```
3540 PRINT "{4 SPACES}HALF INCH ASPHA
     LT BOARD = 2.4"
3550 PRINT "{3 SPACES}1/2 IN GYPSUM O
     R PLASTER = 1.39"
3560 PRINT "{4 SPACES}1/4 IN WOOD FIB
     ER BOARD = 1.12"
3570 PRINT "{6 SPACES}FIR OR PINE SHE
     ATHING = 1.92"
3580 PRINT "{6 SPACES}3/4 IN PLYWOOD
     PANELS = 1.88"
3590 PRINT "{13 SPACES}1/2 IN PLYWOOD
     = 1.57"
3600 PRINT :PRINT
3610 RETURN
4000 REM CEILING ROUTINE
4010 I=4
4020 HI=0.61:HO=0.61:IF PK>1 THEN 406
     0
4030 PRINT "{CLEAR}{DOWN}WHAT IS TOTA
     L CEILING AREA"
4040 PRINT "OF THE HOUSE";
4050 INPUT TEMP:A(I)=TEMP
4060 PRINT "HOW MANY INCHES OF INSULA
     TION IN CEILING";
4070 INPUT CI
4080 PRINT "TYPE OF INSULATING MATERI
     AL"
4090 PRINT "{DOWN}  1. FIBERGLASS"
4100 PRINT "  2. MINERAL WOOL"
4110 PRINT "  3. VERMICULITE OR PERLI
     TE"
4120 PRINT "  4. CELLULOSE FIBER"
4130 PRINT "  5. U-F FOAM{DOWN}"
4140 INPUT T
4150 RM=CI*IC(T)
4160 R(I)=HO+RM+HI
4170 Q(I)=A(I)*DT/R(I)
4180 RETURN
5000 REM FLOOR ROUTINE
5010 I=5:IF PK>1 THEN 5040
5020 PRINT "{CLEAR}{DOWN}WHAT IS TOTA
     L FLOOR AREA";
5030 INPUT TEMP:A(I)=TEMP
5040 PRINT "HOW MANY ITEMS OF INSULAT
     ION IN FLOOR";
5050 INPUT FI:IF PK>1 THEN 5110
5060 PRINT "TYPE OF FOUNDATION"
5070 PRINT "{3 SPACES}1. OPEN CRAWLSP
     ACE"
5080 PRINT "{3 SPACES}2. ENCLOSED CRA
     WLSPACE OR BASEMENT"
5090 PRINT "{3 SPACES}3. CONCRETE SLA
     B"
5100 INPUT TF
5110 R(I)=HO+FI*3.1+RF(TF)+HI
5120 Q(I)=A(I)*(DT-TC(TF))/R(I)
5130 RETURN
5200 REM DUCTS
5210 DI=0.1
5220 IF TF=3 THEN KD=3:RETURN
5230 PRINT "{DOWN}IS YOUR DUCTWORK IN
     SULATED";
5240 INPUT D$:IF PK>1 THEN 5310
5250 PRINT "{DOWN}LOCATION OF HEAT DU
     CTS:"
5260 PRINT "{4 SPACES}1. ATTIC OR CRA
     WLSPACE"
5270 PRINT "{4 SPACES}2. UNCONDITIONE
     D BASEMENT"
5280 PRINT "{4 SPACES}3. IN SLAB FLOO
     R"
5290 PRINT "{4 SPACES}4. INSIDE CONDI
     TIONED SPACE"
5300 INPUT KD
5310 RETURN
6000 REM WRITE A REPORT
6010 PRINT "{CLEAR}","HEAT LOSS EVALU
     ATION"
6020 PRINT :PRINT :TQ=0
6030 PRINT " ITEM"," AREA"," R-VALUE"
     ,"HEAT LOSS"
6040 PRINT ," SQ.FT.",," BTU/HR":PRIN
     T
6050 FOR I=1 TO 5
6060 A(I)=INT(A(I)*100+0.5)/100
6070 R(I)=INT(R(I)*100+0.5)/100
6080 Q(I)=INT(Q(I)+0.5)
6090 PRINT N$(I*10-9,(I-1)*10+NL(I)),
     A(I),R(I),Q(I)
6100 TA=TA+A(I):TQ=TQ+Q(I)
6110 NEXT I
6120 REM PRINT INFILTRATION LOSS
6130 PRINT "INFILTRATION",,INT((IN(1)
     +IN(2))/2+0.5)
6140 TQ=TQ+(IN(1)+IN(2))/2
6150 REM CALCULATE DUCT LOSS
6160 X=TQ/(A(5)*CH*NT):J=3:K=3
6170 IF X<45 THEN K=2
6180 IF X<35 THEN K=1
6190 DI=0.15+0.05*(3-K)
6200 IF D$="N" AND KD<2 THEN 6240
6205 IF KD>2 THEN DI=0:GOTO 6240
6210 IF OT<15 THEN J=2
6220 IF OT<0 THEN J=1
6230 DI=DM(KD,J+K*4)
6240 PRINT "DUCT LOSS",,,INT(DI*TQ+0.
     5)
6250 TQ=TQ+TQ*DI
6260 PRINT ,"{8 R}",,"{8 R}":?
6270 PRINT " TOTAL  ",INT(TA),,INT(TQ
     )
6280 PRINT
6290 PRINT "DESIGN CONDITIONS:"
6300 PRINT "{3 SPACES}OUTSIDE DESIGN
     TEMP:";OT
6310 PRINT "{4 SPACES}INSIDE DESIGN T
     EMP:";IT
6320 PRINT "TEMPERATURE DIFFERENCE:";
     DT
6330 RETURN
7000 REM FIND SAVINGS USING DEGREE-DA
     YS
7010 DD=2961:DD$="ATLANTA GA"
7012 E1=INT(Q1*DD*24)
7014 E2=INT(Q2*DD*24)
7030 PRINT "{CLEAR}TYPE OF HEATING FU
     EL USED"
7040 PRINT "  1. ELECTRICITY"
7050 PRINT "  2. NATURAL GAS"
7060 PRINT "  3. FUEL OIL"
7070 INPUT FT:PC=0.55
7080 ON FT GOTO 7100,7200,7300
7090 GOTO 7030
7100 REM ELECTRICITY
7110 PRINT "IS HEATING UNIT A HEAT PU
     MP";
7120 INPUT HP$:ER=3413
7130 IF HP$<>"Y" THEN 7150
7140 PRINT "ENTER EER OF HEAT PUMP";:
     INPUT ER:ER=ER*1000
7150 PRINT "AVERAGE $ COST PER KWH";:
     INPUT CO:FU$="KWH"
7160 E1=INT(E1/ER+0.5)
7165 M1=E1*CO
7170 E2=INT(E2/ER+0.5)
7175 M2=E2*CO
7180 MS=M1-M2
```

```
7190 GOTO 7400
7200 REM NATURAL GAS
7210 PRINT "AVERAGE $ COST PER THERM
     OF NATURAL GAS";:INPUT CO
7220 E1=INT(E1/(103000*PC)+0.5)
7225 M1=E1*CO
7230 E2=INT(E2/(103000*PC)+0.5)
7235 M2=E2*CO
7240 MS=M1-M2
7250 FU$="THERMS":GOTO 7400
7300 REM FUEL OIL
7310 PRINT "AVERAGE $ COST PER GALLON
      OF FUEL OIL";:INPUT CO
7320 E1=INT(E1/(138000*PC)+0.5)
7325 M1=E1*CO
7330 E2=INT(E2/(138000*PC)+0.5)
7335 M2=E2*CO
7340 MS=M1-M2:FU$="GALLONS"
7400 REM GIVE RESULTS
7410 M1=INT(M1*100)/100
7420 M2=INT(M2*100)/100
7430 MS=INT(MS*100)/100:IF MS=0 THEN
     MS=1.0E-05
7440 PRINT "{DOWN}TOTAL $ COST OF YOU
     R IMPROVEMENTS";:INPUT CI
7450 PB=INT(CI/MS*1000)/1000
7460 REM REPORT SAVINGS AND PAYBACK
7470 PRINT "{CLEAR}","ANALYSIS OF IMP
     ROVEMENTS"
7480 PRINT :PRINT
7490 PRINT ,,"ENERGY NEEDED"
7500 PRINT "ORIGINAL HOUSE ",E1;" ";F
     U$
7510 PRINT "IMPROVED HOUSE ",E2;" ";F
     U$
7520 PRINT ,,"{9 R}":?
7530 PRINT ,"SAVINGS",E1-E2;" ";FU$
7540 PRINT
7550 PRINT ,,"OPER. COSTS"
7560 PRINT "ORIGINAL HOUSE","$";M1
7570 PRINT "IMPROVED HOUSE","$";M2
7580 PRINT ,,"{9 R}":?
7590 PRINT ,"SAVINGS","$";MS
7600 PRINT :PRINT ,"PAYBACK ",PB;" YE
     ARS"
7610 PRINT :PRINT
7620 PRINT "ABOVE IS BASED ON ONE YEA
     R OF OPERATION"
7630 PRINT "IN ";DD$
7640 RETURN
8000 REM DRAW HOUSE
8010 PRINT :PRINT :PRINT :PRINT
8020 POKE 85,8:? ;"{5 SPACES}{I}{O}"
8030 POKE 85,8:? "{H}{10 SPACES}{J}"
8040 POKE 85,8:? "{V}{I}{O}{I}{O}{I}
     {O}{I}{O}{I}{O}{B}"
8050 POKE 85,8:? "{V}{K}{L}{K}{L}{K}
     {L}{K}{L}{K}{L}{B}{5 SPACES}{J}"
8060 POKE 85,8:? "{V}{I}{O}{I}{O}{I}
     {O}{I}{O}{I}{O}{B}{I}{4 U}{B}"
8070 POKE 85,8:? "{V}{K}{L}{K}{L}{Y}
     {Y}{K}{L}{K}{L}{B}{Y}{4 SPACES}
     {B}"
8080 POKE 85,7:? "{21 M}"
8090 RETURN
```
©

To receive additional information from advertisers in this issue, use the handy reader service cards in the back of the magazine.

Warehouse Automation With Personal Computers

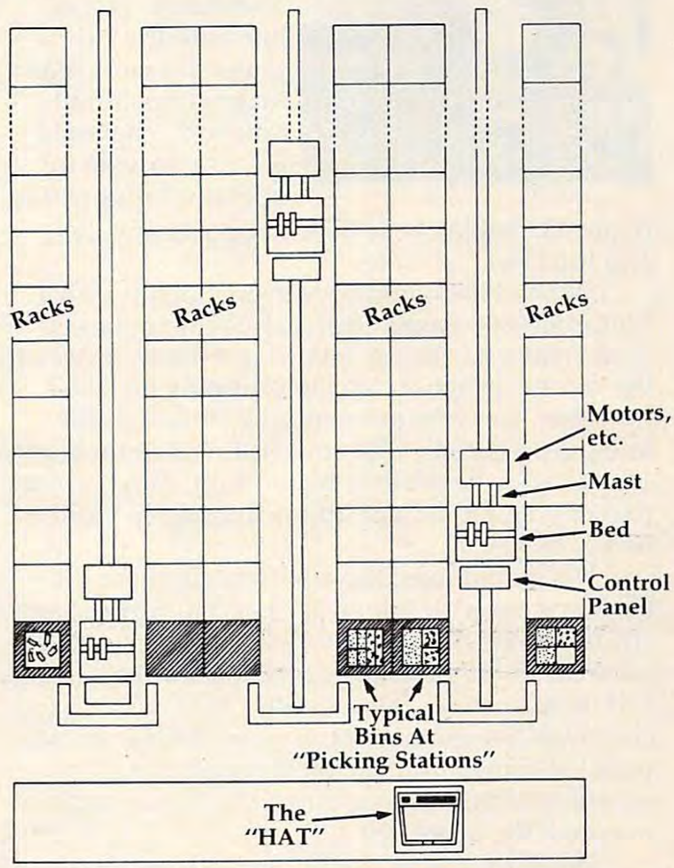
Timothy Stryker, Pompano Beach, FL

While this is not a home application – it does demonstrate the capabilities and sophistication of today's personal computers. This is the story of how a Commodore 8032 runs a large warehouse.

Maybe you use your personal computer for balancing your checkbook, or maybe you use it mainly for playing games. Maybe you even use it to control your furnace or air conditioning system. But did you ever use a personal computer to control a sophisticated industrial automated warehousing system? We did, and it worked out great.

Today's personal computers are sometimes belittled by professional engineers who feel that any computer suitable for home use could not possibly be appropriate for use in an industrial environment. In many cases, they are right: some personal computers are not designed, mechanically, for a great deal of wear and tear, and others have numerous games-related features that would prove less than valuable in an industrial setting. Some personal machines, however, are reliable enough, both mechanically and electrically, not ony to survive, but also to excel when used in industry. The Commodore CBM 8032 is one such machine.

Figure 1: The Warehouse Layout



The Project

Our customer originally started out with an ancient (vintage 1974) automated warehousing system based largely on hard-wired controls. The system consisted of three automatic "cranes" running along tracks in the floors of three separate "aisles," one crane per aisle (see Figure 1).

Each crane could move horizontally up and down its aisle, and had a "bed" that could be moved vertically up and down a 20 foot "mast" to reach packages at different heights (see Figure 2). Packages were arrayed in racks on both the left and the right face of each aisle, and each bed incorporated a "shuttle" which could move left and right, energizing an electromagnet when necessary to pull a package on board.

Each of the three cranes communicated, over a 300-baud asynchronous link, with a single master controller, which sent out commands to the cranes to retrieve various packages, replace others, and so forth. The packages being stored and retrieved contained bins of loose parts. The idea was to fill orders for these loose parts by bringing their respective bins to the front of the aisles, where human "parts pickers" would extract the appropriate parts and then send the bins back into the racks until the next time they were needed.

Our mission was to replace virtually all of the electronic portions of this system with modern, programmable equipment. This involved replacing each of the hard-wired control systems in the cranes with microprocessor-based hardware, as well as coming up with a new master controller, a so-called "Head-of-Aisle Terminal" (HAT).

The HAT would have to communicate with

Figure 2:
A crane, as seen from the front of an aisle, with its bed partway up its mast

each of the three cranes, and would in addition have to communicate over a 4800-baud "bisync" line with the customer's remote IBM mainframe. Simultaneously, it would have to provide for operator input of bin requests, display of system status, diagnostics, and the like.

We had had good results previously with a Motorola 6800-based approach to the replacement of the crane hardware, so this is what we used for the cranes. When it came to replacing the HAT, however, we were momentarily stumped. We knew we wanted a CRT console for operator input, and it had to be reliable. In addition, the computer part of it had to be easy for us to program, and it had to be fast.

We would need the ability to augment the basic system with things like special-purpose synchronous (bisync) communications hardware. We considered the possibility of using an M6800 in an EXOR-bus configuration, with a high-speed asynchronous line going out to a semi-intelligent terminal of some kind, but we realized that this would tend both to bog down the display and to overload the processor.

What we eventually settled on was, of course, the CBM 8032. This fine personal computer combines a 6502 processor, 32K of RAM, a couple of spare 2532-compatible ROM sockets, and a complete memory-mapped video driver circuit, including the CRT tube itself, in a single enclosure.

One major advantage of the 8032, from my standpoint, was that I could use it to write most of the software needed in RPL, a language of my own design that I had specifically optimized for the development of high-speed, memory-critical applications like this one. You may have seen Robert Baker's review of RPL in the February 1982 issue of *Microcomputing;* everything he says in that review is the gospel truth.

By using RPL, I could see that I would have the luxury of writing virtually all of the code in high-level terms, resorting to assembly language only for extraordinarily time-critical functions such as interrupt servicing, block POKEs to the screen, and the like. In addition, the use of the RPL Symbolic Debugger would allow me to test and debug the software in record time, which, since I was working on a fixed-price contract basis, was important.

So, we cut our costs significantly by making use of the machine we planned to sell as its own development system. The fact that the video was memory-mapped meant that updating of the display could proceed at processor speed, yielding instant, random screen updating without any interrupt overhead. The fact that the computer and its display were integrated in a single cabinet lessened the potential mechanical problems of upset and breakage.

Also, the 8032's "memory expansion bus" allowed us to augment the machine's built-in capabilities with boards and other components of our own design, physically located in a separate enclosure. Figure 3 shows the external appearance of the resulting HAT: sharp and professional looking enough for the most discriminating of tastes.

Trials And Tribulations

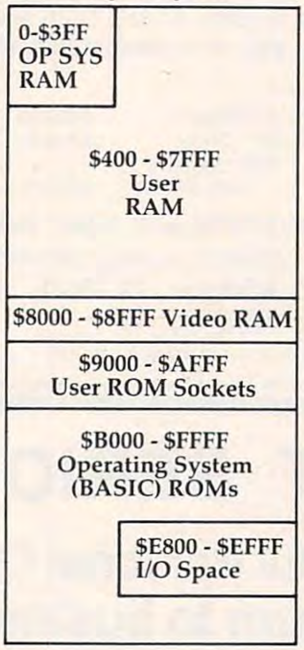
No project of this magnitude, of course, is without its share of problems. The first was that we would need at least 8K of ROM space, thus consuming the two spare 4K ROM sockets, for our control software. Not only did this consume the available

Figure 3: **The HAT**

sockets, but, more importantly, it also consumed (so we thought) the only remaining, non-dedicated portion of the 8032's address space. As shown in Figure 4, Commodore supplies the 8032 with 32K of user RAM occupying addresses 0 through $7FFF, the memory-mapped video display RAM from $8000 to $8FFF, and the main operating system and BASIC interpreter ROMs from $B000 through $FFFF.

Now, there is actually a little gap, running from $E800 to $EFFF, which the 8032 decodes as I/O space rather than ROM. It is in this range that Commodore has placed its PIAs ("Peripheral Interface Adapters", i.e., parallel I/O ports) and VIAs ("Versatile Interface Adapters", another flavor of same) for communicating with the outside world via keyboard, tapes, the IEEE bus, etc.

Figure 4: 8032 Memory Map



What we did not realize at first is that not all of the address space up there is consumed by Commodore's built-in devices: in particular, the range from $EA00 up would appear to be available for user use. By the time we perceived this, though, we had already committed to a bank-switched approach involving the $C000 ROM (*why* it had to be the $C000 ROM is too complicated to go into here).

In so doing, however, we ran afoul of another little peculiarity of the 8032's design, which you may need to know about if you attempt anything like this yourself. This applies whether you use the $EA00 area for I/O or not. What it is, is that the 74LS244's (8-bit tri-state bus drivers) that Commodore uses to bidirectionally buffer the memory expansion data bus are hard-wired to point in the CPU-write direction for all memory accesses to addresses $A000 and above (for addresses in the $9000 to $9FFF range, interestingly enough, this is jumper-selectable).

This leads to the disconcerting problem that, whenever you try to *read* from anything above $A000 on the memory expansion bus, the data comes in as garbage because the bidirectional bus drivers are stupidly driving the wrong way. Our solution to this was to jumper the direction-control inputs of the 74LS244's to a signal we generated off-board which "knew" when a read to the expansion bus was occurring and inverted the drive direction appropriately.

Another little quirk worth noting (forgive me if I'm being overly technical here, but these little insights will be worth about $2000 apiece to you if you ever get involved in this sort of thing) is that the R/W inputs to the ROMs in the 8032 are hard-wired to V_{cc}, implying that you cannot expect a *write* to a ROM address to disable the ROM, or for that matter to do anything reasonable at all.

Why would you want to write to ROM, you may ask? Well, suppose you wanted to substitute a 2532-compatible RAM chip for one of the ROMs, for example? It won't work unless you jumper the socket's R/W line to the CPU's R/W line (and, of course, cut the V_{cc} trace while you're at it). Remembering this can save you days of frenzy and heartburn, when the time comes.

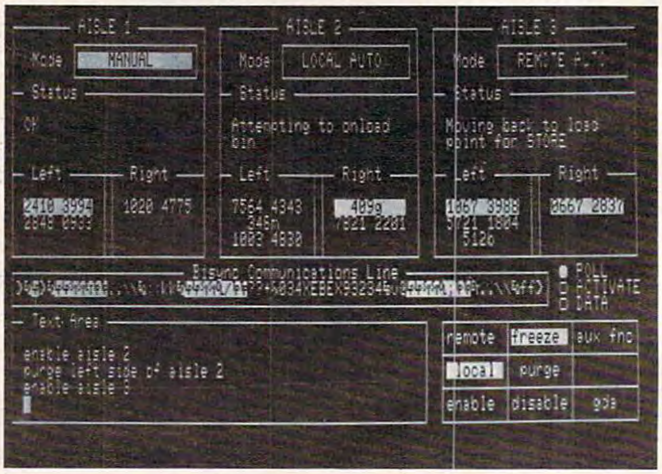
Once these problems were out of the way, the rest of the project went fairly smoothly. Figure 5 shows what the display looked like during actual operation. If you are a Commodore buff, you will notice that some of the characters in the "Bisync Communications Line" box are not part of the standard Commodore character set – this was achieved by substituting a custom 2532 EPROM for the standard character generator chip in the 8032.

This is remarkably easy to do if your character generator chip is socketed (some of the newer ones, unfortunately, are not): you just pop out the existing chip, copy it, with whatever changes you like, into a 2532 (or, if you like, a 2716), and pop the 2532 into the socket. We used a model 2704 PROM programmer/editor from Micro-Link, Inc., of Carmel, Indiana, and found it easy as pie and very satisfying at that.

Software Design

As you can see in Figure 5, our HAT has to keep track of a fair number of things at once. In order to make the system easy for operators to control, I used the "soft-key" technique for input of com-

Figure 5: A Closeup of the HAT's Screen



mands. This unsung hero of a technique, used by Hewlett-Packard in much of their equipment, will, I predict, become the standard command-input technique of the future. The idea behind it is to combine the best features of menu-driven operation with the best features of random-command-driven operation by providing a set of "soft-keys," which, in effect, present the operator with a menu of the currently allowable command options at all times.

Normally, a keyboard must be designed with soft-key operation in mind in order to be so used, but one can sometimes improvise. In the case shown here, for example, the soft-keys are the keys 1 through 9 in the 8032's numeric keypad. The little block of legends in the lower right-hand corner of the display can be thought of as appearing directly on the keys themselves, and the only difference between these legends and normal, "hard" legends is that these legends *change* once you have hit a particular soft-key.

For example, if you hit the "enable" soft-key on the HAT, the word "enable" is echoed onto the bottom line of the "Text Area" shown to the left, and the soft-key legends change to offer you the option of enabling aisle one, aisle two, or aisle three. When you select the aisle you want enabled, your choice is echoed to the Text Area, and the soft-key legends change again to tell you that the only thing you can now do is to hit RETURN (or CLEAR, in case you have changed your mind).

Once you hit RETURN, the command is executed, and the Text Area display scrolls up, retaining a record for you of what you did, in the king's English. Very little possibility for confusion here, especially when you consider that this arrangement makes it *physically impossible* for you to enter a command with invalid syntax!

A fair amount of "human-engineering" (that always sounds to me like android design) also went into the rest of the display shown in Figure 5. The "Bisync Communications Line" box acts as a continuous window onto the line connecting the HAT with its remote IBM mainframe, so that communications problems can be easily diagnosed and corrected. Bytes received are displayed here in normal field (green on black), while bytes transmitted are displayed in reverse field.

Since the HAT is on a multi-drop, shared communications line, it is "polled" in various ways by a network controller. These polling sequences cause the little circles beside the legends POLL, ACTIVATE, and DATA to light up from time to time. This ability to easily simulate LEDs, discrete scrolling windows, and so forth – as though designing a real, mechanical front panel – is one of the big advantages of a direct memory-mapped video display. I made the most of it.

The upper part of the HAT display is devoted to system status, which is logically grouped by aisle. A queue of pending bin requests is maintained for each side of each aisle, and bins currently active in each case are highlighted through the use of reverse field. Aisle mode and status are displayed in English, with highlighting where appropriate.

These fields are all continuously updated in realtime on the screen – the memory-mapped video of the 8032 makes it possible to maintain a high data rate to the tube with very little processor overhead, while the graphics and reverse field capabilities of the machine allow this information to be organized and presented in a clear and aesthetically pleasing way. The fact that the 8032's display is a full 80 columns wide was certainly a boon to us too, as you can see.

No static photograph can really give you an adequate idea of what the HAT screen looks like when the system is in operation. The bytes in the bisync window are constantly whizzing around, while the status fields for each of the aisles are continually changing as the cranes go through their paces. Meanwhile, the bin queues are constantly filling up and emptying out, allowing you to watch as each bin request gradually moves up in sequence until it becomes current.

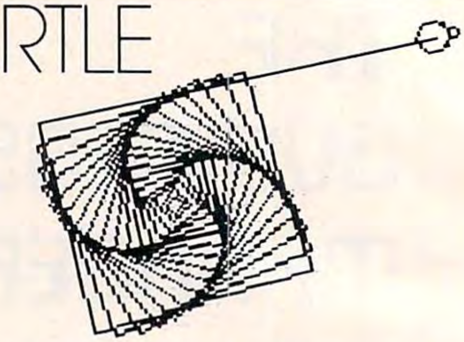
The little POLL, ACTIVATE, and DATA lights blink on and off like mad. And, in spite of all this activity, the operator can enter commands to add new queue entries, purge old ones, enable and disable aisles, and so forth whenever he or she likes, without regard to what else the system might be doing at the time. Operating the HAT actually has a lot in common with playing a realtime video arcade game. It's (dare I say it?) actually fun.

Everybody's Happy

There is a good deal more to the HAT, and to the system as a whole, than I have been able to address here, but I hope that this has given you some feel for what a personal computer is really capable of when pressed to its limits. The HAT and its M6800-controlled cranes have been in operation for several months now and, so far, have run virtually trouble-free. Our customer reports system throughput on the order of double what it was before, and his operators are happy because their jobs are now easier and less confusing.

All of this makes my boss happy that we went with the CBM 8032, and that makes me happy because I'm the one that talked him into it. I'm also happy that I could use this project to demonstrate the viability of RPL in a highly demanding and cost-sensitive application. The next time you're designing a process control system for serious industrial use, think twice about the possibility of rounding it out with a "personal" computer. ©

FRIENDS OF THE TURTLE



David D Thornburg, Associate Editor

Turtle Graphics For The VIC

Judging from the amount of mail I have received on this topic, there are thousands of VIC owners who are waiting for the chance to see turtle graphics on their computer. It was thus with excitement that I viewed the chance to try the Turtle Graphics program cartridge from HES.

This cartridge comes nicely packaged with a thorough manual that contains both a tutorial and a reference section. The cartridge contains an 8K byte program that loaded immediately when the VIC was turned on. Since the program worked on my vintage VIC from Japan, I'm sure it works on every VIC ever made.

I have used this program for a month and am quite ambivalent about it. As someone who uses turtle graphics packages of all sizes and vintages for a host of computers, I confess to having a predefined set of expectations. In order to help you assess my review, it is only fair that I list what I feel are the important aspects of WSFN, Atari PILOT, Apple SuperPILOT, Logo, and the Big Track toy with a felt-tip pen attached to the back:

1. Turtle graphics is just that – a graphics environment. It is capable of drawing continuous lines on a screen (or a sheet of paper).

2. Turtle graphics is richly endowed with commands that allow the incremental movement along a heading and the incremental rotation of the heading by amounts whose values can be stored in variables.

3. The highly interactive and experimental nature of those who use turtle graphics most effectively requires that graphics commands and user-defined procedures be capable of execution directly from the keyboard. A true immediate mode is present in all my favorite turtle systems – including the $40 Big Trak.

Unfortunately, the Turtle Graphics package from HES fails all these tests – and I could have made the list longer with the same result. My biggest complaint is that this program does not use the VIC graphics mode at all, but builds pictures by printing trails of characters on the 22 column by 23 row display screen. This restricts one to very primitive pictures.

This does not mean that the program isn't useful. I feel that, under a new name, this program can find tremendous application in another field, but more on that later.

Structurally, the program is quite nice. It contains its own line editor that performs some error checking before accepting each line. The language itself uses English language commands that, in most cases, are instantly understood by the user. For example, TURTLE COLOR RED changes the turtle's color to red.

When the system is turned on, the user is presented with a menu that allows the addition, insertion, deletion and replacement of program lines, the listing and printing of programs, and the execution and tracing of programs. All aspects of this menu driven system work well.

To get a feel for the language's syntax, let's examine a simple program from the manual:

```
SCREEN COLOR YELLOW
BORDER COLOR PURPLE
TURTLE COLOR BLUE
PEN DOWN
TEXT HI THERE
PEN UP
CHARACTER TO +
TURTLE COLOR GREEN
MOVE TO 6-3
PEN DOWN
RIGHT 5
DOWN 5
LEFT 5
UP 5
STOP
```

The first three commands set colors. Any of the standard VIC colors can be used. The turtle starts with the pen up (unlike all the other turtle systems with which I am familiar), so if you want to see something, you must remember to put the pen down first. The TEXT command functions somewhat like the PILOT T: command in that it prints whatever follows the command. It does not, however, allow you to print out the contents of variables, so it can't be used to print out the results of calculations. Also, unfortunately, there is no INPUT command to allow data to be entered interactively during the execution of a program.

Apple* ★ Atari* ★ Commodore* Franklin* ★ IBM* ★ TRS-80*

Krell's College Board

SAT*

Preparation Series

New for '83

A COMPREHENSIVE PREPARATION PACKAGE / MORE THAN 40 PROGRAMS

1. Diagnostic analysis
2. Prescription of individual study plans
3. Coverage of all SAT* skills
4. Unlimited drill and practice
5. SAT* Exam Question simulator
6. All questions in SAT* format and at SAT* difficulty level
7. Instantaneous answers, explanations and scoring for problems
8. Worksheet generation and performance monitoring - (optional)
9. A complete record management system- (optional)
10. Systematic instruction in pertinent math, verbal & test taking skills - (optional)

Krell's unique logical design provides personalized instruction for each student according to individual needs.

Krell's College Board SAT* 81/82 Prep. Series has demonstrated a mean combined math and verbal score increase of more than 100 points by using sophisticated drill & practice techniques alone. This new series is much expanded and updated. In addition extra cost options are now available to provide worksheet generation, record management, and systematic instruction in all math and verbal areas tested by the SAT*.

All versions of Krell College Board SAT* Preparation Series provide answers, explanations and instantaneous raw scoring. Standard package $299.95

APPLE, ATARI, COMMODORE, CP/M, IBM, AND RADIO SHACK*

SWORD OF ZEDEK

Fight to overthrow Ra, The Master of Evil. Treachery. deceit, and witchcraft must be faced in your struggles as you encounter wolves, dwarves, elves, dragons, etc. Each of the twelve treasures will enhance your power by making you invisible, invulnerable, etc. Each game is unique in this spectacular and complex world of fantasy. $24.95

LOGO

©M.I.T. 1981

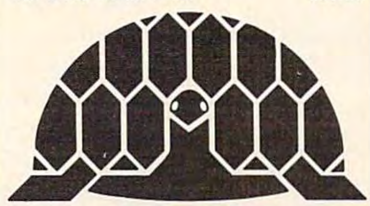
$99

NO FRILLS for Apple II®

NO FRILLS PAK

1. Two copies of Krells LOGO for Apple II*
2. Utility Disk with M.I.T.'s valuable demo programs including Dynatrack
3. The official M.I.T technical manual LOGO FOR APPLE II by H. Abelson & L. Klotz

No Frills Turtle Price $99.00



LOGO FRILLS ONLY

FRILLS FOR LOGO / Support Pak for: M.I.T. LOGO, KRELL & TERRAPIN INC.

1. Krell utility disk
2. Alice in LOGOLAND
3. LOGO for Apple II - by H. Abelson
4. Alice in Logoland Primer
5. Comprehensive wall chart
6. LOGO & Educational Computing Journal

$89.95

NO FRILLS LOGO AND ALL THE FRILLS COMBO FOR APPLE II* $149.95

SPRITES NOW AVAILABLE FOR LOGO Call for information

Prices slightly higher outside U.S.

TIME TRAVELER

Using the Time Machine, players must face a challenging series of environments that include: The Athens of Pericles, Imperial Rome, Nebuchadnezzar's Babylon, Ikhnaton's Egypt, Jerusalem at the time of the crucifixion, The Crusades, Machiavelli's Italy, the French Revolution, the American Revolution, and the English Civil War. Deal with Hitler's Third Reich, Vikings, etc. Players may set level of difficulty. Each game is unique. $24.95

Shelby Lyman Chess*

Shelby's Socrates Chess Tutorial Series uses the latest AI techniques to customize each lesson for you. Instruction modules cover every aspect of the game. For all micro's. Call or write for details.

Amazing Ben The Royal Road to Artificial Intelligence.

$79.95

Connections

A complete game system. Learn the principles of scientific reasoning in your choice of game formats. Expand minds at all ages. Subject areas include: Geography, Biology, Everyday Objects, etc.

Game System $99.95

Data Base, $24.95 per subject, 3 for $50

Isaac Newton + F.G. Newton

Perhaps the most fascinating and valuable educational game ever devised - ISAAC NEWTON challenges the players to assemble evidence and discern the underlying "Laws of Nature". FULL GRAPHICS NEWTON - presents all data in graphic form. This game is suitable for children. Players may select difficulty levels challenging to the most skilled adults. Both Games $49.95

Pythagoras and the Dragon

Mathematics in a fantasy game context. Based on THE SWORD OF ZEDEK introduces Pythagoras as a mentor to the player when called on for aid, Pythagoras poses math questions and depending on the speed and accuracy of response, confers secret information. With Pythagoras as an ally, the quest to overthrow Ra, The Master of Evil, assumes a new dimension of complexity. Depending on the level chosen problems range from arithmetic through plane geometry. APPLE, TRS-80, PET, ATARI* 32K $39.95

KRELL® SOFTWARE CORP.

The state of the art in educational computing

1320 Stony Brook Road / Stony Brook NY 11790
Telephone 516-751-5139

Krell Software Corp. has no official ties with the College Entrance Examination Board or the Educational Testing Service. Krell is, however, a supplier of products to the E.T.S.

*Trademarks of Apple Comp. Corp., Tandy Corp., Commodore Corp., Digital Research Corp., I.B.M., Atari Corp., Terrapin Inc., Socrates Chess Corp.

N.Y.S. residents add sales tax

The command CHARACTER TO + tells the turtle to leave a trail of + marks as it moves. It is better to think of the turtle as carrying a rubber stamp rather than a pen. As it moves, it stamps images of characters on the screen. The command MOVE TO 6-3 moves the turtle to the sixth column and third row of the screen. The commands RIGHT, DOWN, LEFT, UP move the turtle the indicated number of spaces in the indicated direction. Note that RIGHT and LEFT are *not* turn commands. They shift the turtle to the right or left. Movement is allowed to take place only along columns or rows, and there is no command that lets the turtle move incrementally along a diagonal.

The user can create labeled procedures with this system and can then invoke the procedures with the USE command. Among other valuable features, the user can have the program check to see if a particular symbol is underneath the turtle. This allows the creation of simple maze-following programs, and is quite valuable.

The conditional branching command, IF, is quite non-standard in its use. Consider this example from the manual:

```
IF (X>2)
SCREEN COLOR BLUE
JUMP DONE
LABEL FALSE
SCREEN COLOR RED
LABEL DONE
```

If the value of X is greater than 2, then the commands immediately following the IF command will be executed. If it is false, execution branches to the next label. I would guess that the reason this was done was to make sure that each program line did only one thing. The IF command then starts to look like the Logo TEST command. I would have been happier if HES had used TEST, IFT and IFF, as does TI Logo. The present construction is quite convoluted and cumbersome, in my opinion.

While no fault of HES, the aspect ratio of the display screen makes any accurate correspondence between a procedure and a drawn figure hard to detect. Suppose, for example, that a child draws a square on a sheet of paper and then translates this square to the procedure:

```
LABEL SQUARE
RIGHT 10
DOWN 10
LEFT 10
UP 10
ROUTINE END
```

When this procedure is executed, a wide rectangle will be drawn on the screen. The procedure *should* give a square, but the aspect ratio of the VIC's character screen will never let you plot a square with this procedure. Does this adversely affect the child's understanding of programming? I think so. This hunch has been reinforced by my limited testing of this package with an eight-year-old boy who certainly knew what a square looked like. This might seem like a minor point to some, unless they are expecting to use this program in an educational environment.

On the positive side, I think that Turtle Graphics is an excellent program for the creation of animated titles and text displays. The user can create musical sounds and can adjust the rate at which characters are placed on the screen. Any VIC owner who makes home video tapes or who needs an inexpensive, eye-catching attraction for a retail display can benefit from this package. In fact, the cost of the VIC plus the $39.95 Turtle Graphics program is far less expensive than the titling systems presently being sold to VCR owners. The VIC's video output connects easily to all VCR's, making this a natural application for the system.

My recommendation is for HES to market this program to the massive number of home video users who can use it to title their recordings. HES should use the proceeds from these sales to develop a *true* turtle graphics package for the VIC.

It is sorely needed.

[*See manufacturer's statement, p. 112.*]

Microworlds For Atari PILOT

I recently heard from Martin Suey, an elementary school teacher in Tulare, California who has been making good use of Atari's turtle graphics in his second grade classroom. After reading about the creation of microworlds – user-controlled environments which one can change at will – he decided to see if he could implement such an environment with Atari PILOT. His program, *Day and Night*, is designed for primary-aged children. The computer displays a scene showing a house with a movable pet (dog or cat) that can be made to walk in front of or behind the house. Pressing the button on the joystick changes the scene slowly from day to night (or from night to day).

Of technical interest to those of you who use this language, Martin's program uses player graphics, priority data registers, and color registers. The program is controlled with a joystick in port #1. Holding the joystick to the left or right moves the player in the indicated direction (with wraparound).

Pushing the joystick up moves the player "behind" the playfield image (behind the house, for example), and pulling the joystick down moves the player to the front of the image. Pressing the button causes the scene to change from day to

Cardco, Inc. announces five All-American ways to . . .

Expand your VIC®
at affordable prices

card?
A

A universal centronics parallel printer interface for the VIC-20 & C-64 computers. Obeys all standard VIC print commands.
Suggested Reatil — $79.95

cardette
1

A universal cassette interface for the VIC-20 & C-64 computers. Emulates all functions of the data cassette.
Suggested Retail — $29.95

cardboard
3

The CARDBOARD 3 is a fuse protected, economy expansion interface designed to allow the user to access more than one of the plug-in-type memory or utility cartridges lnow available. It will accept up to three cartridges at once.
Suggested Retail — $29.95

cardriter
1

A light pen for the VIC-20 and C-64 computers with a switch on the barrel and 6 good programs.
Suggested Retail — $29.95

cardboard
6

The CARDBOARD 6 is a fuse protected expansion interface designed to allow the user to access more than one of the plug-in-type memory or utility cartridges now available. Additionally it allows switch selection of games and other programs now available in the cartridge format, without the necessity of turning the computer off and on again, thereby saving a great deal of stress on your VIC-20 and on your television or monitor.
Suggested Retail — $139.95

All Cardco products are **Made in the U.S.A.** and are individually tested to ensure quality and reliability. Superior technological engineering optimizes the value/performance ratio of all of our products.

Specifications and prices subject to change.

Cardco, Inc. • 3135 Bayberry • Wichita, KS 67226 • (316) 685-9536
Compuserve® E-Mail Address: Cardco, Inc. • 73575, 1325

Dealer inquiries invited.

Manufacturer's Reply

In the interests of providing readers with a fair and balanced report on the features of a product, we asked the manufacturer of VIC Turtle Graphics to reply to David Thornburg's remarks. The following comments are from the author of VIC Turtle Graphics, David Malmberg, and from Human Engineered Software.

While I was developing VIC Turtle Graphics, I had a number of design objectives in mind. Specifically, I wanted TURTLE to be:

- Inexpensive to buy, yet offer a good value for the price,
- Easy to use, yet "bullet proof" and friendly enough for first-time programmers not to get frustrated when they make a mistake,
- Usable with a standard VIC with only 3.5K of available user memory and no disk drive,
- Capable of fully exploiting the VIC's great sound, color and graphics characters,
- An effective vehicle for teaching programming concepts to children and other first-time programmers,
- Fun and educational to use.

On the whole, I believe these objectives have been met or exceeded in the final TURTLE product. Specifically, at $39.95 for a cartridge-based system with a 72-page tutorial instruction manual, TURTLE is an economical and highly effective means of introducing programming concepts.

However, several of the above design objectives are clearly incompatible with Mr. Thornburg's "predefined set of expectations." The most significant incompatibility is obviously TURTLE's lack of high-resolution graphics. This omission was a conscious decision on my part. Hi-res graphics would have required that approximately 4K of the 3.5K available to the user in a standard VIC be set aside for a bit-mapped video display area. This would mean that it would take extra memory and a cartridge slot expander for it to work. None of the Atari or Apple Logo or PILOT systems that Mr. Thornburg cites as his standards of comparison had to deal with the VIC's limited memory – all have at least 16K of usable memory and several are 64K with a required system disk drive. As a result, such a comparison is quite unfair. Had the developers of these systems had only 3.5K to work with, they probably would not have opted for hi-res either.

Furthermore, it is not clear that hi-res is as important to the child who is being introduced to programming concepts as it is to Mr. Thornburg. Does the novice programmer learn more and/or have more fun if his turtle draws a line in hi-res than if it draws a "line" made of VIC graphic characters, such as red hearts or green diamonds? I think not. The acts of planning and debugging the drawing seem to me to be much more important to developing skills in the child than the aesthetics or resolution of the lines used.

Let me correct a few possible misunderstandings that might result from reading the review:

1. The VIC, like all Commodore computers, has an extensive graphic character set. When combined with the VIC's palette of eight colors, it is possible to create some dramatic displays. Only if you lack imagination would you be restricted to "very primitive pictures."

2. TURTLE does have turn commands; and the square in the example could also have been drawn with:

```
LOOP 4
FORWARD 5
TURN RIGHT (OR TURN LEFT)
LOOP END
```

3. Movement on the diagonal is possible by using the MOVE TO ROW – COLUMN (not COLUMN – ROW as explained by Mr. Thornburg). This command causes the Turtle to move to the specified location by the most direct path. If the PEN is DOWN, a trail of graphic characters will be left behind.

4. If someone is bothered by the VIC display screen's aspect ratio and its inability to draw perfectly square squares, I suggest they relabel the procedure as BOX, rather than SQUARE. The aspect ratio is certainly not enough of a problem to conclude that TURTLE has no value in an educational environment – as Mr. Thornburg implies.

Even though I disagree with the overall tenor and conclusions of the review, Mr. Thornburg did make several good observations. His identification of the need for INPUT and PRINT statements that can handle variables is quite valid. His suggestions on ways to improve the IF statement are good. These and other improvements are currently being incorporated into a version of Turtle Graphics for the Commodore 64.

TYPE-SHARE TYPESETTING
EDUCATIONAL SPECIALISTS
SILICON OFFICE SPECIALISTS

ALL UNDER ONE ROOF

SOUTHERN CALIFORNIA
MAINTENANCE CENTER
FOR COMMODORE EQUIPMENT

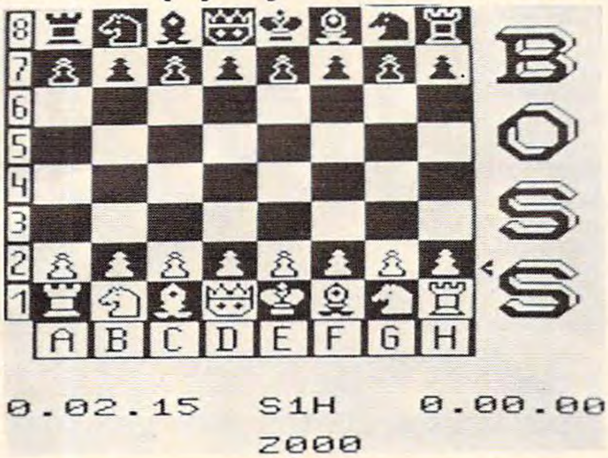
ONE STOP CENTER
for
commodore

VIC-VILLE ™ SOFTWARE

division of Data Equipment Supply Corporation

Exclusive distributors of Kavan Software

BOSS (c) by Kavan Software

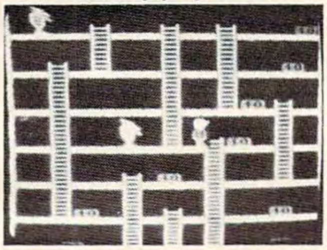


The Definitive Chess Game for the VIC-20

- ★ 10 Levels of Play
- ☆ Beats Sargon II
- ★ Two Clocks
- ☆ Wide range of opening moves
- ★ En passant, queening, castling
- ☆ Change screen and board colors
- ★ Cassette
- ☆ Requires 8K minimum expansion
- ★ 100% machine language

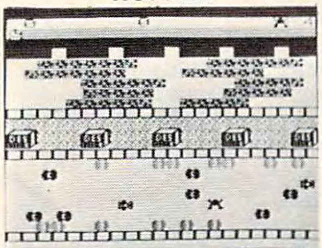
$39.95

BONZO (c) by Kavan



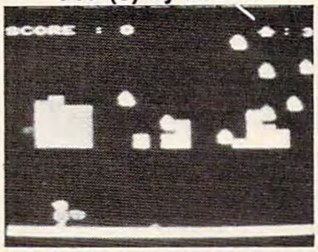
One of the most popular games in Europe. You control BONZO as he climbs the ladders and picks up the point blocks. Watch out for the alien guards. 100% machine language, cassette based. Joystick or keyboard, minimum 8k expansion. **$20.00**

HOPPER



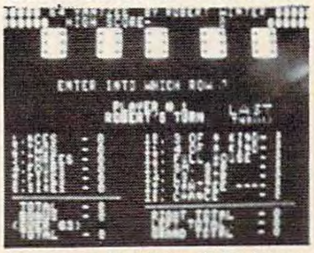
Avoid the cars, dragsters, buildings, logs and other obstacles to bring the frog safely home. Machine language for fast and smooth arcade action. Joystick, standard VIC. **$20.00**

PIT (c) by Kavan



BONZO strikes again as he takes money bags out of the pit. Avoid the alien rain by standing under the shields. Every successfully removed bag of money reinforces your sheilds. 100% machine language, cassette based. Joystick or keyboard, standard VIC. **$18.00**

Commodore 64 YAHTZEE



Commodore 64 version of the famous dice game. 10 player capacity. Watch dice roll across the screen. Automatic tabulation of score and bonuses. Sprite graphics and sound. Cassette based. **$20.00**

Night Crawler **$25.00**
by Interesting Software
Shoot down centipedes, spiders, mushrooms and all kinds of bugs before they get you. Machine language arcade action on standard VIC with joystick.

The Black Castle **$20.00**
Adventure, travel the countryside, fight demons, buy goods, storm the castle. Requires 3k or more expansion.

A Maze Ing **$12.00**
Travel through the maze. Game of skill and tense action. Standard VIC.

Gobbler **$11.00**
Sounds easy? You have 25 seconds to get him and the time gets shorter at each higher level. Standard VIC.

Hang U **$12.00**
Traditional Hangman plays against the VIC's 250 word dictionary or another person. Standard VIC.

Coggle **$11.00**
Computerized version of Boggle. Standard VIC.

Gold Brick **$14.00**
Many levels of play, sound, and color.

3-D Labyrinth **$14.00**
Escape from the labyrinth. Shown in 3-D perspective view with randomly generated mazes. Standard VIC.

Air Strike **$11.00**
Fly the new super bomber V-20 on a mission. Standard VIC.

Attack on Silo III **$12.00**
You are the commander of Silo III. Defend your country. Standard VIC.

Baseball Strategy **$12.00**
The excitement of baseball as a video strategic game. Standard Vic.

Vic Poker **$14.00**
Play poker against the VIC. Hi-res graphics and sound. Standard VIC.

Frogger by (c) Kavan **$14.00**
Eat the flies and avoid the car. Standard VIC.

Space Phreeks **$25.00**
by Interesting Software
Pilot the spaceship "Infinity" and fight the "Space Phreeks". 15 different attack patterns, 33 levels. Machine language, arcade quality. Standard VIC, joystick.

Mailing List **$25.00**
Keep mailing list, print reports, labels. 8k expansion or 16k expansion required.

Astro-Miners **$17.00**
Hi-res graphics and sound space game. Requires 3k or 8k expansion.

Panzer Attack **$14.00**
Enemy tanks are attacking and you must destroy them. Hi-res graphics. Standard VIC.

Pedestrian Polo **$14.00**
Drive your car thru the streets. Based upon Death Race. Standard VIC.

Yahtzee **$12.00**
Solitaire version of this famous dice game. Standard VIC.

Commodore 64 Software **Available Now !!**
64 Monopoly from AP Software
64 Mailing List from VIC-VILLE™ Software
64 Finance from VIC-VILLE™ Software
64 Time Manager 2.0 from TOTL Software

Look for more 64 Software from VIC-VILLE™ & get on our mailing list for all 64 updates and users' group.

Complete descriptive catalog $2.00

Dealers Welcome - Authors Wanted !

Add $3.00 for shipping & handling

Network your CBM, VIC and COMMODORE 64 with the PET SWITCH and VIC SWITCH from DATATRONICS.

