get only the letters "E" and "T". Thirteen will give you the whole alphabet and twenty will give you the whole alphabet, all the numbers, and punctuation marks as well.

When you press "ENTER" the computer will begin to send you code. It will be sent as ten, five-letter words. Each word will consist of five random characters. The screen will remain blank during this time.

When the ten words have been sent, the question, "READY FOR LIST?" will appear on the screen. When you are ready, enter "Y" and the ten words will be printed down the screen.

Then the question, "RUN IT AGAIN?" will be printed across the bottom of the screen. If you want another ten words of the same difficulty and speed, enter "Y" and the computer will begin sending again. If you want to change speed or difficulty level press "ENTER" and then RUN it again.

How The Program Works

The first line sets the C$ array to fifty to hold the value of each letter sent. Lines 20 through 80 input your choice of speed and difficulty level. In line 50 the value of I is set. I is used to determine the length of space between the dots and dashes within each letter.

Line 80 blanks the screen while the code is being sent. Line 90 repeats the program to generate the ten words. Line 100 puts the pause in at the end of each word and line 120 the pause at the end of each letter.

Line 110 repeats for the five letters within each word. Line 120 generates a random letter. Lines 150 and 160 select the data to be used to send the letter and store it's value in the string. Lines 170 through 190 read through A$ and sound the dots and dashes! one is a dot and three is a dash. Line 200 puts the pauses between dots and dashes within each letter.

Line 210 takes the last character of the data word and stores it in D$ for display later. Each data word contains the information to send one character. A list of all the characters and the difficulty level is shown in Figure 1. For each difficulty level the two characters shown plus all above it will be sent. The required dots (1's) and dashes (3's) form all but the last digit which is the actual character sent.

When all ten words have been sent, line 330 asks if you want the list displayed. If not, the program branches to line 430. Lines 340 through 420 display the list on the screen. Line 370 centers the list. Finally, lines 430 on end the program.

I know it's not very colorful and there aren't any pretty pictures, but it does send code. It can get you up to 13 words per minute without ever buying a code practice tape again. When you need to go faster you will have to get Extended Color Basic and use the PLAY instruction. I'm working on that now and maybe you'll see that program listed here sometime in the future.

Figure 1. Difficulty level

DIFFICULTY LEVEL	CHARACTER SENT
1	E T
2	A I
3	N M
4	S C
5	H R
6	D L
7	O P
8	F W
9	V K
10	B G
11	J Q
12	U X

```
13          Y Z
14          1 2
15          3 4
16          5 6
17          7 8
18          9 0
19          " ,
20          ? /
```

Figure 2. Listing of Variables

```
H = SPEED AT WHICH CODE IS SENT
I = SPACING BETWEEN DOTS AND DASHES WITHIN
CHARACTER
K = DEGREE OF DIFFICULTY
Z = NUMBER OF WORDS
T = TIME DELAY
J = NUMBER OF LETTERS IN WORD
B = RANDOM NUMBER OF CHARACTER SENT
E = TOTAL NUMBER OF CHARACTERS SENT
C = POSITION OF WORD IN DATA
A$ = WORD CONTAINING INFORMATION OF
CHARACTER TO BE SENT
S = LENGTH OF CHARACTER TO BE SENT
A = INDIVIDUAL DOTS AND DASHES IN CHARACTER
D$ = REPLY TO QUESTION
```

Figure 3. Program Listing

```
1 ' COLOR CODE PRACTICE
2 ' BY C T WEYE
10 CLS: DIM C$(50)
20 PRINT@39, "COLOR CODE PRACTICE"
30 PRINT: PRINT "WHAT SPEED DO YOU WANT?
INPUT"
40 INPUT "(1 TO 5) THEN PRESS <ENTER>", H
50 IF H<4 THEN I=1 ELSE I=2
60 PRINT "WHAT DEGREE OF DIFFICULTY, INPUT";
70 INPUT "(1 TO 20) THEN <ENTER>";K
80 CLS
90 FOR Z = 1 TO 10
100 FOR T = 1 TO 460/(H*H): NEXT T
110 FOR J = 1 TO 5
120 FOR T = 1 TO 460/(H*H): NEXT T
130 B = RND (K*2)
140 E = E + 1
150 FORC = 1 TO B
160 READ A$: NEXT C
170 FORS = 1 TO LEN (A$)-1
180 A = VAL (MID$(A$,S,1))
190 SOUND 190,A
200 FOR T = 1 TO 40/I: NEXT T,S
210 C$(E) = RIGHT$ (A$,1)
220 RESTORE
230 NEXT J,Z
240 DATA 1E, 3T, 13A, 11I, 31N, 33M, 111S, 313C,
250 DATA 1111H, 131R, 311D, 1311L, 3330
260 DATA 13331P, 1131F, 133U, 1113V, 313K
270 DATA 3111B, 331G, 1333J, 33130, 113U
280 DATA 3113X, 3133Y, 33112, 133331, 113332
290 DATA 111333, 111134, 111115, 311116, 331117
300 DATA 333118, 333319, 333330
310 DATA 131313, "331133," 1133112, 31131/
320 PRINT @233, "READY FOR LIST";: INPUT D$
330 IF LEFT$ (D$,1) <> "Y" THEN 430
340 CLS
350 PRINT: PRINT: PRINT: E = 0
360 FOR Z = 1 TO 10
370 PRINT "          ";
380 FOR J = 1 TO 5
390 E = E + 1
400 PRINT C$(E);
410 NEXT J: PRINT
420 NEXT Z
430 PRINT @489, "RUN IT AGAIN";: INPUT D$
440 IF LEFT$ (D$,1) <> "Y" THEN END
450 E = 0: GOTO 80
```

QUALITY SOFTWARE FOR TRS-80 COLOR!

ADVENTURES!!!!

For TRS-80, and COLOR-80. These Adventures are written in BASIC, are full featured, fast action, full plotted adventures that take 30-50 hours to play. (Adventures are inter-active fantasies. It's like reading a book except that you are the main character as you give the computer commands like "Look in the Coffin" and "Light the torch.")

Adventures require 16K on COLOR-80 and TRS-80. They sell for $14.95 each.

ESCAPE FROM MARS
(by Rodger Olsen)
This ADVENTURE takes place on the RED PLANET. You'll have to explore a Martian city and deal with possibly hostile aliens to survive this one. A good first adventure.

PYRAMID (by Rodger Olsen)
This is our most challenging ADVENTURE. It is a treasure hunt in a pyramid full of problems. Exciting and tough!

TREK ADVENTURE (by Bob Retelle)
This one takes place aboard a familiar starship. The crew has left for good reasons - but they forgot to take you, and now you are in deep trouble.

DEATH SHIP (by Rodger Olsen)
Our first and original ADVENTURE, this one takes place aboard a cruise ship - but it ain't the Love Boat.

VAMPIRE CASTLE (by Mike Bassman)
This is a contest between you and old Drac - and it's getting a little dark outside. $14.95 each.

SPACE SHUTTLE
ONE OR TWO PLAYER HIGH RES GAME – Your mission is to dock with an orbiting space platform – but you may have to land on the planetary surface for refueling first. A real value in a high res real time game. $6.95.

KILLERBOT – (Available in 4K) – Real time action at 20 levels of difficulty as you run, sneak, and dodge your way through a bloody field of Killer Robots. Get across or die! Joysticks or Keyboard controls. TRS-80 COLOR (ANY BASIC 4K or more.). $9.95.

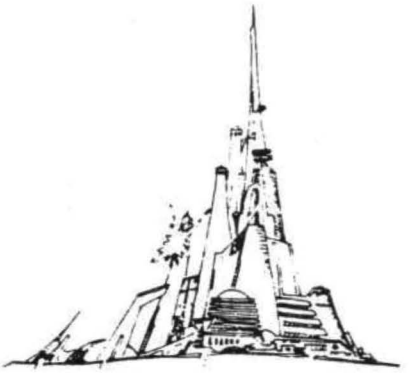
SLASHBALL (Available in 4K) – This one is best described as a thinkers arcade game. It rewards fast reflexes and clear thinking – like nothing you have ever seen before. It is one of our best family games for one or two players. $9.95.

TIMETREK - A REAL TIME, REAL GRAPHICS STARTRECK. See your torpedoes hit and watch your instruments work in real time. No more unrealistic scrolling displays! $14.95.

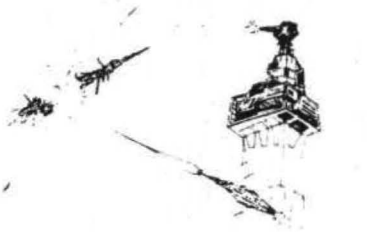
STARFIGHTER - This one man space war game pits you against spacecruisers, battlewagons, and one man fighters, you have the view from your cockpit window, a real time working instrument panel, and your wits. Another real time goody. $9.95

BATTLEFLEET - This grown up version of Battleship is the toughest thinking game available on OSI or 80 computers. There is no luck involved as you seek out the computers hidden fleet. A topographical toughie. $9.95

LABYRINTH – 16K EXTENDED COLOR BASIC – With amazing 3D graphics, you fight your way through a maze facing real time monsters. The graphics are real enogh to cause claustrophobia. The most realistic game that I have ever seen on either system. $14.95.