Distributors for Datatronic AB

(714) 778-5455 | D E S | **Data Equipment Supply Corp. 8315 Firestone Blvd., Downey, CA 90241** | **(213) 923-9361**

night or vice versa. Michael is interested in hearing from those of you interested in these types of educational applications. I will gladly forward any comments to him.

```
10 GR:QUIT
20 R: WRITTEN SEPTEMBER 11,1982
30 C:@B1373=16
40 C:@B1374=2
50 WRITE:S;
60 WRITE:S;
70 WRITE:S;
80 WRITE:S;{4 SPACES}Day and Night
90 WRITE:S;
100 WRITE:S;{9 SPACES}by
110 WRITE:S;
120 WRITE:S;{5 SPACES}Martin Suey
130 *COUNT
140 C:#C=#C+1
150 J(#C=500):*CLEAR
160 J:*COUNT
170 *CLEAR
180 GR:CLEAR
190 *TURTLE
200 C:@B708=20
210 C:@B709=196
220 C:@710=16
230 C:#L=126
240 *LITE
250 C:#P=0
260 C:#L=#L+2
270 C:@B712=#L
280 J(#L=134):*DAY
290 *KNT
300 C:#P=#P+1
310 J(#P=100):*LITE
320 J:*KNT
330 *DAY
340 GR:PEN ERASE

350 GR:GOTO -74,10;5(DRAW 4;TURN 144)
360 GR:GOTO -50,30;5(DRAW 8;TURN 144)
370 GR:GOTO -30,15;5(DRAW 8;TURN 144)
380 GR:GOTO 0,40;5(DRAW 6;TURN 144)
390 GR:GOTO 50,25;5(DRAW 8;TURN 144)
400 GR:GOTO 74,42;5(DRAW 4;TURN 144)
410 GR:PEN RED
420 GR:GOTO 10,-20;TURNTO 0
430 GR:2(DRAW 30;TURN 90;DRAW 40;TURN
    90)
440 GR:PEN BLUE
450 GR:GO 30;TURN 30
460 GR:3(DRAW 40;TURN 120)
470 GR:FILL 39
480 GR:PEN YELLOW
490 GR:GOTO 26,-20;TURNTO 0
500 GR:2(DRAW 10;TURN 90;DRAW 10;TURN
    90)
510 GR:GOTO -79,-32;FILL 11

520 GR:GOTO 26,-20;FILL 10
530 GR:PEN RED
540 GR:GOTO 10,-20;FILL 30
550 GR:GOTO 36,-20;FILL 10
560 GR:GOTO -30,-20;TURNTO 0
570 GR:10(DRAW 5;TURN 90;GO 2;TURN 90
    ;DRAW 5;TURN -90;GO 2;TURN -90)
580 GR:GOTO 51,-20;TURNTO 0
590 GR:8(DRAW 5;TURN 90;GO 2;TURN 90;
    DRAW 5;TURN -90;GO 2;TURN -90)
600 C:@B559=62
610 C:#I=@B106-32
620 C:@B54279=#I
630 C:@B53277=3
640 C:@B704=208
650 C:@B53256=0
660 C:@B53248=125
670 C:#J=#I*256+1024
680 C:#Y=#J+160
690 C:@B#Y=3
700 C:#Y=#J+161
710 C:@B#Y=5
720 C:#Y=#J+162
730 C:@B#Y=6
740 C:#Y=#J+163
750 C:@B#Y=58
760 C:#Y=#J+164
770 C:@B#Y=100
780 C:#Y=#J+165
790 C:@B#Y=212
800 C:#Y=#J+166
810 C:@B#Y=76
820 C:#Y=#J+167
830 C:@B#Y=126
840 C:#Y=#J+168
850 C:@B#Y=202
860 C:#X=125
870 *MOVET
880 J(%T8=1):*CAT
890 J(%J0=2):*PLAYERT
900 J(%J0=1):*PFT
910 J(%J0=4):*LEFTT
920 J(%J0=8):*RIGHTT
930 J:*MOVET
940 *LEFTT
950 C:#X=#X-1
960 C:@B53248=#X
970 J:*MOVET
980 *RIGHTT
990 C:#X=#X+1
1000 C:@B53248=#X
1010 J:*MOVET
1020 *PLAYERT

1030 C:@B623=1
1040 J:*MOVET
1050 *PFT
1060 C:@B623=8
1070 J:*MOVET
1080 *CAT
1090 C:@B708=228
1100 C:@B709=192
1110 C:@B710=16
1120 C:#D=134
1130 *DARK
1140 C:#P=0
1150 C:#D=#D-2
1160 C:@B712=#D
1170 J(#D=128):*NITE
1180 *KOUNT
1190 C:#P=#P+1
1200 J(#P=100):*DARK
1210 J:*KOUNT
1220 *NITE
1230 C:#B=4
1240 *BLACK
1250 C:#P=0
1260 C:#B=#B-2
1270 C:@B712=#B
1280 J(#B=0):*SCENE
1290 *CNT
```

```
1300 C:#P=#P+1
1310 J(#P=100):*BLACK
1320 J:*CNT
1330 *SCENE
1340 GR:PEN RED
1350 GR:GOTO -74,10;5(DRAW 4;TURN 144
     )
1360 GR:GOTO -50,30;5(DRAW 8;TURN 144
     )
1370 GR:GOTO -30,15;5(DRAW 8;TURN 144
     )
1380 GR:GOTO 0,40;5(DRAW 6;TURN 144)
1390 GR:GOTO 50,25;5(DRAW 8;TURN 144)
1400 GR:GOTO 74,42;5(DRAW 4;TURN 144)
1410 GR:GOTO 10,-20;TURNTO 0
1420 GR:2(DRAW 30;TURN 90;DRAW 40;TUR
     N 90)
1430 GR:PEN ERASE
1440 GR:GOTO -30,-20;TURNTO 0
1450 GR:10(DRAW 5;TURN 90;GO 2;TURN 9
     0;DRAW 5;TURN -90;GO 2;TURN -90)
1460 GR:GOTO 51,-20;TURNTO 0
1470 GR:8(DRAW 5;TURN 90;GO 2;TURN 90
     ;DRAW 5;TURN -90;GO 2;TURN -90)
1480 GR:PEN BLUE
1490 GR:GO 30;TURN 30
1500 GR:3(DRAW 40;TURN 120)
1510 GR:FILL 39
1520 GR:PEN YELLOW
1530 GR:GOTO 26,-20;TURNTO 0
1540 GR:2(DRAW 10;TURN 90;DRAW 10;TUR
     N 90)
1550 GR:GOTO -79,-32;FILL 11
1560 GR:GOTO 26,-20;FILL 10
1570 GR:PEN RED
1580 GR:GOTO 10,-20;FILL 30
1590 GR:GOTO 36,-20;FILL 10
1600 C:@B559=62
1610 C:#I=@B106-32
1620 C:@B54279=#I
1630 C:@B53277=3
1640 C:@B704=6
1650 C:@B53256=0
1660 C:@B53248=125
1670 C:#J=#I*256+1024
1680 C:#Y=#J+160
1690 C:@B#Y=10
1700 C:#Y=#J+161
1710 C:@B#Y=14
1720 C:#Y=#J+162
1730 C:@B#Y=21
1740 C:#Y=#J+163
1750 C:@B#Y=219
1760 C:#Y=#J+164
1770 C:@B#Y=68
1780 C:#Y=#J+165
1790 C:@B#Y=124
1800 C:#Y=#J+166
1810 C:@B#Y=124
1820 C:#Y=#J+167
1830 C:@B#Y=68
1840 C:#Y=#J+169
1850 C:@B#Y=170
1860 C:#Y=#J+168
1870 C:@B#Y=202
1880 C:#X=125
1890 *MOVE
1900 J(%T8=1):*TURTLE
1910 J(%J0=2):*PLAYER
1920 J(%J0=1):*PF
1930 J(%J0=4):*LEFT
1940 J(%J0=8):*RIGHT
1950 J:*MOVE
1960 *LEFT
1970 C:#X=#X-1
1980 C:@B53248=#X
1990 J:*MOVE
2000 *RIGHT
2010 C:#X=#X+1
2020 C:@B53248=#X
2030 J:*MOVE
2040 *PLAYER
2050 C:@B623=1
2060 J:*MOVE
2070 *PF
2080 C:@B623=8
2090 J:*MOVE
```

©

A New, Improved Computer Friend For Your Apple

Fred D'Ignazio, Associate Editor

This column catches me in the midst of a move from Chapel Hill, North Carolina, to Roanoke, Virginia. Most of my dozen computers are still in Chapel Hill being looked after by a trusted babysitter. My roof leaks, my shower floods the bathroom floor, my study is buried in boxes, and Catie and Eric just came down with ear infections.

In addition to their sore ears, Catie and Eric are going through something you might call "computer-starvation shock." They think they are still a multi-computer family, and they brag about it to all their friends. They gather a horde of neighborhood kids with the promise of a dozen beeping, flashing computers. They climb the stairs, peek into my study, and what do they see? One lone computer. A rather sad-looking machine, vintage 1977. It doesn't talk, doesn't make pictures, doesn't play music. What a letdown!

Thanks, Chuck!

It's times like these when you readers come in handy. Thanks to one reader – Chuck Johnston of Manhattan Beach, California – I can still provide you with a useful column this month.

Chuck recently sent me a program he wrote that modifies my "Talking Head" program for the Apple. In my opinion, Chuck's program is a substantial improvement on the original version. It's exactly the kind of feedback I'd like to get from my readers. Thanks, Chuck!

Below is Chuck's letter and his program:

I am writing in regard to your column which appears in the September issue of **COMPUTE!** *Magazine. I found your article interesting, but the changes you suggest for the Apple II were, in my opinion, inadequate.*

The Apple is incapable of printing a reverse slash (as is this ancient typewriter), so the head shape you designed does not work. Also, you suggest deleting the sound subroutine, but it makes the program much more interesting. I have revised your program to run on the Apple and thought you might like to see it. I also failed to understand why you didn't draw the head using graphics; as you can see, the resulting animation is much more effective.

Included also is a sound driver program for the Apple in line 20, since, as we know, the Apple is only capable of rudimentary buzzes and clicks in Applesoft. It is POKEd into memory at $0300 and the POKEs in the sound subroutines are as follows: POKE 768,x (where x is a number between 1 and 255) sets the tone frequency. POKE 769,y (y also between 1 and 255) sets the tone duration. In the program enclosed I used the same values as the original program; whether it sounds the same is unlikely, but with some adjustment it could come close. Well, I hope you like the program and thank you for your time.

Chuck Johnston

```
10  REM *** POKE SOUND DRIVER INTO MEMORY
20  FOR I = 770 TO 795: READ M: POKE I,M: NEXT
```

Fred D'Ignazio is a computer enthusiast and author of several books on computers for young people. He is presently working on two major projects: he is writing a series of books on how to create graphics-and-sound adventure games. He is also working on a computer mystery-and-adventure series for young people.

As the father of two young children, Fred has become concerned with introducing the computer to children as a wonderful tool rather than as a forbidding electronic device. His column appears monthly in **COMPUTE!**.

THE LEARNING CENTER

"Perhaps the finest educational software that I have ever had the pleasure of reviewing. It's easy for kids to use and effective in teaching basic concepts and skills.... My kids are learning with it right now!"

Fred D'Ignazio, Associate Editor—*Compute!*, Associate Editor—*Softside*, Author of bestseller—*Katie and the Computer*

We hope you've been using your **ATARI** for more than just games ... it is, and can be, a valuable educational tool for you and your children.

Bruce and Dianne Mitchell realized this potential and designed a series of programs for use in their Small World Preschool & Kindergarten located in Durham, N.C. Presented on TV's *PM Magazine* these unique educational programs will introduce your 3-9 year olds to the era of learning with computers. Using the graphics and sound capabilities of the Atari, each program develops a particular skill and reinforces correct responses with happy faces and music.

Beginning with basic concepts such as colors, shapes and the alphabet, your child will progress to an understanding of counting, arithmetic, and language skills.

Widely acclaimed, classroom designed and tested, these unique educational tools are now available to you ... ***for your children.***

SPECIAL SKILLS	Preschool	K	1	2
* Color For The Non-Reader	●	●		
* Name That Color		●	●	
* Like Shape Identification	●	●	●	
* Different Shape Identification	●	●	●	
Cave Game	●	●	●	●
MATH AND NUMBER SKILLS				
Count With Me	●	●	●	
Number Recognition	●	●	●	●
Addition		●	●	
Subtraction		●	●	
Add.—Vertical/Horizontal		●	●	
Sub.—Vertical/Horizontal		●	●	
Advanced Addition/Subtraction		●	●	●
Ones and Tens			●	●
LANGUAGE SKILLS				
* Alphabet Recognition			●	●
* Letter Sequence	●	●	●	
* Like Symbol Discrimination	●	●		
* Different Symbol Discrimination	●	●		

Pricing Information	Cass.	Disk		Cass.	Disk
Single Program	6.95	9.95	Pre-School	36.95	41.95
Special Skills	24.95	29.95	Kindergarten	54.95	59.95
Language Skills	19.95	24.95	First Grade	49.95	54.95
Math and Number Skills	34.95	39.95	Second Grade	21.95	26.95
Complete Set	59.95	64.95			
Edumate™ Light Pen	19.95		Add $2.00 for postage & handling		

* Compatible with our Edumate™ Light Pen
NOTE: All software requires 8K cassette/16K disk

Start competing with your 4-year-old for computer time. Enroll in **THE LEARNING CENTER**, it could be the best investment you ever make ... ***in your children's future.*** See your local dealer or order direct.

Available now for the ATARI 400/800
COMING SOON for the TRS-80 Model I, III, Color Apple, VIC, TI-99
Free Catalog Upon Request

THE PROGRAMMER'S INSTITUTE
—a division of **FUTUREHOUSE™**
P.O. Box 3191 Dept C
Chapel Hill, NC 27514

1-919-967-0861 10 am - 9 pm, Monday - Saturday

```
40  GR : HOME
50  FOR P = 1 TO 800: NEXT
100  REM ***DIMENSION VARIABLES
120 N = 1: REM *MESSAGE POINTER
500  REM  *** FRIEND MASTER
510  GOSUB 1010: REM  *FRIEND WAKE UP
520  GOSUB 2010: REM  *FRIEND TALK
530  GOSUB 3210: REM  *STORE CHILD'S NAME
540  GOSUB 2010: REM  *FRIEND TALK
550  PRINT : PRINT : PRINT : PRINT : PRINT :
     END
1000  REM *** FRIEND WAKE UP
1010  GOSUB 5010: REM  *DRAW FACE
1020  GOSUB 5410: REM  *DRAW SLEEPING EYES
1035  FOR P = 1 TO 800: NEXT
1040  GOSUB 4000: REM  *WAKE UP BELL
1050  GOSUB 5460: REM  *DRAW OPEN EYES
1060  FOR P = 1 TO 600: NEXT
1070  GOSUB 5320: REM  *WINK EYE
1080  FOR P = 1 TO 100: NEXT
1085 M = 0: GOSUB 4820: REM  *WINK NOISE
1090  GOSUB 5460: REM  *DRAW OPEN EYES
1100  FOR P = 1 TO 800: NEXT
1110  RETURN
2000  REM *** FRIEND TALK
2005  REM  * SELECT MESSAGE
2006 N = N + 1: REM  * SET POINTER TO NEXT M
     ESSAGE
2010  READ SNUM: REM  * SNUM = NO. OF MESSAG
     ES IN SET
2015  FOR K = 1 TO SNUM
2020  GOSUB 3010: REM  *FRIEND TALK--1 SCREEN
2033  FOR P = 1 TO 1000: NEXT
2035  GOSUB 5510: REM * CLEAR MESSAGE WINDOW
2040  NEXT
2050  RETURN
3000  REM *** FRIEND TALKING--1 SCREEN
3010 PY = 21: REM  *SET VERTICAL TAB FOR TEXT
3040  READ M$
3050  IF M$ = "-1" THEN  RETURN
3051  IF M$ = "*" THEN M$ = N$
3060  VTAB PY
3070  PRINT M$;" ";: GOSUB 5250
3075  GOSUB 4810: REM  *FRIEND SOUND
3080  FOR P = 1 TO 50: NEXT : REM  *KEEP MOU
     TH OPEN
3090  GOSUB 5200: REM  *CLOSE MOUTH
3095  FOR P = 1 TO 100: NEXT : REM  *KEEP MO
     UTH CLOSED
3110  GOTO 3040
3200  REM *** FRIEND ASKS CHILD'S NAME
3210  REM
3220  VTAB 21: HTAB 10: INPUT N$
3265  FOR P = 1 TO 75: NEXT
3267  GOSUB 5510: REM * CLEAR MESSAGE WINDOW
3270  RETURN
4000  REM ***WAKE UP BELL
4010  POKE 768,30: POKE 769,105: CALL 770
4020  FOR P = 1 TO 100: NEXT
4030  POKE 768,20: POKE 769,132: CALL 770
4040  RETURN
4080  RETURN
4625  FOR P = 1 TO 15: NEXT
4800  REM *** FRIEND'S VOICE
4810 M =  INT ( RND (1) * 51) + 15
4820  FOR A = M + 25 TO M STEP  - 8
4830  POKE 768,3: POKE 769,A: CALL 770
4840  NEXT
4880  RETURN
5000  REM *** FRIEND'S FACE
5010  GR
5020  COLOR= 9: PLOT 20,10: PLOT 20,12: HLIN
     19,21 AT 11
5025  COLOR= 7: PLOT 20,13: HLIN 19,21 AT 14
     : HLIN 17,23 AT 16: HLIN 17,23 AT 18
5030  COLOR= 2: HLIN 18,22 AT 15: HLIN 17,23
     AT 17: HLIN 17,23 AT 19
5035  COLOR= 11: FOR I = 20 TO 31: HLIN 17,2
     3 AT I: NEXT : PLOT 16,24: PLOT 24,24: HL
     IN 18,22 AT 32: HLIN 18,22 AT 33
5040  COLOR= 12: HLIN 16,24 AT 34: HLIN 15,2
     5 AT 35: HLIN 15,25 AT 36: HLIN 14,26 AT
     37
5050  COLOR= 1: HLIN 19,21 AT 28
5200  REM  ***CLOSE MOUTH
5220  POKE 1852,177
5230  RETURN
5250  REM *** OPEN MOUTH
5260  POKE 1852,16
5280  RETURN
5300  REM  ***LEFT EYE WINK
5320  POKE 1467,176: POKE 1469,190
5330  FOR I = 1 TO 150: NEXT
5340  RETURN
5400  REM  ***EYES ASLEEP
5410  POKE 1467,190: POKE 1469,190
5420  RETURN
5450  REM  ***EYES AWAKE
5460  POKE 1467,176: POKE 1469,176
5470  RETURN
5500  REM  ***CLEAR MESSAGE WINDOW
5510  HOME
5550  RETURN
5600  REM *** SOUND SUBROUTINE
5610  DATA 172,01,03,174,01,03,169,04,32,168
     ,252,173,48,192,232,208,253,136,208,239
     ,206,0,03,208,231,96
6000  REM  ***MESSAGES
6010  DATA 3
6011  DATA    HI,I'M,GEB,-1
6012  DATA  YOU,TURNED,ME,ON,-1
6013  DATA       WHO'S,OUT,THERE?,-1
6020  DATA 2
6021  DATA   I'M,SO,HAPPY,-1
6022  DATA TO,SEE,YOU,*,-1
```

©

Glenn M. Kleiman

Gentle Introductions To Programming

Everyone should understand the fundamentals of programming; learning about programming is an important step towards becoming computer literate. Without a good concept of programming, one cannot really understand the nature of computers, their capabilities, and their limitations.

In addition, programming is an excellent vehicle for developing thinking skills. Many teachers have reported that when children learn to program, their work in other subjects improves. The teachers attribute this general improvement to the students learning to approach problems more systematically and to pay greater attention to detail.

While they acknowledge its importance, few teachers are experienced programmers, and fewer still are well prepared to teach programming. Many dedicated teachers, realizing the need for computer literacy, are making extraordinary efforts to learn about computers and programming so that they can help their students learn. In this month's column, I discuss two courseware packages which can be extremely valuable for such teachers.

The two packages, *Kidstuff* and *Karel the Robot*, are designed to be "gentle introductions to programming." They each contain a simplified programming language and a book with step-by-step lessons for teaching it. With *Kidstuff* or *Karel the Robot*, students (and teachers) can learn many of the fundamental concepts of programming. Both packages can also serve as stepping-stones for students who want to go on to learn BASIC, Logo, PILOT or Pascal.

Kidstuff and *Karel* are not the only existing gentle introductions to programming, and I expect more will be developed in the next few years. Therefore, before turning to the specifics of these two packages, I will discuss in general what we might expect from courseware designed to introduce students to programming.

Structuring Programming

We can think of creating a computer program as involving three main activities. First, we must *design* the program. Recommended approaches to design have been labeled structured programming, successive refinement, top-down programming, and modular programming. In brief, the recommended approach is to start with the most general aims of the program and successively refine them into more and more specific sub-tasks. We want to design the program so that we can work on one sub-task at a time, handling each in a separate module of the program. The modules are then combined to form the program.

Some languages encourage structured programming more than others. For example, in some languages, variables can be local to a module, so you do not have to worry about using the same variable name in different modules of a program. Introductions to programming should encourage structured programming so that students acquire proper habits from the beginning.

The second main activity of programming is to *code* the instructions – translate them into a language the computer can follow. We can discuss computer languages in terms of three types of elements: (1) Commands, such as those which print words and text on the screen, accept inputs from the users of programs, perform mathematical operations, and manipulate text; (2) Control Elements, which are used to iterate (repeat) sets of commands (e.g., FOR/NEXT loops in BASIC), follow commands when a given condition is true (e.g., IF-THEN conditionals), branch to other parts of the program (e.g., GOTO), or use a module (subroutine or procedure) and then return to the current part of the program (e.g., GOSUB); and (3) Data Structures, or ways information can be organized and stored (e.g., simple variables, subscripted variables, hierarchical trees).

DYNACOMP

Quality software for:*

APPLE II Plus
OSBORNE-1
NORTH STAR***
ATARI
PET/CBM
ALTAIR****
NEC PC-8000
TRS-80 (Level II)**
SUPERBRAIN*****
CP/M Disks/Diskettes
(See Availability box MBASIC/CBASIC)

DYNACOMP OFFERS THE FOLLOWING

- Widest variety
- Guaranteed quality
- Fastest delivery
- Friendly service
- Free catalog*
- Toll free order phone

*$2.00 U.S. funds for foreign mailing.

CARD GAMES

NEW

BRIDGE MASTER (Available for all computers) — Price: $21.95 Diskette

If you liked DYNACOMP'S BRIDGE 2.0, you will absolutely love BRIDGE MASTER. BRIDGE MASTER is a comprehensive bridge program designed to provide hours of challenging competition. Bidding features include the Blackwood convention, Stayman convention, pre-emptive openings, and recognition of demand bids and jump-shift responses. After playing a specific hand, you may replay the same hand, with the option of switching cards with your computer opponents. This feature allows you to compare your bidding and playing skills to BRIDGE MASTER. Bonuses for game contracts and slams are awarded as in duplicate bridge. Doubled contracts are scored based upon a computer assigned vulnerability. A score card is displayed at the conclusion of each hand. The score card displays a summary of total hands played, total points scored, number of contracts made and set, and % bids made. BRIDGE MASTER is clearly the best computer bridge program available.

DYNACOMP'S previous BRIDGE 2.0 customers may upgrade to BRIDGEMASTER for a nominal charge of $5.00 plus postage and handling (see ordering information box). Original cassette/diskette must be returned.

BACCARAT (Atari only) — Price: $18.95 Cassette/$22.95 Diskette

This is the European card game which is the favorite of the Monte Carlo jet set. Imagine yourself at the gaming table with 007 to your left and Goldfinger to your right. Learn and play BACCARAT at your leisure on the Atari. Contains full high resolution color graphics and matching sound. Runs in 16K. Requires one joystick.

GIN RUMMY (Apple diskette only) — Price: $22.95 Diskette

This is the best micro computer implementation of GIN RUMMY existing. The computer plays exceptionally well, and the HIRES graphics are superb. What else can be said?

POKER PARTY (Available for all computers) — Price: $19.95 Cassette/$23.95 Diskette

POKER PARTY is a draw poker simulation based on the book, POKER, by Oswald Jacoby. This is the most comprehensive version available for microcomputers. The party consists of yourself and six other (computer) players. Each of these players (you will get to know them) has a different personality in the form of a varying propensity to bluff or fold under pressure. Practice with POKER PARTY before going to that expensive game tonight! Apple cassette and diskette versions require a 32K (or larger) Apple II.

GO FISH (Available for all computers) — Price: $14.95 Cassette/$18.95 Diskette

GO FISH is a classic children's card game. The opponent is a friendly computer with user inputs that are simple enough for small children to easily master. The Apple and Atari versions employ high resolution graphics for the display of hands. A must for children! Runs in 16K.

NEW

BLACKJACK COACH (32K TRS-80 only) — Price: $29.95 Cassette/$33.95 Diskette

BLACKJACK COACH teaches and evaluates professional playing methods. This program will coach you using the Basic and the Complete Card Counting Methods. The BLACKJACK COACH can be used in automatic, unattended play to test the playing and betting strategies you select. Extensive summary reports pinpoint the strengths and weaknesses of various methods of play. All the standard player choices are included: Insurance, splitting pairs, double downs and surrender (optional). A line printer may be used to collect data. If you risk money at the tables, increase your skills with the BLACKJACK COACH!

THOUGHT PROVOKERS

MANAGEMENT SIMULATOR (Available for all computers) — Price: $25.95 Cassette/$29.95 Diskette

This program is both an excellent teaching tool as well as a stimulating intellectual game. Based upon similar games played at graduate business schools, each player or team controls a company which manufactures three products. Each player attempts to outperform his competitors by setting selling prices, production volumes, marketing and design expenditures etc. The most successful firm is the one with the highest stock price when the simulation ends.

FLIGHT SIMULATOR (Available for all computers) — Price: $19.95 Cassette/$23.95 Diskette

A realistic and extensive mathematical simulation of take-off, flight and landing. The program utilizes aerodynamic equations and the characteristics of a real airfoil. You can practice instrument approaches and navigation using radials and compass headings. The more advanced flyer can also perform loops, half-rolls and similar aerobatic maneuvers. Although this program does not employ graphics, it is exciting and very addictive. See the software review in COMPUTRONICS. Runs in 16K Atari.

VALDEZ (Available for all computers) — Price: $17.95 Cassette/$21.95 Diskette

VALDEZ is a computer simulation of supertanker navigation in the Prince William Sound/Valdez Narrows region of Alaska. Included in this simulation is a realistic and extensive 256 x 256 element map, portions of which may be viewed using the ship's alphanumeric radar display. The motion of the ship itself is accurately modelled mathematically. The simulation also contains a model for the tidal patterns in the region, as well as other traffic (outgoing tankers and drifting icebergs). Chart your course from the Gulf of Alaska to Valdez Harbor! See the software reviews in 80 Software Critique, Personal Computing and Creative Computing.

BACKGAMMON 2.0 (Available for all computers) — Price: $19.95 Cassette/$23.95 Diskette

This program tests your backgammon skills and will also improve your game. A human can compete against a computer or against another human. The computer can even play against itself. Either the human or the computer can double or generate dice rolls. Board positions can be created or saved for replay. BACKGAMMON 2.0 plays in accordance with the official rules of backgammon and is sure to provide many fascinating sessions of backgammon play.

NEW

FROG MASTER (Atari only) — Price: $17.95 Cassette/$21.95 Diskette

The Atari APEX first prize winner. FROG MASTER contains exciting arcade features in addition to being a highly educational program. It is a fast-moving high-concentration game for 1-4 players. You score by making touchdowns on the opponents' goal line – if his goalie doesn't get there first. But your players (tadpoles and frogs) must be trained. This is accomplished by giving them a reward at just the right moment when they do something right. This takes precise timing and judgement. Your critters must penetrate barriers and avoid evil line backs if they are to score. Many will fall by the wayside, but some will get through. As they learn you can look inside their heads to see how they think. As you reward them, they reward you (the "thought processes" simulated demonstrate the basic type of animal learning - operant conditioning - widely studied in high school and college courses). As you teach them they teach you how learning takes place! Great graphics! Runs in 16K. Requires two joysticks.

FOREST FIRE! (Atari only) — Price: $14.95 Cassette/$18.95 Diskette

Using excellent graphics and sound effects, this simulation puts you in the middle of a forest fire. Your job is to direct operations to put out the fire while compensating for changes in wind, weather and terrain. Not protecting valuable structures can result in startling penalities. Life-like variables are provided to make FOREST FIRE! very suspenseful and challenging. No two games have the same setting and there are 3 levels of difficulty.

CRANSTON MANOR ADVENTURE (North Star, SuperBrain and CP/M only) — Price: $19.95 Diskette

At last! A comprehensive Adventure game for North Star and CP/M systems. CRANSTON MANOR ADVENTURE takes you into mysterious CRANSTON MANOR where you attempt to gather fabulous treasures. Lurking in the manor are wild animals and robots who will not give up the treasures without a fight. The number of rooms is greater and the associated descriptions are much more elaborate than the current popular series of Adventure programs, making this game the top in its class. Play can be stopped at any time and the status stored on diskette.

SPACE EVACUATION! (Available for all computers) — Price: $15.95 Cassette/$19.95 Diskette

Can you colonize the galaxy and evacuate the Earth before the sun explodes? Your computer becomes the ship's computer as you explore the universe to relocate millions of people. This simulation is particularly interesting as it combines many of the exciting elements of classic space games with the mystery challenge of ADVENTURE.

MONARCH (Atari only) — Price: $14.95 Cassette/$18.95 Diskette

MONARCH is a fascinating economic simulation requiring you to survive an 8-year term as your nation's leader. You determine the amount of acreage devoted to industrial and agricultural use, how much food to distribute to the populace and how much should be spent on pollution control. You will find that all decisions involve a compromise and that it is not easy to make everyone happy. Runs in 16K Atari.

NEW

RUBIK'S CUBE SOLVER (Available for all computers) — Price: $14.95 Cassette/$18.95 Diskette/$21.45 Disk

Solving the Rubik's cube puzzle is an exercise in algorithimic logic, and is a "natural" for computer calculation. The RUBIK'S CUBE SOLVER permits you to input the starting state of the 24 facing elements of the cube. It then solves the problem one step at a time, with each step shown as a unfolded view of the cube. Can you solve the cube in fewer steps. In any case, it sure beats disassembling the cube or peeling off and replacing the colors! Requires 16K.

AVAILABILITY

DYNACOMP software is supplied with complete documentation containing clear explanations and examples. Unless otherwise specified, all programs will run within 16K program memory space (ATARI requires 24K). Except where noted, programs are available on ATARI, PET, TRS-80 (Level II), NEC and Apple (Applesoft) cassette and diskette as well as North Star single density (double density compatible) diskette. Additionally, most programs can be obtained on standard (IBM 3740 single density/double density compatible format) 8" CP/M floppy disks for systems running under MBASIC or CBASIC (for example, Altos, Xerox 820, Heath/Zenith and many others). 5¼" CP/M diskettes are available for the North Star, SuperBrain and Osborne computer systems.

AND MORE...

STARBASE 3.2 (Available for all computers) — Price: $13.95 Cassette/$17.95 Diskette

This is the classic space simulation, but with several new features. For example, the Croylins now shoot at the Invincible without warning while also attacking starbases in other quadrants. The Croylins also attack with both light and heavy cruisers and move when shot at! The situation is hectic when the Invincible is besieged by three heavy cruisers and a starbase S.O.S. is received! The Croylins get even! See the software reviews in A.N.A.L.O.G., 80 Software Critique and Game Merchandising.

LIL' MEN FROM MARS (Atari only) — Price: $19.95 Cassette/$23.95 Diskette

Defend yourself! The little men from Mars are out to get you if you don't get them first. This is a hilarious high resolution animated graphics (arcade) game which exercises much of the Atari's power. Requires one joystick.

NEW

ALVIN (Atari only) — Price: $17.95 Cassette/$21.95 Diskette

ALVIN is a great arcade game. You are commanding a highly maneuverable ship seeking to destroy several enemy cities. You are attempting to bomb these cities while at the same time trying to avoid their defensive fire (MISSILE COMMAND* in reverse?). Also, your radar has been damaged so that you can only see downwards. This would normally not be much of a problem except that you also have to contend with high-flying enemy aircraft. As long as you are above these aircraft you have the advantage and are safe. However, high level bombing takes considerable skill. Therefore to achieve your goal the best strategy is to swoop down for a bombing run while the enemy craft is out of range, and quickly retreat to the skies. A fun game. Requires 16K.

ESCAPE FROM VOLANTIUM (Atari only) — Price: $15.95 Cassette/$19.95 Diskette

Bring the action and excitement of an arcade into your home with ESCAPE FROM VOLANTIUM! To escape you must maneuver your space ship around obstacles and laser blast the guardian (without being eaten). If he is killed with a direct shot (not just a leg lopped off), a door opens to the outside. However, the door does not stay open indefinitely. If you fail to escape in time, the door closes and a new guardian appears. Sometimes you can smash through the door by repeatedly chipping away at it. Other times it is impervious. At the higher levels of play more obstacles and guardians appear, adding to the excitement. Uses high resolution graphics and sound. Runs in 16K.

ALPHA FIGHTER (Atari only) — Price: $13.95 Cassette/$17.95 Diskette

Two excellent graphics and action programs in one! ALPHA FIGHTER requires you to destroy the alien starships passing through your sector of the galaxy. ALPHA BASE is in the path of an alien UFO invasion; let five UFO's get by and the game ends. Both games require the joystick and get progressively more difficult the higher you score! ALPHA FIGHTER will run on 16K systems.

THE RINGS OF THE EMPIRE (Atari only) — Price: $14.95 Cassette/$18.95 Diskette

The empire has developed a new battle station protected by rotating rings of energy. Each time you blast through the rings and destroy the station, the empire develops a new station with more protective rings. This exciting game runs on 16K systems, employs extensive graphics and sound and can be played by one or two players.

INTRUDER ALERT (Atari only) — Price: $15.95 Cassette/$19.95 Diskette

This is a fast paced graphics game which places you in the middle of the "Dreadstar" having just stolen its plans. The droids have been alerted and are directed to destroy you at all costs. You must find and enter your ship to escape with the plans. Five levels of difficulty are provided. INTRUDER ALERT requires a joystick and will run on 16K systems.

MIDWAY (Atari 32K only) — Price: $14.95 Cassette/$18.95 Diskette

MIDWAY is an exciting extension of the game of Battleship. It mixes the challenges of strategy and chance. Your opponent can be another human or the computer. Color graphics and sound are both included. Runs in 16K.

NEW

GOLF PRO (Atari only) — Price: $17.95 Cassette/$21.95 Diskette

Both realism and beautiful graphics are joined together in GOLF PRO to produce the best golf simulation available. To really appreciate this game, you should have a color TV so that you can see the green of the fairway, the blue of the water hazards, and the white sand of the traps. You tee off with a wood, use your wedge in the sand trap, and putt on the green just as would be done on the course. Show off the Atari to your friends with GOLF PRO. Requires 16K and one joystick.

GAMES PACK I (Available for all computers) — Price: $14.95 Cassette/$18.95 Diskette

GAMES PACK I contains the classic computer games of BLACKJACK, LUNAR LANDER, CRAPS, HORSERACE, SWITCH and more. These games have been combined into one large program for ease in loading. They are individually accessed by a convenient menu. This collection is worth the price just for the DYNACOMP version of BLACKJACK.

GAMES PACK II (Available for all computers) — Price: $14.95 Cassette/$18.95 Diskette

GAMES PACK II includes the games CRAZY EIGHTS, JOTTO, ACEY-DUCEY, LIFE, WUMPUS and others. As with GAMES PACK I, all the games are loaded as one program and are called from a menu. You will particularly enjoy DYNACOMP's version of CRAZY EIGHTS.

Why pay $9.95 or more per program when you can buy a DYNACOMP collection for just $14.95?

MOON PROBE (Available for all computers) — Price: $12.95 Cassette/$16.95 Diskette

This is an extremely challenging "lunar lander" program. The user must drop from orbit to land at a predetermined target on the moon's surface. You control the thrust and orientation of your craft plus direct the rate of descent and approach angle. Runs in 16K Atari.

SPACE TRAP (Atari only, 16K) — Price: $14.95 Cassette/$18.95 Diskette

This galactic "shoot 'em up" arcade game places you near a black hole. You control your spacecraft using the joystick and attempt to blast as many of the alien ships as possible before the black hole closes about you.

SUPER SUB CHASE (Atari only) — Price: $19.95 Cassette/$23.95 Diskette

SUPER SUB CHASE simulates a search and destroy mission. Set your course and keep an eye on the sonar readings as you hunt for the hidden submarine. Set the depth charge explosion depth and watch them sink towards the sub. This is an addictive game which takes advantage of the Atari's graphics and sound capabilities. One or two players. Joystick(s) required.

TWO PLAYER GAMES

NEW

TWO PLAYER GAMES (Available for all computers; 32K disk/diskette only)

DYNACOMP has acquired the distribution rights to the best eight of Xitron's war games. These two-player games were originally written for the North Star computer, but have since been converted to play on all of the computers currently supported by DYNACOMP. Because our licensing and development costs were so low, DYNACOMP offers these programs two to a disk(ette) for only $19.95/diskette, $23.95/disk. If you like war games, then this is a bargain you can not pass up.

Set #1: PANZER and BLITZKRIEG

PANZER

Date: 23 Nov. 1943 Place: Several miles west of Kiev, Russia. The Russians have just liberated Kiev and are moving quickly to reach the German forces which are preparing for a last desperate attempt to halt the Russian advance.

BLITZKRIEG

Date: Spring 1940 Place: Northern France. The German blitzkrieg in the east was complete. Germany had turned its attention to the west: France. The German forces has penetrated the Ardennes and Meuse. The heroism of Dunkirk, the defense of the Aisne-Somme position, and the final collapse of the French armies in the south has all passed. And, now, the drive on Paris . . .

Set #2: STARSHIP TROOPERS and INVASION OF THE MUD PEOPLE

STARSHIP TROOPERS

Date: Fortieth Century Place: Arachnid planet of Sheol. The first all-out battle on the planet Sheol which will match equal forces of Terran and alien units. The outcome will set the course of the conflict, for the planet of Sheol is a key position in the solar war.

INVASION OF THE MUD PEOPLE

A Peruvian army battalion has been dispatched to a remote village area to investigate the destruction of many local dwellings and the disappearance of most of the villagers. Eye-witnesses have reported strange creatures appearing from scores of slimy mud holes which have oddly begun forming across the terrain.

Set #3: FALL OF THE THIRD REICH and ARMORCAR

FALL OF THE THIRD REICH

Date: March, 1945 Place: Remagen, Germany. The allies under General Eisenhower had reached the Rhine. The Germans had failed in destroying the Ludendorff railroad bridge, allowing several allied divisions to cross before it finally collapsed on March 17 ... and so, the allies began their drive on Berlin.

ARMORCAR

Date: 2 Feb. 1944 Place: Minsk, Russia. A German front-line unit is hard pressed for radio equipment and medical supplies. A relief convoy of armored cars must reach them through partisan-infested territory.

Set #4: MOUNT SURIBACHI and MIDDLE EARTH

MOUNT SURIBACHI

Date: 16 Feb. 1945 Place: Iwo Jima. The Japanese opened fire from Mount Suribachi as the marines landed on the porkchop-shaped island. Gunfire from the hill could cover the entire island, thus it was a critical objective if the Americans were to capture and utilize the all-important air field. Mount Suribachi proved to be one of the most strongly defended positions in the Japanese theatre of war.

MIDDLE EARTH

Date: 1997 Place: MIDDLE EARTH. Through a maze of tunnels, crevices, and rocky passageways discovered leading from an inactive volcano in South America, a team of United Nations' researchers have undertaken a mission to an uncharted frontier: the center of the Earth. After a perilous journey spanning a period of several months, the mission has arrived at the Earth's core, a land of flames, steam, oceans, and unforeseen vegetation. And then the creatures of MIDDLE EARTH appeared . . . unmatched by the most frightening horror stories created by man . . .

MISCELLANEOUS

CRYSTALS (Atari only) — Price: $14.95 Cassette/$18.95 Diskette

A unique algorithm randomly produces fascinating graphics displays accompanied with tones which vary as the patterns are built. No two patterns are the same, and the combined effect of the sound and graphics are mesmerizing. CRYSTALS has been used in local stores to demonstrate the sound and color features of the Atari. Runs in 16K.

NORTH STAR SOFTWARE EXCHANGE (NSSE) LIBRARY

DYNACOMP now distributes the 23 volume NSSE library. These diskettes each contain many programs and offer an outstanding value for the purchase price. They should be part of every North Star user's collection. Call or write DYNACOMP for details regarding the contents of the NSSE collection.

Price: $9.95 each/$8.95 each (4 or more)

The complete collection may be purchased for $159.95.

NEW

5¼" DISKETTES (soft sectored/ten sectored) — Price: $39.95/20 Diskettes

As you might imagine, DYNACOMP purchases diskettes in large quantities and at wholesale prices. We want to pass the savings along to you!

**ATARI, PET CBM, NORTH STAR, CP/M, IBM, OSBORNE, SUPERBRAIN, NEC PC-8000 and XEROX are registered tradenames and/or trademarks.*

***Except where noted, all TRS-80 Model I software is available on cassette (only) for the TRS-80 Model III. Exceptions: VALDEZ, CRIBBAGE, GRAFIX, CHESSMASTER. TRS-80 diskettes are not supplied with either DOS or BASIC.*

****For most North Star disk-based systems.*

*****For Altair systems having Microsoft BASIC.*

******For SUBERBRAIN systems running under MBASIC or CBASIC (state which).*

BUSINESS and UTILITIES

NEW

PORTFOLIO MANAGEMENT (Apple only) — Price: $69.95 two Diskettes

The PORTFOLIO MANAGEMENT package was written by a stock broker to help manage portfolios for individual customers. With this program data files can be easily created and kept up to date. A variety of reports can be generated for clients which are attractively and professionally laid out. The user may define his/her own investment catagories. PORTFOLIO MANAGEMENT is a top quality, professional tool which will not only provide you with new conveniences but will also serve to enhance your appearance as an efficient and up to date advisor to your clients. Comes complete on two diskettes along with a 30 page instruction manual.

NEW

PERSONAL FINANCE SYSTEM (Available for all computers) — Price: $39.95 Diskette

PFS is a single diskette, menu-oriented system composed of ten different programs. Besides recording your expenses and tax deductible items, PFS will sort and summarize expenses by payee, and display information on expenditures by any of 26 user defined codes by month or by payee. PFS will even produce monthly bar graphs of your expenses by category! This powerful package requires only one disk drive, minimal memory (24K Atari, 32K North Star) and will store up to 600 records per disk (and over 1000 records per disk by making a few simple changes to the programs). You can record checks plus cash expenses so that you can finally see where your money goes and eliminate guesswork and tedious hand calculations. Contains high speed machine language sort. PFS has been demonstrated on network (CBS) TV!

FAMILY BUDGET (Apple and Atari only) — Price: $34.95 Diskette

FAMILY BUDGET is a very convenient financial record-keeping program. You will be able to keep track of cash and credit expenditures as well as income on a daily basis. You can record tax deductible items and charitable donations. FAMILY BUDGET also provides a continuous record of all credit transactions. You can make daily cash and charge entries to any of 21 different expense accounts as well as to 5 payroll and tax accounts. Data are easily retrieved giving the user complete control over an otherwise complicated (and unorganized!) subject.

NEW

TEXT MASTER (Apple 32K, diskette only) — Price: $49.95 Diskette

TEXT MASTER is a general purpose text editor for the Apple II computer. It features powerful, English-oriented commands which permit the complete manipulation of textual information. The information treated may be correspondence, computer programs, data to be used by other programs, and more. TEXT MASTER also interfaces with any printer connected to your Apple. The minimum system requirements are 32K of RAM, Applesoft on ROM, at least one disk, and a lower case adapter. TEXT MASTER can process any length file segment by segment. Thus it is possible to process files as large as a diskette. Comes complete with an efficient 16 page manual. TEXT MASTER is equivalent in capability to many word processing systems costing 3 x or more in price. The commands available in TEXT MASTER are: COLLECT, USE, EXECUTE, NONUM, INSERT, MERGE, STOP, RENUM, DELETE, SAVE, COPY, FREE, REPLACE, WAIT, MOVE, LENGTH, LIST, RESAVE, CLEAR, AUTO, RETRY, APPEND, SET, MANUAL, CHANGE, SCRATCH, SHOW, CATALOG, MODIFY, COMPARE, NUM, DISPLAY, HELP.

INTELINK (Atari only) — Price: $49.95 Diskette

This software package contains a menu-driven collection of programs for facilitating efficient two-way communications through a full duplex modem (required for use). In one mode of operation you may connect to a data service (e.g., The SOURCE or MicroNet) and quickly load data such as stock quotations onto your diskette for later viewing. This greatly reduces "connect time" and thus the service charge. You may also record the complete contents of a communications session. Additionally, programs written in BASIC, FORTRAN, etc. may be built off-line using the support text editor and later "uploaded" to another computer, making the Atari a very smart terminal. Even Atari BASIC programs may be uploaded. Further, a command file may be built off-line and used later as controlling input for a time-share system. That is, you can set up your sequence of time-share commands and programs, and the Atari will transmit them as needed; batch processing. All this adds up to saving both connect time and your time.

NEW

PAYFIVE (Apple II plus diskette, two drives required) — Price: $149.95

This is an enormously flexible employee payroll system with extraordinarily good human engineering features. PAYFIVE prints checks and compiles the required federal, state and local forms for up to 148 employees. The pay methods may be hourly, salary, commission or any combination. There are multiple options for pay periods, and they also can be used in any combination. PAYFIVE includes many other features and comes extremely well documented with a 200 page manual. The manual may be purchased separately for $30, and that payment later applied to the software purchase.

SHOPPING LIST (Atari only) — Price: $12.95 Cassette/$16.95 Diskette

SHOPPING LIST stores information on items you purchase at the supermarket. Before going shopping, it will remind you of all the things you might need, and then display (or optionally print) your shopping list and the total cost. Adding, deleting, changing and storing data is very easy. Runs with 16K.

NEW

TAX OPTIMIZER (Available for all computers) — Price: $59.95 Diskette

The TAX OPTIMIZER is an easy-to-use, menu-oriented software package which provides a convenient means for analyzing various income tax strategies. The program is designed to provide a quick and easy data entry. Income tax is computed by all tax methods (regular, income averaging, maximum and alternate minimum tax). The user may immediately observe the tax effect of critical financial decisions. TAX OPTIMIZER has been thoroughly field tested in CPA offices and comes complete with the current tax tables in its data files. TAX OPTIMIZER is tax deductible!

STOCK MASTER/STOCK PLOT (Apple 48K) — Price: $59.95 Diskette

This is a full-featured stock portfolio management and analysis system. Ten years of records on up to thirty stocks may be maintained. You may record prices, revenue, earnings/share, ROE, quarterly earnings and dividends, transactions, long/short term gains, P/E, indexes (DJ, S & P, NASDAQ) and MORE! You may plot (HIRES) the price history of any stock against any other, time, or the indexes. Portfolio value may be evaluated at any time. Comes complete with superbly written instructions and sample files on a second diskette.

TURNKEY AND MENU (Atari only) — Price: $17.95 Diskette

TURNKEY is a utility program which allows you to create autoboot/autorun diskettes easily. Simply load and run TURNKEY, load the program diskette to be modified, and answer the questions! The TURNKEY diskette also comes with DOS 2.0 and includes another program, MENU. MENU lists the contents of your diskette alphabetically, and permits the running of any BASIC program on the diskette by typing a single key. TURNKEY and MENU provide you with the ability to run any program on your diskette by simply turning on the computer and pressing a single key.

STOCKAID (Atari only) — Price: $29.95 Diskette

STOCKAID provides a powerful set of tools for stock market analysis. With STOCKAID you can display point and figure charts, as well as bar charts with oscillators. You can also examine long term moving averages and on-balance volume features. STOCKAID allows you to input daily data with a single diskette storage capability of 239 days x 16 stocks. Included are stock dividend and split adjustment capabilities. A very professional package!

NEW

NYINDEX (Atari only) — Price: $29.95 Diskette

NYINDEX is a comprehensive software package for storing, retrieving and plotting New York Stock Exchange information. The daily data treated includes the composite index, advances, declines, new highs and new lows. Graphical displays include the above plus the index oscillator, cumulative advances/declines and moving averages. Data entry and editing is easy. The diskette includes more than two years of daily data. NYINDEX is an excellent companion to STOCKAID.

PLAYER-MISSILE GRAPHICS TABLET (Atari only) — Price: $19.95 Diskette

The PLAYER-MISSILE GRAPHICS TABLET was designed to take the drudgery out of developing four color displays in GRAPHICS MODE 7. No longer will you have to read the locations of those tiny blocks on your graph paper and calculate PLOTs and DRAWTOs. With PMG you will be able to easily design colorful graphic displays with your joystick and save them on diskette for later recall.

LIFE CYCLE ANALYSIS AND DEPRECIATION (Apple diskette only) — Price: $39.95 Diskette

This software package creates a data file of business expenses for equipment which can later be used to calculate and display a variety of reports. You may project annual costs, find the present worth, create depreciation schedules and justify tax deductions. The evaluation techniques conform to standards set by federal agencies. This is an invaluable package for any businessman who has invested in equipment. LIFE CYCLE ANALYSIS features an easy to use data file creation section and provides formatted hardcopy reports for use in presentations or for tax record keeping purposes. When used for generating tax information, this package is tax deductible! Requires 48K. Comes on two diskettes.

NEW

MICROMAGIC (Apple diskette only) — Price: $39.95 Diskette

The emphasis of this program is clearly the MAGIC! MICROMAGIC offers outstanding versatility in its ability to function as a stand alone entertainment package or as a utility program to create stunning *animated* graphics for use in other programs. The secret lies in MICROMAGIC's special on screen graphics editor. You control a graphics cursor directly from the keyboard, creating high resolution images using all 16 available colors. When you are done with a picture, it can be saved on disk with a single key command. Up to 24 images can be saved as "frames" of a movie, and then played back at high speed to creat short animated sequences. The effects are truly stunning.

This package comes complete with demonstration programs so that even novice users can get immediate results. No programming skills are necessary to use MICROMAGIC. If you have been frustrated by the effort required to create graphics images with your computer, MICROMAGIC will delight you.

SHAPE MAGICIAN (Apple II, 48K, diskette only) — Price: $29.95

At last! An utility for painlessly creating graphics shapes for the Apple. Create, edit and save up to 30 shapes which can then be used to develop arcade games or to simply enhance your programs. Add that professional touch!

ORDERING INFORMATION

All orders are processed and shipped within 48 hours. Please enclose payment with order and include the appropriate computer information. If paying by VISA or Master Card, include all numbers on card. Purchase orders accepted.

Shipping and Handling Charges
Within North America: Add $2.00
Outside North America: Add 15% (Air Mail)

Delivery
All orders (excluding books) are sent First Class.

Quantity Discounts
Deduct 10% when ordering 3 or more programs; 20% when ordering 5 or more. Dealer discount schedules available upon request.

8" CP/M Disks
Add $2.50 to the listed diskette price for each 8" floppy disk (IBM soft sectored CP/M format). Programs run under Microsoft MBASIC or BASIC-80.

5¼" CP/M Disks
All software available on 8" CP/M disks is also available on 5¼" disks. North Star, Osborne, Superbrain and NEC format.

Ask for DYNACOMP programs at your local software dealer. Write for detailed descriptions of these and other programs from DYNACOMP.

DYNACOMP, Inc. (Dept. E)
1427 Monroe Avenue
Rochester, New York 14618
24 hour message and order phone: (716) 442-8731
Toll free order phones: (800) 828-6772
(800) 828-6773
Office phone (9AM-5PM EST): (716) 442-8960

New York State residents please add 7% NYS sales tax.

EDUCATION

HODGE PODGE (Apple 48K only) — Price: $14.95 Cassette/$18.95 Diskette

Let HODGE PODGE be your child's teacher. Pressing any key on your Apple will result in a different and intriguing "happening" related to the letter or number of the chosen key. The program's graphics, color and sound are a delight for children from ages 1½ to 7. HODGE PODGE is a non-intimidating teaching device which brings a new dimension to the use of computers in education. See the excellent reviews of this very popular program in INFOWORLD and SOFTALK.

TEACHER'S AIDE (Atari and PET only) — Price: $13.95 Cassette/$17.95 Diskette

TEACHER'S AIDE consists of three basic modules contained in one program. The first module provides addition and subtraction exercises of varying levels of difficulty. The second module consists of multiplication problems in which the student may be tested both on the final answer and/or on the subtotal answers in the long hand procedure. Several levels of complexity are provided here as well. The third module consists of division problems; one particularly nice feature of the division module is that the long hand division steps can be displayed along with the remainder in order to clearly demonstrate the procedure by which the remainder is derived. Using TEACHER'S AIDE is not merely a drill, but rather a learning experience.

STATISTICS and ENGINEERING

DIGITAL FILTER (Available for all computers) — Price: $39.95 Cassette/$43.95 Diskette

DIGITAL FILTER is a comprehensive data processing program which permits the user to design his own filter function or choose from a menu of filter forms. In the explicit design mode the shape of the frequency transfer function is specified by directly entering points along the desired filter curve. In the menu mode, ideal low pass, high pass and bandpass filters may be approximated to varying degrees according to the number of points used in the calculation. These filters may optionally also be smoothed with a Hanning function. In addition, multi-stage Butterworth filters may be selected. Features of DIGITAL FILTER include plotting of the data before and after filtering, as well as display of the chosen filter functions. Also included are convenient data storage, retrieval and editing procedures.

DATA SMOOTHER (Not available for Atari) — Price: $19.95 Cassette/$23.95 Diskette

This special data smoothing program may be used to rapidly derive useful information from noisy business and engineering data which are equally spaced. The software features choice in degree and range of fit, as well as smoothed first and second derivative calculation. Also included is automatic plotting of the input data and smoothed results.

FOURIER ANALYZER (Available for all computers) — Price: $19.95 Cassette/$23.95 Diskette

Use this program to examine the frequency spectra of limited duration signals. The program features automatic scaling and plotting of the input data and results. Practical applications include the analysis of complicated patterns in such fields as electronics, communications and business.

TFA (Transfer Function Analyzer) — Price: $19.95/$23.95 Diskette

This is a special software package which may be used to evaluate the transfer functions of systems such as hi-fi amplifiers and filters by examining their response to pulsed inputs. TFA is a major modification of FOURIER ANALYZER and contains an engineering-oriented decibel versus log-frequency plot as well as data editing features. Whereas FOURIER ANALYZER is designed for educational and scientific use, TFA is an engineering tool. Available for all computers.

HARMONIC ANALYZER (Available for all computers) — Price: $24.95 Cassette/$28.95 Diskette

HARMONIC ANALYZER was designed for the spectrum analysis of repetitive waveforms. Features include data file generation, editing and storage/retrieval as well as data and spectrum plotting. One particularly unique facility is that the input data need not be equally spaced or in order. The original data is sorted and a cubic spline interpolation is used to create the data file required by the FFT algorithm.

FOURIER ANALYZER, TFA and HARMONIC ANALYZER may be purchased together for a combined price of $51.95 (three cassettes) and $63.95 (three diskettes).

REGRESSION I (Available for all computers) — Price: $19.95 Cassette/$23.95 Diskette

REGRESSION I is a unique and exceptionally versatile one-dimensional least squares "polynomial" curve fitting program. Features include very high accuracy; an automatic degree determination option; an extensive internal library of fitting functions; data editing; automatic data, curve and residual plotting; a statistical analysis (e.g.: standard deviation, correlation, coefficient, etc.) and much more. In addition, new fits may be tried without reentering the data. REGRESSION I is certainly the cornerstone program in any data analysis software library.

REGRESSION II (Available for all computers) — Price: $19.95 Cassette/$23.95 Diskette

PARAFIT is designed to handle those cases in which the parameters a imbedded (possibly nonlinearly) in the fitting function. The user simply inserts the functional form, including the parameters (A(1), A(2), etc.) as one or more BASIC statement lines. Data, results and residuals may be manipulated and plotted as with REGRESSION I. Use REGRESSION I for polynomial fitting, and PARAFIT for those complicated functions.

MULTILINEAR REGRESSION (MLR) (Available for all computers) — Price: $24.95 Cassette/$28.95 Diskette

MLR is a professional software package for analyzing data sets containing two or more linearly independent variables. Besides performing the basic regression calculation, this program also provides easy to use data entry, storage, retrieval and editing functions. In addition, the user may interrogate the solution by supplying values for the independent variables. The number of variables and data size is limited only by the available memory.

REGRESSION I, II and MULTILINEAR REGRESSION may be purchased together for $51.95 (three cassettes) or $63.95 (three diskettes).

ANOVA (Not available on Atari cassette or for PET/CBM) — Price: $39.95 Cassette/$43.95 Diskette

In the past the ANOVA (analysis of variance) procedure has been limited to the large mainframe computers. Now DYNACOMP has brought the power of this method to small systems. For those conversant with ANOVA, the DYNACOMP software package includes the 1-way, 2-way and N-way procedures. Also provided are the Yates 2^{K-P} factorial designs. For those unfamiliar with ANOVA, do not worry. The accompanying documentation was written in a tutorial fashion (by a professor in the subject) and serves as an excellent introduction to the subject. Accompanying ANOVA is a support program for building the data base. Included are several convenient features including data editing, deleting and appending.

BASIC SCIENTIFIC SUBROUTINES, Volumes 1 and 2 (Not available for Atari)

DYNACOMP is the exclusive distributor for the software keyed to the popular texts *BASIC SCIENTIFIC SUBROUTINES, Volumes 1 and 2* by F. Ruckdeschel (see advertisements in BYTE magazine). These subroutines have been assembled according to chapter. Included with each collection is a menu program which selects and demonstrates each subroutine.

Volume 1
Collection #1: Chapters 2 and 3 - Data and function plotting; complex variables and functions.
Collection #2: Chapter 4 - Extended matrix and vector operations.
Collection #3: Chapters 5 and 6 - Random number generators (Poisson, Gaussian, etc.); series approximations.
Price per collection: $16.95 Cassette/$20.95 Diskette*
All three collections are available for $44.95 (three cassettes) and $53.95 (three diskettes).

Volume 2
Collection #1: Chapter 1 - Linear, polynomial, multidimensional, parametric least squares.
Collection #2: Chapter 2 - Series approximation techniques (economization, inversion, reversion, shifting, etc.)
Collection #3: Chapter 3 - Functional approximations by iteration and recursion.
Collection #4: Chapter 4 - CORDIC approximations to trigonometric, hyperbolic, exponential and logarithmic functions.
Collection #5: Chapter 5 - Table interpolation, differentiation and integration (Newton, LaGrange, splines).
Collection #6: Chapter 6 - Methods for finding the real roots of functions.
Collection #7: Chapter 7 - Methods for finding the complex roots of functions.
Collection #8: Chapter 8 - Optimization by steepest descent.
Price per collection: $14.95 Cassette/$18.95 Diskette

All eight collections are available for $99.95 (eight cassettes) and $129.95 (eight diskettes).

Because the texts are a vital part of the documentation, *BASIC SCIENTIFIC SUBROUTINES, Volumes 1 and 2* are available from DYNACOMP:

BASIC SCIENTIFIC SUBROUTINES, Vol. 1 (319 pages): $19.95 + 75¢ postage
BASIC SCIENTIFIC SUBROUTINES, Vol. 2 (790 pages): $23.95 + $1.50 postage

See reviews in KILOBAUD, Dr. Dobbs, and ACCESS.

NEW

SOFTNET (Apple II and TRS-80 48K diskette only) — Price: $129.95

SOFTNET may be used to create models of liquid pipeline systems to evaluate their flow performance. Up to 150 nodes with up to 150 connecting elements may be simulated, and models may be combined to form yet larger models. If you are involved in water distribution systems, chemical fluid flow problems, building plumbing, or similar situations, this is an ideal analysis tool.

NEW

FILTER ANALYSIS (Apple only) — Price: $19.95 Cassette/$23.95 Diskette

FILTER ANALYSIS is the ideal program for determining the frequency response of passive filters. Any number of RLC components may be included, and any number of poles treated. FILTER ANALYSIS features its own mini-language which makes circuit description simple. Results may be printed in tabular form or plotted in HIRES graphics (decibels versus log-frequency).

ACTIVE CIRCUIT ANALYSIS (Available for all computers) — Price: $35.95 Cassette/$39.95 Diskette

With ACAP you may analyze the response of an active or passive component circuit. The circuit may be probed at equal steps in frequency, and the resulting complex voltages at each component juncture examined; the frequency response of a filter or amplifier may be completely determined with respect to both amplitude and phase. In addition, ACAP prints a statistical analysis of the range of voltage responses which result from tolerance variations in the components. ACAP is easy to learn and use. Circuit descriptions may be saved onto cassette or diskette to be recalled at a later time for execution or editing. ACAP should be part of every circuit designer's program library. Requires 48K.

LOGIC SIMULATOR (Available for all computers) — Price: $35.95 Cassette/$39.95 Diskette

Test your complicated digital logic design with respect to given set of inputs to determine how well the circuit will operate. The elements which may be simulated include multiple input AND, OR, NOR, EXOR, EXNOR and NAND gates, as well as inverters, J-K and D flip-flops, and one-shots. Inputs may be clocked in with varying clock cycle lengths/displacements and delays may be introduced to probe for glitches and race conditions. A timing diagram for any given set of nodes may be plotted. Save your breadboarding until the circuit is checked by LOGIC SIMULATOR. Requires 48K.

NEW

BEAM DEFLECTION (Available for all computers) (disk/diskette only) — Price: $29.95 Diskette/$32.45 Disk

BEAM DEFLECTION Is the first in DYNACOMP's new series of structural analysis software packages. It consists of two programs. The first program permits the development of data files which describe the problem. For example, the ends of the beam may be pinned, clamped or free. The beam may be uniformly supported by an elastic bed, or held up by springs variously placed and having differing spring constants. The elasticity and cross section of the beam may vary along its length. The load may be uniformly distributed or it may include discrete forces. The beam may be pinned at various points along its length. And so on. All this information may be easily entered and edited using the data input program.

Following this the analysis program is called. The calculated results are the stress and deflections of the beam, both in numerical and graphical form. since the input data is saved, cases may be easily re-run with modification, thereby permitting iterative design.

The documentation which comes with BEAM DEFLECTION clearly shows how to use the software. In addition, three text problems are described and demonstrated to ensure that you understand how to use the program. Also, helpful theoretical information is supplied in the appendix.

STATTEST (Not available on Atari cassette or for PET/CBM) — Price: $19.95 Cassette/$23.95 Diskette

This is a statistical inference package which helps you make wise decisions in the face of uncertainty. In an interactive fashion you can build and edit data files and test the differences in means, variances and proportions. STATTEST will also perform data analysis as well as do linear correlation and regression. This menu-directed statistical workhorse is rounded out with a chi-square contingency test and a (uniform and normal) random sample generator. The documentation is written by a college professor who guides you through the various tests.

ABOUT DYNACOMP

DYNACOMP is a leading distributor of small system software with sales spanning the world (currently in excess of 50 countries). During the past three years we have greatly enlarged the DYNACOMP product line, but have maintained and improved our high level of quality and customer support. The achievement in quality is apparent from our many repeat customers and the software reviews in such publications as COMPUTRONICS, 80 Software Critique, A.N.A.L.O.G., Softalk, Creative Computing and Kilobaud. DYNACOMP software has also been chosen for demonstration on network television. Our customer support is as close as your phone; it is always friendly. The staff is highly trained and always willing to discuss products or give advice.

No matter which language is being taught, in introducing programming to students we generally use a few simple commands, control structures for iteration, conditionals, branching and modules, and only the simplest (if any) data structures. These are the elements we would expect to make up a simplified language.

The third activity of programming is *testing* and *debugging*. Beginners often suffer a great deal of frustration in finding and correcting errors. Some programming languages facilitate debugging by such things as catching syntax errors as the program is entered and allowing the program to be run one step at a time so it can be analyzed carefully. Languages designed to introduce programming should contain such debugging aids.

Kidstuff and *Karel the Robot* provide simple languages so students can learn programming fundamentals with a minimal amount of frustration and delay. They can help students master recommended principles of program design while making the coding and debugging stages as painless as possible.

Kidstuff

Kidstuff, by Thomas R. Smith, is suitable for children as young as first or second grades. It is also appropriate as an easy introduction to programming for older children. *Kidstuff* operates on PET computers, and a version for Commodore 64 computers is being developed.

The commands of the *Kidstuff* language let children write programs to create pictures on the computer screen and play music. The language itself is a mix of turtle graphics-like commands (e.g., DF for draw forward, TR for turn right), modules like those in Logo, branches and loops similar to those of BASIC, a command to use any of the PET graphics symbols, a music command, and special features to aid debugging.

This sounds like a mish-mash, but it has been blended into a coherent teaching tool. A particularly good feature of the *Kidstuff* package is the manual, which contains 13 tutorial lessons, demonstration programs, and suggested projects. The *Kidstuff* language and manual make it possible for all teachers to introduce programming to their students. The manual can be an extremely valuable aid for teachers who are not themselves knowledgeable about programming.

The commands of the *Kidstuff* language are:

DF – draw forward
JF – jump forward without drawing
TR – turn right 90 degrees
TL – turn left 90 degrees
P – select a symbol for drawing (any letter, number, or PET graphics symbol can be used)
B# – play a note of a specified pitch and duration

There are also two control elements similar to GOTO branches and FOR/NEXT loops in BASIC, as well as two simple variables, X and Y.

In addition, *Kidstuff* lets you "teach" the computer new commands. For example, you can tell the computer how to draw a square of size 5:

```
TO SQUARE
BL      [begin a loop]
DF5     [draw forward 5 steps]
TR      [turn right 90 degrees]
RL4     [repeat the loop 4 times]
```

Once this is entered, SQUARE can be used just like any of the built-in commands. This capability, similar to the use of procedures in Logo, encourages modular programming.

Kidstuff has several features to facilitate debugging. First, syntax errors are caught as the program is entered, and friendly, clear error messages are given. It's much easier for children to deal with an error message which says "OOPS! THE COMPUTER DOESN'T UNDERSTAND" or "OOPS, LINE NUMBER ERROR" than messages such as "SYNTAX ERROR" or "ERROR 112" found in other languages.

Also, *Kidstuff* has a WALK option which tells the computer to follow the instructions in the program one at a time. When walking, the computer displays an instruction, follows it, and then waits for the child to press the SPACE BAR before going on to the next instruction. This option, similar to TRACE or STEP options in some versions of other languages, is very valuable for helping children analyze their programs and find bugs.

This simple language (I have described all of it) can introduce children to most of the fundamental concepts of programming. The only main concept missing is that there are no conditional (IF-THEN) commands.

Kidstuff is not a powerful language. It is very limited in the number of variables, loops, and new commands possible. However, these limits do not distract from its intended purpose. Once children find the limits of *Kidstuff* constraining, they are ready to go on to learn BASIC or Logo. Having mastered *Kidstuff* first, they will find it easier to learn other languages.

Karel The Robot

Karel the Robot, by Richard E. Pattis, is designed for high school and college students. It teaches concepts of structured programming and can serve as an excellent bridge to learning Pascal, a language now taught in many colleges and universities and becoming increasingly popular in high schools. There is a book about *Karel the Robot*'s language and a "simulator" for Apple II computers that lets you explore the language.

Karel the Robot's world consists of a grid of streets and avenues, walls which block Karel's paths, and beepers which Karel can pick up, carry, and place on street corners. Karel, like all well-behaved robots, obeys simple commands. These are:

MOVE – go forward 1 block
TURNLEFT – pivot 90 degrees to the left
PICKBEEPER – pick up a beeper
PUTBEEPER – put a beeper on a corner
TURNOFF – end the program

In addition, Karel's language contains control elements for repeating instructions (the Pascal ITERATE-TIMES and WHILE-DO commands), conditional tests (IF-THEN-ELSE), and grouping instructions into blocks (BEGIN/END). It also lets you define new instructions. These are some of the most important elements of Pascal, and Karel's language also uses Pascal-like syntax.

The *Karel the Robot* book contains six chapters which present Karel's world and language with example programs, suggested problems, and valuable information about good programming practices. The book is very well done and can be used without the simulator. However, the simulator adds a great deal.

The Karel simulator has a number of excellent features. After you enter your program, it is checked for syntax errors, and useful diagnostic messages are given. There is even a spelling check routine – for example, if you type "MIVE" the computer will display a message saying "I ASSUME YOU MEAN MOVE". Once your program is syntactically correct, you can create the world in which you want Karel to run your program.

You assign Karel a starting location and specify the locations of walls and beepers. You then have many options as to how to run the program. For example, you can select high, medium, or low speed, and you can have Karel leave a trail as he moves. Karel's world is stark, with Karel looking like an arrowhead on the screen. But watching Karel move clearly shows how your program operates.

A "monitor mode" option is an extremely valuable learning aid. In this mode, you can control exactly how Karel proceeds through your program. You can tell Karel how many steps to execute; he does so and then pauses for your next command. At any point you can tell Karel to run the program in reverse, display each command as it is executed, run the program until reaching a specified command, and use other options which make it easy to analyze and debug programs.

Karel the Robot is a well-designed, gentle introduction to programming, as well as a solid stepping-stone for people interested in learning Pascal.

Course disks, which contain solutions to all the problems in the Karel book, are available for $150. The *Karel the Robot* book, published by John Wiley and Sons, is also available separately.

Kidstuff
Thomas R. Smith
P.O. Box 345
Dedham, MA 02026
$59.95 cassette
$69.95 disk (including backup)

Karel The Robot
Cybertronics
999 Mount Kemble Ave.
Morristown, NJ 07960
$85

©

Susan Semancik & C. Marshall Curtis

Developing A Communications Program

This is Part 3 of a series of columns to help the handicapped communicate. The program is for the Apple, PET, and VIC.

The following outline shows the progress we've made so far in developing a program to help non-verbal, motor impaired individuals in their communication needs:

I. Introduction (**COMPUTE!**, April 1982)
II. Menu setup (**COMPUTE!**, June 1982)
III. Selection process
  A. Menu selection
    1. Menu storage
      a. Subscripted variables
      b. DATA lists
      c. Screen values
      d. Peripheral devices
      e. Memory blocks
    2. Multiple menus
      a. Access
      b. Display

Selection Of Menu

Example 9 uses subscripted variables and adds lines to the programs of Example 8 (June, 1982) in order to display a chosen entry of the menu at the top of the screen. The user indicates an entry by giving its menu row number and column number (menu row numbers start at one and increase from top to bottom, and menu column numbers start at one and increase from left to right).

In general, the following changes to Example 8 will produce Example 9: change lines 5, 25, 70, and 130: and add lines 300, 310, and 360. Also, the menu is changed so it will fit on all three computers. Example 9 will work on the VIC if W is changed to 22 in line 20. The Apple computer requires the following changes.

```
10 TEXT: HOME: REM CLEAR TEXT SCREEN
75 P=S(C) + TP
95 IF BR=0 THEN TP=TP + W:IF TP>39 THEN
   TP=0:REM UPDATE TAB IF LINE ENDS WITH
   NO LF
300 VTAB 2: INPUT"ROW #, COLUMN #? "
   ;RN,CN: REM INPUT ON 2ND LINE
310 VTAB 1: PRINT M$(RN,CN): REM DESIRED
   MENU ENTRY PRINTED ON TOP LINE
```

Since most home computers use memory-mapped video, when you PRINT characters on the computer's display screen, these characters are stored as screen values in a block of memory. Usually, changing the values within this block of memory will change the screen contents faster than using PRINT commands will.

If you PRINT a menu to the screen from DATA statements, as is done in Example 9, you actually have the menu stored in memory twice: once within the program storage area of memory, and once in the video-mapped area of memory. If you also save the menu using subscripted variables, then you've increased the memory areas to three, since it is now also stored in the variables-storage area of memory. This can be disastrous if your computer has only a small amount of memory and/or your program or menu is large!

Entry Selection

We can eliminate subscripted variables by using the RESTORE statement to pick the words out from the program's DATA statements when we need them. This is implemented for the PET and VIC computers in Example 10, which lists the changes to be made to Example 9. Make the same changes to the Apple version of Example 9, along with the following change:

```
350 READ M$: VTAB 1: PRINT M$: REM DESIRED
   MENU ENTRY PRINTED ON TOP LINE
```

This program also allows an entry to be selected by its menu row and column numbers and displayed at the top of the screen. The advantage here is that no extra memory is required to store the entries as would be needed by using subscripted variables. The disadvantage is that no other DATA statements can be read in the program without careful checking on where the RESTORE and rereading have left the DATA pointers.

Note that previous non-menu DATA entries are bypassed in line 310 of Example 10. If there had been any non-menu DATA entries needed after the menu selection, we would have had to read through the rest of the menu to get to the right DATA statement after it. (Some computers don't have this problem, since their extended BASIC allows restoration to a particular DATA statement.)

An alternative to the use of the RESTORE statement and its possible DATA pointer problem is to pick the selected menu entries from the video-mapped area of memory. This will, however, cause loss of program mobility between different home computers, since this area of memory is not a standardized location. This can be seen by examining the differences between the versions of Example 11, which uses the PEEK statement to pick up the screen values of the selected entry, and uses the POKE statement to display the entry at the top of the screen. In particular, the VIC needs to add the following lines, the last of which is used to set the color register for text to be visible when POKEing the top line of the screen:

```
130 SP=7680: P= SP+(SR-1)*W: GOTO 300
312 CL=PEEK(646): FOR I=38400 TO 38422: POKE
    I,CL: NEXT I
```

The Apple also needs to add the following lines, the last of which is used to help account for the non-linear mapping of the screen:

```
130 SP=1024: GOTO 300
312 R=SR+RN-1+(RN-1)*BR: REM R=SCREEN
    ROW #
315 P1=SP+128*(R-1)-984*INT(R/8)+980*INT(R/24)
```

The DATA statements in Example 11 are used only once to initially display the menu. We should be able to save this memory space by eliminating the DATA statements and entering the menu directly to the screen from peripherals, such as tape recorders or disk units. This concept will be further explored in our next article.

Example 9: For the PET computer – displays a menu by rows from DATA statements, and uses subscripted variables to allow a user to select by menu row and column numbers an entry for display at the top of the screen.

```
5 REM EXAMPLE 9A) PET COMPUTER
10 PRINT CHR$(147);:REM CLEAR TEXT SCREEN
20 W=40:RM=6:BR=1:CM=4:BC=1:RI=2:SR=3:SC=1:RE
   M SET MENU PARAMETERS
25 DIM S(CM),L(CM),M$(RM,CM):S(1)=SC
30 DATA 3,3,5,8:REM COLUMN WIDTHS
35 IF CM=1 THEN 50
38 REM CALCULATE STARTING POSITION OF EACH CO
   LUMN
40 FOR I=2 TO CM:READ L(I-1):S(I)=S(I-1)+L(I-
   1)+BC:NEXT I:READ L(CM)
50 IF SR=1 THEN 70
60 FOR X=1 TO SR-1:PRINT:NEXT X:REM POSITION ~
   CURSOR TO 1ST ROW OF MENU
65 LP=S(CM)+L(CM)-1:IF LP>W THEN 200
70 TP=0:FOR R=1 TO RM:FOR C=1 TO CM:READ M$:M
   $(R,C)=M$
75 P=S(C)-1+TP
80 PRINT TAB(P);M$;:NEXT C
90 IF S(CM)+LEN(M$)-1<W THEN PRINT:TP=0:GOTO1
   00:REM WRAPAROUND ADVANCES A LINE
95 IF BR=0 THEN TP=TP+W:IF TP>87 THEN TP=0:RE
   M UPDATE TAB IF LINE ENDS W/NO LF
100 IF BR=0 THEN 120
110 FOR B=1 TO BR:PRINT:NEXT B:REM SKIP BLANK ~
    ROWS BETWN COLUMN ENTRIES
120 NEXT R
130 GOTO 300
139 REM ENTER DATA BY ROWS
140 DATA DR.,IS,COLD,INGEDS12
145 DATA I,AM,WHEN," AOTFR34"
150 DATA YOU,ARE,DRINK,.ULHCP56
155 DATA MOM,EAT,WANT,?MYWKB78
160 DATA DAD,NO,TIME,",VJQZX90"
165 DATA HOT,YES,SLEEP,";$%()'+-"
200 PRINT "MENU SIZE ERROR!":END
300 PRINT CHR$(19):INPUT "ROW #, COLUMN #"; RN
    ,CN:REM INPUT ON 2ND LINE
310 PRINT CHR$(19);M$(RN,CN):REM DESIRED MENU ~
    ENTRY PRINTED ON TOP LINE
360 GOTO 360:REM DISPLAY ISN'T DISTURBED UNTIL
     USER BREAKS PROGRAM
```

Example 10: For the PET computer – changes to Ex. 9, so RESTORE can be used instead of subscripted variables.

```
25 DIM S(CM),L(CM):S(1)=SC
70 TP=0:FOR R=1 TO RM:FOR C=1 TO CM:READ M$
310 RESTORE:FOR I=1 TO CM:READ N:NEXT I:REM BY
    PASS PREVIOUS DATA
315 IF RN=1 THEN 330:REM BYPASS PREVIOUS MENU ~
    ROWS
320 FOR R=1 TO RN-1:FOR C=1 TO CM:READ M$:NEXT
     C:NEXT R
330 IF CN=1 THEN 350:REM BYPASS PREVIOUS COLUM
    N ENTRIES
340 FOR C=1 TO CN-1:READ M$:NEXT C
350 READ M$:PRINT CHR$(19);M$:REM DESIRED MENU
     ENTRY PRINTED ON TOP LINE
```

Example 11: For the PET computer – changes to Ex. 10, so PEEK and POKE can be used instead of RESTORE.

```
129 REM SP=STARTING MEMORY AREA FOR SCREEN, P=
    STARTING SCREEN POSITION FOR MENU
130 SP=32768:P=SP+(SR-1)*W:GOTO 300
310 REM P1=STARTING SCREEN POSITION FOR DESIRE
    D ENTRY
315 P1=P+(RN-1)*W+(RN-1)*BR*W
320 P1=P1+S(CN)-1
330 REM P2=ENDING SCREEN POSITION FOR DESIRED ~
    ENTRY
340 P2=P1+L(CN)-1
350 J=0:FOR I=P1 TO P2:POKE SP+J,PEEK(I):J=J+1
    :NEXT I
```

©

Christmas Bird Count

Jean B. Rogers
Eugene, OR

Personal computers can make any hobby more rewarding. Here's how a PET contributed to the author's bird watching, along with some hints on effective preplanning when writing large programs.

Every year, during a two-week period near Christmas, thousands of bird watchers spend whole days surveying all the birds around them. This event, the annual Audubon Christmas Bird Count, provides large amounts of information about bird populations throughout North and Central America.

The first Christmas Bird Count (CBC) was held on Christmas Day, 1900, when 27 birders noted all the birds they saw during the day. Those birders covered 25 different areas, mostly in cities in the Northeastern USA. CBC's have been held every year since then; currently about 34,000 birders survey nearly 1360 different count areas each year. Results from these CBC's are submitted to the National Audubon Society and are published in its journal, *American Birds.*

In 1979, my teen-aged son and my husband decided to establish a CBC in the area near our home, Port Orford, Oregon. To initiate a CBC, a circular area 15 miles in diameter is chosen, separate from an existing CBC area. This circle is then subdivided into sections, and a group of people is assigned to scour each section, recording every bird identified by sight or sound. Each group tallies the birds according to the number of each species seen. After the count, the number of different species seen by each party of observers and the number of species seen by the total group are counted. For CBC's held in 1979, these totals varied from the Atlantic area of the Panama Canal Zone with 320 species, to Bethel, Alaska, with 4. A reasonable expectation for the Port Orford area is 100 to 120.

Additionally, the observers record the number of individual birds of each species seen on the count. For some species such as Screech Owl, only one individual might be found in the whole count area. Others, like the American Robin or the Common Murre, might be tallied in the thousands. The main data processing task related to a CBC thus is a tabulation of sums of species and individuals seen. A count report including this information is provided for each participant in the count as well as being sent to the National Audubon Society.

Since we wanted an easy-to-read, attractive report, and needed to do some simple numerical calculations, I concluded that this would be a very reasonable task for a microcomputer using BASIC. I had available a PET with 8K of memory and cassette for storage, and a CBM printer. We designed software that worked successfully for the 1979 Port Orford CBC and have used the same programs for CBC's since.

Designing The Project

I think that many amateur programmers have a bigger problem analyzing the project they've undertaken than they do coding it. Thus, I propose to explain how I attacked the problem rather than to provide the BASIC code I used. While the code might be useful to some people with projects very much like mine, the information on problem analysis will possibly be helpful to many people with a wide variety of interests.

The first step in working on the project was to sit down with my son and find out specifically what information he wanted on the output report, as well as approximately what he expected it to look like. The report would essentially consist of a list of the names of birds seen on the count, the number of individuals of that species seen by each of the parties (people assigned to a sub-area of the count circle), and the total seen by the whole group.

In discussing the report, we realized that, with little additional work, we could produce a field form for use on the count. This is a recording sheet listing the birds one might expect to see, with spaces for tallying the number of individuals of each species seen. Each party has one person designated as recorder who keeps track of the tallies. On the field form and on the final report, birds are listed in phylogenetic order. This is a standard order based on evolutionary progression, and is used in field guides, ornithological research, and scientific documents.

So the overall task was divided down into subtasks: build a bird list, tabulate the results of the count, print the report. A basic list of the birds one might expect to see in our area, then, was the first thing we would need. Having this list on a separate file stored on a tape would make it easily available for whatever future need we had of it.

The Master Bird List

The program to build the list and store it on the

YOU ARE GOING TO SMILE!

--When you see over 2000 books, programs, and accessories we carry for All Major Brands in our Computer 'SOURCE BOOK©'. From introductory level to professional, we cover it all. Listed Below is just a sample of what we carry.
DEALER INQUIRIES WELCOME

INTRODUCTORY BOOKS

Introduction to Word Processing. Plain language book for those considering a purchase.
4795-000076 300 pgs,140 illus $12.95

How to Buy the Right Small Business Computer. Easy to follow book with 'checklist'.
4925-008494 225 pgs, $8.95

Business System Buyers Guide. This guide cuts thru the jargon and simplifies your task.
4665-000047 166 pgs $7.95

Using Micros in Business. Essential background briefing for any buyer of hard.or software
4410-005152 192 pgs $9.95

Your Home Computer. Meant for the person who has no technical background,-buying tips!
4250-000022 211 pgs $12.95

Home Computers: Beginners Glossery & Guide. Terminology,number systems,symbols
4250-000002 147 pgs, 20 illus. $10.95

Why do YOU need a Personal Computer? Learn the 100's of everyday applications.
4925-004784 320 pages $8.95

Computer Dictionary & Handbook. More than 22,000 definitions,acronyms,abbreviations.
4760-021632 928 pages $34.95

Computer Dictionary. More than 12,000 definitions clear up almost any question you have.
4760-021652 624 pages, $15.95

How to Computerize Your Small Business. Shows how to plan a tailored computer system
4690-403857 171 pages $7.95

How to make MONEY with Your Mjcro. This book gives you dozens of money making ideas.
4250-000074 154 pages $12.95

Playing the Stock & Bond Markets w/ Micros. Learn the principals of investing in the markets
4800-001251 308 pgs,35 illus $9.95

GENERAL INTEREST BOOKS

Background Math for a Computer World. All the basic math facts,techniques,& concepts
4925-008086 308 pages $7.95

Micro's and the 3R's. This book educates the educator on ways to teach with computers.
4410-005113 192 pages $9.75

Intro to Computer Music. The book explains in detail computer music, non-technical 'how to'.
4925-086839 313 pages $14.95

1001 Things to do with your Personal Comp Time saving,money saving & just plain fun ideas
4800-000160 336 pages $7.95

Computer Graphics Primer. Only book on the market that shows how to do comp. animation!
4760-021650 184 pgs. $14.95

How to Troubleshoot and Repair Micro's. How to locate & repair common problems.
4725-002981 290 pages $7.95

Visicalc™: Home & Office Companion. A compendium of 50 VisiCalc™ models & samples
4665-000050 182 pgs,12 illus $15.99

Wordstar™ Made Easy. In 14 east lessons this handy book will teach you the powerful features
5664-000069 125 pages $11.95

ATARI™ BOOKS

rs030A Guide to Atari™ 400/800 Computers Provides invaluable material for Atari™ users.
4665-000065 458 pages $16.95

PAINT. Create 'paintings' of your own on the 400/800 systems.Both begining and pro levels
4725-005425 $29.95

The Atari™ Assembler. Master assembly language,you will need some BASIC experience.
4690-000236 270 pages $9.95

Atari™ Games & Recreations. Provides beginning or advanced users with games & 'how to'.
4690-000242 252 pgs,225 illus $12.95

Computers For People. This book outlines the benefits and uses of Atari™ 400 or

Computers for People. This book outlines the benefits and uses for Atari™ 400 & 800 systems
4250-000064 200 pgs, 47 illus. $7.95

Atari™ Basic. A fine intro to BASIC programming. Turn from novice to pro in week or so.
4925-006496 $8.95

Atari™ Sound & Graphics. This is a self-paced book to fully explore the expressive potentials.
4925-009593 160 pages $9.95

The VisiCalc™ Book: Atari™ Edition. Make the most of VisiCalc™ capabilities on the Atari™
4725-008393 400 pages $14.95

Inside Atari™ DOS. Step by Step guide to the DOS software. Complete with source code.
4105-000002 120 pges $19.95

GOT SOMETHING YOU WANT TO SELL?

JMC is always looking for new items to add to our Computer 'Source Book™'. If you have a book, or software for one of the popular personal/home computer systems please write to us with details. Perhaps we can add it to the thousands of items we are currently selling to computer retailers all over the USA

If you have a product you would like to market -but don't want all the hassle of packaging, promoting,and producing the product---let us know,perhaps we can work out a generous Royalty arrangement with you.

TRS-80™ COLOR COMPUTER BOOKS

55 Color Computer Programs for the Home, School & Office. Includes lots of color graphics
4080-000005 128 pgs $9.95

55 More Color Computer Programs. Companion volume to above,more useful programs.
4080-000008 112 pgs $9.95

Love Poems from the Digital Heart of an electronic Computer, An ideal Valentine!
4080-000013 96 pgs $4.95

TRS-80™ Color Computer Graphics. Comprehensive coverage of color and graphics avail.
4725-007864 $14.95

Color Computer Graphics. A must have book for owners of the TRS-80™ Color Computer.
4080-000012 128 pgs $9.95

Color Computer Songbook. Book of favorite songs and classics to be played on Color Comp.
4080-000011 . 96 pgs $7.95

TRS-80™ Color Computer Applications. A beginners handbook loaded with examples.
4690-007870 240 pgs $12.95

TRS-80™ Color BASIC. Self-teaching guide leads novices step by step into programming

TRS-80™Color BASIC. Self-teaching guide leads novices step by step into programming.
4925-009644 256 pgs $9.95

TRS-80™ Color Programs. 37 programs are fully documented in this easy to follow book.
4250-000061 323 pgs $19.95

101 Color Computer Programming Tips & Tricks. Practical,useful techniques & shortcuts
4080-000007 128 pgs $7.95

LANGUAGE BOOKS

Introduction to BASIC. Extremely popular introduction to BASIC,emphasis on fundamentals
4515-000022 206 pgs,20 illus $10.95

50 More Programs in Basic for the Home, School & Office. All fresh programs in this vol.
4080-000003 96 pgs $9.95

BASIC Programming Primer. User-friendly fundamentals book for both novices and pros.
4760-021586 240 pgs $11.95

Sixty Challenging Problems with BASIC solution. Learn more by solving problems.
4410-005180 144 pgs $8.50

Beat the Odds. Micro simulation of Casino Games. Painless way to gamble your money
4410-005181 128 pgs $8.70

Introduction to Texas Instrument BASIC. A must for any TI™ owner.
4410-005185 320 pgs $11.95

BASIC for Home Computers. A sensible and complete beginning guide to BASIC.
4925-003204 336 pgs $9.50

BASIC,Vol II. Picks up where above book left off,needs no special math or science knowledge
4925-003500 325 pgs $9.95

Inside BASIC games. Teaches the reader how to design error-free interative games,etc.
4795-000055 350 pgs,120 illus $14.95

BASIC Programs for Home Financial Management. 33 Useful programs for you.
4690-066514 250 pgs. illus $12.95

BASIC-BASIC-English Dictionary. This book will help you translate between BASIC dialects
4250-000054 150 pgs $10.95

Basic Programming Worksheets. Printed worksheet forms,40 sheets to pad. 80 columns
4080-000802 For Color Computer $2.95
4080-000801 For Pocket BASIC $2.95
4080-000803 For Apple™ $2.95
4080-000804 For IBM™ PC $2.95
4080-000805 Universal BASIC form $2.95

BASIC for Business. Shows how easy it is to use a computer for general business tasks.
4795-000080 250 pgs $12.95

A to Z Book of Computer Games. Here are 26 exciting and instructive BASIC game programs
4800-001062 308 pgs,73 illus. $7.95

40 Computer Games. Unbeatable game collection from Kilobaud Micro Magazine.
4910-000023 $7.95

The BASIC conversions Handbook For Pet, Apple™,TRS-80™ users. Guide to converting.
4410-005534 80 pgs $7.95

SINCLAIR ZX-81™ BOOKS

Programming the ZX-81 for Real Applications. Practical programs to show ability of ZX-81
4250-000090 166 pages $9.95

Making the Most of Your ZX-81. Information on how to write programs games,tricks,etc.
4725-004188 128 pages $10.95

The ZX-81 Pocket Book. Handy programming manual that gives you everything you need.
4725-009524 128 pages $9.95

Mastering Machine Code on Your ZX-81. First simple english guide to machine code.
4725-004261 180 pages $12.95

49 Explosive Games for the ZX-81. Complete easy to read game rules and program instruct.
4725-002086 140 pages $10.95

NEW

NEW Vic 20™ GAMES

Direct from England, JMC is proud to announce that we have been appointed the exclusive U.S. importer for the following Vic-20 games.

MINEFIELD,A game for the 5K Vic 20. For Ages 6 and over. Your task is to drive an ambulance around a battle ground and pick up injured soldiers--racing the clock,and avoiding touching off the buried mines. You can adjust the complexity of the game by choosing to play against 25 to 100 mines.
4002-000001 $15.98

MAZE OF DEATH. For 5K Vic-20. For ages 8 and over. You have to travel across the screen thru a maze of walls blocking your path. Unseen hobgobblens and monsters may capture you before you reach your safe haven-on the other hand you may find buried treasure to add to your fortune. Full details scroll on the screen when you start game. 7 skill levels and 10 sensitivity levels to choose from.
4002-000002 $15.98

SPLOTTER© For 5K Vic-20. For ages 8 and over. You control a worm type animal that moves over the screen to gobble up the ugle spots--a race against time. If you bump into a wall, or turn back on yourself to quickly--you LOSE. You choose skill level from #1 (hardest) to #20 (easiest). If you really want to drive someone crazy, buy him this game!
4002-000003 $15.98

GUZZLER© For 5K Vic-20. For ages 8 and over. You must pilot you space shuttle thru a maze of asteroids to your mothership. But if you are not careful you will awaken the Guzzler©'s space bouys. You have to reach your mothership before running out of fuel, GOOD LUCK.
4002-000004 $15.98

DELUX 6-PAK© For 5K Vic-20. Here is your chance to buy **6 games** on one cassette for the price of one game! You get all six of the following games for ONE low price!
War. You are defending against waves of enemy tanks. 3 levels of play...we bet you can't win at level 3!
Smashout. This version of bricks game tests skill,not reaction time,plan ahead!
Blackjack Popular card game '21',if you practice long enough maybe you can plan a system to beat Las Vegas!
Logic: You have 12 tries to break the code of 5 balls in random color sequence.
Pickupgame. You have to remove logs from the pile without disturbing any other logs.
Alarm Clock: Use the Computer as a REAL 24 hour alarm clock, bet you can't ignore its call when the alarm goes off!
4003-000001 $19.98

ANT RAIDERS© For 5K Vic-20. Ages 8 and up. An army of ants is marching its way across the land leading to your home. The land is littered with rocks which make the ants change direction often. You must shoot the ants before one gets to your door. If you shoot a rock it will divide into more rocks. If you trap all the ants, then more will come until you trap all the group, then more will come until...well you get the idea...have FUN!
4003-000003 $15.98

MORE GAMES COMING! We are working on more Vic-20™ games which we will be importing in the near future...watch our ads for details!

VIC-20™ BOOKS

Understanding Your Vic.Vol 1. Beginners guide to programing on the Vic-20™. Tips & tech.
4840-000003 148 pages $11.95
4840-000004 Cassette Supplement $7.95

Compute!'s First Book of Vic. Collection of the best articles from Compute! Magazine.
4105-000007 Fall 1982 $12.95

Starting with BASIC on the Vic-20™.Helps novices learn all the capablities of the Vic-20
4725-007070 128 pgs,due Fall82 $12.95

IBM™BOOKS

IBM's™ Personal Computer. Background on the giants entry into field,with evaluations.
4700-000111 303 pages $14.95

Using the IBM™ Personal Computer. A complete handbook showing what it can do.
4725-008183 300 pages $12.95

CONFUSED? No one book may offer all the answers,we suggest buying more than one book on a subject to get as many different views and opinions as possible. Books are cheap if you consider the frustrations they can avert.

APPLE™ BOOKS

Apple II™ Users Guide. This guide is the key to unlocking the full power of your Apple II.
4665-000046 385 pgs $16.95

APPLE™! Collection of BASIC programs that have been converted to run on the Apple II.
4665-000068 200 pages. $14.99

Assembly Lanquage Programming for the Apple II™. Comprehensive introduction.
4665-000051 $12.95

Apple™ Machine Language. A machine language book that's fun and entertaining.
4690-000230 296 pages/ illus. $12.95

Apple™ Interfacing. Helps you to interface your Apple™ to a variety of electronic devices.
4760-021862 208 pgs $10.95

Apple™ Programming Exercies. 17 Easy to read,easy to use exercies including music.
4925-086598 192 pages $9.95

Computer 'Source Book™'. Over 2000 books programs,and accessories for all micro brands
0001-198207 72 pgs $2.00 ppd.

FREE

--With ANY order of $50.00 or more you can receive a 1983 Computer Calendar FREE,--A $ 7.95 VALUE!

COUPON

QTY	NUMBER	DESCRIPTION OF ITEM	COST	TOTAL

(Orders Under $25 Please Add $2 50 Post & Handling -- Illinois Residents Please Add 5% Sales Tax ALL CATALOGS SHIPPED POST PAID) TOTAL

PAYMENT ENCLOSED: ☐CASH ☐CHECK ☐MONEY ORDER

PLEASE CHARGE TO MY ☐MASTERCARD ☐VISA (Min Chg $25)

CARD NUMBER________________

EXPIRES________ INTRBNK #________

SHIP TO________________

STREET ADDR________________

CITY________ STATE________ ZIP________

PRICES AND AVAILABILITY SUBJECT TO CHANGE

DEPARTMENT: G-8
1025 INDUSTRIAL DR., BENSENVILLE IL 60106-1297

ATARI ™

800 48K . . $619

400 16K . . $259

410 Recorder	$ 75
810 Disk Drive	$419
825 Printer	$579
830 Modem	$155
850 Interface	$165
481 Entertainer	$ 79
482 Educator	$119
483 Programmer	$ 55
484 Communicator	$299
853 16K Ram	$ 75
The Bookkeeper Kit	$169

ATARI Software

CX4104 Mailing List	$ 19
CX404 Word Processor	$105
CXL4007 Music Composer	$ 45
Programming 2 & 3	$ 22
Conversational Languages	$ 45
CX4018 Pilot	$ 59
CX405 Pilot	$ 99
CXL4003 Assembler Editor	$ 45
CX8126 Microsoft Basic	$ 67
CXL4022 Pac-Man	$ 33
CX8130 Caverns of Mars	$ 29
CXL4020 Centipede	$ 33
CXL4006 Super Breakout	$ 28
CXL4008 Space Invaders	$ 28
CXL4009 Computer Chess	$ 28
CXL4011 Star Raiders	$ 33
CXL4012 Missile Command	$ 28
CXL4013 Asteroids	$ 28
The Bookeeper	$105
Home Filing Manager	$ 36
Atari Speed Reading	$ 54
My First Alphabet	$ 26

Business & Utilities

Visicalc	$169
Mail Merge	$ 20
Data Perfect	$ 75
Letter Perfect	$105
Text Wizard	$ 65
Disk Detective	$ 20
Datasm 65 2.0	$ 59
File Manager 800 +	$ 65
Syn Assembler	$ 34
Page 6	$ 20
Atari World	$ 39
K-Dos	$ 59
Micropainter	$ 23
Color Print	$ 27
Lisp Interpreter	$ 79
Bishops Square	$ 20
Graphic Master	$ 27
Graphic Generator	$ 17
Basic Compiler	$ 65

Programming Techniques

Display Lists	$ 17
Horiz/Vert Scroll	$ 17
Page Flipping	$ 17
Basics of Animation	$ 17
Player Missile Graphics	$ 24
Sound	$ 17
Data Files	$ 24

For Fast Delivery, send certified or cashier checks, money orders, or direct bank wire transfers. Personal checks allow 2 to 3 weeks to clear. Prices reflect a cash discount only and are subject to change. Shipping—Software ($2.00 Minimum). Hardware—call. Foreign inquiries invited — add 15% for shipping. Nevada residents add sales tax.

NEW LOWER PRICES

TOP SELLERS

Atari

Temple of Apshai	$ 27
Raster Blaster	$ 20
Apple Panic	$ 20
Crossfire	$ 20
Threshold	$ 27
Mousekattack	$ 23
Krazy Shootout	$ 34
Deadline	$ 34
Tumble Bugs	$ 20
Pool 1.5	$ 23
Crypts of Terror	$ 23
Richochet	$ 15
Empire of the Overmind	$ 23
Tanktics	$ 20
Match Racers	$ 20
Wiz & Princess	$ 22
Mission: Asteroid	$ 17
Ali Baba & the Forty Thieves	$ 22
The Shattered Alliance	$ 27
Bug Attack	$ 20
Canyon Climber	$ 20
Shooting Arcade	$ 20
Pacific Coast Highway	$ 20
Clowns & Balloons	$ 20
Ghost Hunter	$ 23
Preppie	$ 20
Rear Guard	$ 17
Lunar Lander	$ 17
War	$ 17
Star Warrior	$ 27
Invasion Orion	$ 17
Dragon's Eye	$ 20
Crush, Crumble & Chomp	$ 20
Jawbreaker	$ 20
Pathfinder	$ 23
Zork I	$ 27
Zork II	$ 27
Action Quest	$ 20
Softporn Adventure	$ 20
Deluxe Invaders	$ 23
Protector	$ 23
Dodge Racer	$ 23
Chicken	$ 23
Nautilus	$ 23
Alien Hell	$ 15
Mar Tesoro	$ 17
Galactic Chase	$ 20
Alien Swarm	$ 23
Intruder	$ 23
Lords of Karma	$ 15
B-1 Nuclear Bomber	$ 12
Rescue at Rigel	$ 20

COMPUTER OUTLET

*** SPECIALS OF THE MONTH ***

ELEPHANT DISKS (BOX)	$ 22
HAYES SMARTMODEM	$209
MOSAIC 32K RAM	$ 99
RAMDISK (128K)	$429
AMDEK COLOR I MONITOR	$309
PERCOM DOUBLE DENSITY DRIVE	$639
NEC 8023A PRINTER	$479
BASIC A +	$ 59
FLIP N' SORT DISKETTE BOX (Holds 50 Diskettes)	$ 21
FLIP-SORT CARTRIDGE BOX (Holds 10 Atari Computer Cartridges)	$ 21
AXIOM GP-100 GRAPHICS PRINTER	$299
AXIOM IMP-4 GRAPHICS PRINTER	$499
MOSAIC 64K RAM	$179
BIT 80 COLUMN BOARD	$289
ALL APX SOFTWARE	15% TO 20% OFF

Computer Outlet

Park Place — Upper Level
1095 E. Twain — (702) 796-0296
Las Vegas, Nevada 89109

Call Toll Free **800-634-6766**
We accept Major Credit Cards
Mon.-Fri. 8 A.M.-6 P.M.
Sat. 9 A.M.-5 P.M.

NEW Atari

Automated Simulations

Upper Reaches of Apshai (D, C)	$ 15
Curse of Ra (D, C)	$ 15
King Arthur's Heir (D)	$ 20
Escape from Vulcan's Isle (D)	$ 20
Crypt of the Undead (D)	$ 20
The Nightmare (D)	$ 20
Danger in Drindisti (D, C)	$ 15
Armor Assault (D)	$ 27
Monster Maze (CT)	$ 27
Alien Garden (CT)	$ 27
Plattermania (CT)	$ 27

Broderbund

David's Midnight Magic (D)	$ 23
Track Attack (D)	$ 20
Star Blazer (D)	$ 22
Choplifter (D)	$ 23
Deadly Secrets (D)	$ 23
Stellar Shuttle (D, C)	$ 20
Genetic Drift (D, C)	$ 20
Labyrinth (D, C)	$ 20
Serpintine (D)	$ 23
Sea Fox (D)	$ 20

Datasoft

Spell Wizard (D)	$ 53
Sands of Egypt (D)	$ 27
O'Riley's Mine (D, C)	$ 23
Rosen's Brigade (D, C)	$ 23
Fathoms Forth (D)	$ 23

Gebelli

Doctor Goodcode's Cavern (D)	$ 20
Firebird (CT)	$ 34
Embargo (CT)	$ 34

Innovative Design

Pool 400 (CT)	$ 27
Speedway Blast (CT)	$ 27

JV Software

Ghost Encounters (D, C)	$ 20

K-Byte

K-razy Kritters (CT)	$ 34
K-Star Patrol (CT)	$ 34
K-Razy Antiks (CT)	$ 34

L & S Computerware

Crossword Magic (D)	$ 34

Lightning Software

Master Type	$ -

On-Line

Frogger (D)	$ 23
Ulysses and The Golden Fleece (D)	$ 23
Ultima I (D)	$ 27
Ultima II (D)	$ 39

Roklan Corp.