QUEST - A NEW IDEA IN ADVENTURE GAMES! Different from all the others, Quest is played on a computer generated map of Alesia. Your job is to gather men and supplies by combat, bargaining, exploration of ruins and temples and outright banditry. When your force is strong enough, you attack the Citadel of Moorlock in a life or death battle to the finish. Playable in 2 to 5 hours, this one is different every time. 16K COLOR-80 OR TRS-80 ONLY $14.95.



SPACE ZAPPER – Protect your central Star Base from ships that attack from all four sides. Fast reflexes are required as the action speeds up. Great for kids or Dads. This game has high speed high resolution graphics and looks as if it just stepped out of the arcades. – 16K extended or 32K disk. BASIC TRS-80 Color only. $14.95.

Please specify system on all orders

This is only a partial listing of what we have to offer. We have arcade and thinking games, utilities and business programs for the OS1 and TRS-80 Color. We add new programs every week. Send $1.00 for our complete catalog.

AARDVARK - 80

TRS 80 **2352 S. Commerce, Walled Lake, MI 48088** **TRS 80 COLOR**

(313) 669-3110

COLOR GRAPHING

by Lane P. Lester

Most computer hackers enjoy showing off the results of their programming efforts. As a college professor I have captive audiences of hundreds of students each year, so naturally I've sought for programs I could use in teaching. Most computer programs written for education are for use by the student at the computer keyboard. What I've been enjoying writing are programs to illustrate concepts during lectures. The Color Computer is an ideal tool for this purpose with its powerful color capabilities. This article deals with a technique for producing line graphs that are impressive and that can be viewed from a distance.

Because the Color Computer uses a modulator and an ordinary TV set rather than direct video input to a monitor, there are limits on the fine detail that can be viewed effectively. Particularly is this the case if one's TV set is not the newest or the best adjusted. When I first wrote a graphic program, the lines were not very sharp, and the different colors not easily distinguished from a distance. To overcome this difficulty the following program, POPGEN, was developed with large areas of solid color to define the areas under the different lines on the graph.

The name of the program refers to POPulation GEN etics, the study of inheritance in populations of creatures, rather than in individuals. Of particular interest are the factors involved in the genetic change of a population through time. Although POPGEN was written specifically to illustrate this process, the graphing procedures can be adapted to other types of information. The program listing is subdivided by REM statements into its different functions, so that you can study those features of interest.

I've also included some programming techniques which are not seen very often. Perhaps you will find some of these useful in your own programming. In lines 360-440 the DRAW strings have no "1"'s in them, since these are superfluous. Lines 700-710 illustrate the use of embedded IF-THENs through two statement lines. Lines 890-910 and 950-960 show how to use the same format in a number of PRINT USINGs.

Sample Problems

For a first run to show the features of the program, one can enter the parameters 240, .95, 0, 0, 0, 0, .9, .9, 1. For a more realistic simulation, the case of sickle cell anemia is interesting. This human mutation is usually lethal for those who carry two alleles for it (A2A2). A1A2 individuals do not suffer from anemia and are resistant to malaria. To simulate this situation in a malaria-infested area, enter the parameters 240, .99, 0, 0, 0, 0, 0.8 (might die from malaria), 1 (resistant to malaria and not anemic), and 0.01 (probably dies from anemia). After determining equilibrium frequencies for A1 and A2 in the preceeding investigation, simulate the situation for slaves brought from Africa to America. Give A1 the equilibrium frequency for the intial frequency and, because malaria is absent in the U.S., make W1 = 1, W2 = 1, and W3 = 0.01. See what happens when the advantage of this mutation is no longer present.

Program Description

60-280 Set Beginning Parameters

In tracing genetic change in population, certain variables influence the outcome. In this demonstration a single gene is involved which has two alleles (forms), A1 and A2. A frequency for A1 between 0 and 1 is input for time unit = 0. Because there are only the two alleles, the frequency of A2 = 1 - (frequency of A1). In a situation where creatures from another population of the same species are invading the population under study, it is necessary to take into account the gene frequencies of the immigrants as well as the proportion of immigrants in the population each generation.

Mutations (genetic mistakes) affect the genetic makeup of a population through time. Allele A1 can change to A2, and vice versa. Typical mutation rates range from 0.00001 to 0.000001. In the simplest situations each creature possesses two genes for each trait, hence one may be A1A1, A1A2, or A2A2. It is often true that one genetic makeup will give its owner some advantage over another. This is called fitness, and refers to the relative ability of each genetic type to reproduce. The genetic makeup with the highest fitness is given a value of 1.00, and the other two makeups are given relatively lower values down to a fitness of 0 for a genetic makeup that is lethal, that never reproduces.

In line 160 all of the above values are entered. 12 sets the beginning of the X-axis; J and J1 are used to scale the X-axis to a maximum of 242 points on the graph. Then comes the high speed POKE, following which are the calculations of the initial frequencies: A2 (Q), immigrant A2 (QM), A1A1 (P2), A1A2 (PQ), and A2A2 (Q2).

300-520 Draw and Label Graph Axes

Note the use of DRAW to add characters to graphics. A full discussion of this technique can be found in "DRAW III" by Don Inman (CCN, January, 1982, p.14)

540-640 Calculate Gene Frequencies

The formulas in this line use the parameters provided at the beginning of the program to compute the change in frequencies of A1A1, A1A2, and A2A2 through time. An explanation of these procedures can be found in most college genetics texts.

690-710 Determine Graph Colors

The colors red, yellow, and blue are used to label the graph frequencies of A1A1 (G1), A1A2 (G2), A2A2 (G3). These two statements determine which frequency is greatest (X), next lower (Y), and lowest (Z) in each generation, so that the graph can be colored appropriately.

730-830 Draw Graph Lines

Using the color of the lowest frequency (Z), a line is drawn up to its value. From that point the line is extended in Y's color up to its value. Then the line is completed in X's color up to its value. Becuase these lines are adjacent, they produce a solid mass of color indicating the change of each genetic makeup.