Gorf	(D) $27, (CT) $30
Wizard of Wor	(D) $27, (CT) $30
Anti Sub Patrol	(D) $20, (C) $15

Sentient

Cyborg (D)	$ 23
Gold Rush (D)	$ 23

Sirius

Space Eggs (D)	$ 20
Sneakers (D)	$ 20
Cyclod (D)	$ 20
Snake Byte (D)	$ 20
Bandits (D)	$ 23
Way Out (D)	$ 27
Fast Eddy (CT)	$ 20
Deadly Duck (CT)	$ 20
World War I (CT)	$ 23
Beanie Bopper (CT)	$ 23

Strategic Simulations

Battle of Shiloh (D, C)	$ 27
Tigers in the Snow (D, C)	$ 27
The Cosmic Balance (D)	$ 27

Big Five Software

Miner 2049er (CT)	$ 34

Bram Inc.

Attack at EP-CYG-4	(D) $22, (C) $20

NEW ATARI

Odesta Corporation
- Chess (D) .... $ 45
- Checkers (D) .... $ 34
- Odin (D) .... $ 34

Spectravision
- Nexar (CT) .... $ 24
- Cave In (CT) .... $ 27
- Number Crunch (CT) .... $ 27
- Reaganomics (CT) .... $ 27

Spinnaker
- Snooper Troops #1 (D) .... $ 30
- Snooper Troops #2 (D) .... $ 30
- Story Machine (D) .... $ 23
- Face Maker (D) .... $ 23

Swifty Software
- Haunted Hill .... (D) $20, (C) $ 17
- Trivia Trek (D) .... $ 20
- Datalink (D) .... $ 27
- Space Shuttle (D) .... $ 20
- Jerry White's Music Lessons (D, C) .. $ 20
- Swifty Tach Master .... (D) $20, (C) $ 17

Synapse
- Projector II .... (D) $23, (CT) $ 29
- Chicken .... (D) $23, (CT) $ 29
- Slime .... (D) $23, (CT) $ 29
- Shamus .... (D) $23, (CT) $ 29
- Picknick Paranoia .... (D) $23, (CT) $ 29
- Claim Jumper .... (D) $23, (CT) $ 25
- Acocalypse (D, C) .... $ 23
- Raptillian (D, C) .... $ 23

Tronix
- Kid Grid (D, C) .... $ 20

Milliken Publishing
- Aliencounter (Face Flash (D, C) .... $ 26
- The Jar Game/Chaos (D, C) .... $ 26
- Gulp/Arrow Graphics (D, C) .... $ 26
- Golf Classic/Compubar .... $ 26
- Frenzy/Flip Flop (D, C) .... $ 26
- Battling Bugs/Concentration (D, C) .. $ 26

Thorn EMI
- Submarine Commander (CT) .... $ 34
- Jumbo Jet Pilot (CT) .... $ 34
- Soccer (CT) .... $ 34
- Kickback (CT) .... $ 34
- Darts (C) .... $ 22
- Snooker and Billiards (C) .... $ 22
- Pool (C) .... $ 22
- Dominoes and Cribbage (C) .... $ 22
- Humpty Dumpty and Jack and Jill (C) .... $ 22
- Hickory Dickory Dock and Baa Baa Black Sheep (C) .... $ 22
- British Heritage Jigsaw Puzzles (C) .... $ 22
- European Scene Jigsaw Puzzles (C) . $ 22
- Owari and Bull and Cow (C) .... $ 22

Avalon Hill
- Andromeda Conquest ... (D) $16, (C) $ 13
- GFS Sorceress .... (D) $23, (C) $ 21
- VC (D) .... $ 17
- Legionnaire (C) .... $ 23

Datamost
- Pig Pen (D) .... $ 20

Infocom
- Starcross (D) .... $ 27
- Zork III (D) .... $ 27

In-Home Software
- The Guardian of Gorm ... (D) $23, (C) $ 20
- Sentinel I .... (D) $23, (C) $ 20
- Baseball .... (D) $23, (C) $ 20

FRIENDLY SERVICE

COMMODORE VIC 20

NEW

Creative Software
- Black Hole (CT) .... $ 36
- Trashman (CT) .... $ 36
- Astroblitz (CT) .... $ 36
- City Bomber & Minefield (CT) .... $ 20
- Apple Panic (CT) .... $ 36
- Choplifter (CT) .... $ 36
- Serpentine (CT) .... $ 36
- Videomania (CT) .... $ 36
- Terraguard (CT) .... $ 36

Thorn EMI
- River Rescue (CT) .... $ 29
- VIC Music Composer CT) .... $ 29

Automated Simulations
- Rescue at Rigel (C) .... $ 20
- Ricochet (C) .... $ 15
- Monster Maze (CT) .... $ 27
- Sword of Fargoal .... $ 27

Spectravision
- Cave In (CT) .... $ 27
- Number Crunch (CT) .... $ 27
- Reaganomics (CT) .... $ 27

Tronix
- Galactic Blitz (C) .... $ 17
- Swarm (C) .... $ 20
- Sidewinder (C) .... $ 20

HES Software
- VIC Forth (CT) .... $ 45
- HES Mon (CT) .... $ 29
- Turtle Graphics (CT) .... $ 29
- HES Writer (CT) .... $ 29
- Aggressor (CT) .... $ 29
- Shamus (CT) .... $ 29
- Protector (CT) .... $ 33
- Synthesound (Music Synthesizer) (CT) .... $ 49
- Skier (C) .... $ 15
- Maze of Mikor (C) .... $ 15
- Tank Wars (C) .... $ 15
- Victrek (C) .... $ 15
- Pinball (C) .... $ 13
- Simon (C) .... $ 13
- Fuel Pirates (C) .... $ 13
- Pak Bomber (C) .... $ 13
- Laser Blitz (C) .... $ 15
- Tank Trap (C) .... $ 15
- Concentration (C) $13
- Dam Bomber (C) $13

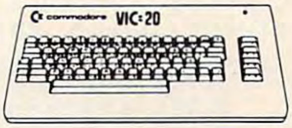
COMPUTER OUTLET

*** SPECIALS OF THE MONTH ***
- SLAGH 24K MEMORY BOARD — VIC 20 .... $ 145
- VERBATIM DISKS (BOX) .... $ 27
- HAYES SMARTMODEM 1200 .... $ 519
- WICO TRACKBALL .... $ 49
- WICO JOYSTICK .... $ 23
- WICO JOYSTICK DELUXE .... $ 26
- WICO FAMOUS RED BALL JOYSTICK .... $ 24
- CARDCO 6 SLOT EXPANSION MOTHER BOARD .... $ 79
- CARDCO 3 SLOT EXPANSION MOTHER BOARD .... $ 39
- CARDRITER LIGHT PEN (VIC 20) .... $ 29
- POINTMASTER JOYSTICK .... $ 15
- USI AMBER MONITOR (12") .... $ 169
- KIDS AND THE VIC (BOOK) .... $ 18
- KIDS AND THE ATARI (BOOK) .... $ 18

Computer Outlet

Park Place — Upper Level
1095 E. Twain — (702) 796-0296
Las Vegas, Nevada 89109

Call Toll Free **800-634-6766**
We accept Major Credit Cards
Mon.-Fri. 8 A.M.-6 P.M.
Sat. 9 A.M.-5 P.M.



commodore

VIC 20 . . . . . $179

- VIC 1530 Datasette .... $ 59
- VIC 1540 Disk Drive $299
- VIC 1525 Graphics printer $309
- VIC 1210 3K Memory Expander .... $ 34
- VIC 1110 8K Memory Expander .... $ 52
- VIC 1111 16K Memory Expander .... $ 92
- VIC 1011 RS 232 Terminal Interface . $ 43
- VIC 1211 Super Expander .... $ 52
- VIC 1212 Programmers Aid Cartridge . $ 45
- VIC 1213 Vicmon Machine Language Monitor .... $ 45
- VL 102 Introduction to Basic Programming .... $ 21
- VT 106A Recreation Pack .... $ 45
- VT 107 A Home Calculation Pack ... $ 45
- VT 164 Programmable Character Set . $ 12
- VIC 1600 Vicmodem .... $ 89
- VIC 1311 Joystick .... $ 8
- VIC 1312 Game Paddles .... $ 16
- VM Programmers Reference Guide .. $ 14

VIC Software
- Avenger .... $ 23
- Superslot .... $ 23
- Super Alien .... $ 23
- Jupiter Lander .... $ 23
- Draw Poker .... $ 23
- Midnight Drive .... $ 23
- Radar Rat Race .... $ 23
- Raid on Fort Knox .... $ 23
- Sargon II Chess .... $ 29
- Super Smash .... $ 23
- Cosmic Cruncher .... $ 23
- Gorf .... $ 29
- Omega Race .... $ 29
- Money Wars .... $ 23
- Menagerie .... $ 23
- Cosmic Jailbreak .... $ 23
- Clowns .... $ 23
- Garden Wars .... $ 23
- Sea Wolf .... $ 23
- Adventureland .... $ 29
- Pirate Cove .... $ 29
- Mission Impossible .... $ 29
- The Count .... $ 29
- Voodoo Castle .... $ 29
- The Sky is Falling .... $ 23
- Mole Attack .... $ 23
- Bingo Speed Math .... $ 23
- Home Babysitter .... $ 23
- Visible Solar System .... $ 23
- Personal Finance .... $ 29

United Microwave
- Spiders of Mars (CT) .... $ 34
- Meteor Run (CT) .... $ 34
- Amok (C) .... $ 17
- Alien Blitz (C) .... $ 17
- Skymath (C) .... $ 12
- Space Division (C) .... $ 12
- Super Hangman (C) .... $ 14
- The Alien (C) .... $ 17
- 3D Maze (C) .... $ 12
- Kosmic Kamikaze (C) .... $ 17
- Sub Chase (C) .... $ 17
- Amok (CT) .... $ 27
- Renaissance (CT) .... $ 34
- Alien Blitz (CT) .... $ 27
- Cloud Burst (CT) .... $ 27
- Satellites and Meteorites (CT) .... $ 34
- Outworld (CT) .... $ 34

The Computer Outlet is an associate of The Computer Learning Center For Children. We are experts in educational technology and can customize educational software curriculums for school districts, individual schools, or for the child at home. Please contact us about your software and equipment requirements and feel free to stop by our school in Las Vegas.

We have one of the world's largest educational software inventories featuring our own Computer Learning Center software.

- Ten Little Robots (ATARI) $12.95
- Pre-School Math (ATARI) $19.95

cassette was very simple.

1. Open the cassette file for writing
2. While there are more birds to go onto the list
   2.1 Input a bird name
   2.2 If the name does not have typos
      2.2.1 Then write the name on the tape file
      2.2.2 Else request re-entering that bird name
3. Close the file

We'd need to be able to make changes in the list when the count results were being tabulated as some unexpected birds would appear, and other expected ones would not.

For creating the field form, however, this list would be used directly. We wanted the complete form to fit on one page for convenience in the field. By reading the whole list into an array in memory, then printing the list in two columns, one from the beginning of the array and one from the midpoint of the array, all the names fit on one sheet.

There was even space for eight unexpected birds to be noted on the bottom of the form. To divide up the tally space, a row of dashes was printed in front of each bird name. The procedure was this:

1. Initialize array space for names
2. Initialize dash string to correct number of dashes
3. Open name file
4. Read complete list into array
5. Close name file
6. Specify format to printer
7. While more copies of the form are needed
   7.1 For number of birds from one to half of total
      7.1.1 Print dash string; bird name (counter); dash string; bird name (counter + halfnumber)
   7.2 Print four lines of pairs of double-dash strings

By using the formatting capability of the CBM printer, it was easy to line up the strings in even rows. This could have been done by padding the name strings with blanks so they would be an even length, as I did later in this project, doing the report.

Processing The Results

After the day of the count we were ready to tabulate the data that had been collected. The primary subtasks of the tabulation and report writing process were these:

1. Get the bird list and edit it
2. Input the numbers of species seen by party
3. Calculate the cross totals and species counts
4. Print the report
5. Save the data for future use

Each of these would be divided further.

The list of names of expected birds was stored on a cassette tape, but some of these birds had not been found on the count day, while a few unexpected others did appear. Using a simple editing program, we read in the original list and wrote out a new list of all the birds sighted on that year's count. The procedure was this:

1. Initialize an array for the names
2. Open the master list file
3. Read names into the array
4. Close master list file
5. Open list file for this year
6. While not at end of list in array
   6.1 Print next name on list for user
   6.2 If a new name should be inserted before next name
      6.2.1 Then accept input of name to be inserted
         Write new name to year file
      6.2.2 Else if next name should be kept
            Then write name to year file
         Move to next name
7. Close the file with this year's list

This procedure deletes birds not seen by simply skipping over them and not writing them on the current year list file.

The next step required entering the data on sightings of each bird by each party. The input mechanism I chose for this portion of the project was the READ-DATA combination. With this method, the data is specified in the program in non-executable statements that look like this:

```
2001 DATA 5,0,4,14,6,3,9
```

They are read by an executable statement (READ) elsewhere in the program. I think of this method as attaching a data file onto the end of the program. When using a PET, there is a very good reason for doing this: the PET screen editor.

The built-in editor on Commodore computers is very flexible and easy to use, not only for correcting typos, but also for duplicating lines or parts of lines. I find that entering a lot of numerical data is hard to do accurately, even when using a number pad. If such data is being input interactively, the user must be asked to confirm each item for correctness, making data entry very boring.

Using the screen editor, however, makes it relatively painless to get a complete set of correct data via DATA lines within the program.

For this project, we needed the number of birds of each species seen by each party. I used one DATA statement for a set of three birds, with the line numbers of the statements keying back to the birds on the list. I then ran the program to combine the names from the cassette file with the data on the sightings.

1. Open name file
2. Initialize array for names
3. Read in names
4. Close name file
5. Open results file on cassette
6. For birds from 1 to end of list in array do
   6.1 Read a set of data from the sequential DATA statements
   6.2 Get the next name from the array
   6.3 Write the name plus the data to the results file
7. Close results file

By now you have noticed that I write intermediate steps of my processing out to cassette files frequently. This is not because I enjoy waiting for the tape read and writes. It is my insurance against radical loss. I am cautious enough about my machine and the perfection of my programs that I never want to get too far away from my last plateau. Additionally, by dividing the total project into chunks, each of the parts did not come up against the size limitation of 8K memory, while a program to do the complete project undoubtedly would have.

Creating The Report

The next step was to actually process the data. This cycle, I read each line of data including the name from the cassette file and processed the numbers in it. I then packed it into a string variable and put it temporarily in an array. It was necessary to do this in sections because the memory is insufficient to hold the complete set of data in the array.

This was still quite convenient, though, because we found that 25 lines of data, plus a heading, fit nicely on a page for the report. We processed it in units of this size, ending up with a report with five pages of results (see the chart).

After each set of 25 was processed, we printed the needed number of copies of that page of the report, then proceeded to the next. The last page was somewhat different because of the totals, but the general process was this:

1. Open input file
2. Initialize
   a) a string array of 25 elements
   b) an eight-element array to read the data into (seen)

Bird Count Results

NUMBER SEEN BY PARTY

	1	2	3	4	5	6	TOTAL
COMMON LOON	- Ø	3	Ø	Ø	Ø	1	4
ARCTIC LOON	- Ø	Ø	Ø	1	Ø	1	2
RED-THROATED LOON	- 1	Ø	Ø	4	3	1	9
LOON SP.	- Ø	4Ø	Ø	15	Ø	Ø	55
RED-NECKED GREBE	- Ø	Ø	Ø	1	Ø	1	2
HORNED GREBE	- 15	2	Ø	5	Ø	2	24
EARED GREBE	- 2	Ø	Ø	1	Ø	Ø	3
WESTERN GREBE	- 6	2	Ø	4	Ø	2	14
PIED-BILLED GREBE	- Ø	7	Ø	2	1	14	24
DBL-CRSTD CORMORANT	- 2	12	1	2	1	1	19
BRANDT'S CORMORANT	- Ø	Ø	Ø	2	Ø	Ø	2
PELAGIC CORMORANT	- Ø	14	Ø	61	Ø	8	83
GREAT BLUE HERON	- 3	2	3	3	1	1	13
GREEN HERON	- Ø	Ø	Ø	Ø	Ø	1	1
GREAT EGRET	- Ø	Ø	Ø	Ø	1	1	2
CATTLE EGRET	- 1	Ø	Ø	Ø	2	Ø	3
WHISTLING SWAN	- Ø	Ø	Ø	Ø	1	Ø	1
GOOSE SP.	- Ø	2Ø	Ø	Ø	Ø	Ø	2Ø
MALLARD	- 28	Ø	37	Ø	Ø	Ø	65
GADWALL	- Ø	6	Ø	Ø	Ø	Ø	6
PINTAIL	- Ø	28	Ø	Ø	Ø	Ø	28
GREEN-WNGD TEAL	- 6	1Ø	Ø	2	Ø	Ø	18
AMERICAN WIGEON	- 8	Ø	23	Ø	Ø	2	33
RING-NECKED DUCK	- 3	Ø	Ø	Ø	Ø	1Ø	13
CANVASBACK	- 1	Ø	Ø	Ø	Ø	2	3
YELLOW-RMPD WARBLER	- 76	34	5	46	1	65Ø	812
PALM WARBLER	- 1	Ø	Ø	Ø	Ø	Ø	1
HOUSE SPARROW	- 25	Ø	Ø	Ø	Ø	2Ø	45
WESTERN MEADOWLARK	- 2Ø	Ø	36	72	12	Ø	14Ø
REDWINGED BLACKBIRD	- Ø	Ø	1ØØ	Ø	6	Ø	1Ø6
BREWER'S BLACKBIRD	- 3	Ø	5	Ø	13	4Ø	61
HOUSE FINCH	- 1	Ø	8	Ø	Ø	6	15
PINE SISKIN	- 16	Ø	Ø	Ø	Ø	5	21
AMERICAN GOLDFINCH	- Ø	Ø	Ø	3	Ø	5	8
RED CROSSBILL	- Ø	Ø	13	Ø	Ø	Ø	13
RUFOUS-SIDE TOWHEE	- 7	2	5	2	1	1	18
SAVANNAH SPARROW	- Ø	Ø	Ø	12	Ø	Ø	12
OREGON JUNCO	- 43	13	7Ø	5Ø	36	53	265
WHITE-CRND SPARROW	- 53	Ø	38	21	13	1Ø	135
GOLDN-CRND SPARROW	- 3	Ø	Ø	1	Ø	Ø	4
FOX SPARROW	- 9	4	11	21	Ø	4	49
LINCOLN'S SPARROW	- Ø	Ø	Ø	1	Ø	Ø	1
SONG SPARROW	- 17	11	4	35	1	9	77
LAPLAND LONGSPUR	- Ø	Ø	Ø	3	Ø	Ø	3
SPECIES SEEN	- 56	54	45	74	38	61	115

TOTAL INDIVIDUALS SEEN- 11944

c) an eight-element array to count species seen by party (count)

3. Create the heading strings
4. For the first hundred birds (four sets) do
   4.1 For 25 data lines do
      4.1.1 Read a data line (name and eight numbers into seen (party))
      4.1.2 For each of the eight parties
         4.1.2.1 If bird seen by the party (not 0) Then increment count (part) by 1
      4.1.3 Sum numbers seen across the eight parties
      4.1.4 Make strings of the numbers seen and the total
      4.1.5 Build a string of the name, number strings, total string
      4.1.6 Place this output string in the string array
      4.1.7 Accumulate grand total of numbers seen
   4.2 For the number of copies of the report needed
      4.2.1 Print heading
      4.2.2 Print the set of 25 output lines
   4.3 Write the set of 25 output lines to a file

The process was repeated in a similar manner for the last page. Here there were fewer data lines, and at the bottom of that page, the total number of species seen by each party and the grand total of individuals and of species seen were printed.

When building the output string, the name and number strings were padded with blanks, effectively formatting the printed output. BASIC's string functions make this quite simple, and storing in one string array again saves space in memory.

Using these programs, we have been able to get reports out to participants within a week of the count. We have been pleased with the quality and attractiveness of the reports, as well as appreciating the use of our personal computer to make another hobby, birding, even more enjoyable. ©

HOFACKER

Books + Software for ATARI VIC-20 OSI SINCLAIR TIMEX

ELCOMP PUBLISHING, INC
53 Redrock Lane
Pomona, CA 91766
Phone: (714) 623 8314

Payment: check, money order, VISA, MASTER-CHARGE, Eurocheck.
Orders from outside USA: add 15% shipping. CA residents add 6.5% tax
*ATARI is a registered trademark of ATARI Inc.
*VIC-20 is a registered trademark of Commodore

PRODUCTS FOR ATARI* 400/800 FROM ELCOMP

BOOKS:

ATARI BASIC – Learning by Using
An excellent book for the beginner. Many short programs and learning exercises. All important features of the ATARI computers are described (screen drawings, special sounds, keys, paddles, joysticks, specialized screen routines, graphics, sound applications, peeks, pokes, and special stuff). Also suggestions are made that challenge you to change and write program routines.
Order #164 — $7.95

Games for the ATARI Computer
This book describes advanced programming techniques like player-missile-graphics and use of the hardware-registers. Contains many ready to run programs in BASIC and one called GUNFIGHT in machine language.
Order #162 — $7.95

Programming in 6502 Machine Language on your PET+CBM
2 complete Editor/Assemblers (Source code 3 hexdump + description plus a powerful machine language monitor (Hexdump)).
Order #166 — $19.95

How to program your ATARI in 6502 machine language
Introduction to machine language for the BASIC programmer
Order #169 — $9.95

SOFTWARE IN BASIC FOR ATARI

Invoice Writing for Small Business
This program makes writing invoices easy. Store your products in DATA statements with order-number, description, and price. The program later retrives the description and price matching to the entered order-number. The shipping cost and the discount may be calculated automatically depending on the quantity ordered or entered manually. The description to the program tells you how to change the program and adapt it to your own needs. Comes with a couple of invoice forms to write your first invoices on to it.

Order #7201	cassette version	$29.95
Order #7200	disk version	$39.95

Mailing List
This menu driven program allows the small business man to keep track of vendors and customers. You can search for a name or address of a certain town or for an address with a certain note. 50 addresses are put into one file.

Order #7212	cassette version	$19.95
Order #7213	disk version	$24.95

Inventory Control
This program is menu driven. It gives you the following options: read/store data, define items, entry editing, inventory maintenance (incoming-outgoing), reports. The products are stored with inventory number, manufacturer, reorder level, present level, code number, description.

Order #7214	cassette version	$19.95
Order #7215	disk version	$24.95

Programs from Book #164
The programs from book no. 164 on cassette. (Book included)
Order #7100 — $29.00

Game Package
Games on cassette. (Bomber, tennis, smart, cannon fodder, etc.)
Order #7216 — $9.95

Microcomputer Hardware Handbook (845 pages)
Descriptions, pinouts and specifications of the most popular microprocessors and support chips.
A MUST for the hardware buff.
Order-No. 29
$14.95

Care and Feeding of the Commodore PET
Eight chapters exploring PET hardware. Includes repair and interfacing information. Programming tricks and schematics.
Order #150 — $9.95

SOFTWARE IN MACHINE LANGUAGE for ATARI

ATMONA-1
This is a machine language monitor that provides you with the most important commands for programming in machine-language. Disassemble, dump (hex and ASCII), change memory location, block transfer, fill memory block, save and load machine-language programs, start programs. Printer option via three different interfaces.

Order #7022	cassette version	$19.95
Order #7023	disk version	$24.95
Order #7024	cartridge version	$59,00

ATMONA-2
This is a tracer (debugger) that lets you explore the ATARI RAM/ROM area. You can stop at previously selected address, opcode, or operand. Also very valuable in understanding the microprocessor. At each stop, all registers of the CPU may be changed. Includes ATMONA-1.

Order #7049	cassette version	$49.95
Order #7050	disk version	$54.00

ATMAS
Macro-Assembler for ATARI-800/48k. One of the most powerful editor assemblers on the market. Versatile editor with scrolling. Up to 17k of source-Code. Very fast, translates 5k source-code in about 5 seconds. Source code can be saved on disk or cassette. (Includes ATMONA-1)

Order #7099	disk version	$89.00
Order #7999	cartridge version	$129.00

ATAS
Same as ATMAS but without macro-capability. Cassette-based.

Order #7098	32k RAM	$49.95
Order #7998	48k RAM	$49.95

ATEXT-1
This wordprocessor is an excellent buy for your money. It features screen oriented editing, scrolling, string search (even nested), left and right margin justification. Over 30 commands. Text can be saved on disk or cassette.

Order #7210	cassette version	$29.95
Order #7216	disk version	$34.95
Order #7217	cartridge version	$69.00

GUNFIGHT
This game (8k machine-language) needs two joysticks! Animation and sound. Two cowboys fight against each other. Comes on a bootable cassette.
Order #7207 — $19.95

FORTH for the ATARI

FORTH from Elcomp Publishing, Inc. is an extended Fig-Forth-version, Editor and I/O package included. Utility package includes decompiler, sector copy, Hex-dump (ASCII), ATARI Filehandling, total graphic and sound, joystick program and player missile. Extremely powerful!
Order #7055 — disk — $39.95

Floating point package with trigonometric functions (0 – 90°).
Order #7230 — disk — $29.95

Learn-FORTH from Elcomp Publishing, Inc.
A subset of Fig-Forth for the beginner. On disk (32k RAM) or on cassette (16k RAM).
Order #7053 — $19.95

Expansion boards for the APPLE II

The Custom Apple + Other Mysteries
A complete guide to customizing the Apple Software und Hardware
Order-No. 680 — $24.95

We also stock the boards which are used in the book "The Custom Apple ..." (barebords)

6522 I/O Board No. 605	$39.00
EPROM Burner No. 607	$49.00
8K EPROM/RAM Board No. 609	$29.00
Prototyping board for the Apple II No. 604	$29.00
Slot repeater board for the Apple II No. 606	$49.00

Order two boards and get the book free!

COMING SOON ! ORDER NOW !

A Look in the future with your ATARI
(Astrology and how to do your own horoscope on the ATARI 800. **Order No. 171** — $9.95

FORTH on the ATARI – Learning by Using
Order No. 170 — $7.95

Hardware – ADD–ONS for ATARI

PRINTER INTERFACE
This construction article comes with printed circuit board and software. You can use the EPSON printer without the ATARI printer interface. (Works with gameports 3 and 4).
Order #7211 — $19.95

RS-232 Interface for your ATARI 400/800
Software with connector and construction article.
Order #7291 — $19.95

EPROM BURNER for ATARI 400/800
Works with gameports. No additional power supply needed. Comes compl. assembled with software (2716, 2732, 2532).
Order #7042 — $179.00

EPROM BURNER for ATARI 400/800 KIT
Printed circuit board incl. Software and extensive construction article.
Order #7292 — $49.00

EPROM BOARD (CARTRIDGE)
Holds two 4k EPROMs (2532). EPROMs not included.
Order #7043 — $29.95

EPROM BOARD KIT
Same as above but bare board only with description.
Order #7224 — $14.95

ATARI, VIC-20, Sinclair, Timex and OSI

New – for your ATARI 400/800

Astrology and Biorythm for ATARI (cass. or disk).
Order #7223 — $29.95

Birth control with the ATARI (Knaus Ogino)
Order #7222 — cass. or disk — $29.95

Books + Software for VIC-20 (requires 3k RAM Exp.)

#4870	Wordprocessor for VIC-20, 8k RAM	$19.95
#4883	Mailing List for VIC-20, 16k RAM	$14.95
#141	Tricks for VICs - The VICstory Progr.	$9.95
#4880	TIC TAC VIC	$9.95
#4881	GAMEPACK I (3 Games)	$14.95
#4885	Dual Joystick Instruction	$9.95

INPUT/OUTPUT Programming with your VIC
Order #4886 — $9.95

#4896	Miniassembler for VIC-20	$19.95
#4881	Tennis, Squash, Break	$9.95
#4894	Runfill for VIC	$9.95

Universal Experimenter Board for the VIC-20
(Save money with this great board). This board plugs right into the expansion slot of the VIC-20. The board contains a large prototyping area for your own circuit design and expansion. The construction article shows you how to build your own 3k RAM expander and ROM-board.
Order #4844 — $18.95

Software for SINCLAIR ZX-81 and TIMEX 1000

#2399	Machine Language Monitor	$9.95
#2398	Mailing List	$19.95

Programming in BASIC and machine language with the ZX-81 (82) or TIMEX 1000.
Order #140 — (book) — $9.95

Books for OSI

#157	The First Book of Ohio	$7.95
#158	The Second Book of Ohio	$7.95
#159	The Third Book of Ohio	$7.95
#160	The Fourth Book of Ohio	$7.95
#161	The Fifth Book of Ohio	$7.95
#151	8K Microsoft BASIC Ref. Man.	$9.95
#152	Expansion Handbook for 6502 and 6802	$9.95
#153	Microcomputer Appl. Notes	$9.95

Complex Sound Generation
New revised applications manual for the Texas Instruments SN 76477 Complex Sound Generator.
Order #154 — $6.95

Small Business Programs Order #156
Complete listings for the business user. Inventory, Invoice Writing, Mailing List and much more. Introduction to Business Applications. — $14.90

High Resolution Turtle Graphics

Connecting The Strobe Pen Plotter To Apple Turtle PILOT

David D Thornburg, Associate Editor

There comes a time when most users of turtle graphics wish they could get higher resolution pictures than those shown on the display screen. The easiest way to accomplish this is to connect the computer to a graphic pen plotter. Pen plotters have been available for many years, but it is only recently that their cost has dropped to the point that they are affordable to home computer users.

Of the various low-price plotters, the Strobe Model 100 has a price of under $800 (including Apple interface card and software), and has a resolution of 0.002 inches in both axes. It uses inexpensive fine-point pens from the corner drugstore, and plots on plain 8½ x 11 paper. With special pens, it can also plot directly onto plastic sheets for overhead transparencies.

While Strobe provides several application packages for various business and other graphic applications, the plotter can also be interfaced to any program written in Applesoft BASIC. In order to use the plotter with your own programs, you must first load the printer driver program (supplied). Since this program resides just above memory location 35071 and is executed with the Applesoft CALL command, I have not found a way to use this plotter directly from Logo. Anyone who has solved this problem is invited to write about it!

Modify PILOT

Devout turtlers need not feel depressed, however, since the Turtle PILOT language by Alan Poole (published in the September 1982 issue of **COMPUTE!**) is written in Applesoft.

This language system consists of two programs – an editor for creating PILOT listings, and a translator that converts the PILOT program to Applesoft and appends the necessary BASIC utilities needed to make everything work properly. To interface the plotter to the language, one needs only to modify two subroutines and add one new subroutine to the translator program. To keep these programs clear, I will show only the changes that are to be made in the program published in Poole's original article. If you used different line numbers in your version, in order to see where you should put them you will have to compare these changes with the original listing.

The modifications to the translator perform three tasks:

1. We must load the plotter driver routine and initialize the system. Since the routine starting at line 50000 is used at the beginning of every translated program, this is where we will add these tasks.

2. We must add plotter commands after the screen drawing commands so that our plotted image will appear at the same time it is being drawn on the screen. The screen drawing routine begins at line 55000, so this is where we will make these changes.

3. Finally, we need to add a routine that scales the plot commands for the paper size and plotter resolution, sets the pen in the up or down position as appropriate, and ships this assemblage of data to the plotter for execution. We will create this routine starting at line 56000.

Because all the changes are in that portion of the translator appended to each translated program, only one tiny change needs to be made in the PILOT programs themselves. As implemented, the command G:GOTO x,y will only be executed when the next G:DRAW command is given. If you are moving the turtle to a new loction X,Y with the pen up, you can execute this on the plotter with the sequence:

```
G:PEN UP;GOTO X,Y;DRAW 0;PEN DOWN
```

The function of the DRAW 0 command is to force the plotter to carry this motion out before setting the pen down.

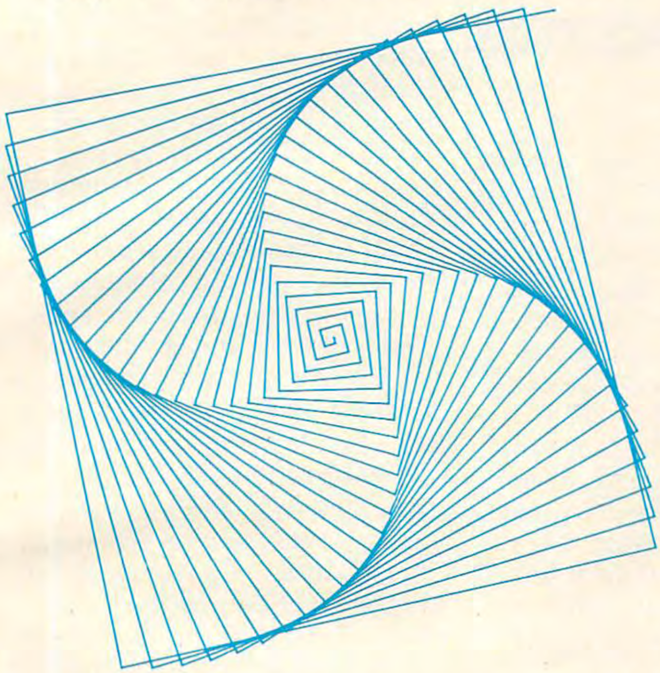
Except for the small inconvenience of adding the extra DRAW 0 commands after each GOTO, any of your existing PILOT turtle graphics programs will run on the plotter as soon as they have been re-translated. I recommend using the original translator for making sure the picture fits on the screen and otherwise does what you want. Once this is done, you can use the modified translator (called, for example, TRANSPLOT) to generate the BASIC program that will both draw pictures on the display and plot them on the plotter at the same time.

To try out the plotter, I entered the following PILOT program:

```
*SQUIRAL
```

```
G: CREAL
C:A=0
*LABEL
G:DRAW A
G:TURN 91
C:A=A+1
J(A<100):*LABEL
E:
```

When this was translated and run, I was able to get a beautiful squiral pattern that was devoid of the jaggies one gets with a raster display screen.

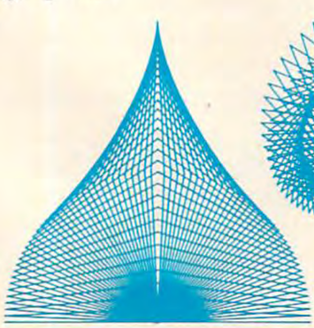


As you create pictures of your own, you will want to change pen colors every so often in the middle of a drawing. An easy way to do this is to use the following procedure when you want to change colors:

```
*QUERY
T:CHANGE PEN AND PRESS RETURN
A:
E:
```

This will stop the execution of the program while you change pens. When you are ready to start plotting again, just press RETURN.

The following figures are but a small indication of the pleasures that await those of you who want to increase the resolution of your turtle graphics.



The changes to be made in the *Translator* program of Apple Turtle PILOT include:

1. Set up procedure:

```
50000 PRINT CHR$(4);"BLOAD PLOT1.8"
50002 HIMEM: 35071
50004 CALL 35081
50006 DIM Q$(25),QS(31)
50008 QP=1:QX=0:QY=0:GOSUB 56000:QP=0
```

2. Modify drawing routine:

```
55004 GOSUB 56000
55045 IF QP=1 THEN GOSUB 56000
55060 HPLOT TO QX+139.0005, -QY+80.0005:
      GOSUB 56000
55070 RETURN
```

3. Add plotter routine:

```
56000 XI=20*(QX+137.5):YI=20*(QY+106.25):
      P%=QP+2
56010 IF XI<0 THEN XI=XI+65536
56020 IX%=XI/256
56030 POKE 35085,IX%
56040 IX%=XI-IX%*256
56050 POKE 35084,IX%
56060 IF YI<0 THEN YI=YI+65536
56070 IY%=YI/256
56080 POKE 35087,IY%
56090 IY%=YI-IY%*256
56100 POKE 35086,IY%
56110 IF P%<0 THEN P%=P%+255
56120 POKE 35088,P%
56130 CALL 35072
56140 RETURN
```

Note: If you are extremely picky about plotting accuracy, add the line:

```
56005 XI=XI*1.0007506:YI=YI*1.0198781
```

Any disk that contains programs generated with *Transplot* also needs to have a copy of the Strobe program *Plot*1.8. To copy this program to your disk, place any Strobe disk in your Apple and enter:

```
BLOAD PLOT1.8
```

Next, insert your program disk and enter:

```
BSAVE PLOT1.8,A$8900,L$6E0
```

©

Apple Educational Games

Sheila Cory, Chapel Hill, NC

If you are either a teacher or a parent of young children and have access to an Apple II+ computer with 48K and a disk drive, there is some software available that you should know about. Produced by The Learning Company, it's specifically designed for preschool and elementary-school youngsters.

This review covers three packages of programs. The first, *Juggles' Rainbow*, is designed for children aged three to six. The second, *Bumble Games*, is for ages four to ten, and the third, *Bumble Plot*, eight to thirteen. All three packages are well designed, and the sequence of the material progresses logically.

Juggles' Rainbow

Juggles' Rainbow is a welcome addition to the small amount of good software for the preschool, kindergarten, and first grade set. Frequently, teachers of very young children are left out when computers are discussed in faculty meetings or workshops, and feel that there's not much that can be done with the computer for children who don't yet have reading skills. It takes great sensitivity to the particular qualities of children of this age to produce software that is interesting, challenging without being too difficult, and educationally sound. *Juggles' Rainbow* shows this sensitivity.

Juggles' Rainbow consists of three programs for children, and one program for teachers or parents. The children's programs are Juggles' Rainbow (the name is used for the entire package and for one of the programs within the package), Juggles' Butterfly, and Juggles' Windmill. The adults' program, called The Big Question Mark, allows the setting of options such as whether sound should be included in the program, whether the child should be given picture and word or just word clues, and gives instructions for dividing the keyboard into halves and quarters for some of the exercises.

Juggles' Rainbow is designed to reinforce the teaching of the concepts of *above* and *below*. The program divides the keyboard into an upper and a lower section with a blue strip of cardboard that is provided with the diskette. A blue line appears on the screen. Children find that when they depress a key below the keyboard divider, a colorful vertical line appears below the blue screen line, and when a key is depressed above the keyboard divider, a colorful line appears above the screen line.

The next segment of this program prompts the user to depress keys above and below the keyboard divider to color in outlined bars above and below the blue screen line. The third segment allows the child to apply his skill with *above* and *below* to create a colorful rainbow.

Juggles' Butterfly reinforces the concepts of *left* and *right*. Again the keyboard is divided with a provided blue strip, but this time the division is in a vertical direction, creating a left and right section of the keyboard. The program works basically the same way as Juggles' Rainbow, but the final segment allows the child to create a marvelous butterfly by applying color to the right and left sides of the butterfly body as keys to the right or left of the keyboard divider are depressed.

Juggles' Windmill takes the learning one step further by having the child depress keys above (or below) the horizontal keyboard divider and to the left (or right) of the vertical divider. The culmination of this activity is the creation of a windmill that would delight a very young child.

Luring our four-year-old visitor, Christopher, away from his LEGO project to try out these programs was difficult, but they quickly absorbed him. This was not only his first opportunity to use the programs, but was also his first time using a computer. A good deal of adult guidance was needed to help him figure out what he was supposed to do, and to extend the learning. This program could make ideal use of a classroom volunteer or older child whose role would be to talk through the concepts, exclaim over the results, and guide the discoveries made using the computer.

One problem Christopher had was keeping the cardboard keyboard dividers in place. I recommend that a piece of heavy blue yarn be used instead of the cardboard. The yarn could be placed between the second and third rows of the keyboard, rather than over the third row as suggested in the manual, and the yarn could be securely taped in place at each end. A similar procedure could be used for the vertical keyboard divider. Christopher's interest in the activity lasted about ten minutes, giving him time to get through Juggles' Rainbow and begin Juggles' Butterfly. His enjoyment of the activity was evident when he asked me if he could play the rainbow game again before he went home.

Bumble Games

Bumble Games introduces the delightful Bumble, who is the central character in all of the programs in *Bumble Games* and *Bumble Plot*. The learning objective in *Bumble Games* is to teach the graphing of positive numbers. Some of the concepts covered are also covered in the MECC (Minnesota Educational Computing Consortium) game of Hurkle. The *Bumble Games* diskette contains six programs, each one progressively more sophisticated. The sequence is excellent, extending the learning by a small degree with each successive game.

The program Find Your Number begins the sequence by giving practice in finding a number between zero and five that Bumble has secretly chosen. The child is shown a horizontal or a vertical number line with the numbers zero to five on it, and makes a guess. Bumble responds with a left or right arrow in the case of a horizontal number line, and an up or down arrow in the case of a vertical number line, indicating whether the next number guessed should be more or less than the present guess. The horizontal and vertical number lines begin preparing children for an X and Y axis that they'll see in a later program. When the number is guessed, the child gets a colorful display of the number, accompanied by tones representing the number. If two children want to play this game, Bumble will select two numbers.

Find the Bumble introduces a four-by-four grid, cleverly differentiating the X and Y axes by labeling one with letters and one with numbers. Bumble hides in one of the boxes formed by the grid, and the child must find Bumble by naming the coordinates of his box. Bumble is very helpful, telling the child to pick bigger or smaller numbers for the Y axis, and letters to the left or to the right for the X axis. When Bumble is found, his friendly, bigger-than-life image appears on the screen.

Butterfly Hunt has Bumble out searching for his lost butterfly. This game works very much like Find the Bumble, but a slightly larger grid prepares the child for the next game, Visit From Space.

Visit From Space introduces the idea that the intersection of two lines in a grid can be named by using a number on the bottom of the grid and one at the side. In this game, both X and Y axes are labeled with numbers. The object of the game is for the child to find Bumble's cousin who has flown in from outer space and is hiding in his spaceship somewhere on the grid. Very clear graphic and written clues help the child learn to locate exactly the intersection he wants to guess. When the spaceship is finally found, it zooms across the screen, making appropriate outer-space noises!

Tic Tac Toc is a game for two players, similar to the more conventional tic tac toe. The idea is for the child to get four markers in a horizontal, vertical, or diagonal line before his or her opponent does. The game screen consists of a five-by-five grid, and a marker is placed by naming the coordinates of the desired position.

The board is somewhat confusing to the beginning player; it would be helpful for the teacher to make a similar board on a transparency and use the overhead projector to play the game a few times with the whole class before children begin to play the game on the computer. The game does give excellent practice in naming points on a grid. It is just different enough from tic tac toe to be interesting.

Bumble Dots extends the grid to ten-by-ten. In this game, Bumble helps the child draw dot-to-dot pictures. A dot appears on the grid, and the child is asked to name it. When the first dot is successfully named, a second dot appears, and when that is successfully named, a line is drawn to connect the dots. This procedure continues until a whole picture is drawn.

The child can also make his own picture by naming coordinates for Bumble to connect. Bumble first asks the child how many dots will be in his picture. Since this is difficult for a child to ascertain in advance, it would be helpful if the teacher had the children first draw a picture using three to nine dots on a piece of graph paper, and then bring that picture to the computer when their turn comes. Children would then be all set to answer Bumble's question about the number of dots needed for the picture. The Learning Company, in developing this program, recognized the fact that generations of children have loved dot-to-dot pictures, and that a natural progression of learning can take place by tapping into this love.

Bumble Plot

Bumble Plot extends the learning about grids to include negative numbers. It consists of five programs, again carefully sequenced to take the child comfortably through the steps culminating with naming points on a ten-by-ten grid where negative numbers are used and the 0,0 point is in the middle. The sequence starts with Trap and Guess, where the child tries to trap Bumble's secret number on a minus three to plus three horizontal or vertical number line. Bumblebug has Bumble hopping around on a grid; the object is to set traps for him to jump into! In Hidden Treasure, the child searches for invisible treasures on a ten-by-ten grid with negative numbers. I found a ship's anchor, a diamond ring, a friendly octopus, and a golden crown! Children would enjoy a worksheet where they could show what they found and where they found it when they played the game. These worksheets could be displayed on the bulletin board above the computer.

Bumble Art is similar to Bumble Dots, but contains negative numbers in the grid. The most action-packed game of the series is Roadblock. The object of this game is to build roadblocks to surround the bank robber before he gets away. This, of course, all takes place on a minus-five by plus-five grid, providing wonderful practice in the skills that have been developed through all of the other games.

All three of the packages reviewed here share some very positive qualities. They all contain excellent graphics; they use sound appropriately to enhance the learning or entertainment value of the program, and sound can be turned off if it provides a distraction in the classroom. The programs are very user friendly, take all kinds of input without bombing, give the user excellent prompts, and have very carefully formatted screen displays.

Manuals are well illustrated and appealing. Each one gives instructions on how to load the diskette and a little information about each program. It would have been very useful to have included suggestions for teachers about things to talk about before each program, and appropriate worksheets for follow-up activities.

The company will send you a set of activity cards for free when you send back the owner registration card. This card also entitles you to purchase a backup diskette for $12. No teacher should ever use software in the classroom without having a backup diskette.

I suspect that schools that purchase software from The Learning Company will have a new little character joining Snoopy and The Cat In The Hat in adorning their bulletin boards. Bumble has great personal appeal and represents software that is both educationally sound and fun to use.

Juggles' Rainbow ($45)
Bumble Games ($60)
Bumble Plot ($60)
The Learning Co.
4370 Alpine Road
Portola Valley, CA 94025

PROMQUEEN (VIC-20 Hardware)

Harvey B. Herman, Associate Editor

The hardware reviewed here will be of interest to a select group of **COMPUTE!** readers. If you own a VIC and have the need to "burn EPROMs," you should consider this cartridge. On the other hand, if you are completely befuddled by the previous sentence, go to the next article; save $200.

I was excited when I received the PROMQUEEN for review, as it was just what I needed. Several pieces of computer-related equipment which I use daily contain EPROMs (Erasable Programmable Read Only Memory chips). What would I do if one failed and I had no way to replace it? The PROMQUEEN promised to solve this potential problem, even for one like myself, who had never programmed an EPROM before.

It is misleading to think of the PROMQUEEN exclusively as hardware. What is visible, of course, is a cartridge (hardware), which plugs into the memory expansion port of the VIC. But it also comes with essential software (actually firmware on EPROM) without which the hardware would be useless. A 25-page instruction manual is included as well.

The major function of this product is to allow the user to conveniently burn EPROMs. That is, data is to be stored into an erased EPROM so that it will be there the next day even after power has been turned off. This data can be copied from a previously programmed EPROM or typed in from scratch using the monitor program provided.

I was a little apprehensive on my maiden EPROM burning session. First, an EPROM had to be erased. No problem here. I used a shortwave ultraviolet mineral lamp (2537 A wavelength). The shortest erase time I tried was 40 minutes. If you don't have one of these lying around, there are several advertised units which should be satisfactory.

Next, I attempted a copy of the EPROM Hexkit program which comes with the PROMQUEEN. The instructions were somewhat confusing. It was not immediately clear that there is both RAM and ROM memory in the package and that you have the option of moving these around by switches. The ROM memory is only there if an EPROM is plugged into the external socket. However, I was using a preliminary manual. The manufacturer promises that an improved manual will be available shortly.

The burning of an EPROM is actually easy once you know what you are doing. The procedure is:

1. Insert the Hexkit EPROM in the zero insertion force socket (ZIF).

2. Set the switches correctly.

3. Transfer the program to the VIC RAM with a SYS call.

4. Insert the EPROM to be copied.

5. Transfer the EPROM data to the PROMQUEEN RAM using the transfer function of Hexkit.

6. Insert the erased EPROM (2716, 2732 or 2732A – a 2532 will not work).

7. Use the burn function of Hexkit. The software first checks for a properly erased EPROM and later verifies the burn. After several false starts, I had successfully burned my first EPROM.

I have not described all of the features of the PROMQUEEN. The Hexkit program has many other features in addition to the burn function. It also uses color effectively: red screen when burning, for example. The hardware has lights (LEDs) which minimize the chance of error. It can also be used to emulate a ROM when working with other computers.

As with most equipment, I can cite good and bad features. The cartridge is well-constructed and easy to use once you know how. However, the version I tested included confusing directions and it is priced at more than twice what a similar system for the PET goes for. Nevertheless, if you need an EPROM burner and already own a VIC, this could be the best way to get one.

PROMQUEEN
Gloucester Computer Bus Co.
6 Brooks Road
Gloucester, MA 01930
$199

©

Preppie! For Atari

Mike Kinnamon, Stillwater, OK

Look out, *Star Raiders*! Move over, *Pac-Man*! *Preppie!* is here. It is time to return from patrolling the deep fringes of space and reduce your gluttonous intake of caloric maze candies. Now you can work off those extra pounds and breathe clean fresh air right here on Earth at your local golf course retrieving golf balls.

Preppie!, an arcade game for one or two players, embodies all the arcade game characteristics that take a player beyond mere enjoyment to truly enthusiastic excitement. The author, Russ Wetmore, is to be highly commended for the thoroughness and detail of his programming efforts. This program easily ranks among the best games to appear for the Atari computer to date.

Superb Graphics

Preppie! fully exploits the Atari graphics capabilities. I have seen no other game use as many different colors as this one. The detail given the objects is superb. The golf carts have steering wheels and bumpers. The treads of the bulldozers rotate. The alligators have wrinkled skin and sharp, white teeth. The logs display growth rings and peeling bark. The frog extends his limbs when leaping. Preppie's knees flex when he jumps. He even has the obligatory knit emblem on his shirt. The blades of the lawnmowers rotate. Even the title page and scoreboard are unique. Add all these details together, and you get a graphics display that commands everyone's attention.

As if graphics weren't enough, there is some fine music as well. As your Preppie moves, a bell rings to indicate passage from one line to the next and increases in your score. Should your Preppie be mauled by a marauding mower or poisoned by the infamous frog, a short funeral march is played. Falling into the water elicits yet another sound, as does retrieving or returning a golf ball from the rough.

Deadly Water Hazards

Enough aesthetics. How do you play *Preppie!*? The object of the game is to maneuver your Preppie across the fairway and the treacherous river into the rough, retrieving golf balls and returning them safely to the greens. You begin play by selecting one or two players. If two are playing, you have the option of using one or two joysticks. Now press the START key and begin the fun.

You start with three Preppies. Your journey is fraught with many dangers. You must negotiate a path that avoids contact with speeding golf carts, razor-sharp lawnmowers, killer bulldozers, and a poisonous frog. Then you must ford the river using moving boats, logs, and alligators as stepping stones. Should you survive to this point, you will find yourself in the rough, where most of the golf balls will appear. Pick up a golf ball, but be careful you don't fall into a water hole.

Now you must make your way back to the green via the same mobile hazards as before. Upon retrieving all the golf balls on the green, you will be advanced to the next higher level and will increase your score according to the amount of time increments remaining on the timer bar. Should the timer expire before you complete your mission, you lose a Preppie and must repeat the level you are on.