850-1030 Print Start and Stop Values

After the delay of line 800, the screen switches to a text presentation of the initial and final parameters. It is then possible to switch back and forth between the graph and the text.

```
10 '        POGEN
20 '     LANE P. LESTER
30 '  LIBERTY BAPTIST COLLEGE
40 '    LYNCHBURG, VA 24506
48 LINE (11, X)-(15, X), PSET
50 'set beginning parameters
60 CLS
70 PRINT @13, "POPGEN"
80 PRINT "ENTER THE FOLOWING PARAMETERS! "
90 PRINT "NO. OF GENERATIONS, "
100 PRINT "FREQ. OF A1 IN POPULATION, "
110 PRINT "FREQ. OF A1 IN IMMIGRANTS, "
120 PRINT "PROP. OF IMMIGRANTS EACH GEN., "
130 PRINT "MUTATION RATE OF A1 TO A2, "
140 PRINT "MUTATION RATE OF A2 TO A1, "
150 PRINT "FITNESS OF A1A1, A1A2, AND A2A2,"
160 INPUT I1, P, PM, M, U, V, W1, W2, W3
170 CLS 0
180 I2 = 16
190 J = 1
200 J1 = INT(GEN/242)+1
210 POKE 65495, 0
220 P1 = P
```

```
230 Q = 1-P
240 Q1 = Q
250 QM = 1-PM
260 P2 = P^2
270 PQ = 2*P*Q
280 Q2 = Q^2
290 'draw and label graph axes
300 PMODE 3
310 PCLS
320 COLOR 2, 1
330 SCREEN 1, 0
340 LINE (15, 176)-(256, 176), PSET
350 LINE (15, 176)-(15, 0), PSET
360 NO$ = "BR4BD5GL2HU4ER2FD4BU5BR4"
370 N1$ = "BR2DLRD5LR2BU6BR5"
380 T$ = "DUR4DUL2D6LR2BU6BR5"
390 I$ = "BRR2LD6LR2BU6BR5"
400 M$ = "D6U6F2E2D6BU6BR4"
410 E$ = "R4L4D3R3L3D3R4L3BD4"
420 F$ = "R4L4D3R2L2D3BL4BD4"
430 R$ = "D6U6R3FDGL2F3BL4BD4"
440 Q$ = "R2FD3G2LHU4D3BR2F2"
450 DATA 172, 130, 88, 46, 4
460 FOR I = 1 TO 5
470 READ X
490 NEXT
500 DRAW "S8BM0, 165XNO$;BM0, 0XN1$;
510 DRAW "BM110, 179XT$;XI$;XM$;XE$;
520 DRAW "BM0, 52XF$, XR$;XE$;XQ$;"
530 'calculate gene frequencies
540 FOR I = 1 TO I1
550 Q = Q+M*(QM=Q)
560 Q = Q+U*P-V*Q
570 P = 1-Q
580 C1 = P^2*W1
590 C2 = 2*P*Q*W2
600 C3 = Q^2*W3
610 W4 = C1+C2+C3
620 G1 = C1/W4
630 G2 = C2/W4
640 G3 = C3/W4
650 P = G1+G2/2
660 Q = G2/2+G3
670 IF J<>J1 THEN J = J+1: GOTO 810
680 'determine graph colors
690 L = 0
700 IF G1>G2 THEN IF G2>G3 THEN X=G1: Y=G2:
Z=G3: CX=4: CY=2: CZ=3: L=1 ELSE IF G1>G3 THEN
X=G1: Y=G3: Z=G2: CX=4: CY=3: CZ=2: L=1 ELSE
X=G3: Y=G1: Z=G2: CX=3: CY=4: CZ=2: L=1
710 IF L=1 THEN 730 ELSE IF G1>G3 THEN X=G2:
Y=G1: Z=G3: CX=2: CY=4: CZ=3 ELSE IF G2>G3 THEN
X=G2: Y=G3: Z=G1: CX=2: CY=3: CZ=4 ELSE X=G3:
Y=G2: Z=G1: CX=3: CY=2: CZ=4
720 'draw graph lines
730 COLOR CZ, 1
740 LINE (12, 172)-(12, 171-Z*167), PSET
750 COLORCY, 1
760 LINE -(12, 171-Y*167), PSET
770 COLORCX, 1
780 LINE -(12, 172-X*168), PSET
790 J = 1
800 I2 = I2+1
810 NEXT I
820 FOR I=1 TO 3000
830 NEXT I
840 'print start and stop values
850 CLS
860 F$ = " = #.### #.###    #.##"
870 PRINT USING " GEN = 0   ####";I1
880 PRINT TAB(22) "FITNESS"
890 PRINT USING "A1A1"+F$;P2, G1, W1
900 PRINT USING "A1A2"+F$;PQ, G2, W2
910 PRINT USING "A2A2"+F$;Q2, G3, W3
920 F$ = " = #.### #.###   #.#^^^^"
940 PRINT TAB(23) "RATE"
950 PRINT USING " A1"+F$;P1, P, U
960 PRINT USING " A2"+F$;Q1, Q, V
```

```
970 IF M=0 THEN 1010
980 PRINT
990 PRINT USING "IMMIGRANT A1 FREQUENCY =
#,###";PM
1000 PRINT USING" IMMIGRANT PROPORTION =
#.###";M
1010 POKE 65494, 0
1020 PRINT
1030 PRINT"PRESS G TO SEE GRAPH, THEN D TO
SEE DATA AGAIN."
1040 K$ = INKEY$
1050 IF K$="" THEN 1040 ELSE IF K$="D" THEN 850
ELSE SCREEN1, 0: GOTO 1040
```

Data & Sorts of all Sorts

By James A. Hornsby

One of the primary uses of a computer is to manage data. A Data Base Managment (DBM) system is a program that will create files containing data we wish to store, retrieve, edit, sort or print. So that we speak the same language, I offer the following definitions!

An item or field is an individual piece of data (name or zipcode).

A record is a collection of related fields (i.e. a club member's name, address, zipcode, and membership expiration date).

A file is a collection of related records (i.e. club membership roster).

A common method of structuring the files used by a DBM would be to set up a 2 dimensional string array, like D$(399,4) with 400 records and 5 fields (0 being a valid member of the array). However, this method uses a lot of memory because each BASIC string requires a 5 byte descriptor and since we have 2,000 strings, we would need 10,000 bytes for string descriptors. Another method of structuring the file is to set up a single dimension string array, D$(399), with 400 records, each string containing a complete record. This method uses 2,000 bytes for string descriptors so we save 8,000 bytes of precious memory. The trade-off is that we must find some way to access the fields.

One method of doing this is to use fields of fixed length. Each field may then be located by its position in the string. By using this technique, any given field must be as long as the longest entry that might appear in that field. We can use the end of the string to hold data which will not need to be sorted, or separated from the string.

Another way to access the fields would be to use delimiters. A delimiter is a character used to mark the end of one field and the beginning of another. This is similar to writing a sentence because you use a space as a delimiter between words. You can locate any given field by searching through the string, and counting delimiters until you arrive at the field needed. Delimiter characters should be chosen with care. If a comma is used as a delimiter then do not use a comma in the field itself, the program will think there are 2 fields instead of 1. Use a character for which you see no other use!

Now, let's look at a simple file program using cassette based sequential files. This program assumes that the file is in memory and that it uses fixed length fields. In this example the first field is always the date, for example! January 2, 1982 would be entered as MONTH 1, DAY 2, YEAR 82). You can set the category and length of the 2nd and 3rd fields. If the field lengths are made larger than 16, the Print to Printer or Screen subroutines will have to be changed. The 4th field is always the amount field (for $999.99, enter 999.99). Records can be input from the keyboard only after New file has been described by command 1, or a file has been input from tape. After a file has been described or loaded from tape, it cannot be redescribed. If you load another file with one in memory, you'll be asked if you wish to save the file already in memory. A "Y" causes a return to main menu a "N" prompts the loading procedure to continue, which will write over the file in memory.

With the file in memory, we can edit records, either changing or deleting them, or you can input more records. When the file capacity is reached, a "FILE IS FULL" message is displayed and you can get a total of all amount fields. You can print records to the screen or printer by starting at any record number and ending with "R" to return to main menu, then you can save the file or files to tape.

Lastly, we can sort by one of 3 fields. This sort speed is not to bad but it does not preserve order in a pre-sorted field and it will sort 400 records made up of 5 random characters in about 4 -5 minutes. Sorting is a fascinating subject and is used in different types of programs.

FILES, GRAPHICS etc. Speed is almost always the most important factor, since all activity is stopped until the sort is finished. The fastest sorts work by comparing 2 records separated by a gap, so the records move in large steps toward their proper position in the file. This gives speed, but prevents order keeping in a presorted field. For order keeping! sort by NAME first, then sort by STATE. You finish with a list of entries alphabetized in order of states, and the names in each state are also in alphabetical order. If order keeping is desirable then use 2 sorts. The fast sort is called on the primary field and the slower linear sort is called on the secondary field. I have written sorts in machine language, which are called from basic with the USR function and are much faster and will sort 400 records consisting of 5 random characters in approximately 3 -5 seconds! These are available through ML-USER SOFTWARE. Order keeping sort algorithms are inherently slower, but can be remarkably fast by using machine language.

This program is made for 32K! for 16K make the following changes!

In line 375 from Y=400 to Y=200.

In line 700 from Z>399 to Z>199.

In 1085 from X)400 to X)200.
In line 90 from D$(199).

```
3 REM        FILE 5.5
4 REM      COPYRIGHT 1982
5 REM     JAMES A. HORNSBY
10 GOTO 1200
15 REM SORT
16 IF PP<1 THEN RETURN ELSE CLS: PRINT @14,
"SORT": PRINT: PRINT: INPUT "FIELD TO SORT
BY";K
18 IF K=1 THEN Z=1: ZZ=8: GOTO23
19 IF K=2 THEN Z=10: ZZ=A: GOTO23
20 IF K=3 THEN Z=10+A: ZZ=B: GOTO23
21 GOTO 16
23 GAP=(P-1)/2
24 FOR N=0 TO (P-1)-GAP
25 J=N
26 IF MID$(D$(J), Z, ZZ)<=MID$(D$(J+GAP), Z, ZZ)
THEN GOTO 30
27 X$=D$(J): D$(J)=D$(J+GAP): D$(J+GAP)=X$
28 J=J-GAP
29 IF J>0 THEN GOTO 26
30 NEXT N
31 GAP=INT(GAP/2): IF GAP>0 THEN GOTO 24
32 RETURN
48 REM TOTAL
49 IF PP<1 THEN RETURN ELSE CLS: PRINT @269,
"ADDING": T=0
50 FOR Z=0 TO P-1
60 T=T+VAL(MID$(D$(Z), (9+A+B)))
70 NEXT Z
80 RETURN
90 CLEAR 12000, 32511: DIM D$(399)
100 FOR I=1 TO 24
110 READ B: POKE 1536+I, B
120 NEXT I
130 DATA 173, 159, 160, 00, 39, 250, 129, 76, 39, 08,
129, 83, 39, 04, 129, 82, 38, 238, 31, 137, 79, 126, 180,
244
140 DEFUSR0=1537
150 CLS: PRINT @12, "FILE 5.5": PRINT @41,
"COPYRIGHT 1982": PRINT @72, "JAMES A.
HORNSBY": PRINT: PRINT "IF YOU SHOULD GET AN
ERROR WHILEUPDATING A FILE, TYPE GOTO 170
AND ENTER IT. DO not TYPE RUN ORTRY TO EDIT
THE PROGRAM UNTIL  THE FILE HAS BEEN
SAVED,"
```

```
151 PRINT "ALWAYS  MAKE A BACKUP COPY OF
EACH     IMPORTANT FILE; ALWAYS VERIFY
THAT THE FILE YOU HAVE JUST     SAVED IS OK
ON TAPE."
152 PRINT @480, "PRESS R TO CONTINUE";
155 IF INKEY$<>"R" THEN GOTO 155
160 P=0: PP=0: B$="PRESS R TO RETURN TO MENU"
170 CLS
180 PRINT @0, "  PRESS <BREAK> IF FILE DONE"
190 PRINT @70, "FUNCTION SELECT"
200 PRINT @133, " 1  DESCRIBE NEW FILE"
220 PRINT @165, " 2  INPUT TO FILE"
230 PRINT @197, " 3  EDIT FILE"
240 PRINT @229, " 4  PRINT RECORDS"
250 PRINT @261, " 5  SORT FILE"
255 PRINT @293, " 6  TOTAL AMOUNT"
260 PRINT @325, " 7  SAVE FILE ON TAPE"
270 PRINT @357, " 8  LOAD FILE FROM TAPE"
280 IF T<>0 THEN PRINT @392, "TOTAL =";: PRINT
USING "$$####.##";T
290 PRINT @448, "FILENAME: ";F$
300 K$=INKEY$: K=VAL(K$)
310 IF K=1 OR K=2 OR K=3 OR K=4 OR K=5 OR K=6
OR K=7 OR K=8 THEN K=K ELSE GOTO 300
320 ON K GOSUB 340, 360, 420, 670, 15, 48, 770, 960
330 GOTO 170
340 REM  DESCRIBE FILE
345 IF PP>0 THEN RETURN ELSE CLS: PRINT @3, B$:
PRINT: PRINT: INPUT "FILENAME";F$: INPUT
"FIELD 2 PROMPT";E$: INPUT "FIELD 2 LENGTH";A:
INPUT "FIELD 3 PROMPT";P$: INPUT "FIELD 2
LENGTH";B: PP=1
350 RETURN
355 REM INPUT
360 IF PP<1 THEN RETURN ELSE CLS: T=0: PRINT
@3, B$
370 PRINT @40, "INPUT ITEMS"
375 IF P=400 THEN PRINT "FILE IS FULL" ELSE
GOTO 380
376 IF INKEY$="" THEN GOTO 376 ELSE RETURN
380 PRINT "RECORD" P
390 INPUT "MON";M$: IF M$="R" OR M$="" THEN
RETURN ELSE M$="0"+M$: M$=RIGHT$(M$, 2): INPUT
"DAY";D$: D$="0"+D$: D$=RIGHT$(D$, 2): INPUT
"YEAR";Y$: S$=M$+"/"+D$+"/"+Y$+" "
```

```
395 PRINT E$;: LINE INPUT "? ";Z$:
T$=Z$+STRING$(A-(LEN(Z$)), 32): PRINT P$;: LINE
INPUT "? ";Z$: U$=Z$+STRING$(B-(LEN(Z$)), 32):
INPUT "AMOUNT";D: D$=" "+STR$(D):
D$(P)=S$+T$+U$+D$
400 P=P+1
410 GOTO 360
420 REM  EDIT RECORDS
430 IF PP<1 THEN RETURN ELSE CLS: PRINT @3, B$:
PRINT @70, "SELECT": PRINT @134, "E  EDIT
RECORD": PRINT @166, "D  DELETE RECORD"
440 K$=INKEY$: IF K$="D" THEN GOTO 540 ELSE IF
K$="R" THEN RETURN ELSE IF K$<>"E" THEN GOTO
440
450 N=0
460 CLS: PRINT @3, B$
470 PRINT @42, "EDIT RECORDS"
480 PRINT: INPUT "ITEM NO. TO EDIT";N$: IF
N$="R" THEN RETURN ELSE N=VAL(N$)
490 IF N >= P THEN GOTO 450 ELSE T=0
520 CLS: PRINT @128, D$(N): LINE INPUT "SEGMENT
TO CHANGE? ";X$: IF X$="" THEN GOTO450 ELSE IF
X$="R" THEN GOTO450
530 PRINT @288, "": LINE INPUT "CHANGE TO? ";Y$:
L=LEN(X$): R=INSTR(D$(N), X$): Z1$=LEFT$(D$(N), 9):
Z2$=MID$(D$(N), 10, A): Z3$=MID$(D$(N), (10+A), B):
Z4$=MID$(D$(N), 10+A+B): IF R<10 THEN GOTO 532
ELSE IF R<(10+A) THEN GOTO 534 ELSE IF
R<(10+A+B) THEN GOTO536
531 IF LEN(Y$)>5 THEN GOTO 539 ELSE
R=INSTR(Z4$, X$): Y1$=LEFT$(Z4$, R-1):
Y2$=MID$(Z4$, R+L): Z4$=Y1$+Y$+Y2$: GOTO 538
532 IF LEN(Y$)>L THEN GOTO 539 ELSE
R=INSTR(Z1$, X$): Y1$=LEFT$(Z1$, R-1):
Y2$=MID$(Z1$, R+L): Z1$=Y1$+Y$+Y2$: GOTO 538
534 R=INSTR(Z2$, X$): Y1$=LEFT$(Z2$, R-1):
Y2$=MID$(Z2$, R+L): ZT$=Y1$+Y$+Y2$: FOR Z=0 TO 1:
L=LEN(ZT$): IF L>A THEN IF RIGHT$(ZT$, 1)=" "
THEN ZT$=LEFT$(ZT$, L-1): Z=0: NEXT Z
535 NEXT Z: IF L>A THEN GOTO 539 ELSE
Z2$=ZT$+STRING$(A-(LEN(ZT$)), 32): GOTO 538
536 R=INSTR(Z3$, X$): Y1$=LEFT$(Z3$, R-1):
Y2$=MID$(Z3$, R+L): ZT$=Y1$+Y$+Y2$: FOR Z=0 TO 1:
L=LEN(ZT$): IF L>B THEN IF RIGHT$(ZT$, 1)=" "
THEN ZT$=LEFT$(ZT$, L-1): Z=0: NEXT Z
537 NEXT Z: IF L>B THEN GOTO 539 ELSE
Z3$=ZT$+STRING$(B-(LEN(ZT$)), 32): GOTO 538
538 D$(N)=Z1$+Z2$+Z3$+Z4$: GOTO 450
539 PRINT @256, "MAKES FIELD TOO LONG!": GOTO
530
540 REM  DELETE RECORDS
550 N=0
560 CLS: PRINT @3, B$
570 PRINT @42, "DELETE RECORDS"
580 PRINT: INPUT "ITEM TO DELETE";N$: IF N$="R"
THEN RETURN ELSE N=VAL(N$)
590 IF N>=P THEN GOTO 560
600 T=0
610 D$(N)=D$(P-1)
640 D$(P-1)=""
650 P=P-1
660 GOTO 540
670 REM  PRINT RECORDS
671 IF PP<1 THEN RETURN ELSE CLS: PRINT @3, B$:
PRINT @138, "DEVICE SELECT": PRINT @202, "P
PRINTER": PRINT @234, "S  SCREEN"
672 K$=INKEY$: IF K$="S" THEN GOTO 680 ELSE IF
K$="P" THEN GOTO 680 ELSE IF K$="R" THEN
RETURN ELSE GOTO 672
680 CLS: PRINT @128, "": INPUT "FIRST ITEM
NUMBER TO PRINT";C
690 FOR X=C TO P-1 STEP 7: PRINT ""
700 FOR Z=X TO X+6: IF Z>999 THEN GOTO 730
705 IF K$="P" THEN PRINT #-2, USING "###";Z;:
PRINT #-2, CHR$(32);: PRINT #-2, LEFT$(D$(Z),
(9+A));: PRINT #-2, STRING$(4, 32);: PRINT #-2,
MID$(D$(Z), (10+A), B);: PRINT #-2, USING
"$$####.##";VAL(MID$(D$(Z), (9+A+B)))
710 IF K$="S" THEN PRINT USING "###";Z;: PRINT
CHR$(32);: PRINT LEFT$(D$(Z), (9+A));: PRINT
STRING$(4, 32);: PRINT MID$(D$(Z), (10+A), B);:
PRINT USING "$$####.##";VAL(MID$(D$(Z), (9+A+B)))
720 NEXT Z
730 PRINT @0, "": PRINT @3, B$: PRINT @480, "";:
PRINT "PRESS (SPACEBAR) TO CONTINUE";
740 C$=INKEY$: IF C$="" THEN GOTO 740 ELSE IF
C$="R" THEN RETURN
750 NEXT X
760 RETURN
770 REM  SAVE FILE ON TAPE
780 IF PP<1 THEN RETURN ELSE CLS: PRINT @3, B$:
PRINT @326, "": INPUT "FILENAME";G$: IF G$="R"
THEN RETURN
790 CLS: PRINT @103, "SAVE FILE ON TAPE"
800 PRINT @170, "POSITION TAPE"
810 PRINT @230, "PRESS PLAY AND RECORD"
820 PRINT @3, B$
830 PRINT @448, "PRESS S WHEN READY"
840 R=USR0(0)
850 IF R=83 THEN GOTO 880
860 IF R=82 THEN RETURN
870 GOTO 840
880 CLS: OPEN "O", #-1, G$
890 PRINT #-1, P, A, B, E$, P$: PRINT E$;" ";A;"
";P$;" ";B;" ";P
910 FOR X=0 TO P-1
920 PRINT #-1, D$(X)
930 NEXT X
950 CLOSE #-1: RETURN
960 REM  LOAD FILE FROM TAPE
965 CLS: PRINT @3, B$: IF PP>0 THEN PRINT: PRINT:
INPUT "DO YOU WISH TO SAVE THE FILE
ALREADY IN MEMORY? (Y/N)";Z$: IF Z$="Y" THEN
RETURN
```