There are ten levels of play in *Preppie!*. You may start the game at level one or choose a higher level up to nine by pressing the appropriate number on your keyboard before hitting the START key. Each successively higher level increases the speed of the hazards. At certain levels you will be confronted with new hazards or an increase in the number of golf balls, up to four. Level ten can be reached only after you have successfully completed level nine. An extra Preppie is awarded when you reach a score of 8000.

Be sure to take the time to read the short story at the front of the instruction book. It gives you an uproariously entertaining and irreverent view of what a Preppie is.

The most surprising aspect of *Preppie!* is that Adventure International is the marketer of the program. AI, famous for its excellent text and graphics adventure programs, makes its first excursion into the world of arcade gaming with *Preppie!*.

The disk version contains a high-resolution picture not found on the tape version at the opening of the game. Disk users are offered a chance to purchase a back-up copy for $3.99.

If you are looking for a game that will please all ages, *Preppie!* is it. I have used this game in my school classrooms with kids ranging in age from five to thirteen, and they all loved to play it. It has replaced *Pac-Man* as their favorite game. Get *Preppie!*. It will entertain the kids and impress your friends.

Preppie!
Adventure International
Box 3435
Longwood, FL 32750
16K Tape
32K Disk
$29.95

©

ATARI PRICE WAR APPLE

WE WILL NOT BE UNDERSOLD

ATARI HARDWARE

ATARI 800 48K **619.95**

810 Disk Drive	419^{95}
400 16K	259^{95}
400 48K	409^{95}
850 Interface	155^{95}
410 Program Recorder	71^{95}

INTEC RAM BOARDS
(Lifetime Warranty)

16K 800	44^{95}
32K 400/800	64^{95}
48K 400	144^{95}

PRINTERS

Epson MX80FT	479^{95}
Epson MX100	669^{95}
NEC 8023AC	459^{95}
Cable for 850 to printer	29^{95}

PERCOM DISK DRIVES

Atari 1st Drive	579^{95}
Atari 2nd Drive	359^{95}

BOOKS

Your ATARI Computer	14^{95}
ATARI Games & Recreation	11^{95}

BLANK DISKS 5¼'' S.S., S.D.

Elephant Box of 10	21^{95}
Verbatim Box of 10	29^{95}

JOYSTICKS

Wico *(1 Year Warranty)*

Red Ball	24^{95}
TrackBall	49^{95}
12 Foot Extension Cord	8^{45}

ATARI SOFTWARE

ATARI, INC.

Pacman Cart	29^{95}
Centipede Cart	29^{95}
Educator	119^{95}
Programmer	54^{95}
Entertainer	69^{95}

BRODERBUND

Choplifter 48KD	24^{95}
David's Midnight Magic 48KD	24^{95}
Star Blazer 32KD	22^{95}

COSMI

Crypts of Plumbous	16KC	7^{95}
	16KD	12^{35}
Spider Invasion	16KC	10^{35}
	16KD	15^{95}

DATASOFT

Text Wizard 32KD	69^{95}
Spell Wizard 48KD	55^{95}
O'Riley's Mine 16K D/C	24^{95}
Fathoms Forty 16K D/C	24^{95}
Rosen's Brigade 16K D/C	24^{95}
Sands of Egypt 16KD	27^{95}
Canyon Climber 16K D/C	21^{95}
Shooting Arcade 16K D/C	21^{95}

DON'T ASK SOFTWARE

S.A.M. (Software Automatic Mouth) 8KD	41^{95}

INFOCOM

Zork I or II 32KD	27^{95}
Deadline 32KD	34^{95}

ON-LINE

Frogger C/D	24^{95}
Ultima I D	27^{95}

ROKLAN

Gorf D	27^{95}
Wizard of Wor 32KD	27^{95}

SIRIUS

Sneakers 48KD	21^{95}
Bandits 48KD	24^{95}
Way Out 48KD	27^{95}

SYNAPSE

File Manager + 40KD	69^{95}
Claim Jumper C/D	24^{95}
Apocalypse C/D	24^{95}
Picknick Paranoia C/D	24^{95}
Raptillian C/D	24^{95}

APPLE SOFTWARE

BRODERBUND

Arcade Machine	41^{95}
David's Midnight Magic	24^{95}
Choplifter	24^{95}
Star Blazer	22^{95}
Track Attack	21^{95}
Payroll	274^{95}
General Ledger W/Payables	344^{95}
General Ledger W/Receivables	344^{95}
Sea Fox	21^{95}
Hard Hat Noah	21^{95}

ON-LINE

Cranston Manor	24^{95}
Crossfire	21^{95}
The Dictionary	69^{95}
Mission Asteroid	15^{95}
Screenwriter II	89^{95}
Frogger	24^{95}
Pest Patrol	21^{95}
Lunar Leeper	21^{95}
Screenwriter Professional	139^{95}

SIRIUS

Way Out	27^{95}
Free Fall	21^{95}
Bandits	24^{95}
Borg	21^{95}
Sneakers	21^{95}
Joy Port	54^{95}

MISC.

Bookeeper Master	62^{95}
Bookeeper Check Writer	27^{95}
Visicalc 3.3	174^{95}
Visifile	174^{95}
Financial Partner	174^{95}
Zork I or II	27^{95}
Wordstar	259^{95}
Data Star	219^{95}

HARDWARE

System Saver	69^{95}
Microsoft Softcard	279^{95}
Microsoft Ramcard	149^{95}
Versawriter	229^{95}

RANA DRIVES

1st Drive W/Controller	389^{95}
2nd Drive	314^{95}

ORDERING INFORMATION:
ABOVE PRICES ARE FOR PREPAID ORDERS.
VISA, M/C ADD 3%.
ADD $2.50 FOR SOFTWARE ORDERS.
3% FOR HARDWARE ORDERS.
FOREIGN INQUIRIES INVITED—ADD 10% SHIPPING
(INCLUDES CANADA)
CALIFORNIA RESIDENTS ADD 6% SALES TAX.
30% DISCOUNTS ON ATARI, APPLE, IBM & TRS-80 SOFTWARE.

SPORT 'N' SOUND

VISA — mastercharge

290 North 10th Street
Post Office Box 1516
Colton, CA 92324-0821

Order Number Call TOLL FREE 800-652-8391
Inside California & All Other Inquiries (714) 783-0556

PRICES ON AD EXPIRE JANUARY 31, 1983

CALL OR WRITE FOR FREE CATALOG, PLEASE SPECIFY COMPUTER TYPE.

Player ZX81
A Tune-Playing Program For The Sinclair/Timex

Arthur B. Hunkins, School of Music, University of North Carolina, Greensboro

Player ZX81 is a 1K tune-playing program available in versions for all Sinclair/Timexes (including 4K ROM). Although each cassette is specific to a single version, the six pages of instructions cover them all. For anyone interested in coding tunes into the Sinclair and playing them back – over and over again if you wish – the $6.95 spent on this program is a bargain.

This is especially true considering the fact that you also get *81 Space Muse-AK*, a program that randomly generates and plays 127 pitches and rhythms, generates another set, etc. In short, it's guaranteed to drive you and your neighbors batty – a kind of mindless musical autopilot.

The procedure in *Player ZX81* is to code a duration number (all durations are relative) and pitch value (from the handy pitch table) for each note. You can have, even on a 1K machine, up to 127 notes. The limitations are these: 1) The single tone color available is a square wave; 2) Only pitches from the B above middle C on up are available (i.e., *high* pitches); 3) You can't easily do rests, but I'll describe a fix for this presently.

The external hardware required is a high-gain amplifier/speaker. A Radio Shack mini- or telephone amplifier/speaker at $10-$12 will do the job nicely. Or you can use a regular hi-fi amplifier, connecting to its phono input.

Versatile Modifications

Several program modifications lend further versatility.

All modifications start with two steps: 1) Omit the recommended protecting of high memory while making the modification; and 2) Immediately after loading, POKE 16544,28 and POKE 16600,28 – this renders the BASIC code accessible.

One thing you may want to do is to relocate the note table – according to how much memory you have and what you may want to add to the program. First, POKE 16549 with the same value you'll POKE into 16389, namely the page number for the beginning of the table. The lowest possible number is 67, the top page of 1K. Each additional K of memory is four pages; for example, the top page of a 2K Timex is 71. Whatever starting page you choose, change the memory location in statements 105 and 120 to 256*page number, and to this value +1 in line 117.

If you want a single play, instead of infinite repeat, delete statement 150. Or insert a PAUSE statement, a do-nothing FOR/NEXT loop, between 140 and 150 in order to space the repeats. Or again, use a statement 145 IF INKEY$ = " " THEN GOTO 145, to wait until you press a key.

If you have more than 1K memory, you may wish to use it to code longer songs. What you have to change here is line 95. The principle is this: a page of memory holds 127 notes; two pages hold 255. Subtract one from these numbers, and multiply by two to get the loop value to plug in line 95. Given available memory, there is no limit to the number of notes you may specify.

I mentioned the problem of rests. Here we have to modify the machine language routine itself, by changing statement 10 (in the process you can lop off the final zero in the statement, if you like). Be sure to follow these instructions *precisely*; any mistakes or omissions will cause a system crash. The idea is to reserve the frequency value of 255 as a flag for the rest specified as the duration value. In other words, every time a frequency of 255 is specified, you get a rest instead.

Bring statement 10 down to the bottom of the screen for editing. Locate the <= character. Add the following sequence of three characters immediately *before* <=: a lower left quarter square graphic, a C, and an upper right quarter square graphic. (Graphics characters are shifted, and must be preceded and followed by pressing the graphics key.) Then, after the RETURN that follows <=, insert a left half square graphic. In addition, change the following three characters: LOAD to FOR, DIM to FAST, and the last PEEK to INT.

INT is a function, and is prepared by pressing the function key. FOR and FAST are keywords, and are registered by pressing THEN, the keyword, and finally deleting THEN. Double check the code, and SAVE the program before RUNning it. If all is well, a frequency of 255 should now produce a (clickless) rest!

Finally, I recommend substituting the following list of "Pitch Nos." for those given by the author. Note that names duplicate Mr. Maples', but add two pitches on the top end: 250, 235, 222, 210, 198, 186, 176, 166, 157, 148, 139, 131, 124, 117, 110, 104,, 98, 93 or 92, 87, 82, 77, 73, 69, 65, 61 (B2), and 58 (C3). These values have been checked with a frequency counter.

Player ZX81 is a most useful program; the above modifications can make it even more so. Anyone interested in a "voice" for the Sinclair/Timex will find this a real value.

Player ZX81
Wm. Maples
688 Moore St.
Lakewood, CO 80215
$6.95

©

PET/CBM Standard Terminal Communications Package

Harvey B. Herman, Associate Editor

"Why did you buy a personal computer?" I have been asked this question several dozen times. **COMPUTE!** readers, as a group, should be able to reply with a dozen answers. One of my answers, "I wanted to use it as a terminal to communicate with other computers," would probably be a popular response.

How so, you ask? Well, the basic ingredients of a terminal are present in any small computer, e.g., a keyboard and a display screen. Thus, by paying a little more for terminal software and hardware, the user has the best of both worlds. That is, the computer can be used standalone for games or word processing, but also for accessing bulletin boards and large data bases over the telephone line.

Where does one find a good terminal program? There are several choices for PET/CBM computers, ranging from gratis (only a copying fee) up to $300 or more. However, the free program, attractive as it may sound, does require construction of hardware, which may be beyond many people's abilities.

The *Standard Terminal Communications Package,* reviewed here, is priced intermediately between those two extremes. It comes with all necessary hardware and software, ready to run on any Commodore system (except the 2001 PET with Original ROMs). The hardware, based on a 6850 ACIA chip, plugs easily into an empty ROM socket; clip on a few wires, connect your modem, and you're ready to run the terminal software supplied on disk. (Specify what computer and disk drive in your initial order to receive the proper version.)

The initial program loaded is written in BASIC, and is used to load, partially configure, and run the machine language terminal program. For example, you are prompted for the printer device number and the communication format (device four and eight bits-no parity, in my case).

Fifteen Different Options

The terminal program also offers a full configuration menu. You can specify full- or half-duplex, set a timer, toggle a printer on or off, etc. I counted 15 options. A status line at the top of the screen informs the user of the current state of the program, just as a much higher-priced, dedicated terminal would.

This package has some very attractive features. The best one is the ease with which it uploads and downloads BASIC programs. I have previously published in **COMPUTE!** two "how-to" articles on this subject. This program simplifies that process to the point where anyone, even a complete novice, can do it. So much for my arcane knowledge!

There are a number of minor points which I didn't like:

1. No end-of-line bell.
2. No option for line feed with an ASCII printer.
3. Does not always tokenize "IF" during downloading.
4. The manual did not make it clear that the modem must be off when configuring.
5. The delete key sends a backspace (change $23DE from $08 to $7F, if desired).

Let there be no mistake – this is a very worthwhile package. It is reasonably priced and is offered by a very reputable company which promises updates and program maintenance. Do not underestimate the importance of this service, particularly if you don't have the time or expertise to do it yourself. My wife and I have been using *STCP* regularly since it was received for review. Unlike most people, we have a choice of terminal software, and the fact that we continue using this program is our highest recommendation.

Standard Terminal Communications Package
Eastern House Software
3239 Linda Drive
Winston-Salem, NC 27106
$129.95

©

COMPUTE!
The Resource.

A Financial Wizard For Atari

Tina Halcomb, Carrollton, TX

If you want to use a finance system, but don't want to spend several days trying to learn how to use one, then *A Financial Wizard* by Computari of Dallas, Texas may be just what you need.

A Financial Wizard is an autoboot program that requires an Atari 400 or 800, an 810 disk drive, and at least 24K of memory. A printer is optional. It works with the Epson with Graftrax, NEC, Prowriter or Centronics 739.

The illustrated manual that comes with this program is clear, direct, and very thorough. It won't take long to get the system set up and working for you. Procedures appear in the manual in the order in which they should be followed, so you can just learn as you go. On the back cover of the manual you will find a (tear-out) command reference card.

To prevent costly errors that will result in data damage and/or loss, error warnings appear throughout the manual and on the screen during program execution.

It appears that this finance system was designed to achieve the best and most comfortable working relationship between the user and the program.

The Check Entry routine (item #1, main menu) is the most attractive feature of this finance system. Data prompts are very clear, and the category item names are displayed at all times during data entry for your convenience. It keeps track of the number of the last check entered and displays it for you. The balance of your checking account is kept current as checks are entered.

Also, if you make a mistake while recording your checks, you can correct it immediately. You are given the ability to split one check into more than one expense category. If you write a check at your bank for cash and want to account for everything it was used for, then you will be allowed to itemize it using this feature; and you will want to use it if you are striving for accuracy in your accounting. Scanning your entries is made possible by pressing START. You can see records very quickly this way. The correction capability is offered during scanning also.

Menu item #2 – Budget Entry allows you to set up a projected budget on a monthly basis. The category names provided by the program are generally used by almost all of us. However, if Spot is sick more often than you are, you could change the "Medical" category to "Vet" with the category change routine in the Utilities program.

The same applies to all categories. If your budgeted allowance for one category is the same for more than one month, you can replicate a budget from one month to the next by entering a "/" next to the category letter. An entire budget plan can also be replicated when you start to set up a new month by entering a "C" and the number of the month you want to copy from in the salary block.

Superior File Searching

The file search capabilities of this program are superior. You are offered seven ways to look up the checks. You may search by Name of Payee, Category Name, Sub-Category Name, a range of Check #'s, a range of Months, a range of Days, and a range of Amounts.

The Tabulation section figures what percentage of your salary is spent on each category. This can be very revealing. The Bargraphs clearly display what you have spent with respect to your planned expenses. A thick colored bar represents your expenses and a thin contrasting line overlays this bar to show your budgeted amounts. You have a choice of seeing either one month's expenses in all categories or one category shown over a 12-month period.

The Checkbalancer routine follows the standard procedure shown on the back side of your bank statement. Marking cancelled checks has been made extremely simple. The checks are addressed sequentially and displayed on the screen for you. To mark a check that has cleared your bank, you type an asterisk next to it. If you accidentally cancel a check that has not cleared the bank, you can reverse it by typing an "X".

The Checkwriter routine will print your checks for you, with custom checks available through Abacus Software.

The Utilities program, item #8, consists of seven utilities that you will need to maintain accurate records. With this program you will transfer records to a permanent storage disk at the end of a financial period and prepare the disk for the new year.

This system is disk intensive. All data is saved automatically and immediately following all routines that either enter data or modify it.

Overall, this is an excellent finance system – entertaining, accurate, and fun to use.

A Financial Wizard
Available From:
On-Line Computer Center
10944A N. May Avenue
Oklahoma City, OK 73120
(405)751-2701
$59.95

(An earlier version of *A Financial Wizard*, known as *Personal Finance for the Atari*, may be upgraded for $10. Users should return their master disk and a check or money order to On-Line Computer Center.) ©

Automate Your Atari

Joseph J. Wrobel, Rochester, NY

The Atari Disk Operating System (DOS) supports the use of a file named AUTORUN.SYS that has a very special characteristic. At system start-up, the DOS loads and runs this file automatically if it exists on the mounted diskette. This allows you to arrange for your Atari to come up smart.

The Potential

The AUTORUN.SYS file could contain a machine language program that loads and runs. It could also contain just a short program to do some routine operations like setting the screen margins or color before passing control to BASIC. However, the major use I've seen for AUTORUN. SYS is to direct the system to load and run a BASIC program. Not only does this type of operation save you some time and effort, but it also allows an unskilled operator, like a student, to turn on the machine and interact with an application program without getting into the details of the LOAD or RUN instructions.

The Problem

So far, so good. Why doesn't everyone use the AUTORUN.SYS file? Apparently the major obstacle to its more widespread use is the fact that it is a machine language routine. Thus, it requires knowledge of 6502 machine language and, for complex operations, some knowledge of the intricacies of the Atari Operating System to create a functional AUTORUN.SYS file. Unless someone were to come up with a program to do it for you.

Automate (Program 1) is just such a program. If you key in this program correctly and run it, Automate will help you create your own personal AUTORUN.SYS file, and it won't hurt a bit. The program starts by asking you to input the series of commands you wish to be executed at start-up. You enter the commands exactly as you would if the machine came up in its normal ready state. The only limit on the number of commands is that the total number of characters entered may not exceed 196 (including the Atari end-of-line character added each time you hit RETURN). The program keeps track of the number of characters entered and will prevent you from exceeding this limit. After you've entered the final command in the sequence, the program will create an AUTORUN.SYS file on the mounted diskette. Note that any previous AUTORUN.SYS file will be overwritten by this operation.

The next time you boot up from the diskette bearing the AUTORUN.SYS file, the AUTORUN.SYS program will be run. This will cause the commands you entered to be executed in the order they were entered (although they will not be displayed), then control will be returned to the system. The commands, of course, must be compatible with the cartridge in use (BASIC, Assembler Editor, etc.) or an error will result. If at any time you wish to boot up from a diskette and circumvent the AUTORUN.SYS file, just hold the OPTION key down until system initialization is complete. The AUTORUN.SYS file created by Automate checks that key and, if it finds it depressed, the command list will not be executed.

A BASIC Example

To demonstrate the use of the program, a single command BASIC example will be presented. Let us suppose there exists a BASIC program entitled BEGIN which you would like to run automatically at start up. Using AUTOMATE, you enter (as Command #) the statement:

```
GR.0:?"Autoboot in progress.":RUN"D:BEGIN"
```

then press RETURN. Assuming you entered the command correctly, you respond to the question:

```
Is that correct (Y/N)?
```

by pressing Y. When the program asks if there are:

```
More commands (Y/N)?
```

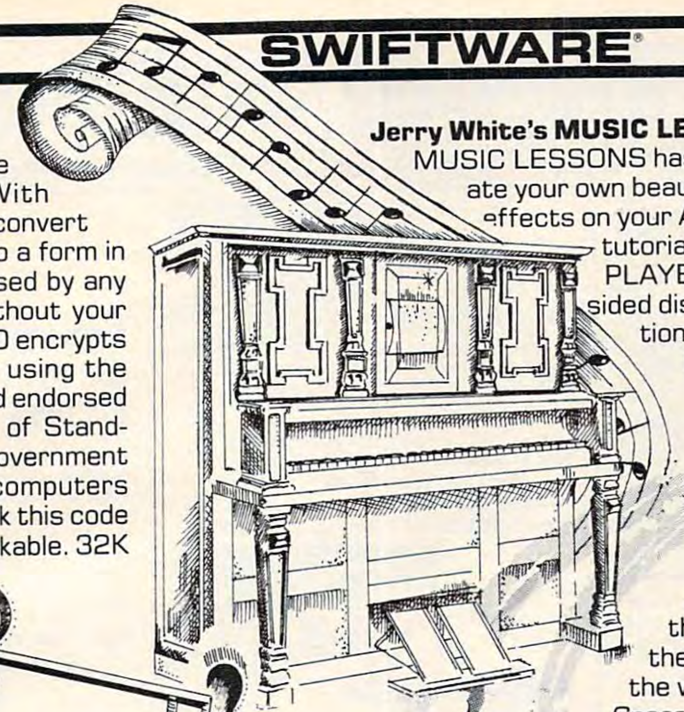
respond by pressing N. The program then creates the AUTORUN.SYS file and displays READY when it's done. If you now turn off your computer and switch it on again, you will find that it "comes up" running program BEGIN. How simple can you get?

Description Of Operation

This section is for those who are not satisfied with just running the program, but are also interested in knowing how it works. Let's first take another look at Program 1. Automate consists of three major sections. The first section (lines 50 through 130) are for documentation and initialization. The program employs two key numeric variables: I, which counts the number of commands entered, and L, which counts the total number of characters in the command list. The second program section (lines 140 through 350) INPUTs the commands one at a time. As each command is entered, the program allows for error correction, checks command list size, packs the command into B$ and tacks on an Atari end-of-line (EOL) character, namely CHR$(155). The third section of the program (lines 360 through 600) actually creates the AUTORUN.SYS file.

Before this third section is discussed, I direct your attention to Program 2. This is the assembly

SWIFTWARE®



CRYPTO 800

Protect your valuable programs and data! With CRYPTO 800 you can convert your programs or data to a form in which they can not be used by any unauthorized parties without your secret key. CRYPTO 800 encrypts and decrypts your files using the data encryption standard endorsed by the National Bureau of Standards for use by many government agencies. The fastest computers would take years to break this code making it virtually uncrackable. 32K Disk. $39.95

Jerry White's MUSIC LESSONS

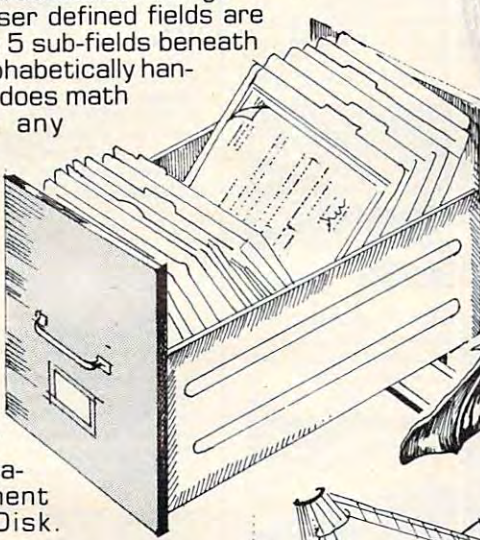
MUSIC LESSONS has everything you need to know to create your own beautiful music and a wide range of sound effects on your ATARI computer. This comprehensive tutorial contains 13 separate programs and PLAYER PIANO on 2 cassettes or 1 double sided disk and includes extensive documentation complete with program listings. 32K DISK 16K CASSETTE (24K required for PLAYER PIANO). $29.95

SPACE SHUTTLE by Paul Kindl

Join the crew of the *Space Shuttle* as they prepare to take the next step into the world of space travel. Take control of the world's first reuseable spaceship, the *Space Shuttle*, and in an accurate full graphic simulation, place yourself in the cockpit. Pilot the *Space Shuttle* through take-off with booster stage separations, orbit, descent down the glide path and landing to touch-down—complete with a chase plane and scrolling runway visible through the cockpit windscreen. You assume command throughout all phases of the mission aided by complete instrumentation. 32K Diskette. $29.95

FILE • IT 2 + by Jerry White

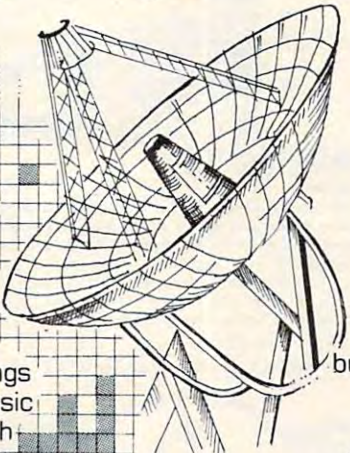
A powerful financial database management system. 6 user defined fields are created with up to 5 sub-fields beneath each main field. Alphabetically handles data and also does math computation on any selected fields. Data files are stored on separate disks with full field and sub-field sorting with file merging. Supports up to 4 drives including the 128K Axlon Ramdisk. The "alternative" to more costly database management systems. 24K Disk. $49.95



HAUNTED HILL by George Richardson

Fight bats and ghosts in the dark of the cemetery. This exciting, all machine language game has arcade quality graphics and speed. Requires Joystick.

$24.95 16K Cassette/$29.95 16K Disk.



P/M 800 by Fred Tedson

The ULTIMATE Player/Missile Editor

P/M 800 gives you complete control of all graphic functions: 5 players, colors, missiles, resolution, priority settings, etc. You create and save players and missiles as strings that can be incorporated into your own basic assembler programs allowing smooth arcade type action. Complete with a users tutorial that takes you step by step, exploring the fantastic graphic capabilities of the ATARI 400/800 computer. 32K Disk. $39.95

DATALINK by Tony Dobre

Top rated by national magazine reviews, purchased by NASA, this ultra-sophisticated menu-driven multioption smart terminal communications package supports uploading/downloading in full-duplex or simplex modes. Compatible with all the commercial services and bulletin boards such as the Source and Compuserve, etc. 24K Disk. $39.95

- **AVAILABLE AT SELECT COMPUTER STORES**
- **MAIL ORDERS:** Send check or money order plus $2.50 shipping and handling. N.Y. Residents add 7¼% sales tax.
- **TELEPHONE ORDERS:** (516) 549-9141

Send for FREE Catalog • Dealer Inquires Invited

SWIFTY SOFTWARE, INC.
64 Broad Hollow Road
Melville, New York 11747

MasterCard VISA

A January Jubilee of

For Your Atari 400/800

rep•u•ta•tion: *a place in the public esteem or regard; a good name.*

The Program Store, one of the original software companies of the '70s, began to develop its solid reputation for selection and service right from the start. Today, it offers the largest selection of Atari software in America, and backs every piece of inventory with the trusted Program Store name.

GOLF CHALLENGE

By Schwab from On-Line

Play an exciting game of golf *without* getting up at dawn! GOLF CHALLENGE features a moveable golfer on your choice of 18 holes or the front or back 9. True to its name, there are lots of challenges: water hazards, sand traps, trees. Screen shows the fairway or a close-up of the green for teeing off and putting. 1-4 players. Requires joystick.

16K Tape, $24.95

LEGIONNAIRE

By Chris Crawford from Avalon Hill

The *NEWEST* from Chris Crawford, famous author of *EASTERN FRONT*, Winner of the Best Computer Game of 1981!

One of the most realistic war games ever created for the home computer! It's 58 B.C., and you're Julius Caesar, commanding your legions against the Barbarian tribes. The battlefield scrolls before you, displaying the 10 square foot map of the battle area. Are you up to the challenge? A real-time war game with hi res graphics and realistic sound effects.

16K Tape, $34.95 32K Disk, $39.95

PROTECTOR II

By Mike Potter from Synapse

You must transport all your people from the city under attack by Fraxullan Slimehordes. You must lift them one by one ***first*** to the City of Hope on the far side of the volcano; ***then*** to safety at the Verdann Fortress. Beware the Xytonic Pulse Trackers, meteoroid showers and the Laser Caves of Fear. But ***hurry***, before the volcano erupts!!

32K Disk, ~~$34.95~~ Save 20%
Now Thru Jan. 31 You Pay Only $27.96

ANDROMEDA CONQUEST

From Avalon Hill

Exciting strategy simulation in which 1-4 players compete to form and protect their galactic empires. Each player starts with 10 resource points, accumulating more through skillful selection and management of far-flung star systems.

32K Tape, $17.95 40K Disk, $22.95

GFS SORCERESS

From Avalon Hill

It's 2582, and you're Joe Justin on the Galactic Federation Starship SORCERESS. Can you clear yourself of the unjust charge of mutiny? And what part will be played by the lovely Selena Sakarov?

40K Tape, $29.95 40K Disk, $34.95

STARCROSS

From Infocom

Tell your ship's computer your course coordinates, and strap in for your exciting adventure into space in search of the legendary black holes. Board the drifting artifact ship; explore its many rooms and ringed corridors. Encounter alien beings like the large spider who bubbles with conversation, and enjoys gadgets. Some, however, are not so friendly!

32K Disk, ~~$39.95~~ Save 20%
Now Thru Jan. 31 You Pay Only $31.96

SURVIVAL OF THE FITTEST

From IDSI

Fascinating program based on the game of "Life." It uses—and extends—the "rules of the universe" from "Life" (included in the package). Both players start with 99 seed cells which they try to "plant" in the most effective shapes on the playfield. The computer keeps continuous count of the cells as shapes grow or shrink according to the rules. Vivid color and sound effects. 2 joysticks required.

ROM Cartridge (16K), ~~$39.95~~ Save 15%
Now Thru Jan. 31 You Pay Only $33.96

VC

From Avalon Hill

Can you change history? You command the South Vietnam army; the computer controls the Viet Cong (VC) and North Vietnamese Army. Can you win the hearts and minds of the people, and destroy the VC units in your province? Challenging operational level combat game with hi res graphics.

40K Disk, $24.95

THE INSTITUTE

By Jyym Pearson from Med Systems **

Five games in one! You begin in an institution, and must successfully negotiate not just the Institute itself, but four other dream adventures as well. Objects in each dream help solve problems in the next. ***Concentrate***, or you may ***never*** escape the nightmare of the Institute!

16K Tape, $19.95
32K Disk, $22.95

TAX DODGE

By Freeman & Westfall from Island Graphics

The "truly taxing" maze game: negotiate the maze, collect your money, survive inflation, and avoid those trying to tax you. Each round represents a year in the life of the taxpayer—you must keep more money each round to go into the next one.

32K Disk, $39.95

Over 1500 Programs for TRS-80, ATARI 400/800, APPLE & IBM.

For Information Call 202-363-9797

Visit our other stores:
829 Bethel Rd., Columbus OH
Seven Corners Center, Falls Church, VA
W. Bell Plaza, 6600 Security Blvd., Baltimore MD

the PROGRAM STORE™

To Order Call Toll-Free 800-424-2738

MAIL ORDERS: Send check or M.O. for total purchase price, plus $2.00 postage & handling. D.C., MD. & VA.: add sales tax. Charge cards: include all embossed information

4200 Wisconsin Avenue, NW, Dept. 10-01-3 Box 9582 Washington, D.C. 20016

Great Games and Books

For Your Atari 400/800

the PROGRAM STORE™

YOUR ATARI COMPUTER

By Poole, McNiff & Cook from Osborne/McGraw-Hill
Comprehensive, all-in-one guide for Atari 400/800 computers that is helpful to the beginning and advanced programmer. Complete operating instructions; detailed BASIC programming, including a handy alphabetical glossary of BASIC statements and functions; advanced BASIC; computer graphics; and tips on hardware, peripherals and compatible software.

Softcover, $16.95

COMPUTE!'S SECOND BOOK OF ATARI

From Compute! Books
An **all new** collection of **never-before** published articles on the Atari for beginner to advanced users. Topics include: Utilities, Programming, Graphics, Applications and Beyond BASIC.

Spiral-bound, $12.95

GAMES & RECREATIONS

From Reston
This new book is packed with games and ideas on how to create your own. You'll discover the special Atari capabilities of the keyboard, graphics, sound and color. The book provides an entertaining way to learn more about general programming, too.
Part I: Learning Through Games. Part II: Games for the Atari. Part III: The Atari Special. Plus seven appendices.

Softcover, $14.95 **Hardcover, $19.95**

MASTER MEMORY MAP

From Santa Cruz Software
If you are serious about programming the Atari, this booklet could become one of your most important tools. A highly detailed map of the Atari's memory, it details thousands of locations and routines. Using this booklet makes easier the use of display list, player/missile, and interrupt graphics. The "Miscellaneous Notes" section contains a wealth of knowledge picked up by Santa Cruz in their explorations of the Atari. Also included are notes on the new GTIA graphics chip.

Softcover, $6.95

VISICALC HOME AND OFFICE COMPANION

By Castlewitz and Chisausky from Osborne/McGraw Hill
Fifty VisiCalc models for home and office, including: investments, inventory, sales forecasts, payroll, personal net worth, home budget planning, family insurance needs. Each comes with model listings, sample printed reports and narrative.

Softcover, $15.95

6502 ASSEMBLY LANGUAGE SUBROUTINES

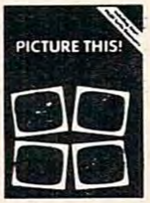
By Leventhal and Saville from Osborne/McGraw-Hill
If you're interested in using assembly language quickly, this book is ideal. It describes general 6502 programming methods, provides code for more than 40 subroutines to help you improve your programming skills, debug or revise an existing program; add instructions and addressing modes.

Softcover, $15.95

ATARI SOUND AND GRAPHICS

By Moore, Lower and Albrecht from John Wiley
This self-paced, self-teaching guide will have you seeing and hearing things on your Atari in no time—even if you're a complete beginner. You'll learn to compose and play melodies, draw cartoons, create sound effects and games. Each section teaches something new in BASIC, the most commonly used computer language.

Softcover, $9.95



PICTURE THIS!

An Introduction to Computer Graphics for Kids of All Ages

By David D. Thornburg from Addison-Wesley
This book promises to become the "modern replacement for coloring books and crayons". It's a learn-by-doing manual that uses PILOT, a simpler language than BASIC, and Turtle Geometry to teach kids to create pictures in full color from simple lines to complex angles and curves. Recommended for use in conjunction with PILOT Cartridge.

Spiral-bound, $14.95

ATARI ASSEMBLER

By Don & Kurt Inman from Reston
While the Atari Assembler Cartridge comes with an operating manual, it assumes that you already know assembly language. If you're new to the Atari or its 6502 processor, this book is a must. The Inmans guide you through the rudiments of this fascinating type of programming in clear, easy steps. Includes full listing and description of 6502 mnemonics and addressing modes. Recommended for use in conjunction with Assembler Cartridge.

Softcover, $12.95

INSIDE ATARI DOS

By Bill Wilkinson from Compute
The comprehensive manual on the disk File Manager System (FMS), commonly known as Atari DOS 2.0S. Contains the only complete and official listing for the system, plus a full description of: the external view, charts & tables, various interfaces and functions of individual subroutines.

Spiral-bound, $19.95

DE RE ATARI

From APX
Translated from Latin, the title of this book is "All About Atari" and it means what it says! Used in combination with Atari's Technical Reference Manual, advanced programmers will be able to learn to exploit the many hardware and operating system features that make the Atari 400/800 so tremendously versatile. Includes a useful discussion of the new GTIA chip. Once you know Atari BASIC and assembler, this book is a must.

Loose leaf (binder not supplied), $19.95

COMPUTERS FOR KIDS

Atari Edition By Sally Larsen from Creative
The BASIC programming manual written for kids, from flowcharts to color graphics, including the sure-to-please program, "Scare Mom with an Elephant." Detailed instructions and sketches plus glossary of statements and commands. With lesson plans and tips for parents and teachers.

Softcover, $4.95

STIMULATING SIMULATIONS,

Atari Version, 2nd Edition

By C.W. Engel from Hayden
A handbook of 12 simulation games including Art Auction, Starship Alpha, Monster Chase and Devil's Dungeon—each complete with listing, sample run, instructions and program documentation, including flowchart and ideas for variations.

Softcover, $5.95

THE PROGRAM STORE • Dept. 10-01-3 • Box 9582 • 4200 Wisconsin Avenue, NW • Washington, D.C. 20016

Item Tape/Disk/Book Price Postage $2.00 Name________
________ Total ________ Address________
________ ☐ CHECK ☐ VISA City________ State________ Zip________
________ ☐ MASTERCARD Card #________ Exp________
________ Computer ________

listing for the core of the AUTORUN.SYS program. What this machine language program does, in a nutshell, is to temporarily take over the task of supplying screen editor data by substituting a new device handler table and "get character" routine for the default ones provided by the operating system. At system start-up while the AUTORUN.SYS program is active, it intercepts all the keyboard entry requests and feeds out, one character at a time, the commands which you have entered. When it has sent out the last character of the last command in the list, it re-installs the default screen editor handler table, and the system takes over from there.

Returning to the section of the BASIC program which creates the AUTORUN.SYS file, you will find that it consists primarily of three loops. Loop one (lines 490 through 510) PUTs the core program and its associated six byte header into the file as READ from the DATA statements in lines 430 through 480.

Note that in line 500 of Automate, two numbers are changed from the values shown in the DATA statements before putting them into the AUTORUN.SYS file. The first is a byte in the AUTORUN.SYS file header which gives the end of the program when loaded in memory. This is the sum of the core program length and the number of bytes in the command list. Automate also alters the value of the immediate argument of the CPY instruction in line 370 of Program 2. This byte is set equal to the total number of characters (including EOL's) in the command list. Loop two (lines 530 through 550) PUTs in the command list which resides in B$. Finally, loop three (lines 580 through 590) adds a twelve byte postscript to the file which provides the system with the initialization and run locations for the routine.

The BASIC program here provides an easy way to create a useful AUTORUN.SYS file. There are dozens of ways this file can be used. It doesn't necessarily have to be a serious application. For example, it's sort of fun just to start up my machine, listen to it go through its disk machinations, then see it automatically display the personalized greeting:

READY WHEN YOU ARE, J.W.!

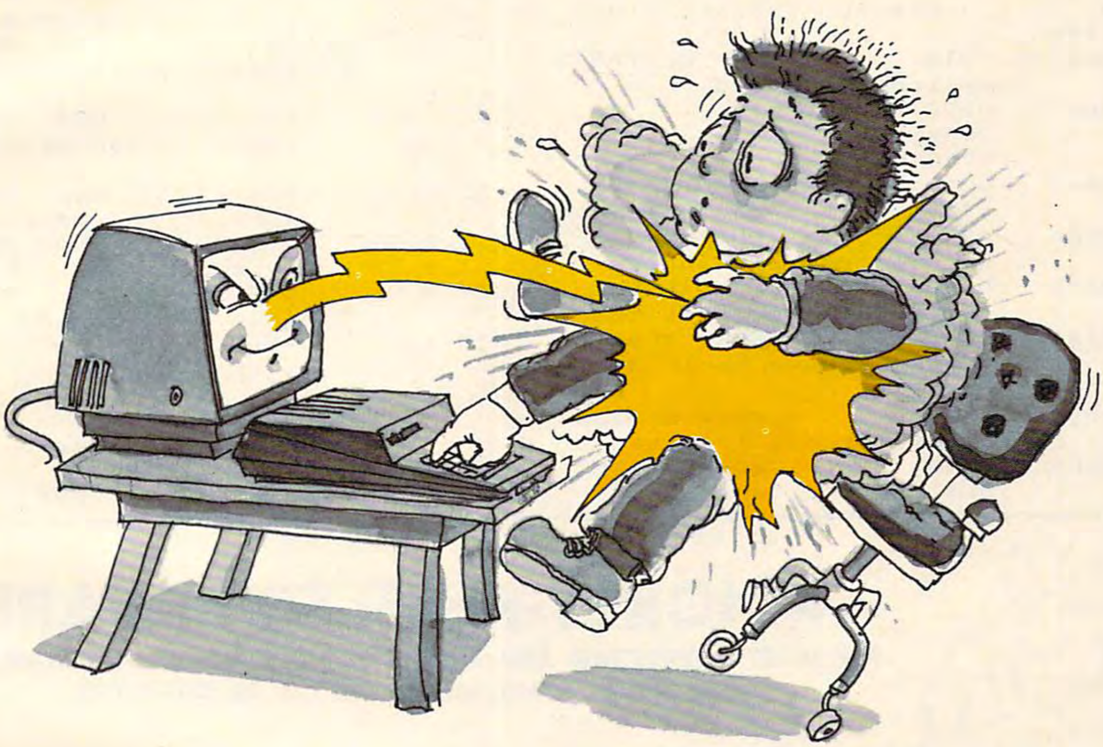
Program 1.

```
50 I=0:L=0:MAX=196
60 DIM A$(MAX),B$(MAX),R$(1)
70 OPEN #1,4,0,"E:":OPEN #2,4,0,"K:"
80 ? "This program helps you to creat
   e"
90 ? " a personalized AUTORUN.SYS fil
   e"
100 ? "  which, following the disk bo
    ot"
110 ? "{3 SPACES}process, automatical
    ly issues"
120 ? "{4 SPACES}a set of commands th
    at YOU"
130 ? "{5 SPACES}specify."
140 I=I+1
150 ? :? "Please enter command #";I;"
    ."
160 ? :INPUT #1;A$
170 POKE 766,1:? :? "Command #";I;":"
    ;A$:POKE 766,0
180 ? :? "Is that correct (Y/N)? ";:G
    ET #2,X:? :R$=CHR$(X)
190 IF R$="Y" OR R$="y" THEN 220
200 IF R$="N" OR R$="n" THEN 150
210 GOTO 170
220 X=L+LEN(A$)+1-MAX
230 IF X<=0 THEN 260
240 ? :? "Command #";I;" is ";X;" cha
    racter(s)"
250 ? "too long.":I=I-1:GOTO 270
260 B$(L+1)=A$:L=LEN(B$):B$(L+1)=CHR$
    (155):L=L+1
270 ? :? "Current command list:"
280 POKE 766,1:? :? B$:POKE 766,0
290 IF L>=MAX-1 THEN ? "Command list
    is full.":? :GOTO 370
300 ? "Command list can hold ";MAX-L-
    1;" more"
310 ? " character(s)."
320 ? :? "More commands (Y/N)? ";:GET
     #2,X:R$=CHR$(X)
330 IF R$="Y" OR R$="y" THEN 140
340 IF R$="N" OR R$="n" THEN 360
350 GOTO 300
360 ? CHR$(125);
370 ? "Mount diskette which is to bea
    r"
380 ? " the AUTORUN.SYS file, then"
390 ? "   press RETURN. ";:GET #2,X:CL
    OSE #1:CLOSE #2
400 ? CHR$(125);:? "Writing AUTORUN.S
    YS file."
410 OPEN #1,8,0,"D:AUTORUN.SYS"
420 REM PUT OUT THE HEADER AND THE CO
    RE MACHINE LANGUAGE PROGRAM
430 DATA 255,255,0,6,59,6
440 DATA 173,31,208,41,4,240,10,169,1
    8,141,33,3
450 DATA 169,6,141,34,3,96,251,243,51
    ,246,33,6
460 DATA 163,246,51,246,60,246,76,228
    ,243,0,238,33
470 DATA 6,172,33,6,192,0,208,10,169,
    0,141,33
480 DATA 3,169,228,141,34,3,185,59,6,
    160,1,96
490 FOR I=1 TO 66:READ X
500 IF I=5 OR I=48 THEN X=X+L
510 PUT #1,X:NEXT I
520 REM ADD THE COMMAND LIST
530 FOR I=1 TO L
540 X=ASC(B$(I,I))
550 PUT #1,X:NEXT I
560 REM APPEND INITIALIZE AND RUN VEC
    TORS
570 DATA 226,2,227,2,0,6,224,2,225,2,
    17,6
580 FOR I=1 TO 12:READ X
590 PUT #1,X:NEXT I
600 CLOSE #1:? CHR$(125);:END
```

Program 2.

```
D01F        0100 CONSOL =     $D01F
0320        0110 DEVTAB =     $0320
E400        0120 OLDDHT =     $E400
```

LEARN TO TYPE OR GET BLOWN TO BITS.



MasterType™ and Atari™ make typing a blast.

Now anyone can learn to type and have hours of fun at the same time. With MasterType. We've combined the fast-action of blow 'em up video games with the best instructional program available. You either learn to type or the aliens ZAP your spaceship into oblivion. The results? Highly motivated and enjoyable learning.

MasterType earns a ten-gun salute.

MasterType is fast becoming the best selling educational software product designed for use with the Atari 800 home computer.

It's no wonder, when you consider what InfoWorld magazine had to say:

"MasterType is an excellent instructional typing game. We had fun reviewing it, and we highly recommend it to those who want to learn typing in an unconventional but motivating way."

InfoWorld also went on to rate MasterType as "excellent" in all four of its categories.

MasterType teaches your fingers to fly.

Face it. If you're like most people, the most inefficient components on your computer are probably your fingers. But you can change all that with MasterType. In just a few easy lessons your fingers will be doing things you thought them incapable of.

MasterType. With 18 explosive learning levels. You'll either learn to type or get blown to pieces.

$39.95. Requires disk drive 32k and basic.

MasterType

Lightning Software ©

P.O. Box 11725, Palo Alto, CA 94306
(415) 327-3280

```
                  ;
0000                     *=     $0600
0600 AD1FD0       INIT   LDA    CONSOL    ;Load the console switch register
0603 2904                AND    #$04      ;and check for the OPTION key.
0605 F00A                BEQ    RUN       ;If it's pressed, branch to the RTS.
0607 A912                LDA    #NEWDHT&$00FF ;Otherwise, install the vector
0609 8D2103              STA    DEVTAB+1  ;to the new device handler table
060C A906                LDA    #NEWDHT/256 ;in the appropriate place in the
060E 8D2203              STA    DEVTAB+2  ;device table and
0611 60           RUN    RTS              ;return.
                  ;
0612 FBF3         NEWDHT .WORD  $F3FB     ;This is the replacement
0614 33F6                .WORD  $F633     ;screen editor handler
0616 2106                .WORD  GET-1     ;vector table.  All the
0618 A3F6                .WORD  $F6A3     ;vectors have their default
061A 33F6                .WORD  $F633     ;values except for the
061C 3CF6                .WORD  $F63C     ;GET routine, which
061E 4C                  .BYTE  $4C       ;points to the replacement
061F E4F3                .WORD  $F3E4     ;routine below.
                  ;
0621 00           COUNTR .BYTE  0         ;character counter
                  ;
0622 EE2106       GET    INC    COUNTR    ;Increment the character
0625 AC2106              LDY    COUNTR    ;counter.  Compare it with
0628 C000                CPY    #ENDLST-BEGLST ;the command list length.
062A D00A                BNE    CONT      ;If not equal, branch to CONT.
062C A900                LDA    #OLDDHT&$00FF ;Otherwise, reinstate the
062E 8D2103              STA    DEVTAB+1  ;default screen editor handler
0631 A9E4                LDA    #OLDDHT/256 ;table vector at the proper
0633 8D2203              STA    DEVTAB+2  ;spot in the device table.
0636 B93B06       CONT   LDA    BEGLST-1,Y ;Fetch the next character
0639 A001                LDY    #1        ;from the command list and
063B 60                  RTS              ;return.
                  ;
                  BEGLST
                  ;The command list goes here.
063C              ENDLST .END
```

©

All About Commodore's WAIT Instruction

Louis F. Sander, Pittsburgh

WAIT is one of Commodore BASIC's most mysterious instructions – seldom seen in programs, rarely mentioned in magazines, and nearly impossible to understand in manuals. But it's available for VIC-20, PET/CBM, and 64 users. To find out how helpful it can be for all kinds of applications (program debugging, single-stepping, even a superior form of the common pause GET K$: IF K$ = " " THEN), read on.

WAIT allows a BASIC program to communicate with hardware and with certain software external to itself. It causes PET to suspend all apparent activity on receipt of a signal from the keyboard, an external device, or the computer's internal timers. PET's normal activity resumes when the signal is removed. Thus, WAIT provides a simple means of pausing until a key is pressed, an interval ends, or contacts open or close. We'll soon get to some useful examples.

When executed, WAIT examines a selected memory location and halts the program if the location contains a specified "trigger value." The program continues if, or as soon as, any other value appears in the selected location. Optionally, WAIT can be made to ignore some of the bits in the location it is testing.

In other words, WAIT halts a program if, and for as long as, selected bits in a chosen location have one specific pattern. Note carefully: the program waits *if* a specific pattern exists, not *for* a specific pattern to appear.

WAIT's format is:

WAIT ADDR, MASK, TRIG

ADDR, MASK, and TRIG can be any numeric constants, expressions, or variables in the range 0-65535 for ADDR, and 0-255 for MASK and TRIG. TRIG and its leading comma may be left out of the statement if desired, in which case TRIG defaults to zero.

Technically speaking, the WAIT statement reads the status of memory location ADDR, exclusive ORs it with TRIG, then ANDs the result with MASK, repeating these steps until a nonzero result is obtained. *Practically* speaking, few human minds can *follow* such logic, let alone comprehend its effect on their programs. If you prefer simplicity, think of WAIT as saying this: "Pause if the MASK bits in the contents of ADDR are the same as those in TRIG. Otherwise, continue." But let's illustrate some of its specific uses.

ADDR is the address of the memory location to be tested. WAIT halts the program if ADDR contains a preselected trigger value, resuming execution if and when ADDR's contents change. It follows that ADDR must be a location whose contents can change independently of the program, or there will be no way to resume program execution. Relatively few memory locations meet this criterion – mainly they are associated with the keyboard, the user and IEEE ports, and the computer's internal timers. Table 1 is a partial listing of such locations.

MASK determines whether WAIT tests all, or only some, of the bits in ADDR. If a given bit in MASK is set to one, the corresponding bit in ADDR will be tested. Otherwise, the bit will be ignored. If the entire contents of ADDR are to be tested, MASK must equal 255; any lower number will cause WAIT to ignore one or more bits. The various powers of two are often used in MASK, to monitor a single bit for a one or a zero. Zero is a legal value for MASK, but should never be used, since it always causes an endless halt. (Any number and zero equals zero.)

TRIG is the value that triggers a halt. If WAIT is executed when ADDR contains TRIG, the program will stop until TRIG is replaced by another value. Of course, if MASK is blocking out one or more bits, any number whose unblocked bits are identical to those in TRIG will have the same effect as TRIG, and will cause the program to halt. TRIG's default value is zero, so when TRIG is omitted from the WAIT statement, a halt occurs whenever all the unblocked bits are zero.