SORTS OF ALL SORTS by Craig Clampitt

```
970 CLS: PRINT @3, B$: PRINT @326, "": INPUT
"FILENAME";G$: IF G$="R" THEN RETURN
980 CLS: PRINT @102, "LOAD FILE FROM TAPE"
990 PRINT @171, "REWIND TAPE"
1000 PRINT @236, "PRESS PLAY"
1010 PRINT @3, B$
1020 PRINT @448, "PRESS L WHEN READY"
1030 R=USR0(0)
1040 IF R=76 THEN GOTO 1070
1050 IF R=82 THEN RETURN
1060 GOTO 1030
1070 CLS: OPEN "I", #-1, G$
1080 INPUT #-1, X, A, B, E$, P$: PRINT E$;" ";A;"
";P$;" ";B;" ";X
1085 IF X>400 THEN CLS: PRINT @3, B$: PRINT @128,
"FILE WOULD EXCEED DIMENSIONED ARRAY."
ELSE GOTO 1090
1086 IF INKEY$="" THEN GOTO 1086 ELSE RETURN
1090 FOR P=0 TO X-1
1100 LINE INPUT #-1, D$(P)
1110 NEXT P
1130 CLOSE #-1: F$=G$: PP=1: RETURN
1200 PCLEAR1: GOTO90
2000 FOR P=0 TO 399: FOR Z=1 TO 5: D$(P) = D$(P)+
CHR$(64+ RND(25)): NEXT Z: NEXT P: STOP
```

Color Computer News

Are you tired of searching the latest magazine for articles about your new Color Computer? When was the last time you saw a great sounding program listing only to discover that it's for the Model I and it's too complex to translate? Do you feel that you are all alone in a sea of Z-80's? On finding an ad for a Color Computer program did you mail your hard earned cash only to receive a turkey because the magazine the ad appeared in doesn't review Color Computer Software? If you have any of these symptoms you're suffering from Color Computer Blues!

But take heart there is a cure!

It's COLOR COMPUTER NEWS.

The monthly magazine for Color Computer owners and only Color Computer owners. CCN contains the full range of essential elements for relief of CC Blues. Ingredients include: comments to the ROMS, games, program listings, product reviews, and general interest articles on such goodies as games, personal finances, a Kid's page and other subjects.

The price for 12 monthly treatments is only $21.00 and is available from:

Mail Today!

REMarkable Software

P.O. Box 1192
Muskegon, MI 49443

NAME _____

ADDRESS _____

CITY _____ State _____ Zip _____

Allow 8-10 weeks for 1st issue.

CCN

Educating Your Preschooler

Paul Lee

I have a 2 1/2 year old daughter that is just completely infatuated with my Color Computer. Whenever I am using it and she is in the room, she will insist on sitting on my lap and watching the computer. Because of this, I felt that the computer would be an excellent tool for teaching my daughter the alphabet.

This program flashes each letter of the alphabet on the TV screen, and then announces what letter it is through the speaker of the TV. I sat my daughter in front of the screen and ran the program. To my surprise, my daughter repeated each letter after it was announced. After several sessions in front of the computer, she is beginning to recongnize several letters of the alphabet and will say them before the TV announces it. She loves watching the program.

The program is very simple. It uses the DRAW statement to draw the letters of the alphabet, and the PLAY statement to make sounds when the letter is flashed on the screen. Each letter is flashed on the screen 10 times using the SCALE subroutine. Each time it is flashed, the letter increases in size from scale S5 to S50 in steps of 5.

Line 240 turns the tape recorder and audio on, and line 250 holds the letter on the screen for approximately 7 seconds. This is plenty of time to repeat the letter three times on the tape. When recording the announcement on the tape, I recommend that you first CSAVE the program on the tape. Then leave the tape recorder on "record" and RUN the program. The program will turn the recorder on and off, and you can "record" the alphabet in this way. The tape recorder will always be automatically set-up if you do it this way.

I made three recordings of the alphabet so that it could be rerun three times without rewinding the tape recorder. This is also about the maximum time for the attention span of my daughter. For the first recording, I repeated the alphabet three times. The second recording I said, "What letter is this? - It's an A." The third recording I said, "A - apple - A." In the third recording, I used words that were familiar to my daughter.

The utility subroutine allows the program to be restarted by pressing any key. My daughter has learned how to do this. Once the program is RUN, she can operate the computer without my assistance.

My next project for my daughter will be a Numbers program. This will display the numbers from 1 to 20 for her to learn. So until next time...

TABLE I

LINES	FUNCTIONS
1-10	Remarks and Clear
20-40	Utility Subroutine
50	DRAW Subroutine
60-110	Scale Subroutine
200-280	Main Program
300-340	DATA Statements

```
5 '   ALPHABET PROGRAM
BY PAUL LEE
136 OLIVE COURT
HERCULES, CALIF 94547
10 CLS: GOTO200
20 PRINT" REWIND TAPE RECORDER AND PRESS
PLAY. THEN TOUCH (ANY KEY) TO REPEAT THE
ALPHABET."
30 A$=INKEY$: IFA$="" THEN 30
40 RESTORE: CLS: RETURN
50 DRAW"BM90, 50; "+LT$: PLAY"O3; L10; T50; A; B;
C; D; E; F; G": RETURN
60 FOR S=5TO50 STEP 5
```

```
70 PCLS: SCREEN1, 1
80 S$="S"+STR$(S)
90 DRAW S$: PCLS: GOSUB50
100 NEXT S
110 RETURN
200 FOR K=1TO29
210 READ LT$
220 PMODE3, 1: PCLS: SCREEN1, 1
230 GOSUB60
240 AUDIOON: MOTORON
250 FOR T=1TO3000: NEXTT
260 MOTOROFF: AUDIOOFF
270 NEXTK
280 GOSUB20: GOTO200
300 DATA BD1ND6E1 R3F1D2NL5D4, ND7R4F1D1G1
NL4F1D2G1L4, BR4NF1L3 G1D5F1R3E1,
R4F1D5G1L4U7, NR5D3 NR4D4R5
310 DATA NR5D3NR4D4, BR4NF1 L3G1D5
F1R3E1U2L2, D3ND4 R5NU3D4, R6L3D7NL3R3,
R6L2D6 G1L2H1U1
320 DATA BR1D4 ND4NE4F4, D7R5, ND7F3E3D7,
ND7F7U7, ED1D5F1R3 E1U5H1L3G1
330 DATA ND7R4F1D1G1L4, BD2E2R2F2 D2G1NH1F1
R1L1H1 G1L2H2U2, ND7R4F1 D1G1NL4 F1D3,
BR4NF1L3 G1D1F1 R3F1D2 G1L3H1, R6L3D7
340 DATA D6F1R3E1U6, BL3F6E6, D7E3F3U7,
BH1F4NE4NG4F4, F3NE3D4, R7G7R7
```

C. C. Writer

[Word Processing for the TRS-80‡ Color Computer]

Features Page Formatting, Block Moves, Tabs, Sentence Deletion and Insertion, Global Search and Replace, Centering, Indenting, Page Pause, ASCII Code Transmission, Justification, Scrolling Review, Keyboard Stops, and File Chaining. 16-32K cassette-$35. 32K Disk-$40 (Shipping late Feb.)

NEW! --> Check Rec Plus

Reconciles your Checkbook AND allows you to keep Memo Entries of cash and credit card expenses without affecting your Checkbook balance. The History Files may be Listed and Totaled by Income or Expense Category for Budgeting or Tax Preparation. On screen Trial Balancing and printed Trial Balances, Audit Trails, and Summaries (REQUIRES PRINTER). Current balance is displayed on the Screen Menu and Reports. Save yourself frustration and perhaps some money too.

--> Prices until April 15th: <--

16K Version-$25, 32K with extensive prompts-$30

16-32K Disk System with Prompts-$35

All versions include System Binder, Cassettes or Disk with storage pockets, and documentation. (Shipping late Feb.)

For information or orders write:

TransTek BSC 2-2
194 Lockwood
Bloomingdale, IL 60108

‡ TRS-80 is a Trademark of Tandy Corporation

The other day my friendly Radio Shack dealer called and said that he had just received a Color Scripsit cartridge and wondered if I would be interested. Well I have a R.S. Disk System and was hesitant to purchase something which would create a problem in terms of plugging and unplugging at the cartridge slot and would be unable to support disk files. After a few minutes of deep thought I decided that any word processor should be better than none,and purchased the Color Scripsit cartridge at $39.95.

The program comes in a neat package with a 40 page reference manual. After spending a few minutes (very few - I hate manuals) of reading the instructions, I unplugged the disk and plugged in the "COLOR SCRIPSIT". I was met with a menu of six selections:

1 = CLEAR MEMORY
2 = EDIT TEXT
3 = SAVE ON TAPE
4 = LOAD FROM TAPE
5 = PRINT
6 = CHANGE STANDARDS

Many of the above commands have subcommands and all give you the option of returning to the main menu should you arrive somewhere in error or change your mind. I elected to get right down to business and do some typing so I pressed <2>. Since I had not set the print standards, the document I was typing appeared 32 characters wide showing 13 lines. (The bottom line is a status line and constantly displayed - more about that later) line 14 is reserved for tab markers. It didn't take me long to make a few typos so I had to resort to the manual for instructions. When editing text, the break key plus one other key is used to manipulate the text. Following is a brief summary of thesecommands:

BREAK <1> = CONTROL OR HELP
BREAK <2> = CHANGE BACKGROUND COLOR
BREAK <3> = OPENS LINE FOR INSERTIONS
BREAK <4> = PRINTS ONE LINE ON PRINTER
BREAK <5> = SETS TABS
BREAK <6> = CHANGE LINE ALIGNMENT (CENTERED, LEFT OR RIGHT)
BREAK <7> = SEARCH AND CHANGE MODE
BREAK <8> = FINDS WORDS FOR POSSIBLE HYPHENATION
BREAK <9> = DELETE BLOCK OF TEXT
BREAK <0> = HEADERS, FOOTERS OR PAGE #
BREAK <!> = COPY OR REPEAT BLOCK OF TEXT

The clear key is used to delete one character and the shift/clear are used to delete words. The cursor is moved about the text with the arrow keys. Pressing the shift/up arrow takes you to the beginning of your text and the shift/down arrow takes you to the end of the text. The shift/right arrow takes you to next tab as will the shift/left takes you back to the last tab. If no tabs are in place the cursor goes to the beginning or end of the line depending on which arrow was used. I encountered no problems using the above functions and found them all to be easily utilized after some practice.

I felt that the manual was well written and adequate for my purposes. I did feel that more explanation may have been provided regarding the print to tape function. This is an option offered on pressing the print command. Apparently Color Scripsit can be used with Color Basic programs if they accept ASCII files. Printing to tape takes much longer than saving on tape with the regular save mode (option 3 of main menu). A few words about saving on tape are in order. The program allows you to name the file (up to 8 characters) and asks which file you want your file recorded after. The screen displays "SCANNING" as the computer searches for the desired file then stops after it to allow you to set the tape player properly for recording. After having a disk system for awhile this seems rather archaic but from what I've read it sure beats the Model I or III systems for tape handling.

Now about the print standards, I found it best to set these after you have entered your text. This allows you to view your message in a continuous manner. After the line width is set greater than 32 the screen divides into two halves (1 to 32 and 33 to ??). If the width is set greater than 63, the whole screen jumps one character at a time to the right for awhile then displays the right portion of the text. This was a little confusing and makes it difficult to read a complete sentence. However, it does allow you to view the document in the form to be printed. On power up the standard for width is set at 32 and this seems to be the best setting for entering text.

I was generally pleased with the program and feel that it is worth its price. I have a R.S. Line Printer VIII and had hoped to use some of its special functions but Color Scripsit does not offer a way to underline or use expanded print etc. My biggest complaint though is what I was afraid it would be - plugging and unplugging the cartridge in order to use the C.C. Disk System. Oh well, I bet one of these days there will be hundreds of good Color Computer Word Processors that work with Disk Files and the new generation printers but meanwhile Color Scripsit gets the job done and is easy to use.

Software Review: GAUNTLET
by Jay R. Hoggins

DESCRIPTION: GAUNTLET is an arcade type game which is designed to run on a 4K (or 16K) Color Computer. The screen has a neat little blue rocket ship at the bottom which you move right and left with your joystick. There is a background field of stars, meteors and evil purple aliens that come down the screen at your ship. This gives you the feeling of forward movement through the field. You have control of your speed through this field by moving the joystick forward or backward. Game is over if you collide with an alien, a meteor or the red bombs that the aliens drop. You may shoot at will, but you only get 100 photon missles. The game has a scoreboard at the bottom edge of the screen. You get 100 points for shooting an alien and one point for each step of movement forward. The program limits you to 10,000 "Zenonian" seconds per game. The faster you move through space the slower the seconds tic off. (Must have something to do with Einstein's theory!)

CONCLUSION: GAUNTLET'S graphics are good. The sound is quite good; the action and game plot are excellent. GAUNTLET is written in machine language and thus is fast and fun. I didn't think that I was an arcade game freak at all until I got my CC and a few games like GAUNTLET. Now I play for hours, ending up with sweaty palms and a stiff neck - that is, if I can get my computer and GAUNTLET away from my 6 year old son. The game is easy enough for my son and still challenging enough for me. Some CC owners seem to think that any software that is written to run on a 4K machine is not possibly good enough to bother with. Their 16K or greater machines require complex, long programs. Nonsense! Many excellent programs will run on 4K machines and will rival many longer more complex programs. GAUNTLET is a good example. Of course the program does not use super hi-res graphics, but the entertainment value is there, and the price is right.

Dear Bill:

I have discovered a serious bug in the "1981 Income Tax" program I submitted for your magazine. The error is the result of a late change with limited testing of parameters. As submitted, Schedule A will be incorrect if the expenditures for "Medicine and Drugs" (line 2) are greater than 1% of the gross adjusted income (line 3).

To correct the error, change line 11080 to: HD = HB-HC

The program was geared to using rounded dollar amounts. Persons who don't choose that route may want to simplify by making the following changes:

11200 HL = HJ-HK
11460 MC = MA-MB

This eliminates a couple of unneeded integer calculations. But do NOT change any other integers-they are critical to the tax calculations in virtually every other case.

Sorry about the goof-all tapes will have the correction.

Not having a printer makes editing long programs a real problem. Here are some more changes which improve the program, delete unnecessary lines, etc.

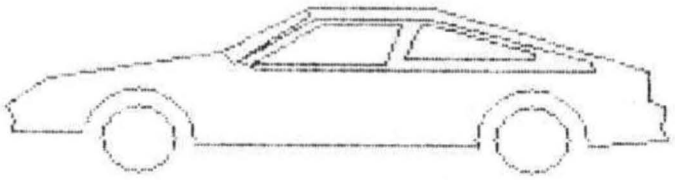
```
6285 PRINT: INPUT "OTHER PAYMENTS"; GP
6290 IF GM+GP=GL THEN CLS: PRINT@ 234,
"NO REFUND BUT": PRINT @262, "NOTHING OWED
EITHER!"
6660 PRINT @230, "BALANCE OWED:$"
GL-(GM+GP);
```

DELETE LINES 6520, 6530, 6540, 6550, 6560, 6570, 6340, 6345

ADD:

```
6470 FOR X = 1 TO 100: NEXT X
6480 PRINT @232, "REFUND=$" GM+GP-GL;
6490 FOR X = 1 TO 250: NEXT X: NEXT Y
6500 PRINT @456, "HOW ABOUT THAT!";
6510 FOR X = 1 TO 1500: NEXT X
6520 RETURN
```