WAIT has three other notable properties. First, just as PRINT can be abbreviated as "?", WAIT can be abbreviated as "W shifted A". You can use this property to save keystrokes and line space. Second, the STOP key will not terminate a WAIT. That can only be done by satisfying the logical conditions in the argument; if a programming error has made this impossible, you must reset your machine to recover. So as soon as you put a WAIT statement into a program, save a copy on tape or disk; that will save *you* if you've made an error. Finally, WAIT does not affect the jiffy clock – TI and TI$ continue during WAITs, even though the computer and the STOP key are ostensibly dead. So by using the memory locations of the jiffy clock, you can precisely control WAIT's pauses.

Real World Applications

Table 2 lists some of WAIT's uses, along with the arguments used to implement them. To demonstrate these applications, the following examples can be inserted as line 25 of this little program:

```
10 TI$="000000"
20 PRINT TI
30 GOTO 20
```

Lines 20 and 30 cause a continuous screen printout while the program is running, making it easy to observe the effects of the WAIT in line 25. (The following examples use ADDResses from Original ROMs; if you have PET Upgrade, 4.0, or a VIC or a 64, use Table 1 to find the right ADDResses for your machine.)

WAIT 59410,255,251 stops the program when SPACE is depressed, and continues execution when it's released. No other key can make the program pause if these arguments are used. Different TRIGs, of course, will activate different "59410 keys."

WAIT 59410,255,255 stops the program *until* SPACE or one of the other "59410 keys" is depressed. Only these keys can change the contents of 59410, and any one of them will do it, thus ending the WAIT.

WAIT 516,255 is a simple way to pause until a key is pressed. Since 516 contains a zero until SHIFT is pressed, this line halts the program until you press SHIFT. Notice that the STOP key has no effect unless SHIFT is pressed and the program is running. Also notice that WAIT 516,1 would have the same effect, using fewer bytes.

WAIT 59411,8,8 waits for a button on the tape drive to be pressed. While all the buttons are up, 59411's eight-bit is set, and the program halts. Depressing PLAY or any other recorder button clears the eight-bit, resuming execution of the program.

WAIT 59411,8 halts the program when the eight-bit is cleared, resuming when it's set. So, unlike the last example, this one stops when a button is *down*. Together, these two examples show how to use a one or a zero in any bit position to stop your program – just block out all the other bits and use TRIG to look for a one or a zero in the position of interest. This technique can be used to wait for a peripheral to signal that it is ready to proceed, assuming that the signal comes

Table 1: Some Useful Memory Locations

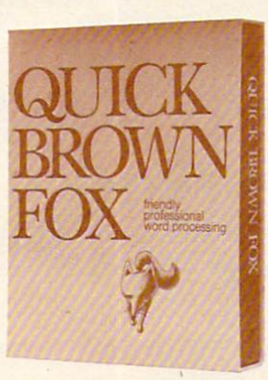
PET Orig. ROMs	PET Upgrade or 4.0 ROMs	VIC-20	64	Contents
512	141	162	162	Increments every jiffy (1/60 second).
513	142	161	161	Increments every 256 jiffies (4.2 seconds).
514	143	160	160	Increments every 65536 jiffies (18.2 minutes). Zeroing TI$ zeros all three clock locations.
515	151	197 (Note: returns keyboard matrix rather than ASCII value)	197	Unique value for the key pressed at the current jiffy. No key=255. Other values differ with ROMs.
516	152	653	653	Status of SHIFT key. Up=0, Down=1.
525	158	198	198	Number of characters in the keyboard buffer (0 to 9).
59410	59410	N/A	N/A	Senses certain keys. The keys vary with keyboards and ROMs, but these are the most common: For BASIC 1.0 and 2.0: RVS = 254; '[' key = 253; SPACE = 251; '<' key = 247; STOP = 239; '.' key = 191; '=' key = 127 For BASIC 4.0: '←' key = 254; '3' key=253; '6' key = 251; '9' key = 247; STOP = 239; ':' key = 223 Note that pressing a key zeros a bit in 59410. Pressing multiple keys zeros multiple bits.
59411	59411	37151 (Button down=62 all buttons up=126)		The eight-bit of this location tells the state of the buttons on TAPE #1. Button down = 0, all buttons up = 1.
59471	59471	37136		Parallel User Port.

Ad designed by Todd Lerner

WORD PROCESSOR?

You bet! Quick Brown Fox word processing software has more features than Word Star and runs on your VIC or Commodore 64. And it can grow and grow. Add memory, 80 column display, disks, even a letter quality printer. We'll show you how a 1st-class word processor can be yours for less than $2000! You can even have a "student" system for less than $700!

It all starts with the Quick Brown Fox at $65.



Call or write for our free brochure
Quick Brown Fox 548 Broadway New York NY 10012 (800) 547-5995 Ext 194
Dealer Inquiries Invited (212) 925-8290

by way of a line going high or low.

WAIT 514,128 pauses until the jiffy counter hits 128. See Table 2 for other valid MASKs for this purpose (WAIT ADDR,T).

WAIT 525,1 : POKE 525,0 waits for any key to be pressed. This is simpler than the more common

```
25 GET A$ : IF A$ = " " THEN 25
```

and it allows other statements to follow it on the same program line. The POKE is there to clear the keyboard buffer and can be omitted if there are no subsequent GETs or INPUTs in your program.

WAIT 525,2 : POKE 525,0 waits for *two* keys to be hit. There is no easy way to wait for three.

WAIT 513,255,PEEK(513) waits for the 4.2 second timer to increment. The screen display will prove that this happens every 256 jiffies.

WAIT 516,1,PEEK(516) illustrates an interesting technique. Whether SHIFT is up or down,

WAITing On The VIC-20 And Commodore 64

Doug Ferguson, Elida, Ohio

Joysticks Can WAIT

One of my pet peeves involves a game that uses a joystick for virtually all movement, but when it's time to indicate whether to play again, I have to put aside the joystick and hit a function key, type Y for YES, or hit the space bar. Why not use the joystick?

End-of-the-program questions are well suited for the WAIT command. To replay or not to replay is hardly a "menu" of choices. With WAIT, the computer "waits" for the replay signal. Even if the player wants to quit, he can always RUN/STOP-RESTORE or turn off the power.

The most suitable replay signal is the fire button, as in this VIC-20 example:

```
6000 PRINT "YOU WIN!!": PRINT" PRESS
     FIRE-BUTTON TO PLAY AGAIN"
6005 WAIT 37137,32: REM IN CASE BUTTON
     IS ALSO USED IN THE GAME ITSELF
6010 WAIT 37137,32,32
6020 RUN
```

Here is a table showing the specific test values not only for the VIC-20 but also for the Commodore 64:

		COMMODORE 64	
	VIC-20	Joystick 1	Joystick 2
FIRE	WAIT 37137,32,32	145,16,16	56464,16,16
LEFT	WAIT 37137,32,32	145,4,4	56464,4,4
DOWN	WAIT 37137,16,16	145,2,2	56464,2,2
UP	WAIT 37137,4,4	145,1,1	56464,1,1
*RIGHT	WAIT 37152,128,128	145,8,8	56464,8,8
ANY (except RIGHT on the VIC)	WAIT 37137,62,62	145,31,31	56464,31,31

*POKE 37154,127 before and POKE 37154,255 after the WAIT statement on the VIC.

This table assumes you want to test if the joystick is *pressed* a certain way. If you want to test that a certain position is *not pressed,* just leave off the last number (as in line 6005).

Tracing With WAIT

Another way to use WAIT is in FOR/NEXT loops in either program or direct mode. For example, to examine the contents of the ROM memory containing BASIC, type in the following program:

```
100 FOR X = 12 * 4096 TO X + 81
    91: PRINT X,PEEK(X)
110 WAIT 197,64
120 NEXT
```

or the direct statement

```
FOR X = 12 * 4096 TO X + 8191:
    PRINT X,PEEK(X):WAIT 197,64:
    NEXT
```

(In both examples, substitute 10*4096 for the Commodore 64.)

A list of memory addresses and contents will begin to scroll by. To stop printing, press any key (except RESTORE, SHIFT, CTRL, or the Commodore key). Printing resumes when the key is released. If the WAIT is changed to WAIT 653,1,1, the SHIFT key alone becomes the control key. This has the advantage of providing a "hands off" pause by using the SHIFT LOCK key.

It is also possible to single-step (go through a program line by line) using the WAIT command. Simply change the WAIT to

```
WAIT 197,64: WAIT 197,64,64
```

for "any key" control or

```
WAIT 653,1,1: WAIT 653,1
```

for SHIFT key control, although the SHIFT LOCK is of no consequence when single-stepping.

Escape from examining memory by hitting the RUN/STOP key.

NOW. The only real limitation for your VIC 20 is imagination.

The ARFON MICRO VIC 20 EXPANSION CHASSIS allows you to fully expand the VIC 20 memory, plug in interfaces, other computer peripherals, cartridges for expanded Basic language functions, programming utilities and even ROM cartidges of your own design to turn the VIC 20 into a sophisticated computer control system. In fact, with your VIC installed in the AFRON MICRO VIC 20 EXPANSION CHASSIS the only real limitation to the uses you can find for the VIC 20 is imagination.

VIC 20 SOFTWARE FROM ARFON MICRO, U.S.

TOTL TIME MANAGER is a set of two programs which allow you to create personal or business schedules, calendars of events, and checklists of activities. Organize reports by: person, project, or activity. Sort reports by: beginning or ending dates, or activity number. Print 56 different bar chart formulas. $25.

RESEARCH ASSISTANT 2.0 is a set of programs which allow you to keep track of reference data and create keyword cross reference lists. Keep data on reference sources: author, title, bibliography. Keep reference notes: page(s), text, up to 12 keyword cross reference, print data and or cross reference lists. $25.

TOTL LABEL 2.0 is a very useful mailing list and label program. TOTL LABEL 2.0 features easy editing, add or delete labels, define your own labels (width, length, number of printed lines), sort alphabetically or numerically and more. $20.

TOTL TEXT 2.0 is a complete word processing program which allows you to create and format professional looking documents. There is no limit to the length of a document. TOTL TEXT features: page numbering, paragraph control, right and left margin control, VIC printer expanded characters, skip to top of page, single, double, or triple spacing, centered title lines, tab position control, character and line spacing, upper and/or lower case, use of graphic characters, full screen editing, full cursor control while editing, scroll up and down through text, add, change and delete characters, insert and delete blocks of text. $25.

TOTL TEXT 2.5 has all the features of TOTL TEXT 2.0, plus up to 4 heading lines per page, footing line every page, footnotes, keyboard imput for form letters (up to 6 80-column lines), special characters for printer, right justify, and 3K additional working memory for editing. $35.

ALL TOTL 2.0 series programs work with tape and/or disk and require VIC 20, 8K expansion, cassette deck and/or disk drive, VIC printer or RS-232 printer.

BALDOR'S CASTLE is a fast-moving real time adventure—can you fight off 11 different types of monsters with just your bare hands, or will you need bow and arrow, magic sword, potions, and more to steal Baldor's gold? Game cartridge by Martin Kennedy. (More fun with joystick.)

ARFON VIC EXPANSION CHASSIS comes complete with aluminum cover/monitor shelf.

TOTL TEXT AND TOTL TIME MANAGER NOW AVAILABLE FOR THE COMMODORE 64

VIC 20 & COMMODORE are registered trademarks of Commodore Business Machines.

Ask your VIC Dealer, or Contact
ARFON MICROELECTRONICS, U.S.
111 Rena Drive, Lafayette, La. 70503
(318) 988-2478

this line waits for it to change. (Careful – if you changed ADDR to accommodate your ROMs, the PEEK must be changed to match it.)

WAIT 515,255,PEEK(515) does the same for any other key.

WAIT 59471,1,1 waits for the PA0 line on the user port to go low. Don't try this or the following examples unless you've configured the port for inputs and can control the lines.

WAIT 59471,1 waits for PA0 to go high.

WAIT 59471,1,PEEK(59471) waits for PA0 to change state.

WAIT 59471,3,2 waits if PA0 is low and PA1 is high. Otherwise, the program continues to run.

Of course, there are hundreds of other ways to use WAIT. If you understand the ones we've looked at here, you're ready to find and exploit the others. Here's one of them, to check your understanding: If X means we don't care whether a bit is 0 or 1, and if we want our program to pause as long as XXXX0101 appears in location 59471, but to continue on any other value, the proper statement is WAIT 59471,15,5. If you understand exactly why 15 and 5 are the proper arguments, you really *do* know all about WAIT.

Table 2:
Some Useful Applications

ADDR is the memory location to be tested.
CONT is ADDR's contents when tested.

Argument	Effect
WAIT ADDR,255,N	For N of 0 - 255, waits while CONT = N. Continues when CONT does not equal N.
WAIT ADDR,255,255	Waits as long as all bits in CONT are ones. Continues when any bit goes to zero.
WAIT ADDR,255	Waits as long as all bits in CONT are zeros. Continues when any bit goes to one.
WAIT ADDR,B,B	Where B is 1,2,4,8,16,32,64 or 128, waits while CONT's B-bit is one. Continues when the B-bit is zero.
WAIT ADDR,B	Where B is as above, waits while CONT's B-bit is zero. Continues when the B-bit is one.
WAIT ADDR,T	Where a counter in ADDR cycles from 0 to 255, and where T is 128, 192, 224, 240, 248, 252, or 254, waits while CONT is less than T. Continues as soon as CONT = T. ©

SJB DISTRIBUTORS. THE MOST COMPETITIVE PRICES ON COMMODORE PRODUCTS.

commodore

INTERFACES

ADA-1450 Serial	$149
ADA-1600 Parallel	149
RS232 cable for Vic or 64, 2m	30
Video/Audio cable for 64 & monitor	25

MONITORS — Great resolution for the CBM 64 or VIC

Panasonic, 13" Color	$375
Amdek Color I	330
NEC JB 1201M, 12" Color	330
NEC JB 1201, 12" green phosphor	170
Amdek Video 300L, green phosphor	175

BUSINESS SOFTWARE

Spellmaster Dictionary (great for WordPro!)	$199
OZZ Data Base System (8050)	240
Silicon Office (database, wp)	995 (New)
Wordcraft 80	289
VisiCalc (new expanded)	199
Dow Jones Portfolio Management System (RS232)	120
WordPro 4+ or 5+	299
The Manager	199
Legal Time Accounting	425
I.R.M.A.	295
BPI A/R, G/L, Job Cost, Inventory, Payroll	325/pkg

SJB will service any VIC or CBM64.

MasterCard, Visa, Money Order, Bank Check
COD (add $5) accepted.
Add 3% surcharge for credit cards.
In stock items shipped within 48 hours, F.O.B., Dallas, TX.
All products shipped with manufacturer's warranty.

TO ORDER CALL TOLL FREE
800-527-4893 800-442-1048 (Within Texas)

SJB will meet any competitive price under similar in-stock conditions.

SJB DISTRIBUTORS, INC.
10520 Plano Road, Suite 206
Dallas, Texas 75238
(214) 343-1328

Prices are subject to change without notice.

Business Hours:
M-F 8 to 6
Sat 10 to 2

SOFTWARE FOR CBM 64

Word Processing	$90
Computer Tutoring Game (COCO)	50
General Ledger	199
Pet Emulator (emulates 4.0 basic)	30
CBM EasyCalc (for the 64)	99
CBM EasyFinance	50
CBM EasyPlot	80
CBM EasyScan (appointment manager)	80
Sprite-Magic (build sprites on screen with Joystick, save to disk or cassette)	30
Assembler Package for CBM 64 (cassette) Editor (creates and updates source code), Assembler, Loader, Disassembler	50
Mail Mate	50
IEEE Interface (64)	100
Parallel Interface	90
RS232 Interface (modems, printers)	45

VIC PRODUCTS

VIC 20 Computer, 5K	$199
Vic Datasette Recorder	60
Vic 1541 Disk Drive	395
VIC MODEM (for CBM 64)	100
VIC 1525 Graphic Printer (for CBM 64)	325
8K Memory Expansion Cartridge	49
16K RAM	99
24K RAM	155
IEEE Interface (VIC)	85
Gorf (great arcade game)	30
Omega Race	30
Midnight Drive	23
VIC 3 slot Expander	43
VIC 6 slot Expander	83
Seawolf	23
Cosmic Cruncher	23

Arcade Joysticks — Heavy duty with 2 firing buttons! Great for the VIC or 64	**$25**
SuperPET (5 languages, 2 processors)	$1409
CBM 8032 Computer, 80 column	1029
CBM Memory Expansion, 64K	359
PET 4032, 40 Column	950
CBM 8050, 1 Mg. Dual Drive	1259
CBM D9060, 5 Mg. Hard Disk	2240
CBM D9090, 7.5 Mg. Hard Disk	2600
CBM 4040, 340K Dual Drive	919
CBM 2031, 170K Single Drive	489

PRINTERS — LETTER QUALITY

CBM 8300,40cps	$1450
Diablo 620,25cps	1350
Nec Spinwriter 7700,55cps	2350
Nec Spinwriter 3500,35cps	1600

PRINTERS — DOT MATRIX

CBM 4022,80cps/graphics	$395
CBM 8023,150 cps/graphics	599
Okidata 82A,120cps/serial or par	449
Nec 8023A(parallel)	499

Apple Machine Language Memory Aid

K. Lourash, Decatur, IL

"ML Helper" is a utility developed to assist fledgling Apple machine language programmers in studying 6502 object code when the original source code is not available, and also in adapting that code to their particular needs and systems. This program also works as is on OSI and can easily be modified for any Microsoft BASIC.

Options are offered in this program to list and modify zero page usage, to list and modify absolute addressing references, and to relocate the code under examination. Although written in Microsoft floating-point BASIC, this utility is readily converted to the other popular dialects. In fact, while my system is OSI, the listing is for Apple simply to involve a wider audience.

You may save ML Helper without REMarks. If you do, notice that line 31 may be incorporated into line 29, and line 35 into line 33, for increased program optimization. However, do not tamper with the "NEXT A" statement of line 51, since ML Helper will exit a loop without completing it; a simple "NEXT" there is insufficient.

In the interest of brevity, I chose to do no error checking of input from the keyboard. Thus it's easy to become careless and obtain seemingly inexplicable program performance. Also, when using hexadecimal notation, I assumed you won't prefix an address with the "$" symbol. Furthermore, leading zeros are harmless, but quite unnecessary. No relocate is foolproof. Hence, ML Helper does not resolve the indirect JMP or the technique of jumping with an RTS once the stack has been prepared. In other words, jump tables and data blocks are moved unchanged.

Disassemble And Relocate

When up and running, ML Helper emulates a disassembler, examining the address range you've specified for valid 6502 operation codes. When they are found, the program logic proceeds to list or modify the zero page references, to list external absolute references, to modify absolute references, or to move code and modify addresses for a successful relocate, whichever option is operative.

Bytes determined to be invalid instruction code sequences are assumed to form data tables. A data table finder, as such, is always active and can actually become an unspecified sixth option to locate unknown data table areas.

At this point I set an arbitrary criterion – namely, that wherever there occurs a block of six or fewer consecutive bytes of executable code, the data table finder should, nonetheless, report that block of code as part of a data table area. If this standard proves unsuitable for your requirements, then change the "A-7" expression in line 350. The absence of data tables is reported as an address range of 0-0 ($0000-0000 hexadecimal).

Menu item 4 may not be immediately clear. The "EXTERNAL" references that ML Helper will list are those absolute addresses referencing memory outside the body of the program module being examined. Displaying all absolute addressing usage produces a counterproductive volume of screen clutter which I thought best to avoid.

Menu item 5, by which you elect to change absolute references, is not similarly restricted. If during a run it appears that interesting data might scroll away, then Apple users are reminded to invoke the CTRL S Stop-List feature of their system; others may have to rely on CTRL C or divert all output to hard copy. Have fun exploring uncharted machine language programs with ML Helper pointing the way.

```
Ø DATA 232,2ØØ,2Ø2,136,72,1Ø4,24,56,96,17Ø,1
   68,138,152,234,1Ø,74,42,1Ø6,186
1 DATA 154,64,12Ø,88,184,248,216,8,4Ø,Ø,2Ø8,
   24Ø,144,176,48,16,8Ø,112,169,162
2 DATA 16Ø,2Ø1,224,192,1Ø5,233,41,9,73,165,1
   66,164,133,134,132,23Ø,198,197,228
3 DATA 196,1Ø1,229,36,37,5
1Ø DATA 69,38,1Ø2,6,7Ø,181,182,18Ø,149,15Ø,14
   8,246,214,213,117
11 DATA 245,53,21,85,54,118,22,86,177,145,2Ø9
   ,113,241,49,17,81,161,129,193,97
12 DATA 225,33,1,65,32,76,1Ø8,44,173,174,172,
   141,142,14Ø,238,2Ø6,2Ø5,236,2Ø4
13 DATA 1Ø9,237,45,13,77,46,11Ø,14,78,189,19Ø
2Ø DATA 188,157,254,222,221,125,253,61,29,93,
   62,126,3Ø,94,185,153,217,121,249
21 DATA 57,25,89: GOTO 53Ø
3Ø REM *** LIST ADDRESSES ***
4Ø IF A(Z) >  = S THEN  IF A(Z) <  = E THEN
   RETURN
5Ø IF Z = Ø GOTO 8Ø
6Ø FOR X = Ø TO Z - 1: IF A(X) = A(Z) THEN  R
   ETURN
68 S(T) = VAL (H$):E(T) = VAL (E$)
7Ø NEXT
8Ø PRINT "ADDR REF'D:  ";: IF H THEN D = A(Z)
   : GOSUB 22Ø: PRINT "$"H$: GOTO 1ØØ
9Ø PRINT A(Z)
```

```
100 Z = Z + 1 - (Z > 29): RETURN
110 REM *** ZERO PAGE CHANGE ***
120 FOR I = Ø TO X: IF C(I) = A(Z) THEN POKE A
    + 1,D(I)
130 NEXT : RETURN
140 REM *** RELOCATE ***
150 IF A(Z) < TS OR A(Z) > TE THEN  RETURN
160 I =  PEEK (A + 1) + T3: IF I > 255 THEN I ~
    = I - N:T4 = T4 + 1
170 POKE A + 1,I: POKE A + 2, PEEK (A + 2) + T
    4: RETURN
180 REM *** CHANGE ABSOLUTE ADDR ***
190 FOR I = Ø TO X: IF C(I) = A(Z) THEN K =  I
    NT (D(I) / N)
195 POKE A + 1, D(I) - N * K: POKE A + 2,K
200 NEXT : RETURN
210 REM *** DEC-HEX ***
220 H$ = "":F = 4096: FOR J = H TO 4:K =  INT ~
    (D / F):D = D - K * F
225 H$ = H$ + MID$ (G$,K + H,H):F = F / 16: NE
    XT : RETURN
230 REM *** HEX-DEC ***
240 D = Ø:F = H: FOR J = LEN (H$) TO H STEP - ~
    H:M = ASC ( MID$ (H$,J,H )) - 48
245 D = D + F * (M - 7 * (M > 9)):F = 16 * F: ~
    NEXT : RETURN
250 REM *** PRINT DATA TABLES ***
260 PRINT "DATA TABLE:  ";: IF H THEN D = T1: ~
    GOSUB 220: PRINT "$"H$"-";:D=T2
265 GOSUB 220: PRINT H$: RETURN
270 PRINT T1"-"T2: RETURN
280 REM *** MAIN ROUTINE ***
290 FOR A = S TO E
300 REM *** SKIP DATA TABLES ***
310 FOR I = Ø TO T: IF S(I) THEN IF A > = S(I)
     THEN A = E(I) + 1:S(I) = Ø
320 NEXT : FOR I = Ø TO 15Ø: READ M: IF PEEK (
    A) = M GOTO 39Ø
330 NEXT
340 REM *** PRINT DATA TABLES ***
350 IF A - 7 > T2 THEN IF T1 THEN GOSUB 26Ø:T1
     = A
360 IF T1 = Ø THEN T1 = A
370 T2 = A: GOTO 51Ø
380 REM *** 1-BYTE IGNORE ***
390 IF I < 29 GOTO 51Ø
400 REM *** 2-BYTE IGNORE ***
410 IF I < 48 GOTO 5ØØ
420 REM *** ZERO PAGE ***
430 IF I > 1Ø2 OR C > 2 GOTO 47Ø
440 IF C < 3 THEN A(Z) = PEEK (A + 1): ON C GO
    SUB 5Ø,12Ø
450 GOTO 5ØØ
460 REM *** 3-BYTE ***
470 IF I < 1Ø3 GOTO 5ØØ
480 IF C > 2 THEN A(Z)=PEEK (A + 1) + PEEK (A ~
    + 2) * N: ON C-2 GOSUB 15Ø,4Ø,19Ø
490 A = A + 1
500 A = A + 1
510 RESTORE : NEXT A: GOSUB 26Ø: END
520 REM *** END OF MAIN ROUTINE ***
530 PRINT "1= LIST ZERO PAGE REFERENCES":PRINT
    "2= CHANGE ZERO PAGE REFERENCES"
531 PRINT "3= RELOCATE": PRINT "4= LIST EXTERN
    AL ABSOLUTE REFERENCES"
532 PRINT "5= CHANGE ABSOLUTE REFERENCES": PRI
    NT: PRINT "CHOOSE ONE: ";: GET H$
533 PRINT H$:C = VAL (H$):PRINT :PRINT "WANT H
    EX  NUMBERS, Y/N? ";: GET H$:PRINT H$
540 PRINT :H = H$ = "Y":N = 256:G$ = "Ø1234567
    89ABCDEF": DIM A(3Ø)
541 INPUT "INPUT START,END ADDRESSES: ";H$,E$:
     PRINT
542 IF H THEN GOSUB 24Ø:S = D:H$ = E$: GOSUB 2
    4Ø:E = D: GOTO 56Ø
550 S = VAL (H$):E = VAL (E$)
560 IF C < > 3 GOTO 66Ø
570 INPUT "INPUT TARGET ADDRESS: ";H$: PRINT :
     IF H THEN GOSUB 24Ø:TS=D:GOTO6ØØ
580 TS = VAL (H$)
590 REM *** CALCULATE OFFSET ***
600 TE = TS + E - S:I = ABS (TS - S):T4 = INT ~
    (I / N):T3 = I - T4 * N
605 IF TS < S THEN T3 = - T3: T4 = - T4
610 REM *** MOVE ROUTINE ***
620 IF T3 > Ø THEN K = TE: FOR I = E TO S STEP
     - 1: POKE K, PEEK (I):K=K-1
625 NEXT : GOTO 65Ø
630 K = TS: FOR I = S TO E: POKE K, PEEK (I): ~
    K = K + 1: NEXT
640 REM *** SWAP TS & S, TE & E ***
650 K = TS:TS = S:S = K:K = TE:TE = E:E = K
660 PRINT "LIST UP TO 11 KNOWN DATA TABLES IN ~
    THE  PROGRAM.  TYPE Ø,Ø WHEN DONE.":P
    RINT
670 PRINT "DATA TABLE "T" START,END:  ";: INPU
    T "";H$,E$
675 IF H THEN GOSUB 24Ø:S(T) = D:H$ = E$: GOSU
    B 24Ø:E(T) = D: GOTO 69Ø
680 S(T) = VAL (H$):E(T) = VAL (E$)
690 IF E(T) THEN I = T3 + T4 * N:S(T) = S(T) +
     I:E(T) = E(T) + I:T = T + 1
695 IF T < 11 GOTO 67Ø
700 IF C < > 2 THEN IF C < > 5 THEN PRINT : GO
    TO 29Ø
710 PRINT :PRINT "LIST UP TO 11 ADDRESSES TO B
    E CHANGED. TYPE Ø,Ø WHEN DONE.":PRINT
720 PRINT "#"X". OLD,NEW ADDRESSES:  ";: INPUT
     "";H$,E$
725 IF H THEN GOSUB 24Ø:C(X) = D:H$ = E$: GOSU
    B 24Ø:D(X) = D: GOTO 74Ø
730 C(X) = VAL (H$):D(X) = VAL (E$)
740 IF C(X) = D(X) OR X = 1Ø THEN PRINT : GOTO
     29Ø
750 X = X + 1: GOTO 72Ø
```

©

Supermon64

Jim Butterfield, Associate Editor

Supermon64 is your gateway to machine language programming on the Commodore 64. Supermon, in several versions, has been popular over the years as a major programming tool for Commodore users. With this adaptation for the 64, a good book on 6502 programming, and patience, you can learn to write programs or subroutines which are capable of running at extraordinary speeds. You can learn machine language programming.

Supermon64 itself is in machine language, but you can type it in without knowing what it means. Using the Tiny Peeker/Poker (Program 1), or via the built-in monitor of a PET, type it in and SAVE it. The fastest way to check for errors is to type in Program 3 on a regular PET (or use the portioning techniques described in the article to make room for the checksum program the same way you made room for the Tiny Peeker). Then load Supermon64 into the PET. It will come in above your BASIC. Then RUN the checksum and it will report the location of any errors.

The easiest way to type in Supermon64 is by using a monitor. Unfortunately, you won't have a monitor until Supermon64 is typed in. This leads to a chicken-and-egg situation. It's no good my suggesting that you should borrow a friend's Supermon64 to type this in; if he's got it, you can just make a copy and save a lot of work. There's gotta be another way.

The Commodore 64 has lots of memory. We can waste some of it to create an easy method which will allow us to enter this program. Maybe we'll even learn something along the way.

You may have noticed that when you power up the 64, it tells you that you have 38911 bytes free. For the moment, I'm not going to tell you what happened to the rest of the 64K (it's there, but BASIC won't use it). The baffling thing is that if you ask the machine to PRINT FRE(0), it will tell you that it has -26627 bytes free. Don't be confused: this is a bug and you can get the right number by adding 65536. Whip out your trusty calculator, do the addition, and do you get 38911? Of course not, but it's close enough (another story for another time).

Now: we're going to trim some memory away from the computer, so as to give ourselves space to stage Supermon64. First, we must prepare the new space so that it will be able to hold a BASIC program. Type POKE 8192,0 and the space is ready. Next, we are going to move BASIC to this new area. Type POKE 44,32 followed by NEW and the deed is done. At this point we seem to have a perfectly normal Commodore 64 machine. Everything will work as before. But, if you ask PRINT FRE(0) you'll find that your free space has dropped to 32765 bytes. We have sectioned off the space where we will plant Supermon64. [*You can use this same technique, then type in Program 3 and run the checksum on your program – Ed.*]

Now we are ready for a simple input program. Enter the following:

Program 1. Tiny Peeker/Poker.

```
100 PRINT "TINY PEEKER/POKER"
110 X$="*":INPUT X$:IF X$="*" THEN END
120 GOSUB 500
130 IF E GOTO 280
140 A=V
150 IF J>LEN(X$) GOTO 300
160 FOR I=0 TO 7
170 P=J:GOSUB 550
180 C(I)=V
190 IF E GOTO 280
200 NEXT I
210 T=0
220 FOR I=0 TO 7
230 POKE A+I,C(I)
240 T=T+C(I)
250 NEXT I
260 PRINT "CHECKSUM=";T
270 GOTO 110
280 PRINT MID$(X$,1,J);"??":GOTO 110
300 T=0
310 FOR I=0 TO 7
320 V=PEEK(A+I)
330 T=T+V
340 V=V/16
350 PRINT " ";
360 FOR J=1 TO 2
370 V%=V
380 V=(V-V%)*16
390 IF V%>9 THEN V%=V%+7
400 PRINT CHR$(V%+48);
410 NEXT J
420 NEXT I
430 PRINT "/";T
440 GOTO 110
500 P=1
```

64 ** ALL NEW!!! ** 64

— — — * * * — — —

SOFTWARE FOR COMMODORE 64

WORD-PAC $74.95
- Print up to 99 pages of text.
- Automatic tabbing/Centering/Underlining.
- Copy Lines/Merge/Plus More!
- Coded in Machine language.

CALC-PAC $74.95
- Interface-Compatible with WORD-PAC & DATA-PAC.
- Coded in our own Unique Spread-Sheet language.
- User-Friendly Mathematical Applications.

DATA-PAC $39.95
- Interface-Compatible with WORD-PAC & CALC-PAC.
- User defined Formats/Search & Sorts.
- Printer compatable.

EDITOR-PAC $69.95
- Complete Programmer's Editor.
- Auto-Number/Renumber including goto & gosub.
- Program Merge/Global Search and Replace.
- Plus Much More!

ASSEMBLER-PAC $59.95
- Programmers take note!
- Mnemonic format to Machine Language.
- Link Modules/External references, More!

HOME-ACCOUNTANT $29.95
- Checkbook with reconciliation routine.
- Hard-Copy listing option.
- Search and Review/Chart of Accounts.
- Income and Expense.

ANNOUNCING . . .

The PCS/8064 Upgrade Module for the 64

On power-up the PCS/8064 provides:
- 80-column video output.
- WORD-PAC word processing.
- CALC-PAC spread sheet mathematics.
- DATA-PAC data base system.
- Exit to BASIC.
- All Applications Interface-Compatible.

Check local dealers or Call Pacific Coast Software for retail pricing.

Plus Full line of Games/Home Software for 64 Free Catalog Offer.

PACIFIC COAST SOFTWARE

3220 S. Brea Canyon Rd.
Diamond Bar, CA 91765
(714) 594-8210

218 S. Main/Box 147
LeSueur, MN 56058
(612) 665-6724

Mid-Eastern Distribution:
PERIPHERALS PLUS .... (215) 687-8540
155 E. Lancaster Ave. - Wayne, Penn. 19087

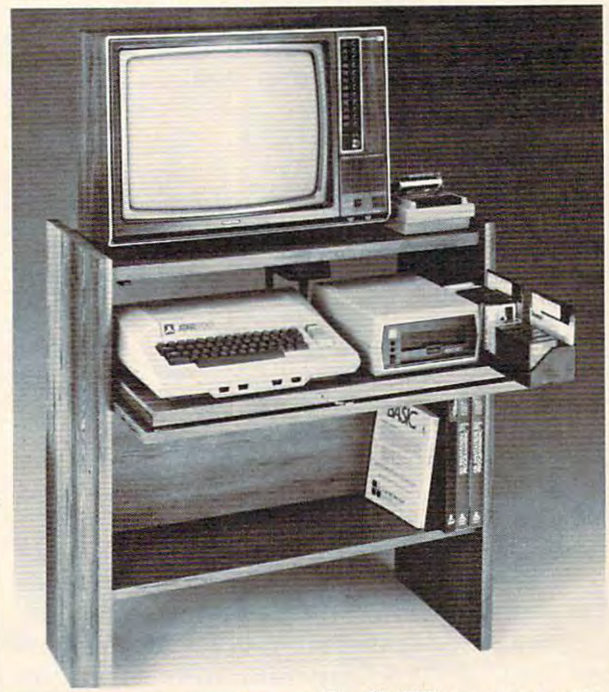
New England Distribution:
OMICRON .... (617) 769-6867
1416 Providence Highway - Norwood, Mass. 02062

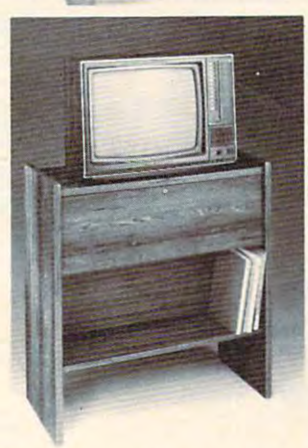
Dealer Inquires Encouraged.

Commodore 64 and 64 are trademarks of Commodore Business Machines.

ARE YOU A SMART BUYER?

For **$89.95** this is a smart buy if you're looking for a place to store your computer, peripherals, and accessories without spending a fortune.





The CS 1632 computer storage cabinets compact yet functional design fits almost anywhere while housing your computer monitor, joysticks, software, books and peripherals all for only $89.95.
The slide out shelf puts the computer at the right height and position for easy comfortable operation.
The fold up locking door keeps unwanted fingers off the key board when not in use.
To store joysticks just turn them upside down and slide them into the inverted storage rack.
Twist tabs on the back of center panel allow for neat concealed grouping of wires, while power packs rest hidden behind center panel on shelf.
The slide out software tray has room for 14 cartridges or cassettes and up to 30 diskettes. Most brands of software will fit between the adjustable partitions with a convenient hook for the spare key at rear.
Stand fits Atari 400 & 800, Commodore 64 & VIC 20, Ti 99/4A and TRS-80.
Cabinet dimensions overall 36" high x 33-7/8" wide x 16" deep.
Cabinet comes unassembled. Assembly requires only a screwdriver, hammer, and a few minutes of your time.
Choice in simulated woodgrain, of warm golden oak or rich natural walnut finish.

To order CS1632, send $89.95 to:

BÖEGNER
INDUSTRIES CORPORATION

P.O. Box 528
Lake Oswego, OR 97034
Phone orders call, (503) 636-6888

Name ______________________
Address ______________________
City ____________ State ________ Zip ________
☐ Golden oak finish ☐ Natural walnut finish
☐ My personal check, cashiers check or money order is enclosed.
☐ Bill my VISA # ______________ Exp. Date ______
☐ Bill my Mastercard # ______________ Exp. Date ______
Card Holders Signature ______________________

Immediate shipment if in stock. If personal check is sent, allow additional 2 weeks. Prices subject to change. Shipment subject to availability. Cabinet shipped unassembled in 2 cartons. Ships UPS frt. collect FOB Portland, Oregon.

```
51Ø L=4
52Ø GOTO 6ØØ
55Ø P=J
56Ø L=2
6ØØ E=Ø
61Ø V=Ø
62Ø FOR J=P TO LEN(X$)
63Ø X=ASC(MID$(X$,J))
64Ø IF X=32 THEN NEXT J
65Ø IF J>LEN(X$) GOTO 79Ø
66Ø P=J
67Ø FOR J=P TO LEN(X$)
68Ø X=ASC(MID$(X$,J))
69Ø IF X<>32 THEN NEXT J
7ØØ IF J-P<>L GOTO 79Ø
71Ø FOR K=P TO J-1
72Ø X=ASC(MID$(X$,K))
73Ø IF X<58 THEN X=X-48
74Ø IF X>64 THEN X=X-55
75Ø IF X<Ø OR X>15 GOTO 79Ø
76Ø V=V*16+X
77Ø NEXT K
78Ø RETURN
79Ø E=-1
8ØØ RETURN
```

This program is a very tiny monitor. It will allow you to enter information into memory, eight bytes at a time. To do this: wait for the question mark, and then type in monitor-format the address and contents:

```
? 0800 00 1A 08 64 00 99 22 93
```

The program will return a checksum value to you, which you can use to insure that you have entered the information correctly. To view memory, type in only the address: the contents will be displayed.

Completing The Job

When you have finished entering all that data, you can make Supermon64 happen quite easily. Three last POKE commmands and a CLR:

```
POKE 44,8
POKE 45,232
POKE 46,17
CLR
```

You have Supermon64. Save it with a conventional BASIC SAVE before you do anything else.

Now you may RUN it – and learn how to use it.

Supermon64 Summary

Commodore Monitor Instructions:

G GO RUN
L LOAD FROM TAPE OR DISK
M MEMORY DISPLAY
R REGISTER DISPLAY
S SAVE TO TAPE OR DISK
X EXIT TO BASIC

Supermon64 Additional Instructions:

A SIMPLE ASSEMBLER
D DISASSEMBLER
F FILL MEMORY
H HUNT MEMORY
P PRINTING DISASSEMBLER
T TRANSFER MEMORY

• Simple assembler

```
.A 2ØØØ LDA #$12
.A 2ØØ2 STA $8ØØØ,X
.A 2ØØ5 (RETURN)
```

In the above example the user started assembly at 2000 hex. The first instruction was load a register with immediate 12 hex. In the second line the user did not need to type the A and address. The simple assembler prompts with the next address. To exit the assembler type a return after the address prompt. Syntax is the same as the disassembler output.

• Disassembler

```
.D 2ØØØ
(SCREEN CLEARS)
2ØØØ A9 12        LDA #$12
2ØØ2 9D ØØ 8Ø     STA $8ØØØ,X
2ØØ5 AA           TAX
2ØØ6 AA           TAX
```

(Full page of instructions)

Disassembles 22 instructions starting at 2000 hex. The three bytes following the address may be modified. Use the CRSR keys to move to and modify the bytes. Hit return and the bytes in memory will be changed. Supermon64 will then disassemble that page again.

• Printing disassembler

```
.P 2ØØØ,2Ø4Ø
2ØØØ A9 12          LDA #$12
2ØØ2 9D ØØ 8Ø       STA $8ØØØ,X
2ØØ5 AA             TAX
    ....
2Ø3F A2 ØØ          LDX #$ØØ
```

To engage printer, set up beforehand:

```
OPEN 4,4:CMD4
```

• Fill memory

```
.F 1ØØØ 11ØØ FF
```

Fills the memory from 1000 hex to 1100 hex with the byte FF hex.

• Go run

```
.G
```

Go to the address in the PC register display and begin RUN code. All the registers will be replaced with the displayed values.

```
.G 1ØØØ
```

Program 2. Supermon64.

```
0800 00 1A 04 64 00 99 22 93
0808 12 1D 1D 1D 1D 53 55 50
0810 45 52 20 36 34 2D 4D 4F
0818 4E 00 31 04 6E 00 99 22
0820 11 20 20 20 20 20 20 20
0828 20 20 20 20 20 20 20 20
0830 00 4B 04 78 00 99 22 11
0838 20 2E 2E 4A 49 4D 20 42
0840 55 54 54 45 52 46 49 45
0848 4C 44 00 66 04 82 00 9E
0850 28 C2 28 34 33 29 AA 32
0858 35 36 AC C2 28 34 34 29
0860 AA 31 32 37 29 00 00 00
0868 AA AA AA AA AA AA AA AA
0870 AA AA AA AA AA AA AA AA
0878 AA AA AA AA AA AA AA AA

0880 A5 2D 85 22 A5 2E 85 23
0888 A5 37 85 24 A5 38 85 25
0890 A0 00 A5 22 D0 02 C6 23
0898 C6 22 B1 22 D0 3C A5 22
08A0 D0 02 C6 23 C6 22 B1 22
08A8 F0 21 85 26 A5 22 D0 02
08B0 C6 23 C6 22 B1 22 18 65
08B8 24 AA A5 26 65 25 48 A5
08C0 37 D0 02 C6 38 C6 37 68
08C8 91 37 8A 48 A5 37 D0 02
08D0 C6 38 C6 37 68 91 37 18
08D8 90 B6 C9 4F D0 ED A5 37
08E0 85 33 A5 38 85 34 6C 37
08E8 00 4F 4F 4F 4F AD E6 FF
08F0 00 8D 16 03 AD E7 FF 00
08F8 8D 17 03 A9 80 20 90 FF

0900 00 00 D8 68 8D 3E 02 68
0908 8D 3D 02 68 8D 3C 02 68
0910 8D 3B 02 68 AA 68 A8 38
0918 8A E9 02 8D 3A 02 98 E9
0920 00 00 8D 39 02 BA 8E 3F
0928 02 20 57 FD 00 A2 42 A9
0930 2A 20 57 FA 00 A9 52 D0
0938 34 E6 C1 D0 06 E6 C2 D0
0940 02 E6 26 60 20 CF FF C9
0948 0D D0 F8 68 68 A9 90 20
0950 D2 FF A9 00 00 85 26 A2
0958 0D A9 2E 20 57 FA 00 A9
0960 05 20 D2 FF 20 3E F8 00
0968 C9 2E F0 F9 C9 20 F0 F5
0970 A2 0E DD B7 FF 00 D0 0C
0978 8A 0A AA BD C7 FF 00 48

0980 BD C6 FF 00 48 60 CA 10
0988 EC 4C ED FA 00 A5 C1 8D
0990 3A 02 A5 C2 8D 39 02 60
0998 A9 08 85 1D A0 00 00 20
09A0 54 FD 00 B1 C1 20 48 FA
09A8 00 20 33 F8 00 C6 1D D0
09B0 F1 60 20 88 FA 00 90 0B
09B8 A2 00 00 81 C1 C1 C1 F0
09C0 03 4C ED FA 00 20 33 F8
09C8 00 C6 1D 60 A9 3B 85 C1
09D0 A9 02 85 C2 A9 05 60 98
09D8 48 20 57 FD 00 68 A2 2E
09E0 4C 57 FA 00 A9 90 20 D2
09E8 FF A2 00 00 BD EA FF 00
09F0 20 D2 FF E8 E0 16 D0 F5
09F8 A0 3B 20 C2 F8 00 AD 39

0A00 02 20 48 FA 00 AD 3A 02
0A08 20 48 FA 00 20 B7 F8 00
0A10 20 8D F8 00 F0 5C 20 3E
0A18 F8 00 20 79 FA 00 90 33
0A20 20 69 FA 00 20 3E F8 00
0A28 20 79 FA 00 90 28 20 69
0A30 FA 00 A9 90 20 D2 FF 20
0A38 E1 FF F0 3C A6 26 D0 38
0A40 A5 C3 C5 C1 A5 C4 E5 C2
0A48 90 2E A0 3A 20 C2 F8 00
0A50 20 41 FA 00 20 8B F8 00
0A58 F0 E0 4C ED FA 00 20 79
0A60 FA 00 90 03 20 80 F8 00
0A68 20 B7 F8 00 D0 07 20 79
0A70 FA 00 90 EB A9 08 85 1D
0A78 20 3E F8 00 20 A1 F8 00

0A80 D0 F8 4C 47 F8 00 20 CF
0A88 FF C9 0D F0 0C C9 20 D0
0A90 D1 20 79 FA 00 90 03 20
0A98 80 F8 00 A9 90 20 D2 FF
0AA0 AE 3F 02 9A 78 AD 39 02
0AA8 48 AD 3A 02 48 AD 3B 02
0AB0 48 AD 3C 02 AE 3D 02 AC
0AB8 3E 02 40 A9 90 20 D2 FF
0AC0 AE 3F 02 9A 6C 02 A0 A0
0AC8 01 84 BA 84 B9 88 84 B7
0AD0 84 90 84 93 A9 40 85 BB
0AD8 A9 02 85 BC 20 CF FF C9
0AE0 20 F0 F9 C9 0D F0 38 C9
0AE8 22 D0 14 20 CF FF C9 22
0AF0 F0 10 C9 0D F0 29 91 BB
0AF8 E6 B7 C8 C0 10 D0 EC 4C

0B00 ED FA 00 20 CF FF C9 0D
0B08 F0 16 C9 2C D0 DC 20 88
0B10 FA 00 29 0F F0 E9 C9 03
0B18 F0 E5 85 BA 20 CF FF C9
0B20 0D 60 6C 30 03 6C 32 03
0B28 20 96 F9 00 D0 D4 A9 90
0B30 20 D2 FF A9 00 00 20 EF
0B38 F9 00 A5 90 29 10 D0 C4
0B40 4C 47 F8 00 20 96 F9 00
0B48 C9 2C D0 BA 20 79 FA 00
0B50 20 69 FA 00 20 CF FF C9
0B58 2C D0 AD 20 79 FA 00 A5
0B60 C1 85 AE A5 C2 85 AF 20
0B68 69 FA 00 20 CF FF C9 0D
0B70 D0 98 A9 90 20 D2 FF 20
0B78 F2 F9 00 4C 47 F8 00 A5

0B80 C2 20 48 FA 00 A5 C1 48
0B88 4A 4A 4A 4A 20 60 FA 00
0B90 AA 68 29 0F 20 60 FA 00
0B98 48 8A 20 D2 FF 68 4C D2
0BA0 FF 09 30 C9 3A 90 02 69
0BA8 06 60 A2 02 B5 C0 48 B5
0BB0 C2 95 C0 68 95 C2 CA D0
0BB8 F3 60 20 88 FA 00 90 02
0BC0 85 C2 20 88 FA 00 90 02
0BC8 85 C1 60 A9 00 00 85 2A
0BD0 20 3E F8 00 C9 20 D0 09
0BD8 20 3E F8 00 C9 20 D0 0E
0BE0 18 60 20 AF FA 00 0A 0A
0BE8 0A 0A 85 2A 20 3E F8 00
0BF0 20 AF FA 00 05 2A 38 60
0BF8 C9 3A 90 02 69 08 29 0F

0C00 60 A2 02 2C A2 00 00 B4
0C08 C1 D0 08 B4 C2 D0 02 E6
0C10 26 D6 C2 D6 C1 60 20 3E
0C18 F8 00 C9 20 F0 F9 60 A9
0C20 00 00 8D 00 00 01 20 CC
0C28 FA 00 20 8F FA 00 20 7C
0C30 FA 00 90 09 60 20 3E F8
0C38 00 20 79 FA 00 B0 DE AE
0C40 3F 02 9A A9 90 20 D2 FF
0C48 A9 3F 20 D2 FF 4C 47 F8
0C50 00 20 54 FD 00 CA D0 FA
0C58 60 E6 C3 D0 02 E6 C4 60
0C60 A2 02 B5 C0 48 B5 27 95
0C68 C0 68 95 27 CA D0 F3 60
0C70 A5 C3 A4 C4 38 E9 02 B0
0C78 0E 88 90 0B A5 28 A4 29

0C80 4C 33 FB 00 A5 C3 A4 C4
0C88 38 E5 C1 85 1E 98 E5 C2
0C90 A8 05 1E 60 20 D4 FA 00
0C98 20 69 FA 00 20 E5 FA 00
0CA0 20 0C FB 00 20 E5 FA 00
0CA8 20 2F FB 00 20 69 FA 00
0CB0 90 15 A6 26 D0 64 20 28
0CB8 FB 00 90 5F A1 C1 81 C3
0CC0 20 05 FB 00 20 33 F8 00
0CC8 D0 EB 20 28 FB 00 18 A5
0CD0 1E 65 C3 85 C3 98 65 C4
0CD8 85 C4 20 0C FB 00 A6 26
0CE0 D0 3D A1 C1 81 C3 20 28
0CE8 FB 00 B0 34 20 B8 FA 00
0CF0 20 BB FA 00 4C 7D FB 00
0CF8 20 D4 FA 00 20 69 FA 00

0D00 20 E5 FA 00 20 69 FA 00
0D08 20 3E F8 00 20 88 FA 00
0D10 90 14 85 1D A6 26 D0 11
0D18 20 2F FB 00 90 0C A5 1D
0D20 81 C1 20 33 F8 00 D0 EE
0D28 4C ED FA 00 4C 47 F8 00
0D30 20 D4 FA 00 20 69 FA 00
0D38 20 E5 FA 00 20 69 FA 00
0D40 20 3E F8 00 A2 00 00 20
0D48 3E F8 00 C9 27 D0 14 20
0D50 3E F8 00 9D 10 02 E8 20
0D58 CF FF C9 0D F0 22 E0 20
0D60 D0 F1 F0 1C 8E 00 00 01
0D68 20 8F FA 00 90 C6 9D 10
0D70 02 E8 20 CF FF C9 0D F0
0D78 09 20 88 FA 00 90 B6 E0

0D80 20 D0 EC 86 1C A9 90 20
0D88 D2 FF 20 57 FD 00 A2 00
0D90 00 A0 00 00 B1 C1 DD 10
0D98 02 D0 0C C8 E8 E4 1C D0
0DA0 F3 20 41 FA 00 20 54 FD
0DA8 00 20 33 F8 00 A6 26 D0
0DB0 8D 20 2F FB 00 B0 DD 4C
0DB8 47 F8 00 20 D4 FA 00 85
0DC0 20 A5 C2 85 21 A2 00 00
0DC8 86 28 A9 93 20 D2 FF A9
0DD0 90 20 D2 FF A9 16 85 1D
0DD8 20 6A FC 00 20 CA FC 00
0DE0 85 C1 84 C2 C6 1D D0 F2
0DE8 A9 91 20 D2 FF 4C 47 F8
0DF0 00 A0 2C 20 C2 F8 00 20
0DF8 54 FD 00 20 41 FA 00 20
```

WHAT'S SMALLER THAN A BUSINESS CARD?