Warren S. Napier
Rd 1, Box 475
Mars, PA 16046



The above picture and the program below were inadvertently left out of Kathy Goebel's printer article in the January issue.

```
5 PMODE 4, 1
10 PCLS
20 SCREEN 1, 0
30 READ X, Y
40 IF X=999 THEN 100
50 IF X<0 THEN X1=-X: Y1=Y: READ X, Y: LINE(X1,
Y1)-(X, Y), PSET
60 IF X>=0 THEN LINE -(X, Y), PSET
70 GOTO 30
100 CIRCLE (48, 60), 12: CIRCLE(172, 60), 12
110 CIRCLE(48, 60), 18, 1, 1, .55, 1
120 CIRCLE(172, 60), 18, 1, 1, .48, 1
130 ' SCREEN PRINT ROUTINE
140 FOR I=0TO6: T(I)=INT(2^I): NEXT
150 PRINT#-2, CHR$(18);
160 FOR Y=12 TO 80 STEP 7
170 FOR X=0 TO 255
180 A=128
190 FOR Z=0 TO 6
200 IF PPOINT(X, Y+Z) THEN A=A+T(Z)
210 NEXT Z
220 PRINT#-2, CHR$(A);
230 NEXT X
240 PRINT#-2
250 NEXT Y
260 PRINT#-2, CHR$(30)
270 GOTO 999
600 DATA -6, 49, 19, 39, 74, 30, 102, 14, 141, 14, 209,
36, 209, 46, 214, 48, 214, 53
610 DATA 213, 55, 215, 60, 180, 62, -154, 62, 66, 62,
-30, 58, 8, 58, 9, 54, 6, 49
620 DATA -74, 30, 78, 34, 102, 17, 101, 15, -104, 18,
140, 18, 191, 33, 192, 36
630 DATA 85, 36, 80, 33, 104, 18, -106, 20, 131, 20,
126, 34, 84, 34, 106, 20
640 DATA -137, 20, 140, 20, 173, 30, 173, 32, 132, 32,
137, 20, 999, 0
999 END
```

DIRECT PLUG ISOLATORS

ELECTRONIC SPECIALISTS expands their patented ISOLATOR Line to include units that plug directly into the wall socket. Designed for installations that do not require extension cords, the Direct Plug ISOLATORS provide the same equipment interaction isolation and power line protection as their popular line cord ISOLATOR series. A convenient retention screw prevents accidental withdrawal from the wall socket.

Direct Plug ISOLATORS can accommodate a total 1875 watt load, with up to 1000 watts per socket. A high capacity Spike/Surge Suppressor is designed into each unit.

Direct Plug SUPER ISOLATOR (Model DP-SIS032) provides 2 super-isolated channels for $96.95.

Electronics Specialists, Inc., 171 S. Main St., Box 389 Natick, Massachusetts 01760 Phone: (617) 655-1532 For more information: Frank Stifter

DATAPACK
Data Communications Package

DATAPACK is a Terminal package program for the COLOR COMPUTER, allowing you to use the color computer as a buffered computer terminal through a modem to a time sharing network or as a direct connect terminal to another computer system at rates up to 3600 baud. This program is more than a standard "Videotext" type program in that it will allow you to save data stored in the buffer either to cassette tape, or output a hard copy to a printer. The data buffer is automatically set to the maximum size of your system memory when entered to allow maximum space for saving data. The program includes features to send control codes and to enable or disable keyboard echo. When the terminal mode is exited the contents of the buffer may be viewed on the screen or saved to tape for later loading. Also the RS-232 port can be used to plug your printer back in for sending the screen buffer to the printer. An additional feature is the ASCII format that is used on tape is compatible with the CER-COMP Text Editor program and BASIC, enabeling you to edit or delete unwanted information.

SPECIAL INTRODUCTORY PRICE: $24.95 ON CASSETTE

CER-COMP
5566 Ricochet Ave.
Las Vegas, Nevada 89110
Ph# (702) 452-0632

Mark Data Products releases CAVE HUNTER, a new action game fo the 16K TRS-80 Color Computer. Maneuver your way to the bottom of a spooky old cave to retrieve the treasures. It's not so easy! Passages lead in all directions and angry cave creatures pursue you relentlessly.

CAVE HUNTER is a fast-paced arcade game using hi-res graphics, sensational colors and a variety of unique sounds. Single or multiple players. Joysticks are required.

Priced at $24.95, this game is available postpaid from:

Mark Data Products
23802 Barquilla
Mission Viejo, Ca 92691
(714)768-1551

TEXTPRO is a complete text edition & text processing program for the Color Computer. The program includes our powerful function text editor plus the added features of a text processor. The entire program utilizes only 6K of memory space including the tape, screen and keyboard buffers. It is extremely fast in editing and processing text files and is compatible with Basic ASCII formatted tape files.

The Editor itself includes 24 commands including string search & replace, line and automatic line edit modes which allow you to insert, delete, change or add characters. Automatic line editing allows you to skip forward and backward for checking and editing, all screen editing immediately updates the screen so you know exactly what you are doing at all times. The Editor also has commands to move or copy single lines or blocks of text from one place to another. Some of the other commands include Tape load, save, and append; Automatic line numbers, delete line, set input line length and printer output. One other nice little feature is during line entry an audible "beep" is output seven characters before the end of line, at the end of line and if an illegal character is entered. This alone can increase your input speed by 100%.

The Text Processor includes 29 commands for formatting the output, some of them include: page length, left margin, top & bottom margin, line length, justify & fill modes, page heading, center line, double width print, margin control, single, multiple & special indent modes, test lines left on page, display & input from keyboard and even special control codes can be sent to the printer for different print densities etc. It even has a repeat command with a next command to redo all of or a portion of the file as many times as needed. TEXTPRO will turn your Color computer into

a full fledged text processing machine at a price you won't believe, available on "CLOADM" compatible cassette. Only $29.95

CER-COMP
5566 Ricochet Ave.
Las Vegas, NE 89110
(702)452-0632

DISK TEXTPRO
Text Editor/Processor
for R.S. Color Disk System

TEXTPRO is a complete text editor and text processing program for the R.S. Color Computer Disk System. The program includes our powerful function text editor plus the added features of a text processor. The entire program utilizes only 6K of memory space including the tape, screen and keyboard buffers. It is extremely fast in editing and processing text files and is compatible with Basic ASCII formatted Tape and Disk files.

The Editor itself includes 24 commands including string search and replace, line and automative line edit modes which allow you to insert, delete, change or add characters. Automatic line editing allows you to skip forward and backward for checking and editing, all screen editing immediately updates the screen so you know exactly what you are doing at all time. The Editor also has commands to move or copy single lines or blocks of text from one place to another. Some of the other commands include Tape and Disk load, save and append; Directory (disk). Automatic line numbers, delete line, set input line length and printer output. Files larger then memory can be edited and processed directly from the disk. One other nice little feature is during line entry an audible "beep" is output seven characters before the end of line, at the end of line and if an illegal character is entered. This alone can increase your input speed by 100%.

The Text Processor includes 29 commands for formatting the output, some of them include: page length, left margin, top and bottom margin, line length, justify and fill modes, page heading, center line, double width print, margin control, single, multiple and special indent modes, footnotes, test lines left on page, display and input from keyboard and even special control codes can be sent to the printer for different print densities etc. It even has a repeat command with a next command to redo all of or a portion of the files as many times as needed. TEXTPRO will turn your Color Computer into a full fledged text processing machine at a price you won't believe. Available on R.S. Color Computer Disk.
PRICE $49.95
CER-COMP
5566 Ricochet LAve.
Las Vegas, NV 89110
(702) 452-0632

COMPUTERWARE
Microcomputer Sales and Software

Computerware introduces MAZERACE for the Radio Shack Color Computer.

MAZERACE is a fun board type game that involves both chance and strategy. The playing field is an eighteen by eighteen hexagon matrix that is partially filled with obstacles. Either one person against the computer or two people can play, with the computer or the players randomly scrambling the playing field to keep the action exciting. MAZERACE uses high resolution graphics and requires joysticks, Extended BASIC, and is a lot of fun!

MAZERACE costs only $17.95 on cassette or $22.95 on disk (plus $2.00 for shipping and handling) and is available directly from Computerware at 1472 Encinitas Blvd., Box 668, Encinitas, CA. 92024, (714) 436-3512.

COMPUTERWARE
Microcomputer Sales and Software

Computerware introduces its 3D Drawing Board for the Radio Shack Color Computer.

3D Drawing Board is a tool for education, entertainment, or serious projects. It helps you draw objects in 3 dimensions and then rotate, change elevation, size, and distance. The drawings can be saved to tape or disk for future use. A complete instruction manual accompanies the program to make it easy to use.

3D Drawing Board costs only $24.95 on cassette or $29.95 on disk (plus $2.00 for shipping and handling) and is available directly from Computerware at 1472 Encinitas Blvd., Box 668, Encinitas, CA. 92024, (714) 436-3512.

SUGAR SOFTWARE
2153 Leah Lane
Reynoldsburg, OH 43068
(614) 861-0565

Silly Syntax is an ingenious and creative story

game with instant appeal to players of all ages. It is written for the TRS-80 Color Computer (16K Extended Basic). Using the Fairy Tales cassette, Silly Syntax can be a family game which provides hours of togetherness and fun as well as a stimulating educational experience. It can also provide unique adult entertainment when the X-Rated story games are played.

Silly Syntax is a menu-driven game. The players select a menu item from the list displayed: (1) Assign players, (2) Load story tape, (3) Play Silly Syntax, (4) Output story (screen or printer), (5) Modify story using the built-in text editor, (6) Create a new Silly Syntax story and (7) Store the story to tape.

The computer asks each player, in turn, to supply a word as required by the story which has been loaded into memory, i.e. noun, adjective, celebrity. When all prompts have been supplied, the story is complete and ready to be read. Even though the same story may be used over and over, the final results will always be different and guarenteed to provide chuckles, snickers and belly-laughs.

Silly Syntax may be ordered from Sugar Software, 2153 Leah Lane, Reynoldsburg, Ohio 43068. For $19.95 you will receive the game, 2 stories and a 12 page instruction booklet. Additional story tapes are available for $9.95 per tape of 10 stories (10% discount on an order of 3 or more story tapes). Topics currently available are Fairy Tales, Science Fiction and Adventure, Current Events, Sing-Along, X-Rated and Gothic Romance. Please add $1.00 per tape for postage and handling. Ohio residents please include 5.5% sales tax.

WANTED: COMPUTER LITERACY TEST SITE FOR COMPUTERTOWN, USA!

ComputerTown, USA!, a computer literacy project funded by the National Science Foundation, is accepting proposals from individuals and organizations interested in becoming an official ComputerTown, USA! test site.

The test site would offer public access to computers within specific community located somewhere in the western U.S., excluding California.

Site activities will focus on public access computer literacy in the ComputerTown spirit of "people helping people learn about computers".

The final deadline for proposals is May 1, 1982. Once selected in early May, the new ComputerTown test site will begin operations by June, 1982.

For deatils on how to submit a proposal, write to Site Selection Coordinator, ComputerTown, USA!, P.O. Box E, Menlo Park, CA. 94025.

BBS 'CONNECTION-80 WOODHAVEN', QUEENS, NY 1 YEAR OLD WITH OVER 15K CALLERS

March 23, 1982 marked one year of operation for Connection-80 of Woodhaven, NY, a BBS in the boro of Queens NYC. By its birthday the BBS had already logged over 15,000 callers. The Bulletin Board System which can be accessed at no charge whatsoever by simply dialing 212-441-3755 24 hours/day is dedicated to the color computer user. The BBS has over 2.6 megabytes of disk storage, download section, bulletin section, merchandise section, chat section and electronic mail. Everyone agrees of course that one of the best features is, that it's FREE. That's hard to beat nowadays!

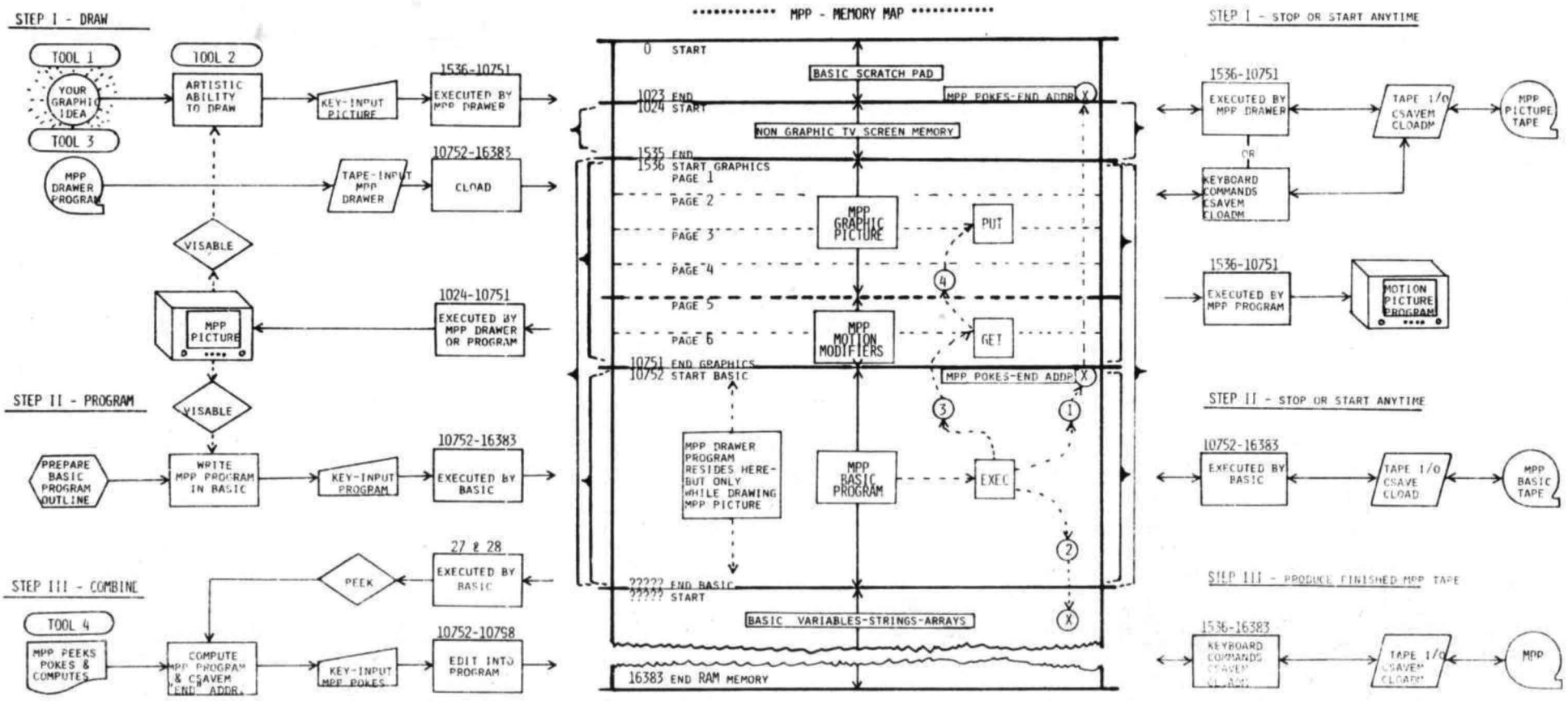
CONTACT: Bob Rosen, SYSOP via terminal at 212-441-3755 24hrs/day via voice at 212-441-2807.

Continued from page 17

SCHEMATIC DIAGRAM TO PRODUCE - MOTION PICTURE PROGRAMING

BY

SUPERIOR GRAPHIC SOFTWARE



NOTE: THE COMPLETE PROCESS IN THE SCHEMATIC DIAGRAM ABOVE IS AVAILABLE ON TAPE FROM SUPERIOR GRAPHIC SOFTWARE @ $ 34.95 AND INCLUDES THE FOLLOWING: SEVERAL MPP GRAPHIC IDEAS TO START WORKING WITH. CHROMASETTE'S CASSETTE (AUGUST 1981 ISSUE) WITH THEIR EXCELLENT DRAWER PROGRAM. MODIFICATIONS FOR THE MPP DRAWER. ALL NECESSARY INSTRUCTIONS. COMPLETER MPP PROGRAMS.

Color computer owners,

32K PLUS DISKS*

$298.00

Yes, that's right - for as little as $298.00 you can add 32K of dynamic RAM, and a disk interface, to your TRS-80 Color Computer! If you just want the extra memory it's only $199.00, and you can add the disk interface later for $99.00.

Just plug the *Color Computer Interface (CCI)*, from Exatron, into your expansion socket and "Hey Presto!" - an extra 32K of memory. No modifications are needed to your computer, so you don't void your Radio Shack warranty, and Exatron give both a 30 day money-back guarantee and full 1 year repair warranty on their interface.

The *CCI* also contains a 2K machine-language monitor, with which you can examine (and change) memory, set break-points, set memory to a constant and block-move memory.

So what about the *CCI Disk Card*? Well as we said it's only an extra $99.00, but you'll probably want Exatron's *CCDOS* which is only $29.95 – unless you want to write your own operating system. The *CCI Disk Card* uses normal TRS-80 Model I type disk drives, and *CCDOS* will even load Model I TRSDOS disks into your color computer – so you can adapt existing TRS-80 BASIC programs.

As a further plus, with the optional *ROM Backup* adaptor, you can dump game cartridges to cassette or disk. Once the ROM cartridge is on cassette, or disk, you can reload, examine and modify the software. The *ROM Backup* adaptor is only $19.95.

For more information, or to place an order, phone Exatron on their Hot Line *800-538 8559* (inside California 408-737 7111), or clip the coupon.

excellence in electronics

exatron

DEALER ENQUIRIES INVITED

Exatron,
181 Commercial Street,
Sunnyvale, CA 94086

☐ Please send a 32K Color Computer Interface for $199.00

☐ Please send a CCI Disk Card for $99.00

☐ Please include CCDOS and manual for $29.95

☐ Also include a ROM Backup adaptor for $19.95

Please add $5.00 for shipping to all orders, and 6 percent sales tax in California.

Name .....

Address .....

City .....

State ..... Zip .....

Charge my:

☐ MasterCard Interbank Code .....

☐ Visa Expiration Date .....

Card .....

☐ Check enclosed for .....

☐ Ship COD ($2.00 extra)

Signature .....

*32K RAM plus Disk Interface ... TRS-80 is a trademark of Tandy