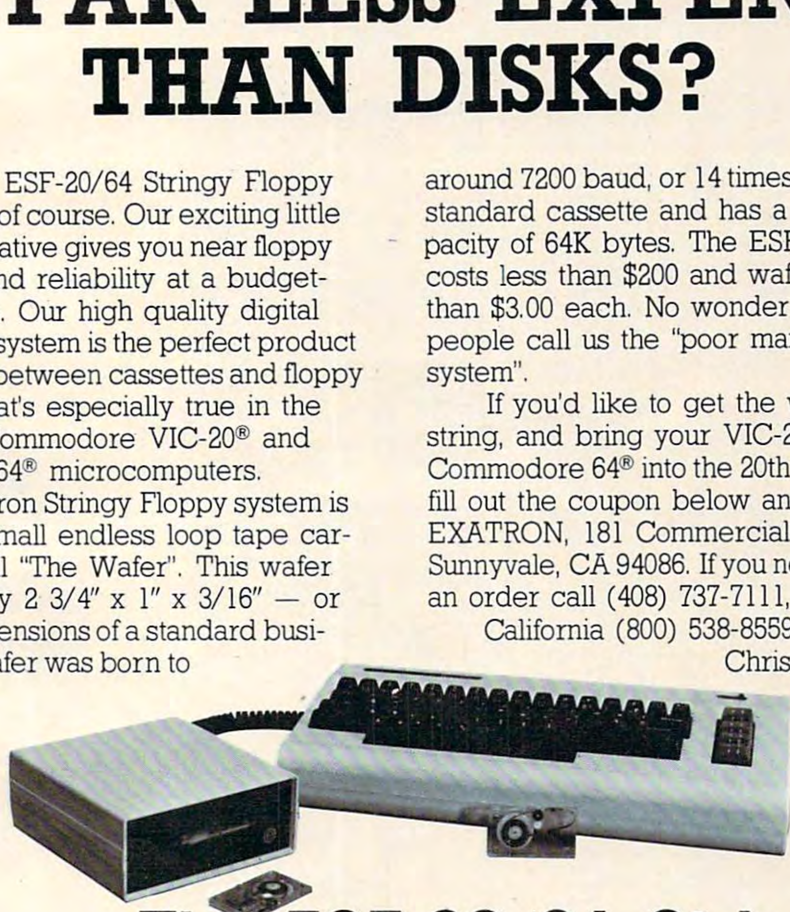
FASTER THAN CASSETTES?

AND FAR LESS EXPENSIVE THAN DISKS?

Why the ESF-20/64 Stringy Floppy from Exatron, of course. Our exciting little storage alternative gives you near floppy disk speed and reliability at a budget-minded price. Our high quality digital mass storage system is the perfect product to fill the gap between cassettes and floppy disks. And that's especially true in the case of the Commodore VIC-20® and Commodore 64® microcomputers.

The Exatron Stringy Floppy system is based on a small endless loop tape cartridge we call "The Wafer". This wafer measures only 2 3/4" x 1" x 3/16" — or about the dimensions of a standard business card. Wafer was born to run fast — at around 7200 baud, or 14 times faster than a standard cassette and has a memory capacity of 64K bytes. The ESF-20/64 system costs less than $200 and wafers are less than $3.00 each. No wonder computer people call us the "poor man's disk system".

If you'd like to get the world on a string, and bring your VIC-20® or Commodore 64® into the 20th Century, just fill out the coupon below and mail it to EXATRON, 181 Commercial Street, Sunnyvale, CA 94086. If you need to place an order call (408) 737-7111, outside California (800) 538-8559 and ask for Chrissy or Natalie.



The ESF-20/64 Stringy Floppy System from Exatron

for use with the Commodore VIC-20® and Commodore 64® microcomputer.

exatron

For exciting alternatives in mass storage.

Name ____________________

Address ____________________

City/State/Zip ____________________

Mail today for information on the ESF-20/64 from EXATRON, 181 Commercial St., Sunnyvale, CA 94086

```
0E00 54 FD 00 A2 00 00 A1 C1
0E08 20 D9 FC 00 48 20 1F FD
0E10 00 68 20 35 FD 00 A2 06
0E18 E0 03 D0 12 A4 1F F0 0E
0E20 A5 2A C9 E8 B1 C1 B0 1C
0E28 20 C2 FC 00 88 D0 F2 06
0E30 2A 90 0E BD 2A FF 00 20
0E38 A5 FD 00 BD 30 FF 00 F0
0E40 03 20 A5 FD 00 CA D0 D5
0E48 60 20 CD FC 00 AA E8 D0
0E50 01 C8 98 20 C2 FC 00 8A
0E58 86 1C 20 48 FA 00 A6 1C
0E60 60 A5 1F 38 A4 C2 AA 10
0E68 01 88 65 C1 90 01 C8 60
0E70 A8 4A 90 0B 4A B0 17 C9
0E78 22 F0 13 29 07 09 80 4A

0E80 AA BD D9 FE 00 B0 04 4A
0E88 4A 4A 4A 29 0F D0 04 A0
0E90 80 A9 00 00 AA BD 1D FF
0E98 00 85 2A 29 03 85 1F 98
0EA0 29 8F AA 98 A0 03 E0 8A
0EA8 F0 0B 4A 90 08 4A 4A 09
0EB0 20 88 D0 FA C8 88 D0 F2
0EB8 60 B1 C1 20 C2 FC 00 A2
0EC0 01 20 FE FA 00 C4 1F C8
0EC8 90 F1 A2 03 C0 04 90 F2
0ED0 60 A8 B9 37 FF 00 85 28
0ED8 B9 77 FF 00 85 29 A9 00
0EE0 00 A0 05 06 29 26 28 2A
0EE8 88 D0 F8 69 3F 20 D2 FF
0EF0 CA D0 EC A9 20 2C A9 0D
0EF8 4C D2 FF 20 D4 FA 00 20

0F00 69 FA 00 20 E5 FA 00 20
0F08 69 FA 00 A2 00 00 86 28
0F10 A9 90 20 D2 FF 20 57 FD
0F18 00 20 72 FC 00 20 CA FC
0F20 00 85 C1 84 C2 20 E1 FF
0F28 F0 05 20 2F FB 00 B0 E9
0F30 4C 47 F8 00 20 D4 FA 00
0F38 A9 03 85 1D 20 3E F8 00
0F40 20 A1 F8 00 D0 F8 A5 20
0F48 85 C1 A5 21 85 C2 4C 46
0F50 FC 00 C5 28 F0 03 20 D2
0F58 FF 60 20 D4 FA 00 20 69
0F60 FA 00 8E 11 02 A2 03 20
0F68 CC FA 00 48 CA D0 F9 A2
0F70 03 68 38 E9 3F A0 05 4A
0F78 6E 11 02 6E 10 02 88 D0

0F80 F6 CA D0 ED A2 02 20 CF
0F88 FF C9 0D F0 1E C9 20 F0
0F90 F5 20 D0 FE 00 B0 0F 20
0F98 9C FA 00 A4 C1 84 C2 85
0FA0 C1 A9 30 9D 10 02 E8 9D
0FA8 10 02 E8 D0 DB 86 28 A2
0FB0 00 00 86 26 F0 04 E6 26
0FB8 F0 75 A2 00 00 86 1D A5
0FC0 26 20 D9 FC 00 A6 2A 86
0FC8 29 AA BC 37 FF 00 BD 77
0FD0 FF 00 20 B9 FE 00 D0 E3
0FD8 A2 06 E0 03 D0 19 A4 1F
0FE0 F0 15 A5 2A C9 E8 A9 30
0FE8 B0 21 20 BF FE 00 D0 CC
0FF0 20 C1 FE 00 D0 C7 88 D0
0FF8 EB 06 2A 90 0B BC 30 FF
```

```
1000 00 BD 2A FF 00 20 B9 FE
1008 00 D0 B5 CA D0 D1 F0 0A
1010 20 B8 FE 00 D0 AB 20 B8
1018 FE 00 D0 A6 A5 28 C5 1D
1020 D0 A0 20 69 FA 00 A4 1F
1028 F0 28 A5 29 C9 9D D0 1A
1030 20 1C FB 00 90 0A 98 D0
1038 04 A5 1E 10 0A 4C ED FA
1040 00 C8 D0 FA A5 1E 10 F6
1048 A4 1F D0 03 B9 C2 00 00
1050 91 C1 88 D0 F8 A5 26 91
1058 C1 20 CA FC 00 85 C1 84
1060 C2 A9 90 20 D2 FF A0 41
1068 20 C2 F8 00 20 54 FD 00
1070 20 41 FA 00 20 54 FD 00
1078 A9 05 20 D2 FF 4C B0 FD

1080 00 A8 20 BF FE 00 D0 11
1088 98 F0 0E 86 1C A6 1D DD
1090 10 02 08 E8 86 1D A6 1C
1098 28 60 C9 30 90 03 C9 47
10A0 60 38 60 40 02 45 03 D0
10A8 08 40 09 30 22 45 33 D0
10B0 08 40 09 40 02 45 33 D0
10B8 08 40 09 40 02 45 B3 D0
10C0 08 40 09 00 00 22 44 33
10C8 D0 8C 44 00 00 11 22 44
10D0 33 D0 8C 44 9A 10 22 44
10D8 33 D0 08 40 09 10 22 44
10E0 33 D0 08 40 09 62 13 78
10E8 A9 00 00 21 81 82 00 00
10F0 00 00 59 4D 91 92 86 4A
10F8 85 9D 2C 29 2C 23 28 24

1100 59 00 00 58 24 24 00 00
1108 1C 8A 1C 23 5D 8B 1B A1
1110 9D 8A 1D 23 9D 8B 1D A1
1118 00 00 29 19 AE 69 A8 19
1120 23 24 53 1B 23 24 53 19
1128 A1 00 00 1A 5B 5B A5 69
1130 24 24 AE AE A8 AD 29 00
1138 00 7C 00 00 15 9C 6D 9C
1140 A5 69 29 53 84 13 34 11
1148 A5 69 23 A0 D8 62 5A 48
1150 26 62 94 88 54 44 C8 54
1158 68 44 E8 94 00 00 B4 08
1160 84 74 B4 28 6E 74 F4 CC
1168 4A 72 F2 A4 8A 00 00 AA
1170 A2 A2 74 74 74 72 44 68
1178 B2 32 B2 00 00 22 00 00

1180 1A 1A 26 26 72 72 88 C8
1188 C4 CA 26 48 44 44 A2 C8
1190 3A 3B 52 4D 47 58 4C 53
1198 54 46 48 44 50 2C 41 42
11A0 F9 00 35 F9 00 CC F8 00
11A8 F7 F8 00 56 F9 00 89 F9
11B0 00 F4 F9 00 0C FA 00 3E
11B8 FB 00 92 FB 00 C0 FB 00
11C0 38 FC 00 5B FD 00 8A FD
11C8 00 AC FD 00 46 F8 00 FF
11D0 F7 00 ED F7 00 0D 20 20
11D8 20 50 43 20 20 53 52 20
11E0 41 43 20 58 52 20 59 52
11E8 20 53 50 AA AA AA AA AA
```

Go to address 1000 hex and begin running code.

- **Hunt memory**

```
.H C000 D000 'READ
```

Hunt through memory from C000 hex to D000 hex for the ASCII string read and print the address where it is found. A maximum of 32 characters may be used.

```
.H C000 D000 20 D2 FF
```

Hunt memory from C000 hex to D000 hex for the sequence of bytes 20 D2 FF and print the address. A maximum of 32 bytes may be used.

- **Load**

```
.L
```

Load any program from cassette #1.

```
.L "RAM TEST"
```

Load from cassette #1 the program named RAM TEST.

```
.L "RAM TEST",08
```

Load from disk (device 8) the program named RAM TEST. This command leaves BASIC pointers unchanged.

- **Memory display**

```
.M 0000 0080
.  0000 00 01 02 03 04 05 06 07
.  0008 08 09 0A 0B 0C 0D 0E 0F
```

Display memory from 0000 hex to 0080 hex. The bytes following the .: can be altered by typing over them, then typing a return.

- **Register display**

```
.R

     PC   IRQ  SR AC XR YR SP
    0000 E62E 01 02 03 04 05
```

Displays the register values saved when Supermon64 was entered. The values may be changed with the edit followed by a return.

- **Save**

```
.S "PROGRAM NAME",01,0800,0C80
```

SAVE to cassette #1 memory from 0800 hex up to but not including 0C80 hex and name it PROGRAM NAME.

```
.S "0:PROGRAM NAME",08,1200,1F50
```

SAVE to disk drive #0 memory from 1200 hex up to but not including 1F50 hex and name it PROGRAM NAME.

- **Transfer memory**

```
.T 1000 1100 5000
```

Transfer memory in the range 1000 hex to 1100 hex and start storing it at address 5000 hex.

- **Exit to BASIC**

```
.X
```

Return to BASIC ready mode. The stack value SAVEd when entered will be restored. Care should be taken that this value is the same as when the monitor was entered. A CLR in BASIC will fix any stack problems.

Program 3. Supermon64 Checksum.

```
100 REM SUPERMON64 CHECKSUM PROGRAM
110 DATA 10170,13676,15404,14997,15136,
    16221,16696,12816,16228,14554
120 DATA14677,15039,14551,15104,15522,
    16414,15914,8958,11945 :S=2048
130 FORB=1TO19:READX:FORI=STOS+128:N=P
    EEK(I):Y=Y+N
140 NEXTI:IFY<>XTHENPRINT"ERROR IN
    BLOCK #"B:G OTO160
150 PRINT"BLOCK #"B" IS CORRECT"
160 S=I:Y=0:NEXTB:REM CHECK LAST SHORT
    BLOCK BY HAND
```

©

INSIGHT: Atari

Bill Wilkinson

No gossip to start with this month. Instead, let's start right off into a whole series of interesting tidbits (and even a few tidbytes).

Which Is It? GTIA Or CTIA?

Several articles have been written on how to tell whether you have a GTIA or CTIA in your system. Most of them suggest that you use a GRAPHICS 9 statement and observe the screen (it turns black with a GTIA, remains blue with a CTIA).

But suppose you want to write a program that takes advantage of all the capabilities of the GTIA. What does the poor user with only a CTIA do? If you are commercially clever, you will have your program sense which chip is in use and adapt itself accordingly. This program will enable you to do just that:

```
100 GRAPHICS 0:REM ALWAYS USE THIS MODE
110 PRINT "NOW TESTING FOR CTIA VERSUS GTIA"
120 PRINT "================================"
130 POKE 559,58:POKE 53277,2:REM ENABLE PLAYERS
140 POKE 54279,240:REM USE ROM FOR PLAYER DATA
145 POKE 53248,80:REM CENTERED PLAYER
150 POKE 53278,0:REM CLEAR COLLISION REGISTERS
160 POKE 623,65:REM ENABLE GTIA, IF IT EXISTS
170 POKE 20,0
180 IF PEEK(20)<2 THEN 180
190 POKE 623,1:REM DISABLE GTIA
200 POKE 559,34:POKE 53277,0:REM TURN OFF
PLAYERS
210 FOR A=53261 TO 53265:POKE A,0:NEXT A:REM
(AND PLAYER DATA)
220 IF PEEK(53252) THEN PRINT "SORRY, ONLY A
CTIA":GOTO 240
230 PRINT "AHA! A GTIA."
240 END
```

First of all, to give credit where it is due, I should mention that I was inspired to try this by a remark I read in one of Craig Chamberlain's articles in the *M.A.C.E.* newsletter (Michigan Atari Computer Enthusiasts, in the Detroit area). Portions of that article were also reprinted recently in **COMPUTE!**.

But let's now discuss the program. First, we must explain why and how it works: there is no way to inquire of the chip itself which it is. Even the operating system does not know which is installed. *But* .... (There *had* to be a "but," or there wouldn't be this article.)

There are a few subtle differences between how the two chips view players and missiles. In particular, the GTIA doesn't believe that players can collide with "printed" characters, so it never reports such a collision. The CTIA, though, considers a character to be just another kind of COLORed (and SETCOLORed) display.

The first thing our listing does is insure that we have some mode zero characters on the screen (lines 100 - 120). Then we enable the player DMA and the players themselves (line 130). And we tell the chips that the player data memory is smack in the middle of the ROMs! (Why? To insure that lots of data bits will be on, forcing lots of collisions between the player and the playfield screen characters.)

With line 145, we place the player somewhere left of the center of the screen, insuring that it will collide with our printed message. Then, after clearing the collision registers (to insure that the later results will be valid), we enable the special modes of the GTIA (lines 150 and 160).

We wait for at least one full screen scan (lines 170 and 180), to be sure that the collision will "take" (if it's going to). Then we turn everything back off again (lines 190 - 210).

Finally, we inspect the collision register for player zero. If a collision did occur, it must be because the older CTIA was installed. If no collision occurred, we presume that we have a GTIA.

All of this is a little complicated, but I sincerely hope that some of you game developers out there will start designing some good GTIA-based games, now that you can have them modify themselves for the CTIA owner.

A Few Abbrev'd REMs. Period.

In his article on "The Atari Wedge" (in the November 1982 **COMPUTE!**), Charles Brannon mentions that BASIC treats a line beginning with a period as a REMark, claiming that it is a lucky fluke. Well, it really isn't a fluke. It's just one of those things that got designed into Atari BASIC and then forgotten about.

The rule for using abbreviations in Atari BASIC (and BASIC A+, naturally) is fairly simple: when a statement begins with an abbreviation (any alphabetic characters followed by a period), BASIC searches the keyword name table for the first statement name which matches the abbreviation, *starting at the first character of the abbreviation and ending at the period.*

This means, for example, that "L." will match "LIST" only because LIST is the first word in the keyword name table that begins with an "L". If "LET" had been placed before "LIST" in this table, then "L." would have been interpreted as a LET statement. Boy, aren't we lucky that LIST comes before LET!

Luck had nothing to do with it. The order of those keywords was carefully chosen to provide the maximum usability of the shortest abbreviations. (Actually, I now believe that there are a few variations in the order that might be more useful; but remember that the order was set by intuition, not experience, since the language didn't then actually exist.)

Anyway, Atari had asked for a very short abbreviation for REMark statements (e.g., "!", as is used by most Microsoft BASICs). But what could be shorter than a single period? It's even easier to use than "!" (no shift key needed). How to produce that result? Trivial! Place REM as the first statement name in the keyword table.

So try it sometime. Why type in three characters ("REM") when one will do? Of course, because of the tokenizing nature of Atari BASIC, any abbreviated statement(s) are LISTed in their full form. So "." will be LISTed as "REM".

And a P.S. for those of you into BASIC internals: note that this implies that the token value for REM must be zero, since the token values relate directly to the order of the names in the keyword table.

Page 6 Preached Again

I kind of promised myself that I would get down off my soap box this month and quit ranting and raving. But I couldn't go one whole column without a little preaching, could I?

Stay out of page 6! I can't believe it! It seems that every other article and/or utility program and/or device driver that I run across wants to place itself in page 6 (memory locations $600 to $6FF, 1536 to 1791 decimal). *It won't work*!

How can I possibly install a printer driver in page 6 and then put my player vertical move routine there and my disk block input and output and .... Ah, come on, folks. Give us a break.

If you are writing a complete "system" (a game, or data base program, or whatever), then you are naturally free to configure memory as you wish, including doing whatever you want to page 6. But if you are going to publish a utility in a magazine or include a device driver with your printer interface board or do anything that others might use or modify, *please* don't make it fixed-assembled in page 6. *Please*.

Besides, it is *not* true that BASIC leaves all of page 6 alone. If you do an INPUT from disk (or cassette or anything other than the screen), and if the data you input exceeds 128 bytes, BASIC will use at least a portion of page 6 as its buffer. (However, it is probably – not surely, just probably – safe to use memory from $680 to $6FF.)

A little history: If you examine your Atari BASIC reference, you will find that there are two memory usage tables. One claims that all of page 6 is available for the user. The other claims that only the upper half is available. In general, you should believe the latter. *It is not a design flaw nor an error* that BASIC sometimes uses the bottom half of page 6. It is necessary and documented.

I think it was someone at Atari (my rumor sources say Chris Crawford, but this is unconfirmed) who began using all of page 6 for assembly language routines. And, as I stated above, there is really nothing wrong with doing so within a "closed" environment (where you write *all* the software, both BASIC and assembler). Just don't do it for public consumption.

So what should you do, instead? The best solution is to write self-relocatable code and load it wherever there is free memory (e.g., in a BASIC string). (Showing how to write self-relocatable code might be an instructive article, in and of itself. Any takers?)

The second best solution is to perform my favorite trick: place your code at LOMEM and move LOMEM up. Even here, though, it is best to use relocatable code, so you can run under a variety of operating system configurations and varying heights of LOMEM (as I documented in last month's column).

And, last but not least, I have some good, practical (and a little bit selfish) reasons for avoiding page 6: BASIC A+ uses a good portion of it ($610 through $642, actually). Does that make us a villain? Perhaps a little, to the article writers. But we aren't that terrible: I understand that Microsoft BASIC uses *all* of page 6. And who knows what other languages and operating systems and peripheral devices and whatever will also use page 6? Why complicate both your and others' lives by putting your routines there also?

Some FORTH-Right Comments

I received a very well written and thought-out letter from Steven Weston, of Del Mar, California, regarding the benchmarks I reported in my September 1982 column. Mr. Weston shares the predilections of some others, considering FORTH to have been slighted in that column (and in the following one, I presume).

First, I should like to report that he translated the BASIC benchmark to FORTH and obtained a time of a little under 118 seconds. Which is interesting, since ValFORTH (the version he used) makes use of the Atari floating point routines, I believe. So why should it be slower than Atari BASIC? If I were guessing (which means I'm about

to take a flyer), I would presume that the floating point words for ValFORTH are written in FORTH words, instead of being written as low-level (assembly language) words. The very operation of stacking and unstacking the floating point numbers must then be relatively slow and painstaking.

If this is indeed true, then my comment is a positive one: the FORTH user indeed has the choice of implementing "commands" (words) either way, with other FORTH words or with assembly language. This flexibility is poorly supported by most other languages. (Although many C compiler implementations come close to having such accessible assembly code. C/65 functions, for example, need very little overhead in the assembly language code to "unstack" their parameters.) Want a faster FORTH instead of a smaller one? Recode some routines in assembly language.

What Benchmarks Really Test

Before going on to the second point of Mr. Weston's letter, I should like to note that I feel that perhaps he (and many other readers) missed part of the point of the benchmarks: I was really trying to show how useless any one benchmark is, since it is so easy to dream up benchmarks which show off the best features of a given language. I would be hard pressed to construct even a set of ten benchmarks which would adequately compare languages.

And even if I thought I succeeded, how much is the human interface to a language worth? PILOT is still the easiest language on the Atari to learn and interface to. *By definition*, it therefore out-benchmarks every other language for beginners. But would anyone seriously propose using PILOT for generating prime numbers? I think not. Benchmarks are usually worth the paper they are printed on and no more.

So now to Mr. Weston's second point. I quote: "...the bottom line on languages is to use that language which is best suited to the task. [With Atari BASIC] the lack of integer based math is a serious deficiency which can preclude its use by professional software authors." He goes on to ask why I don't provide a "toolbox" of integer math routines to be interfaced to Atari BASIC "instead of defending an inadequate situation."

Well. Kudos and jibes all in one it seems. Anyway, he is absolutely right: pick the language that fits the job instead of making the job fit the language. You will remember, I hope, that in a recent column I mentioned that I collect languages like some people collect games. I keep hoping to find one that will be useful to me.

But now let me disagree a little on a couple of points. And I do so because I have received too many comments in this same vein. (1) Integer math is *not* needed by all "professional software authors." The person writing a financial package needs integer math about as much as the game writer needs floating point. If you need integer math, choose a language which supports it. (2) BASIC is, unfortunately, a non-extensible language. Sure, we could put integer math routines in memory somewhere and use them from BASIC. But BASIC would still insist on thinking of its variables and constants as floating point, and the conversion time (from floating point to integer to floating point, *ad nauseam*) would wipe out all speed advantages gained. (3) I don't think Atari BASIC is an "inadequate situation." Sure, I think there are other solutions. Why else would our company produce languages such as BASIC A+ and C/65 (and probably more to come)? But "inadequate"? I think not, if it is used for and how it was meant to be used. (If anything is inadequate, it is the 6502 microprocessor, which does not lend itself to the implementation of powerful language compilers.)

But, if you are a beginner, don't let anyone (including me) pressure you into trying to learn a new language before you are ready. It is true that you are not going to write "Super Invading Packers with Tronic Fighters" with Atari BASIC. But just look at what you *can* write! Ten years ago, a computer fanatic would have sacrificed his left thumb for what we now take for granted. Seven short years ago, the "hot" computer game that everybody was rewriting (to make it fit in their expanded memory 8K byte gigantic machine) was Wumpus. Today, I seldom see a published program that doesn't make Wumpus look like something out of the dark ages. Hang in there, folks, you ain't seen nuthin' yet.

The New BASIC Standard?

Well, I finally got time to take a long, hard look at the new ANSI BASIC specification. Whew! I think the tower of Babel must have seemed organized by comparison. Even ADA and PL/1 look like closely designed languages compared to ANSI BASIC. I think that the rule in designing it was "If someone wants it, let's put it in."

You certainly won't see any microcomputer interpreter implementations of it in the near future. I estimate it would take over 80K bytes of Z-80 code to do it (which translates to maybe 100K to 120K of 6502 code). It is definitely designed to be compiled, not interpreted, and then only by big machines.

The error descriptions alone would take a few kilobytes (and they are required!). And what do lines like the following mean?

NEW FOR YOUR ATARI

MAGIC MAIL*

An exciting New Data Base Mailing and Personal Information System for your Atari 400/800** Personal Computer. Never before have all these features been designed into a single Mailing and Information System, so easy to use and so powerful.

COMPARE THESE FEATURES:

- Data Fields: Name, Address, City, State, Zip Code, Area Code, Phone Number, Birthday (m/d/y), and Entry Type.
- Up to 1027 records per Diskette!
- Less than 1 second Search Time over 95% of your Data, and 2-4 seconds over the other 5%.***
- Up to 62 SEPARATE and UNIQUE files permitted per Diskette.
- Single or Double Disk Drive Capability.
- Built in SORT, capable of Sorting ANY Data File by ANY Field with up to 10 Sub-fields.
- Command-Driven with simple English-like commands: FIND, MODIFY, PRINT, ADD, SORT, etc.
- Position Independent & Position Dependent Fields.
- Files compatible with Atari DOS II.**
- Recorded on High-Quality Verbatim Diskettes.
- Every Diskette Pre-tested for Bad Sectors and recording integrity.
- Versions for: Epson MX-80, Prowriter, Centronics
- Print Single-width Mailing Labels or Directory Listing.
- 100% Machine Language.
- 16 K Minimum Memory.
- 60-Page User Manual.
- 90 Day Buyer Protection Guarantee.

Call or write for FREE additional information.

*MAGIC MAIL is a Trademark of A-BIT-BETTER Software
**ATARI is a Registered Trademark of Atari Inc.
***Search time is for an Indexed File.

ABBS
A-Bit-Better Software
Post Office Box 28
Laurel Md 20707
(301) 953-7256

Dealer Inquiries Invited.

ONLY $59.95
*Plus tax, shipping and handling.
VISA AND MASTERCARD ACCEPTED.

Use the handy reader service cards in the back of the magazine for information on products advertised in **COMPUTE!**

Maxell Floppy Disks

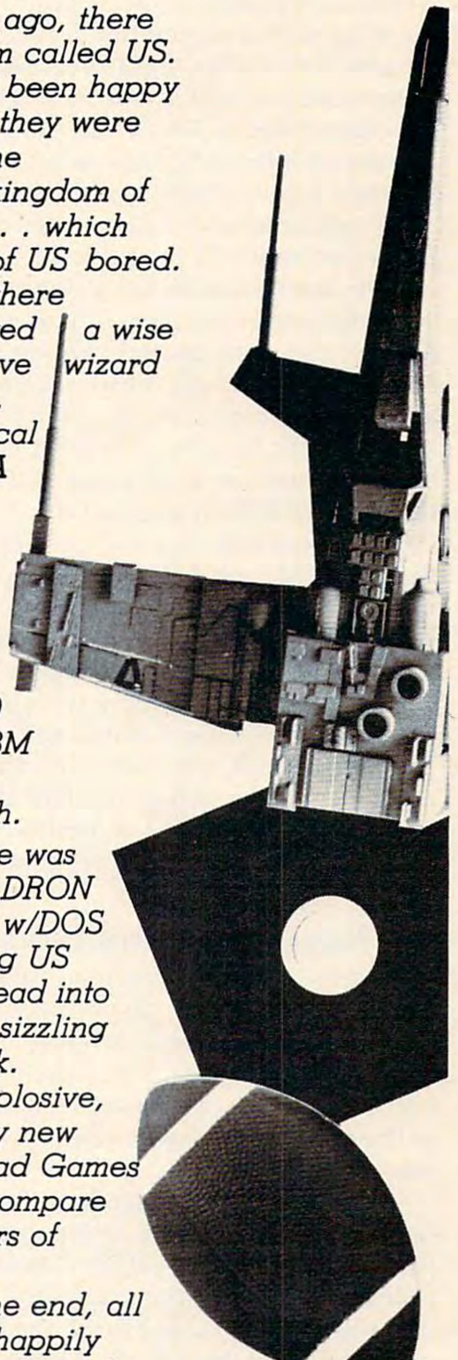
The Mini-Disks with maximum quality.

Dealer inquiries invited. C.O.D's accepted.
Call FREE (800) 235-4137.

PACIFIC EXCHANGES
100 Foothill Blvd.
San Luis Obispo, CA 93401
In Cal. call (800) 592-5935 or (805)543-1037

MasterCard VISA

A True Story



Not so long ago, there was a kingdom called US. All of US had been happy until one day they were attacked by the neighboring kingdom of Bad Games . . . which rendered all of US bored.

*East of US there fortunately lived a wise and imaginative wizard named **NEXA**. With his magical powers, **NEXA** created two games for US. There was SUPERBOWL FOOTBALL (Atari 400/800 w/ 48K and IBM PC) with pro-ball punch. And then there was DELTA SQUADRON (Apple II 64K w/DOS 3.3), launching US light years ahead into the heart of a sizzling starbase attack.*

*X-citing, X-plosive, and X-emplary new games. The Bad Games just couldn't compare with the powers of **NEXA**.*

*And so in the end, all of US played happily ever after. (Fortunately for us, **NEXA** created 7 other games that year.)*

NEXA
NEXA CORPORATION
P.O. Box 26468 San Francisco, CA 94126-6468

```
MAT PRINT #N, SAME, IF THERE EXIT FOR: A$;
   B$, C
ASK #3: access outin, organization org$, rectype
   internal, pointer p$
```

I am disappointed. I had hoped that the committee would distill the best of the various BASICs and come up with a somewhat enhanced version of the original ANSI standard BASIC. Instead, they seem to have distilled out the biggest features of the biggest BASICs they can find. And who will use the standard? Not the micros. (At least not in the near future. I understand that Microsoft's representative on the committee dropped out. From frustration? I would have.) Not those who need to contract with the government. (Soon, you will *have* to use ADA if you work with the defense department and various allied agencies.) Not the big business computer users. (They can't afford to go from COBOL, a clumsy but eminently maintainable language, to a BASIC as kludged up by the committee, with a lack of the data structures that made COBOL successful.)

I guess I believed that the only BASIC users that would be left in a few years would be the hobbyists and the time-sharing companies. Now, I think the only ANSI BASIC users will be the time-sharing companies. Maybe.

As much as I disagree with much of what Microsoft has done, I would rather have seen Microsoft BASIC (version 5, on the CP/M machines) become the standard than the hodgepodge the ANSI committee has selected. ANSI, on a scale of 10, I give you a 2.

The New Atari Computers

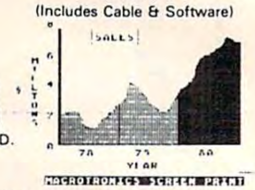
Perhaps by the time you read this, the new Atari computers will be on display at the Consumer Electronics Show (early January, in Las Vegas). Don't expect any real surprises. I expect to hear of a 64K machine (with no software to take advantage of the extra 16K). And probably a low-end 16K machine.

Obviously, Atari needs to get in there and fight with Commodore, both on price and features. Price is easy. Features? Well, if Commodore follows through as they claim they will, it could be a tough fight. And I think the 400 replacement might outstrip the VIC-20. I guess I should note that I am not as much of an Atari loyalist as this paragraph makes me sound. It's just that I like a good, competitive race. The consumer is bound to win.

Oh, yes, one more thing. No more right-hand cartridge slot in the new machines. And no memory board slots at all. Ouch? I don't know. I hope there will be a good way to expand the new machines, but we will all have to wait to see what it is.

Basically BASIC

All this talk about benchmarks and ANSI BASIC has made me regain interest in a project I thought of doing a while back. So, starting next month, we will begin writing a BASIC interpreter right here in this column. And we will write it in BASIC. Interested? I am. ©



ComputAbility™ presents the Newest in Software Products for Atari

ATARI

SUPER SPECIALS

GORF $29.95 cartridge	KICKBACK $34.95 cartridge	JUMBO JET PILOT $34.95 cartridge
WIZARD OF WOR $29.95 cartridge	SOCCER $34.95 cartridge	SUBMARINE COMMANDER $34.95 cartridge

Prices effective January 1 through January 31, 1983

Buy any 3 programs and receive a *FREE** SLIK STIK

ATARI
- CONVERSATIONAL LANGUAGES - T 43.95
- INVITATION TO PROGRAMMING 2 & 3 - T ... 21.95
- MUSIC COMPOSER - C 32.95
- MY FIRST ALPHABET - D ... 28.95
- TOUCH TYPING - T 18.95
- WORD PROCESSOR - D ... 112.95
- BOOKKEEPER - D 112.95
- HOME FILING MANAGER - D 37.95
- MAILING LIST - T 18.95
- ASTEROIDS - C 26.95
- CAVERNS OF MARS - D...... 28.95
- COMPUTER CHESS - C 26.95
- MISSILE COMMAND - C 26.95
- SPACE INVADERS - C 26.95
- SUPER BREAKOUT - C 26.95
- STAR RAIDERS - C 32.95
- ASSEMBLY EDITOR - C 44.95
- BASIC - C 44.95
- MACROASSEMBLER - D 65.95
- MICROSOFT BASIC - D 65.95
- PILOT (HOME PACKAGE) - C .58.95

ADVENTURE INTERNATIONAL
- REAR GUARD - D 20.95
- PREPPIE - D/T 23.95
- S.A.G.A. ADVENTURES - D . 23.95

ANALOG
- RACE IN SPACE - D/T 20.95
- CARNIVAL - D/T 20.95
- SUNDAY DRIVER - D/T 23.95
- CRASH DIVE! - D/T 23.95

BIG FIVE
- COAL MINER 2049'ER - Cart . 39.95

APX
- ALL ITEMS - 20% OFF RETAIL. WE CARRY THE ENTIRE APX CATALOG!

AUTOMATED SIMULATIONS
- TEMPLE OF APSHAI - D/T ... 31.95
- UPPER REACHES OF APSHAI - D/T 16.95
- CRUSH, CRUMBLE & CHOMP - D/T 23.95
- RICOCHET - D/T 16.95
- CRYPT OF THE UNDEAD - D 23.95
- ARMOR ASSAULT - D 31.95
- DRAGON'S EYE - D......... 23.95
- ESCAPE FROM VULCAN'S ISLE - D 23.95
- KING ARTHUR'S HEIR - D ... 23.95
- MONSTER MAZE - Cart 31.95
- PLATTERMANIA - Cart 31.95
- THE NIGHTMARE - D 23.95

DATASOFT
- SHOOTING ARCADE - D/T . 23.95
- PACIFIC COAST HIGHWAY - D/T 23.95
- CLOWNS & BALLOONS - D/T 23.95
- MICROPAINTER - D 27.95
- CANYON CLIMBER - D/T ... 23.95
- FATHOMS FORTY - D 27.95
- O'RILEY'S MINE - D/T 27.95
- ROSEN'S BRIGADE - D/T ... 27.95
- SANDS OF EGYPT - D 31.95
- SPELL WIZARD - D 63.95

BRODERBUND
- APPLE PANIC - D/T 23.95
- STELLAR SHUTTLE - D/T ... 23.95
- DAVID'S MIDNIGHT MAGIC - D 27.95
- STAR BLAZER - D 25.50
- TRACK ATTACK - D 23.95
- LABYRINTH - D/T 23.95
- SERPENTINE - D 23.95
- DUELING DIGITS - D 23.95
- DEADLY SECRETS - D 27.95
- CHOPLIFTER - D 27.95
- GENETIC DRIFT - D/T 23.95
- SEAFOX - D 23.95

COSMI
- AZTEC CHALLENGER - D... 16.95
- AZTEC CHALLENGER - T ... 11.95
- CRYPTS OF PLUMBOUS - D 13.95
- CRYPTS OF PLUMBOUS - T .. 8.95
- GALACTIC AVENGER - D ... 13.95
- GALACTIC AVENGER - T 8.95
- SPIDER INVASION - D 16.95
- SPIDER INVASION - T 11.95

FIRST STAR
- ASTRO CHASE - D/T 23.95

ON-LINE
- JAWBREAKER - D/T 23.95
- ULTIMA I - D 31.95
- CROSSFIRE - D/T 23.95
- THRESHOLD - D 31.95
- ULTIMA II - D 44.95
- MOUSKATTACK - D 27.95
- FROGGER - D/T 27.95

***COUPON PROGRAM**

The purchase of each program (with the exception of Super Specials, Atari and APX) will earn you 1 COMPUTABILITY DIVIDEND COUPON. Save 3 coupons and redeem them for your choice of Slik Stik, a Lefty Adaptor, or an Extension Cable. You pay only a $2.50 shipping & handling charge.

I.D.S.I.
- POOL 1.5 - D 27.95
- POOL 400 - Cart 31.95
- SPEEDWAY BLAST - Cart ... 31.95

ROKLAN
- DELUXE INVADERS - Cart .. 31.95
- GORF - Cart 35.95
- WIZARD OF WOR - Cart..... 35.95

SENTIENT
- CYBORG - D............... 27.95
- GOLD RUSH - D/T 27.95

SIRIUS
- SNAKE BYTE - D 23.95
- SPACE EGGS - D 23.95
- CYCLOD - D............... 23.95
- SNEAKERS - D 23.95
- BANDITS - D............... 27.95
- WAY OUT - D 31.95

SYNAPSE
- PROTECTOR - D/T 23.95
- SLIME - D/T............... 23.95
- SHAMUS - D/T 23.95
- FORT APOCALYPSE - D/T .. 23.95
- CLAIM JUMPER - D/T....... 27.95
- NAUTILUS - D/T 23.95
- PICNIC PARANOIA - D/T.... 27.95
- RAPTILIAN - D/T 27.95

D - Disk   T - Cassette   C - Cartridge

WE CARRY HUNDREDS OF ITEMS FOR ATARI 400/800, ASK FOR OUR FREE CATALOG.

ATARI is a trademark of ATARI, Inc.

Starfighter

The Ultimate Joystick
- 2 Year Warranty
- More Accurate
- Easier to Hold

$16.95

SLIK STIK
- 90 Day Warranty
- Easy Ball Top Control.

Suncom

$9.95

LEFTY JOYSTICK ADAPTOR
Adapts to *any* Atari Joystick.
Moves fire button to top Right. $9.95

EXTENSION CABLE (5 ft.)
Adapts to *any* Atari controller. $6.95

ALL JOYSTICKS WORK WITH: Atari VCS, Sears Telegame, Commodore VIC 20, Atari 400 800 (All products have registered trademarks)

30 Day Money Back Guarantee on all Suncom Products

PERCOM D/D DISK DRIVE
$579.00

MOSAIC 32K RAM BOARD
$97.95

Mastercard/VISA Order Toll Free 800-558-0003

In Wisc. Call 414/351-2007

ORDERING INFORMATION

To order by mail send money order, certified check or personal check (allow 14 days to clear) to COMPUTABILITY. Include $2.00 shipping on software orders and $2.50 shipping on hardware orders (FREE OR PURCHASED). Mastercard & VISA please include card number and expiration date. WI residents please add 5% sales tax. Outside of continental U.S.A. please add 15% shipping (U.S. Funds only). Prices subject to change without notice.

Order Hours
Monday - Friday
12 pm - 9 pm C.S.T.

ComputAbility
P.O. Box 17882
Milwaukee, WI 53217

Michael Day

Computers And Communication

Like the brains of animals, computers have evolved from dedicated, single-mindedness into general-purpose information processors. This month's column takes us through a brief survey of the changes in the capabilities of micro-, mini-, and mainframe computers, concluding with how telecommunications are generally used with each type of computer.

When computers were first introduced, they were designed and *wired* to perform a particular job. When the job was done, either it was performed again with new information inserted, or the computer was *rebuilt* to run a different job. A patch board reprogrammed the computer by reconfiguring the way the hardware was put together. Such reprogramming was necessary because there was just not enough memory available to hold both the program and data at the same time. (1K of memory was a lot back then.)

Batch Processing

Later, as the available memory size increased (to a whole 4K), software programmable machines were built. This improvement allowed the machine to be automatically "built" by the same mechanism that was used to load the data into the machine. Since most of the time spent on the computer was in "building" it for the particular job at hand, this improvement also permitted an interesting approach to processing, called *batch processing*.

Batch processing involves loading the program and data into the machine from a mass storage device (usually a tape drive) and running the program. The results are then saved (printed, put back on the tape or on punch cards). The program and data are then purged from the system, and a new program/data job is loaded into the machine. Batch processing helped increase the popularity of these very expensive machines. But they required intensive use to make them worth the cost.

Using The Computer's Time

As computer costs increased, even batch processing was insufficient to offset the costs of the computer. Analysis of computer operations showed that much of the computer's time was spent waiting for information to be given to it. If the computer could be subdivided into individually operating parts (or *subprocessors*), it would be possible to request the information from a slow external device, such as a tape drive, and while the information was being retrieved, another job could be loaded into the computer and operated on. The processor could later return to the original job and finish it.

Eliminating the computer's inactive or waiting times greatly contributed to the efficient utilization of the computer. By sharing the computer's resources, several different jobs could run at the same time. Careful control of access to the various parts of the system could actually make the computer work like several different computers at the same time. Several users could therefore use the computer without interfering with or being aware of each other. This brought into being *time-share* computers. Since a user seldom uses the system continuously, someone else could use it when it would otherwise be idle.

Patterns Of Development

When the minicomputer (bigger than a "micro," smaller than a "mainframe") came into being, it went through the same sequence of development. It started out as a computer designed to solve a particular problem and developed into a general purpose machine. The difference: by the time the minicomputer was developed, it was cheaper to design it to be program-controlled rather than to have fixed control. This was true because many parts of the machine could be shared by many parts of the program. Because it was not necessary to have individual parts available for each action the computer performed, the computer could actually "rebuild" itself on the fly.

The result was a shift from the mainframe concept of computing. Since large mainframe computers operate best where there are large chunks of data to be processed, they tend to be run mostly as batch processing machines where an entire job, or a large portion of it, is operated on before moving on to something else. The minicomputer, however, is more suitable to applications where the job requirements are varied and rapidly shifting. They are thus most often found in time-share applications where the ability to handle a large number of jobs simultaneously is more important than the actual processing time. The minicomputer can't meet the raw crunch power of the mainframe, but it surpasses the mainframe in adaptability.

A Rapid Change In Microcomputers

When the microcomputer came along, again the same development pattern was followed. Like the mainframes and minicomputers, the microcomputer was initially developed for single-job applications. But it moved on to more generalized applications more rapidly than either of the other computers. Since the microcomputer was developed as a result of Large Scale Integration (LSI chips), the computers could be created at a very low development cost and an unbelievably low production cost.

The microcomputer too does not have the crunch power of the mainframe, nor does it have the adaptability of the mini. What it does have is low cost of implementation, which makes it the first computer ideally suited to fixed job applications. Some of these applications are found in the calculator, smart thermostats, microwave oven controllers, etc.

In between these fixed applications and the minicomputer are the high level microcomputers (which are coming to be called personal computers). These computers, though sometimes not suitable to the rapidly changing job environment of the mini, do have general processing capabilities. This makes them ideal for personal computing since only a single job generally needs to be run at one time, but the types of jobs that the computer is required to perform are varied.

Telecommunications Needs

You might be wondering, "That's all very fine, but what has this got to do with telecommunications?"

Actually, there is a very definite relationship between the type of computer and its needs in telecommunications. Large mainframes seldom need extensive telecommunications. When they do have such a need, it generally involves special communication circuits designed specifically for the computer system, such as airline or hotel reservation systems, or banking systems. Minicomputers, because they are highly adaptive, tend to use a wide variety of communications capabilities. Examples are the many time-share systems and service bureaus.

Microcomputers, as opposed to *personal* computers, generally don't have a need for telecommunications. When they do, the telecommunications tend to be specific to the device or application. In fact, in some applications, the microcomputer is the communications device, as it is with some of the high-powered modems available.

Finally, with the personal computer, communications vary depending on the use to which the computer is put. Generally, the application consists of machine to machine communications between users or connection to a large data base service like Micronet or The Source.

These are only generalizations, of course, and it is quite easy to find exceptions to the rule. You can find microcomputers handling multiple communications devices, and fully dedicated minicomputers that have no outside communications at all. As a general rule, however, these basic patterns prevail. ©

Jim Butterfield, Associate Editor

Speed Demon

Machine language programs are fast. So fast, in fact, that for many applications we can consider them to be instantaneous. That's good, of course, but sometimes we have to take steps to restrain the program's speed.

The first moon lander program that I wrote was carefully coded, and the calculations were carefully checked to see that they were correct. I was rather taken aback to discover that the instant I pressed GO I found myself crashed on the lunar surface. All the calculations had taken place correctly, but everything worked too fast.

Waiting For The World

In most cases, your machine language program is controlled by the speed of external events. If you're waiting for a user to type a line at the keyboard, chances are that machine language is running at about a ten thousandth of its potential speed. It can do nothing until the next key is pressed; and chances are that it will do little until a line has been completed by pressing the RETURN (or ENTER) key.

Even when we're not waiting for the operator, we are usually waiting for some external process. Using the printer? Your program will spend most of its time waiting for the printer to be ready for the next character. Disk? Same thing. Communications lines fall into the same category, but there's a difference: even though the transfer rate of characters to and from the communications interface is relatively slow, there will often be a need to check it very frequently.

The result is that your program speed is usually determined by the speed of external events. In this case, the "instantaneous" assumption is quite legitimate.

Let's take another example: you're printing material on the screen. Now you can deliver characters at blinding speed; but there's a limit to how fast a user can read. Better slow it down, or your program will be useless.

Compute Bound

Programs that spend most of their time waiting for external events are called *I/O-bound*. Sounds like a good name for a sailing ship, but it really means that if you could get a faster printer, disk, or whatever, your program would run significantly faster. Your speed is bound to the speed of these devices.

There are other programs that do a great deal of computation: they tend to be compute-bound. No, that doesn't mean that you plan to submit them to your favorite magazine; it means that if you could calculate faster, you'd get more work done. Compute-bound programs are often mathematical in nature: to calculate the millionth prime number you won't care much about your printer speed; you want the computation to be fast. Sorting programs are often compute-bound: there's a lot of calculation needed there.

It's often wise to think about your program in terms of its potential: will it be I/O-bound or compute-bound? It will give you an idea of where you might place extra effort in order to speed things up.

Slowing Down

There are many cases where we deliberately wish to slow down the speed of a machine language program. Animation is a prime example: you don't want your space ships, bombs, or cats to always travel at supersonic speed. Indeed, if you used maximum speed you'd never see them.

There are technical reasons to want to slow

COMPUTE! Back Issues

Here are some of the applications, tutorials, and games from available back issues of **COMPUTE!**. Each issue contains much, much more than there's space here to list, but here are some highlights:

February 1981: Simulating PRINT USING, Using the Atari as a Terminal for Telecommunications, Attach a Printer to the Atari, Double Density Graphing on C1P, Commodore Disk Systems, PET Crash Prevention, A 25¢ Apple II Clock.

May 1981: Named GOSUB/GOTO in Applesoft, Generating Lower Case Text on Apple II, Copy Atari Screens to the Printer, Disk Directory Printer for Atari, Realtime Clock on Atari, PET BASIC Delete Utility, PET Calculated Bar Graphs, Running 40 Column Programs on a CBM 8032.

June 1981: Computer Using Educators (CUE) on Software Pricing, Apple II Hires Character Generator, Ever- expanding Apple Power, Color Burst for Atari, Mixing Atari Graphics Modes 0 and 8, Relocating PET BASIC Programs, An Assembler In BASIC for PET, QuadraPET: Multitasking?

July 1981: Home Heating and Cooling, Animating Integer BASIC Lores Graphics, The Apple Hires Shape Writer, Adding a Voice Track to Atari Programs, Machine Language Atari Joystick Driver, Four Screen Utilities for the PET, Saving Machine Language Programs on PET Tape Headers, Commodore ROM Systems, The Voracious Butterfly on OSI.

August 1981: Minimize Code and Maximize Speed, Apple Disk Motor Control, A Cassette Tape Monitor for the Apple, Easy Reading of the Atari Joystick, Blockade Game for the Atari, Atari Sound Utility, The CBM "Fat 40," Keyword for PET, CBM/PET Loading, Chaining, and Overlaying.

October 1981: Automatic DATA Statements for CBM and Atari, VIC News, Undeletable Lines on Apple, PET, VIC, Budgeting on the Apple, Switching Cleanly from Text to Graphics on Apple, Atari Cassette Boot-tapes, Atari Variable Name Utility, Atari Program Library, Train your PET to Run VIC Programs, Interface a BSR Remote Control System to PET, A General Purpose BCD to Binary Routine, Converting to Fat-40 PET.

December 1981: Saving Fuel $$ (Multiple Computers: versions for Apple, PET, and Atari), Unscramble Game (multiple computers), Maze Generator (multiple computers), Animating Applesoft Graphics, A Simple Printer Interface for the Apple II, A Simple Atari Wordprocessor, Adding High Speed Vertical Positioning to Atari P/M Graphics, OSI Supercursor, A Look At SuperPET, Supermon for PET/CBM, PET Mine Maze Game.

January 1982: Invest (multiple computers), Developing a Business Algorithm (multiple computers), Apple Addresses, Lowercase with Unmodified Apple, Cryptogram Game for Atari, Superfont: Design Special Character Sets on Atari, PET Repairs for the Amateur, Micromon for PET, Self-modifying Programs in PET BASIC, Tinymon: a VIC Monitor, Vic Color Tips, VIC Memory Map, ZAP: A VIC Game.

February 1982: Insurance Inventory (multiple computers), Musical Transposition (multiple computers), Multitasking Emulator (multiple computers), Disassemble Apple Programs from BASIC, Plotting Polar Graphs on Apple, Atari P/M Graphics Made Easy, Atari PILOT, Put A Rainbow in your Atari, Marquee for PET, PET Disk Disassembler, VIC Paddles and Keyboard, VIC Timekeeping.

March 1982: Word Hunt Game (multiple computers), Infinite Precision Multiply (multiple computers), Atari Concentration Game, VIC Starfight Game, CBM BASIC 4.0 To Upgrade Conversion Kit, Apple Addresses, VIC Maps, EPROM Reliability, Atari Ghost Programming, Atari Machine Language Sort, Random Music Composition on PET, Comment Your Apple II Catalog.

April 1982: Track Down Those Memory Bugs (multiple computers), Shooting Stars Game (multiple computers), Intelligent Input Subroutines (multiple computers), Ultracube for Atari, Customizing Apple's Copy Program, Using PET/CBM In The High School Physics Lab, Grading Exams on a Microcomputer (multiple computers), Atari Mailing List, Renumber VIC Programs The Easy Way, Browsing the VIC Chip, Disk Checkout for PET/CBM.

May 1982: VIC Meteor Maze Game, Atari Disk Drive Speed Check, Modifying Apple's Floating Point BASIC, Fast Sort For PET/CBM, Extra Atari Colors Through Artifacting, Life Insurance Estimator (multiple computers), PET Screen Input, Getting The Most Out Of VIC's 5000 Bytes.

June 1982: Outpost Game (multiple computers), Apple Pascal Lister, Income Property (multiple computers), VIC Intelligent Videodisc System, Atari Disk Operating Systems, PET/Apple Search, A Self-modifying Atari P/M Utility, Use Atari Joysticks with VIC, VIC/PET Program Transfers.

July 1982: Gold Miner Game (Atari and VIC), IRA Planner (multiple computers), Atari Video Graphics, Apple DOS Changer, Super QuadraPET, VIC Overview, Maze Race (multiple computers), Direct Access File Editor (PET and Atari), VIC Super Expander Memory Map, Using The 6560 Video Interface Chip, PET Compactor, Headless FORTH Metacompilation, Test RAM Nondestructively (multiple computers).

August 1982: The New Wave Of Personal Computers, Household Budget Manager (multiple computers), Word Games (multiple computers), Color Computer Home Energy Monitor, Intelligent Apple Filing Cabinet, Guess That Animal (multiple computers), PET/CBM Inner BASIC, VIC Communications, Keyprint Compendium, Animation With Atari, VIC Curiosities, Atari Substring Search, PET and VIC Electric Eraser.

September 1982: Apple and Atari and the Sounds of *TRON*, Commodore Automatic Disk Boot, VIC Joysticks, Three Atari GTIA Articles, Color Computer Graphics, The Apple Pilot Language, Sprites and Sound on the Commodore 64, Peripheral Vision Exerciser (multiple computers), Banish INPUT Statements (multiple computers), Charades (multiple computers), PET Pointer Sort, VIC Pause, Mapping Machine Language, Editing Atari BASIC With the Assembler Cartridge, Process Any Apple Disk File.

Home and Educational COMPUTING! (Fall 1981 and Summer 1981 – count as one back issue): Exploring The Rainbow Machine, VIC As Super Calculator, Custom Characters, Alternate Screens, Automatic Line Numbers, Using The Joystick (Spacewar Game), Fast Tape Locater, Window, VIC Memory Map.

Back issues are $3 each or six for $15. Price includes freight in the US. Outside the US add $1 per magazine ordered for surface postage. $4 per magazine for air mail postage. All back issues subject to availability.

In the Continental US call
TOLL FREE 800-334-0868
(In NC Call 919-275-9809)

Or write to **COMPUTE!** Back Issues, P.O. Box 5406, Greensboro, NC 27403 USA. Prepayment required in US funds. MasterCard, Visa and American Express accepted. North Carolina Residents add 4% sales tax.

down certain activities. Some types of interfaces want you to hold a voltage at a certain level for a minimum amount of time before you take it away again. You may need to "stall" for a few instructions (or a few dozen) to make sure that you're doing the job right.

The Stall Loop

The simplest way to slow things down is to kill time in a stall loop. If you're not doing anything else anyway, this is quite sensible and easy to do. We might code:

```
        LDX #$00
LOOP    DEX
        BNE LOOP
```

At a typical clock rate of 1 Mhz (a million cycles per second) the above routine will waste a little over a millisecond of time. You could make the time shorter by changing the LDX value at the start. For longer delays, you use a loop within a loop:

```
        LDY #$00
        LDX #$00
LOOP    DEX
        BNE LOOP
        DEY
        BNE LOOP
```

This will waste almost a third of a second as written above; change the LDY to reduce the delay.

The Timer

If the time is moderately long and you have other things to do, you may set the desired time into a timer and check it occasionally to see how the time has been going. Timers are part of the interface chips – the 6522 VIA has two of them, for example. They work a little like kitchen timers: you put the desired time in and it runs downward toward zero, showing you the time remaining. Time runs very quickly in these, however: the maximum time is often something like a fifteenth of a second. Don't try to time a boiled egg unless you either call the timer many times or you like it really soft-boiled.

When you have more than one event to time, it's nevertheless often best to stay with just one timer. Juggling various timers can be more work than just setting the next expected event into a single one. When you have numerous different things going on, you can often still work by a single timer, as we'll explain.

Countdowns

It's often convenient to have a single timer, and clock all events on a "countdown" basis. The timer can run at fixed intervals – on the PET, you can often use the interrupt timing of 1/60 second to clock many events.

The trick is this: whenever your timer signals, count one for each event you have going. You can count up or down; but when you have counted a fixed value, it's time to handle that particular process.

An example: you have a game involving tanks, planes, bombs and bullets (the usual destructive thing). On a sixtieth-of-a-second timer, you might move a tank every 20 time units; a plane, every 10 time units; a bomb, every eight; and a bullet every five. You don't need a dozen different timers: every time the bullet counter reaches five, you move it to the next spot of the screen and see what you've shot down.

Machine language is fast, and often seems instantaneous. It's often so much faster than other processes in the computer that we don't need to worry about speed calculations at all.

Sometimes machine language is too fast. When that happens, there are ways of slowing it down.

It's hard to believe that you can be so speed rich that you have to rein back your program, but it can happen. ©

PROGRAMMING THE TI

C. Regena

We are happy to welcome C. Regena and her new, monthly TI-99/4A column to the pages of **COMPUTE!**. *She has extensive experience in personal and educational computing and has written numerous articles on TI computers. To start things off, here is an overview of hardware, software, and miscellaneous resources for the TI.*

Welcome to the world of the TI-99/4A computer! (This column is also addressed to TI-99/4 owners and users, but since the "A" is the newer and more plentiful model, I'll refer to both computers when I write "TI-99/4A.") For home, personal, and educational applications, the TI-99/4A computer is a very powerful machine. In this column I'd like to illustrate some of the features unique to this microcomputer.

Extraordinary Graphics And Sound

Graphics. You may easily define your own high-resolution (detailed) graphics characters. There are 16 colors, and you may use all 16 on the screen *at the same time* in high-resolution graphics (unlike other computers). You may also use text anywhere on the screen at the same time you use high resolution graphics. Most other microcomputers are limited when combining text with graphics.

Music. You may play up to three notes and one noise for a specified time using *one* statement. The music is specified by a number which represents a frequency of 110 Hz to 44733 Hz, tones from low A on the bass clef up to out of human hearing range. The tone may be between regular musical notes. An example which plays a three-note, C-major chord for three seconds is:

```
CALL SOUND(3000,262,6,330,4,440,2)
```

The first number is the duration in milliseconds, in this case 3000. The next numbers are frequency and loudness for each note. You may also add a "frequency" of -1 through -8 and a loudness for the noise generator. You may combine tones and noises for all kinds of sounds – everything from classical music to sound effects from outer space.

Combining music and graphics. "Computer choreography" is possible because other statements (including graphics) may be executed while music is played. You may illustrate a song, for example. Or if you have a game program, you may make calculations while you are making a noise. The computer will play music and execute statements until the duration runs out or until the program comes to another CALL SOUND statement with a positive duration. A negative number for the duration will start that CALL SOUND statement even if the first duration has not finished. Try using a FOR/NEXT loop to vary any of the parameters for special effects. Here is a sample using just one tone:

```
100 FOR N=500 TO 880 STEP 20
110 CALL SOUND(-99,N,2)
120 NEXT N
130 FOR N=880 TO 500 STEP -20
140 CALL SOUND(-99,N,2)
150 NEXT N
160 GOTO 100
```

Noises. Using negative durations and combinations of music and noise numbers for frequency, you can make all sorts of synthesized noises. Quite often with noises you will want to use a FOR/NEXT loop and vary the loudness parameter.

Built-in BASIC. The programming language of TI BASIC is built into the main console – nothing extra to buy. The TI BASIC language is an excellent language for learning how to program, yet it is powerful enough for an experienced mathematician because of the built-in functions.

String manipulations. String (non-number) manipulations are also very powerful. Here is a sample program to print a phrase A$ on the screen starting at row R and column C:

```
100 FOR I=1 TO LEN(A$)
110 CALL HCHAR(R,C+I-1,ASC(SEG$(A$,I,1)))
120 NEXT I
```

The loop will go from 1 to the LENgth of the phrase A$. String variable names must always end with a dollar sign. SEG$ takes a SEGment of the phrase. In this case we are starting at the left side and taking one letter at a time. ASC gets the ASCII character code value of the character in the phrase. CALL HCHAR uses a graphic method to place the character on the screen at a certain row and column.

No Variable Name Worries

Variable naming. In your own programming on the TI-99/4A you may use meaningful variable names, although in many microcomputers the BASIC language recognizes only two letters – or a letter and a number – for a variable name. For example, if you have a program with the variable name BLUE and another variable name BLACK, other computers may recognize only one variable, BL, but the TI-99/4A knows you are using two variables. You also do *not* have to worry about embedded reserved words in variable names.

Documentation. Two excellent manuals are included with the computer. One teaches you programming in TI BASIC. The manual is very easy to understand, and a person with no previous computer experience can learn to program with this book. Also included is the *User's Reference Manual*, which may cost over $15 for other computers. The reference manual, which is in looseleaf form, includes all the commands along with explanations and sample programs.

Plug-in modules. The easiest way to use the TI-99/4A is to insert a command module which contains a program. Modules are available for a variety of applications. The variation in price is largely dependent on the amount of memory built into the module. The modules actually add memory to the computer while they are being used.

Speech. Even though this feature is not built in, I am going to include speech in this list of unique features of the TI-99/4A because it is very easy to use and because, if you purchase six command modules before January 31, you can get the TI Speech Synthesizer free. The speech synthesizer is a small box that attaches to the side of your console. Command modules are available for you to program your own speech.

16-bit microprocessor. The TI-99/4A uses a 9900, 16-bit microprocessor, which offers more computing power and greater expansion and configuration flexibility than an eight-bit microprocessor. You can get higher numeric precision and simplified memory addressing.

Programmer's aids. Programmers will enjoy the easy line editing features. Various function keys allow you to insert or delete characters or to erase or clear a line. There is also a TRACE command to help in debugging.

Another feature programmers like is the built-in automatic numbering. Just type in NUM, press ENTER, and you can start programming. The line numbers start with 100 and automatically increment by 10. You may specify any starting number and increment. NUM 5,2 will start with line 5 then increment by 2.

After you have programmed and added or deleted statements here and there, you'll enjoy the automatic resequencing command, RES, which will automatically renumber your statements, including all statement numbers referenced by other statements.

There is a lot built into the TI-99/4A, and I have only touched on some features this month. Future columns will go into more detail, and I hope to be able to answer your questions and present programs and ideas to help you really enjoy your computer.

Since many readers may be new TI-99/4A owners and users, let's also describe some *peripherals* – hardware you can add on to your basic console. You may have noticed that buying a computer is much like buying a house – you can buy the basic house (computer), but then you need to add furniture (programs or software) to make it livable (usable), and soon you want to make major improvements (add peripherals).

Using The Cassette Recorder

Cassette. Probably one of the first items you'll need is a cassette cable to connect a cassette recorder to the computer to save your own programs or to use cassette programs available on a variety of subjects and applications. Nearly any cassette recorder is acceptable; however, the TI-99/4A is more critical on how you set the volume control than is the TI-99/4. In general, a battery-operated recorder does not work well enough for accurate data retrieval. Also, your recorder should have a tone control and a volume control. I have had the greatest success using the Panasonic RQ2309A cassette recorder.

Page I-9 in the *User's Reference Guide* tells how to connect the cassette cable, and the pages following describe how to save and load data from modules. Page II-42 shows an example of how to load a program that you have saved or purchased. Some other hints for using the cassette recorder are:

Turn the tone control to the highest setting.

Start with the volume about mid-range.

Follow the instructions after you type in OLD CS1.

If you get the message NO DATA FOUND, increase the volume.

If you get the message ERROR IN DATA, decrease the volume.

Sometimes a fraction of a change in volume can make all the difference in your success in reading a program. Once in a while, if I alternate between the two error messages at a volume setting near 2 or 3, I turn the volume to about 8 or 9 and the program will load.

The smallest jack of the cassette cable goes into the remote switch of the cassette recorder so the computer can turn the recorder on and off automatically. If the recorder does not turn on and off properly, simply remove the remote jack from the plug. You can operate the cassette recorder manually to save and load programs. For programs using the cassette recorder for data entry, you will need the remote capability. An adapter is available for the remote switch.

Disk drives. You can save and retrieve data or programs on a diskette much more quickly than by using a cassette system. The TI-99/4A uses 5¼-inch, single-sided, soft-sectored diskettes. To connect a disk drive, you also need a disk controller. One disk controller can handle up to three disk drives. Many business applications require two disk drives.

Memory Expansion. The TI Memory Expansion is for 32K RAM, and you need a module that will access it. You cannot use it with console BASIC. Extended BASIC does not require the memory expansion but can use it. Pascal, TI Logo, and Editor/Assembler require the memory expansion.

Peripheral Box. The "old" method had each peripheral in a separate "box" connected to the computer or the previous peripheral; each had its own power cord. The "new" system is the peripheral box, which has its own power supply and slots for cards for the RS-232 interface, memory expansion, disk controller, P-code, one disk drive, and possible future cards.

Monitor. Although the TI-99/4A may be connected to your regular television set, Texas Instruments has a very attractive, ten-inch, color monitor. The monitor gives a very clear, sharp picture and may be connected to other microcomputers as well as the TI-99/4A.

Making The Computer Speak

Speech. The TI Speech Synthesizer allows you to hear the computer speak to you. You will need a command module with built-in speech to hear the computer speak.

To program your own speech or to use any cassette or disk programs that use speech, you will need a module. Speech Editor and Extended BASIC have speech capabilities with a given list of words. Terminal Emulator II allows unlimited speech; the accompanying documentation gives you ideas for programming speech using this module. You may vary the pitch and slope and inflections. You may use allophones to create words, or you may have the computer speak words which you spell phonetically.

Telecommunications And Languages

Terminal. The Terminal Emulator II command module (or Terminal Emulator I, which does not have speech) allows you to use your TI-99/4A to act as a terminal either to another computer or to a large telecommunications service. You will also need the TI RS-232 Interface and a telephone modem.

The SOURCE is an on-line information service from Source Telecomputing Corporation. TEXNET is a special edition of The SOURCE especially for the Texas Instruments home computer.

Printer. You may use a number of different brands of printers with your microcomputer. To connect your TI-99/4A to a printer, you'll need the TI RS-232 Interface and a cable to go from the interface to the printer (the cable is usually sold with the printer).

RS-232. The RS-232 Interface has two ports so you may be connected to a modem and a printer at the same time. An instruction book comes with the RS-232 so you'll know how to operate the computer under different conditions.

Extended BASIC. TI Extended BASIC (XBASIC) is a programming language contained on a module. A manual (over 200 pages) and a programmer's reference card come with the module. No other peripherals are necessary to use XBASIC. If a program has been written in XBASIC, the XBASIC module must be inserted for the program to run. Some of the advantages of XBASIC are multi-statement lines, complex IF-THEN-ELSE logic, subroutine and MERGE capabilities, DISPLAY AT and PRINT USING, program security (SAVE protection), speech (with speech synthesizer), and moving sprites with greater graphics capabilities.

Logo. TI Logo is a fascinating programming language designed especially for young children. TI Logo is contained in a module, and the 32K memory expansion is required. Logo I can print using the TI thermal printer only. Logo II has music and also RS-232 capability so you can print listings on a regular printer.

Editor/Assembler. For machine language programmers, it requires the memory expansion, disk controller, and one disk drive.

USCD PASCAL. This language requires the memory expansion, P-code peripheral card, disk controller, and at least one disk drive.

Software

Software. I've mentioned software (programs) last, although it's probably the first extra purchase you will make for your computer. Software is what you need to use your computer. Software is available on command modules, cassettes, and diskettes, and in a variety of subjects. Scott, Foresman educational courseware is available for grade levels kindergarten through eighth grade, Texas Instruments has several educational modules, and other educational and publishing companies are also developing modules for all grade levels.

In addition, there are modules for all types of home use (budget, finances, decision making, record keeping) and, of course, games from chess to soccer, from *Hunt the Wumpus* to *TI Invaders*.

Cassette and diskette programs are available for many applications, including programs for two-year-olds learning colors to sophisticated business programs. When you purchase, a software's documentation should tell you what hardware is required. For example, much of the business software requires a printer and two disk drives (and thus the peripheral box, RS-232 Interface, and disk controller) plus perhaps the Extended BASIC module and/or the 32K memory expansion.

Current literature. Texas Instruments sends an informative newsletter to all owners (be sure to send in your registration card). Many user groups have formed which have their own newsletters and catalogs. Other magazines are available that support the TI-99/4A. Now **COMPUTE!** will offer you a monthly column dedicated to the TI-99/4A, and other articles and programs to help you enjoy your TI-99/4A home computer to the fullest. ©

Copy VIC Disk Files

Roger L. Smith, Mesa, AZ

The "Copy 2031 Files" program by G. H. Watson (**COMPUTE!**, August 1982) was greatly appreciated. It was suggested that the program might run on the VIC-20 if appropriate changes were made. The following program makes those changes.

The program will work with either VIC-1540 or the new VIC-1541 disk drives. However, your VIC *must* have a 3K memory expansion plugged in since the program uses memory beginning at the expansion start-of-BASIC address ($0400) to handle the file transfer.

As noted in the original article, the program will handle BASIC program files. To copy sequential files (data files, ASM/TED files), replace the appended P with an S by using POKE 7672,83. Program files use POKE 7672,80. If you want to transfer more than one file, you will have to type SYS 7354 each time.

```
500 FOR ADRES=7354TO7674:READ DATTA
    :POKE ADRES,DATTA:NEXT ADR
    ES
600 SYS7354
634 DATA 169,1,133,184,32,195
640 DATA 255,169,8,133,186,169
646 DATA 15,133,185,169,0,133
652 DATA 183,32,192,255,160,29
658 DATA 169,179,32,30,203,32
664 DATA 96,197,169,0,133,187
670 DATA 169,2,133,188,160,255
676 DATA 200,177,187,208,251,162
682 DATA 4,189,245,29,145,187
688 DATA 200,202,208,247,132,183
694 DATA 169,2,133,184,32,195
700 DATA 255,169,8,133,186,169
706 DATA 2,133,185,32,192,255
712 DATA 32,141,29,162,2,32
718 DATA 199,242,169,4,133,1
724 DATA 169,3,133,0,160,0
730 DATA 32,14,242,145,0,166
736 DATA 144,208,7,200,208,244
742 DATA 230,1,208,240,132,5
748 DATA 165,1,133,6,169,2
754 DATA 32,195,255,32,204,255
760 DATA 160,29,169,191,32,30
766 DATA 203,32,228,255,240,251
772 DATA 164,183,136,169,87,145
778 DATA 187,32,192,255,32,141
784 DATA 29,162,2,32,9,243
790 DATA 169,4,133,1,169,3
796 DATA 133,0,160,0,177,0
802 DATA 32,122,242,165,1,197
808 DATA 6,208,4,196,5,240
814 DATA 14,200,208,238,230,1
820 DATA 208,234,160,29,169,228
826 DATA 32,30,203,169,2,32
832 DATA 195,255,169,1,32,195
838 DATA 255,32,204,255,76,116
844 DATA 196,162,1,32,199,242
850 DATA 32,14,242,141,240,29
856 DATA 32,14,242,141,241,29
862 DATA 32,204,255,173,240,29
868 DATA 201,48,208,206,173,241
874 DATA 29,201,48,208,199,32
880 DATA 204,255,96,13,70,73
886 DATA 76,69,78,65,77,69
892 DATA 63,32,0,13,83,87
898 DATA 73,84,67,72,32,68
904 DATA 73,83,75,69,84,84
910 DATA 69,83,44,13,84,72
916 DATA 69,78,32,72,73,84
922 DATA 32,82,69,84,85,82
928 DATA 78,46,13,0,13,68
934 DATA 73,83,75,32,69,82
940 DATA 82,79,82,32,0,0
946 DATA 33,13,0,0,82,44
952 DATA 80,44,246,230,1,76
```
©

Keith Falkner

Tap Applesoft's Heartbeat

You can use machine language routines to enable Applesoft to read and rapidly process incoming data.

Imagine that your Apple is connected to some gizmo which feeds the Apple some data rapidly. The device could be, for example, a modem or some newfangled digital geiger counter monitoring an atomic reactor. In an example below, we will simulate this device with the game paddle buttons, or, if you have no paddles, with a mere piece of wire. The essential idea is that the attached device offers data to the Apple sporadically, and the data will be lost if it is not noticed and processed within a few milliseconds.

If you try to support this device with a program written in Applesoft BASIC, you will likely miss some of the data offered by the device, because Applesoft is rather slow. Assuming that such a problem does arise and must be solved, here's how.

Machine Language Patch Into CHRGET

Here is an intriguing exercise: type in and run the listing in Program 1. If you type it correctly, it will say "OK"; make sure you fix it if it says "OOPS." This program installs, but does not run, three tiny machine language routines. Now type CALL 909 and then run the program again. Inexplicably, it will make an irritating buzz for the 0.37 seconds it takes to run. Indeed, you can load and run almost any Applesoft program and listen to it run.

You may notice that difficult computations and lengthy array references are accompanied by buzzes, whereas fast-running code such as FOR/NEXT loops that do little more than count will produce brief musical tones. I do not suggest that this is a useful effect, but I hope it sparks your interest, for what is coming is a bit dull and difficult but results in a very powerful technique which you can harness to produce utterly amazing results at zero cost.

By the way, you can deactivate the noise-making routine and restore your Apple to normal by typing CALL 896. The DOS command FP is even more powerful; issue that if your Apple seems confused.

A Look Into CHRGET

Here is how the noise is caused. The Applesoft interpreter uses a tiny routine to fetch each byte of your program in turn as the program runs. The (valid) BASIC statement IF BAD THEN STOP is stored as six bytes, specifically the token for IF, the letters B, A, and D, and the tokens for THEN and STOP. The character-getting routine, which is known by the name CHRGET, will be invoked a total of seven times to execute all of this statement (the token for THEN is fetched twice, once to detect the end of the variable name BAD, and once to be executed).

Program 1 and the routine installed at location 909 introduce a detour into CHRGET so that the Apple's speaker is tweaked each time a character of the program is fetched. This of course makes the noise and accounts for the various buzzes and squeaks made by slow- and fast-running code. To see the actual machine language routine, enter the monitor via CALL -151 and enter 380L (number 380 followed by letter L) to see the routines at 896 ($0380), 909 ($038D), and 922 ($039A).

The CHRGET routine starts in location 177 ($00B1), and can be listed by B1L (letter B, digit 1, letter L). You can verify if you wish that CALL 909 installs a JMP instruction at location 186 ($00BA), and CALL 896 restores the CMP and BCS instructions which belong there. You can return from the monitor to Applesoft by typing CTRL-C and pressing RETURN.

Now let's put this technique to use. If you have game paddles, identify PDL (1) and skip the rest of this paragraph. To simulate the button on PDL(1), you will need a piece of slender wire at least two feet long. Solid wire works better than multi-strand. Strip about one-eighth inch from each end. You shuld now *turn off* the Apple and open the cover carefully. Locate the GAME I/O connector at coordinates J8 on the motherboard, and stick an end of the wire into hole number three, which is third from the front on the right side.

Do be careful with this, because disaster awaits you if you pick the wrong hole, or are careless with the other end of the wire. Now close the cover of the Apple, letting the free end of the wire hang down away from the computer. Reach under the front edge of the keyboard and you will find the heads of some bolts. You will be touching the free end of that wire to one of these to simulate a press of the button. If you choose, you can loosen one of these, attach another piece of wire, tighten the bolt, and attach the two loose ends of wire to any type of switch, but this is not essential. When these preparations are complete, turn the Apple on again.

Catching Every Count

Now type in Program 2 and run it. Please note the lengthy loop in lines 130-140. This takes over half a minute to run and obviously contains none of the PEEK statements necessary to test for a press of button number one. Those tests are done by the machine language routine patched into CHRGET, at locations 922 through 965. When the program is running, press the button (or touch the wire to the bolt) as fast as you can count, and you will find that the Apple catches every single one. Actually, when you try to touch the wire to the bolt once, you almost certainly cause it to bounce and touch the bolt more than once, so the count will be higher than you expect, and never lower.

In this example the switch was tested by a few instructions in machine language. This powerful technique is possible only in machine language. Perhaps it is possible to devise a routine that would permit a few lines of BASIC to be invoked by the routine which interrupts CHRGET, but what would be the point? Our objective here is to support a rapid-fire device, and any attempt to do this in BASIC will, it is assumed, lead to missed data. At least that is where this article started.

Using The Keyboard Buffer

A totally practical application of intercepting CHRGET is a keyboard buffer, except for one troublesome detail. From time to time, in any program which handles strings, Applesoft must pause to accomplish "garbage collection" – in other words, to make available again some memory which has been used for storage of strings which were later discarded. This process usually takes from one to thirty seconds, but in an artificial and extreme case it could take over an hour!

During "garbage collect," Applesoft is totally out of touch with all external events, so the keyboard buffering routine has no way to service the keyboard. Nonetheless, the routine is of genuine help when a speedy typist is using a slow data entry program. In fact, even a moderately slow hunt-and-peck typist like me can occasionally leave Applesoft behind. With the buffer running, I never lose a key.

There are two other limitations. During processing of the LIST command, Applesoft is not using CHRGET, so the buffering routine has no chance at the keyboard. Also, when DOS is active, all BASIC functions are inactive, so again the keyboard cannot be serviced.

Program 3 shows the complete keyboard buffer program. The program occupies the first 512 points of the BASIC program area, so it destroys any Applesoft program already present.

Briefly, here is how the program works. A preliminary test verifies that Applesoft is active, for this program is inapplicable to Integer BASIC. Next, the program sees if the beginning-of-BASIC pointer has been altered to $0A01 (from the usual $0801). If so, a warm start is done, retaining the current Applesoft program; if not, the pointer is so altered, and the new routine of Applesoft is called. Then the "patch" to CHRGET is made, as in Programs 1 and 2.

The next step is a connection to the keyboard-servicing routine at the "hook" known as KSW. Whenever such a connection is what you need, you must let DOS know your intentions, or it will patiently remove your connection and restore its own hook. This is very easy – just CALL 1002 (or JSR $3EA in machine language). The program ends by entering Applesoft at the warm-start entry $E003.

By this point, the program really has not done anything except insinuate itself into the system and protect itself from harm. The actual buffer is the 256-byte area from $0900 to $09FF (2304 to 2559), and two one-byte counters look after data in the buffer. The counter BIX points to the next place where a key can be stored, and the counter BOX points to the next byte to be sent to whoever asks for a key.

For example, if BIX contains $2E and BOX contains $28, the operator has keyed six bytes ahead, and they are stored in locations $0928 through $092D. If the operator now keys exactly 250 more bytes before the running program asks for any more, the keyed bytes will be stored in $092E through $09FF, then the buffer will "wrap around" and more keys will be stored in $0900 through $0926. By this time the value in BIX will be $27, one less than that in BOX. That's 249 in addition to the six already there, and now the buffer is full, so the buffering routine will sound the "bell" when it cannot store the last byte keyed.

At this point the operator must pause and wait for the program to catch up. I think this event is very unlikely.

Keys are detected and stored by the routine patched into CHRGET. A word of caution to anyone patching CHRGET: since BASIC uses this routine dozens or thousands of times a second, the patch must execute as fast as possible, else the program may be slowed to an unacceptable degree.

Does It Function?

When a key is wanted, the code at INLINK sees if one is in the buffer. If not, the standard ROM routine is called. If a key is available in the buffer, it is delivered, and the counter, BOX, is updated to account for the departed key. It is all very simple, mainly because of the eight-bit indexing automatically provided by the 6502's X-register. Indeed, if the buffer were any size but 256 bytes, the program would have been noticeably harder to write and debug.

OK, how do you key this program into your Apple? You could CALL -151 to get to the monitor, then type in all the hex stuff, 803:4C 09 08 4C 99 08, and so on. If you did the "homework" I assigned in last month's column, there is an easier way. Key in the pure Applesoft program in Program 4, then SAVE it, RUN it, and finally EXEC GEN KEYBUF. This final step will invoke the mini-assembler to build KEYBUF, save the result, and return control to the keyboard eventually. This process must destroy any Applesoft program in memory, so be sure you have saved Program 4 before typing the EXEC command!

To verify all this work, peer closely at the screen – the command JMP $083C should be in location 08A8. The acid test, of course, is "does it work?" Follow the instructions below to test your work, and when you actually make it work, you'll have a potent and versatile tool which makes your Apple a little bit better than it was before!

Homework Assignment. Boot your System Master and LOAD BRIAN'S THEME. That is the program which displays pretty moiré patterns in high resolution. Here is some code to add a fascinating effect! Type in the few lines in Program 5 and RUN the changed program. When the display starts acting oddly, play with the keyboard. The most recently pressed key controls the timing in a tiny machine language routine at location 600 ($258).

In my particular Apple, the keys W, K, 8, question mark, and especially CTRL-D, produce interesting effects. The machine language routine is completely relocatable, so it can be used without change in any place in memory where 26 bytes are free. So if you wish to use the routine in another program, change the variable ML to whatever suits you. The timing is so delicate that the effects change greatly when ML is just under a multiple of 256, so that a branch instruction crosses a page boundary. To stop this demonstration, you must press RESET, because the machine language routine treats CTRL-C as any other key.

Table: How to use the Keyboard Buffer

1. To load and initialize the routine,
   BRUN KEYBUF
2. Now use your Apple as usual, but be sure that you do not switch to Integer BASIC!
3. To suspend use of the buffer,
   CALL 2054
4. To resume use of the buffer,
   CALL 2051
5. To recover memory used by the buffer, after suspending it via CALL 2054,
   FP (or INT, if you choose)
6. To copy the routine from disk to disk,
   BLOAD KEYBUF
   Insert the disk to receive a copy.
   BSAVE KEYBUF,A$803,L$F8

Program 1.

```
10 REM   'TAP' DEMO 1
20 FOR I = 896 TO 935
30 READ X
40 Z = Z + X
50 POKE I,X
60 NEXT
70 IF Z <  > 5155 GOTO 90
80 PRINT "OK": END
90 PRINT "OOPS. Z=";Z: END
896 DATA 169,201,133,186,169,58
902 DATA 133,187,169,176,133,188
908 DATA 96, 169,76,133,186,169
914 DATA 154,133,187,169,3,133
920 DATA 188,96,141,48,192,201
926 DATA 58,176,3,76,190,0
932 DATA 76,200,0,0
```

Program 2.

```
10 REM   'TAP' DEMO 2
20 REM
30 FOR I = 896 TO 955
40 READ X
50 Z = Z + X
60 POKE I,X
70 NEXT
80 IF Z < > 7425 THEN PRINT "OOPS. Z=";Z: STOP
90 HOME : GR
100 PRINT "WHILE I SCRIBBLE AIMLESSLY,"
110 PRINT "PRESS BUTTON 1 SEVERAL TIMES."
120 POKE 24,0: POKE 25,0: POKE 26,0: CALL 909
130 FOR I = 1 TO 1000: COLOR= 16 * RND (I)
140 PLOT 40 * RND (I),40 * RND (I): NEXT
150 CALL 896:T =  PEEK (25) + 256 * PEEK (26)
160 TEXT : PRINT CHR$ (7): REM BELL!
170 HOME : PRINT "YOU PRESSED IT "; INT (T / 2
    );" TIMES."
```

```
896 DATA 169,201,133,186,169,58
902 DATA 133,187,169,176,133,188
908 DATA 96, 169,76,133,186,169
914 DATA 154,133,187,169,3,133
920 DATA 188,96,72,152,72,173
926 DATA 98,192,41,128,197,24
932 DATA 240,8,133,24,230,25
938 DATA 208,2,230,26,104,168
944 DATA 104,201,58,176,3,76
950 DATA 190,0,76,200,0,0
```

Program 3.

```
0000               ; THIS PROGRAM USES 512 BYTES FROM
0000               ; 2048 TO 2559 TO CONTAIN AND LOOK
0000               ; AFTER A 256-BYTE KEYBOARD BUFFER.
0000               ;
0000               ; 'BRUN KEYBUF' TO CREATE THE BUFFER.
0000               ; 'CALL 2054' TO DISABLE THE BUFFER.
0000               ; 'CALL 2051' TO RE-ENABLE THE BUFFER.
0000               ;
0000               ; HOW TO SAVE THE PROGRAM:
0000               ; BSAVE KEYBUF,A$803,L$F8
0000               ;
0000                      *=$803          ;START AT 2051.
0803               ;
0803               ; JUMP-TABLE OF ENTRY-POINTS:
0803               ;
0803 4C0908               JMP STARTS      ;ENABLE BUFFER
0806 4C9908               JMP CANCEL      ;DISABLE BUFFER
0809               ;
0809 AD00E0        STARTS LDA $E000       ;WHICH LANGUAGE?
080C C94C                 CMP #$4C        ;APPLESOFT?
080E D036                 BNE STEXIT      ;NO, SO QUIT!
0810               ;
0810 A90A                 LDA #>BASIC     ;-> NEW START
0812 A001                 LDY #1          ;OF BASIC (+1)
0814               ;
0814 C467                 CPY $67         ;WARM ENTRY TO ME?
0816 D004                 BNE STCOLD      ;NO
0818 C568                 CMP $68         ;WARM FOR SURE?
081A F00C                 BEQ STLINK      ;YES!
081C               ;
081C 8467          STCOLD STY $67         ;SET UP THE NEW
081E 8568                 STA $68         ;START-OF-BASIC
0820 A900                 LDA #0
0822 8D000A               STA BASIC       ;TRADITION
0825 204BD6               JSR $D64B       ;EXECUTE 'NEW'.
0828               ;
0828 A94C          STLINK LDA #$4C
082A 85BA                 STA $BA
082C A949                 LDA #<CHLINK    ;TIE IN TO
082E 85BB                 STA $BB         ;CHRGET.
0830 A908                 LDA #>CHLINK
0832 85BC                 STA $BC
0834               ;
0834 A97C                 LDA #<INLINK
0836 8538                 STA $38         ;TIE IN TO THE
0838 A908                 LDA #>INLINK    ;INPUT HOOK 'KSW'
083A 8539                 STA $39
083C               ;
083C ADEA03        STTIES LDA $3EA
083F C94C                 CMP #$4C        ;IS DOS PRESENT?
0841 D003                 BNE STEXIT      ;NO, NO DISK HERE!
0843 20EA03               JSR $3EA        ;TELL DOS ABOUT TIE-IN
0846 4C03E0        STEXIT JMP $E003       ;WARM START
0849               ; THIS ROUTINE IS ENTERED EVERY TIME
0849               ; APPLESOFT FETCHES A BYTE OF BASIC.
0849               ;
0849 2C00C0        CHLINK BIT $C000       ;KEY PRESSED?
084C 1026                 BPL CHCOLO      ;NO, NOT YET
084E 48                   PHA             ;SAVE BASIC BYTE
084F 8A                   TXA             ;SAVE X-REGISTER
0850 48                   PHA
0851 AEAC08               LDX BIX         ;GET INPUT POINTER
0854 E8                   INX             ;PREPARE TO STEP UP
0855 ECAD08               CPX BOX         ;BUT IS BUFFER FULL?
0858 D00A                 BNE CHSTOW      ;NO, GO & STASH
085A 98                   TYA             ;BUFFER FULL:
085B 48                   PHA             ;(BELL USES Y-REG)
085C 20E2FB               JSR $FBE2       ;RING THE BELL!
085F 68                   PLA
0860 A8                   TAY
0861 4C6E08               JMP CHRETR
0864               ;
0864 8EAC08        CHSTOW STX BIX         ;SAVE NEW POINTER
0867 CA                   DEX             ;-> PLACE FOR THE KEY
0868 AD00C0               LDA $C000       ;GET THE KEY
086B 9D0009               STA BUF,X       ;SAVE IN BUFFER
086E               ;
086E 8D10C0        CHRETR STA $C010       ;RESET KEYBOARD
0871 68                   PLA
0872 AA                   TAX             ;RECOVER X-REG
0873 68                   PLA             ;& BYTE OF BASIC
0874               ;
0874 C93A          CHCOLO CMP #$3A        ;(CHRGET REPLACEMENT)
0876 B003                 BCS CHBACK      ;"
0878 4CBE00               JMP $BE         ;"
087B 60            CHBACK RTS             ;"
087C               ;
087C               ; THIS ROUTINE IS USED WHENEVER A
087C               ; KEY IS NEEDED FROM THE KEYBOARD.
087C               ;
087C 8EAB08        INLINK STX SAVX        ;SAVE IT
087F AEAD08               LDX BOX         ;GET OUTPUT POINTER
0882 ECAC08               CPX BIX         ;ANYTHING IN BUFFER?
0885 D006                 BNE INSEND      ;YES, GO SEND IT!
0887 AEAB08               LDX SAVX        ;NO, RESTORE X-REG
088A 4C1BFD               JMP $FD1B       ;NORMAL KEY HANDLER
088D               ;
088D 9128          INSEND STA ($28),Y     ;STOP FLASHING
088F BD0009               LDA BUF,X       ;GET KEY FROM BUFFER
0892 EEAD08               INC BOX         ;UPDATE POINTER
0895 AEAB08               LDX SAVX        ;RESTORE X-REG
0898 60                   RTS
0899               ; DISABLE THE KEYBOARD BUFFER
0899               ;
0899 A9C9          CANCEL LDA #$C9
089B 85BA                 STA $BA
089D A93A                 LDA #$3A        ;RESTORE CHRGET
089F 85BB                 STA $BB         ;ORIGINAL STUFF
08A1 A9B0                 LDA #$B0
08A3 85BC                 STA $BC
08A5               ;
08A5 2089FE               JSR $FE89       ;EXECUTE "IN#0".
08A8 4C3C08               JMP STTIES
08AB               ;
08AB 00            SAVX   .BYT 0          ;SAVE AREA FOR X-REG
08AC 00            BIX    .BYT 0          ;-> PLACE FOR NEXT BYTE
08AD 00            BOX    .BYT 0          ;-> NEXT ONE TO DELIVER
08AE               ;
08AE               ; (BIX=BOX) MEANS BUF IS EMPTY
08AE               ; (BIX+1=BOX) MEANS IT'S FULL!
08AE               ;
08AE               ; THE ABOVE MUST END BY $8FF
08AE               ; OR IT WILL BE OVERWRITTEN!
08AE               ;
08AE               BUF=$900               ;BUFFER IS $900-$9FF
08AE               BASIC=$A00             ;NEW START-OF-BASIC
08AE               ;
08AE                      .END
```

SYMBOL TABLE

SYMBOL	VALUE				
BASIC	0A00	BIX	08AC	BOX	08AD
BUF	0900	CANCEL	0899	CHBACK	087B
CHCOLO	0874	CHLINK	0849	CHRETR	086E
CHSTOW	0864	INLINK	087C	INSEND	088D
SAVX	08AB	STARTS	0809	STCOLD	081C
STEXIT	0846	STLINK	0828	STTIES	083C

Program 4.

```
100 REM MAKE "GEN KEYBUF"
110 D$ = CHR$ (4)
120 F$ = "GEN KEYBUF"
130 PRINT D$"OPEN "F$
140 PRINT D$"WRITE"F$
150 PRINT "FP"
160 PRINT "MON I"
170 PRINT "BRUN MINI-ASSM"
180 PRINT "803:";: REM NOTICE SEMICOLON
190 READ Z$
200 IF Z$ = "END" GOTO 230
210 PRINT " "Z$
220 GOTO 190
230 PRINT "FP"
240 PRINT "BSAVE KEYBUF,A$803,L$F8"
250 PRINT D$"CLOSE"
260 END
270 DATA JMP809,JMP899,LDAE000
280 DATA CMP#4C,BNE846,LDA#A
290 DATA LDY#1,CPY67,BNE81C
300 DATA CMP68,BEQ828,STY67
310 DATA STA68,LDA#0,STAA00
320 DATA JSRD64B,LDA#4C,STABA
330 DATA LDA#49,STABB,LDA#8
340 DATA STABC,LDA#7C,STA38
350 DATA LDA#8,STA39,LDA3EA
360 DATA CMP#4C,BNE846,JSR3EA
370 DATA JMPE003,BITC000,BPL874
380 DATA PHA,TXA,PHA
390 DATA LDX8AC,INX,CPX8AD
```

```
400 DATA BNE864,TYA,PHA
410 DATA JSRFBE2,PLA,TAY
420 DATA JMP86E,STX8AC,DEX
430 DATA LDAC000,"STA900,X",STAC010
440 DATA PLA,TAX,PLA
450 DATA CMP#3A,BCS87B,JMPBE
460 DATA RTS,STX8AB,LDX8AD
470 DATA CPX8AC,BNE88D,LDX8AB
480 DATA JMPFD1B,"STA(28),Y","LDA900,X"
490 DATA INC8AD,LDX8AB,RTS
500 DATA LDA#C9,STABA,LDA#3A
510 DATA STABB,LDA#B0,STABC
520 DATA JSRFE89,JMP83C,BRK
530 DATA BRK,BRK,END
```

If you have Integer BASIC in ROM or in a Language Card, substitute:

```
150 PRINT "INT"
170 PRINT "CALL -2667" :REM MINI-
     ASSM
```

Program 5.

```
460 ML = 600 : FOR I = ML TO ML+25
470 READ X: POKE I,X: NEXT
480 LIST  (... OR PRINT SOME STUFF)
490 CALL ML
500 DATA 173,80,192,173,0,192
510 DATA 41,127,170,202,208,253
520 DATA 173,81,192,173,0,192
530 DATA 41,127,170,202,208,253
540 DATA 240,230
```

©

Atari Lister

LeRoy J. Baxter, Milwaukie, OR

Debugging a long program listing can be tedious. Most of us have typed in a long program and then had to hunt for errors when it wouldn't RUN. This utility routine can make the job a little easier.

Make a copy of this program and LIST it to tape or disk. When you need it, load it with the ENTER command (the line numbers shouldn't conflict). Then type GOTO 32700. A prompt will appear. Press RETURN, and the first set of six program lines will be LISTed to the screen, regardless of their line numbers, with spaces between the lines. Then with a touch of RETURN, it LISTs the next set of six lines.

Enter "EDIT," and the program goes to the Editing Subroutine. It asks for the line number of the offending line, then LISTs the line and prints the command "CONT" below it. You can then edit the line using the screen editor keys.

When you press RETURN, the line will be entered into the program. You can enter or delete complete lines using standard techniques. Simply move the cursor up and enter your new line between the LISTed line and CONT. When you are done, enter "ERASE," and the utility program will erase itself.

```
32700 DIM A$(5):T=0
32705 Z=0:INPUT A$:ON (A$="EDIT")+(A$
      ="ERASE")*2 GOTO 32730,32745
32710 ? CHR$(125):ADDR=PEEK(136)+PEEK
      (137)*256:FOR X=0 TO T:ADDR=ADD
      R+PEEK(ADDR+2)*(T>0):NEXT X
32715 LINENO=PEEK(ADDR)+PEEK(ADDR+1)*
      256:Z=Z+1:IF LINENO>=32700 THEN
       ? "* END OF LISTING *":GOTO 32
      710
32720 LIST LINENO:T=T+1:ADDR=ADDR+PEE
      K(ADDR+2):IF Z<6 THEN 32715
32725 GOTO 32710
32730 ? "WHAT LINE #";:INPUT X
32735 ? CHR$(125):POSITION 2,4:LIST X
      :? :? :? :? "CONT":INPUT A$:POS
      ITION 2,0:POKE 842,13:STOP
32740 POKE 842,12:T=T-6:GOTO 32705
32745 ? CHR$(125):POSITION 2,4:FOR X=
      32700 TO 32750 STEP 5:? X:NEXT
      X:? "POKE 842,12"
32750 POSITION 2,0:POKE 842,13:STOP
```

©

Perfect Commodore INPUTs

A one-line cure for accidental program exits during keyboard input.

Blaine D. Standage, Orange, CA

Solve the problem of inputting from the VIC, PET/CBM, 64 keyboard once and for all with one simple line of BASIC code. Why clutter your program with complex subroutines when there is a better way?

I waited a long time for someone to write this article, but no one did. Meanwhile, I kept seeing involved subroutines offered as solutions to the "input problem."

The "problem" is that when you hit the RETURN key in response to an INPUT statement without first giving a Commodore computer some data, it promptly dumps you out of the program – often a very undesirable result. The same thing happens if you accidentally hit the STOP key while the computer is in a GET loop waiting for data. (10 GET A$: IF A$ = "" THEN 10). Since most of the proposed solutions use GET loops, it seems that they only move the problem by a keywidth rather than solve it.

Following the KISS (Keep It Simple, Sam) method, let's define our ideal objectives and try to take a completely fresh approach to the problem.

Avoid Subroutines

The highest probability of an operator disrupting a program occurs when he is responding to an INPUT or a GET command. At that time we need to simplify his task by preventing him from accidentally halting the computer.

To keep it simple for the programmer, we need to avoid subroutines, particularly large ones or those written in machine language which may require some form of special handling.

As to the fresh approach, remember that when a thing doesn't work the way you want it to, one alternative is to simply not use it! Oddly enough, that is exactly the way out of this problem. Don't use INPUT or GET.

When we remove INPUT and GET from the instruction list, we are left with only two commands which might take their place, INPUT# and GET#. But aren't they for use with external devices like tape or disk drives? Not exactly. They are the complete form of the commands for inputting. INPUT and GET are just simplified forms which allow easy access to the keyboard.

Using GET#

The results of investigating GET# showed that, overall, it operates very much like GET. We can't completely solve our problems with it because the program can always be halted with the STOP key.

The most obvious difference occurs when we press a non-numeric key while trying to get a numeric value (i.e., GET#1 A). The computer responds with:

```
?SYNTAX ERROR
READY.
```

"SYNTAX ERROR" doesn't seem a reasonable response, and I haven't found out why it happens. Nor can I see any way to take advantage of it, but maybe someone else can.

Perhaps the biggest potential advantage to GET# becomes evident by recalling that we commonly open output files to both the printer and the screen. We then direct the PRINT# output by selecting the appropriate file with a variable value. Similarly, the keyboard could be included in a selectable group of input devices through the use of GET# or INPUT# with a variable defining the file number.

How About INPUT# ?

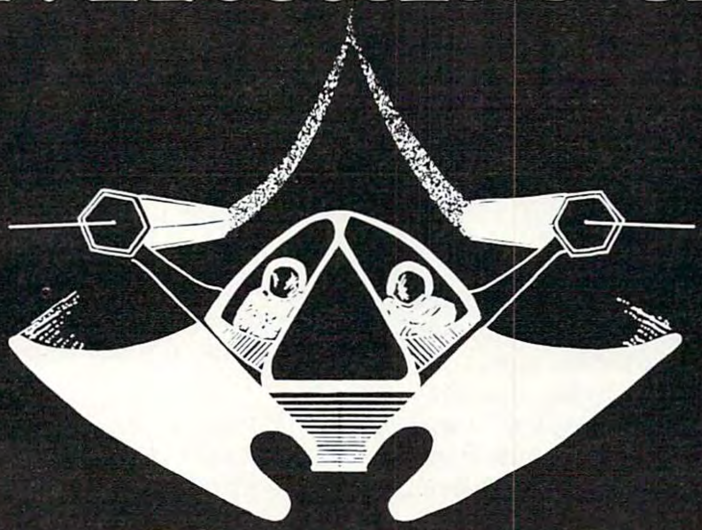
Using INPUT# proved to be the answer, but a lot was learned along the way.

When you enter and run the simple program:

```
10 OPEN 1,0
20 INPUT#1,A$: PRINT A$: GOTO 20
```

all the inputs are echoed back, and there seems to be no way out of the program. RETURNs are ignored, and the STOP key has no effect. Most of my test group (victims, to hear them tell it) decided there was no way to regain control short of cycling computer power. Only the most determined ones discovered that the SHIFTed RUN/STOP would cause a break. (On VIC, this would be Commodore Key/RUN\STOP.) This combination is so unlikely in an input situation that we can almost disregard it as an accidental response.

GAME PROGRAM DEVELOPMENT KIT



for the

COMMODORE VIC - 20

VIC - 20 is a registered trademark of Commodore Business Machines, Inc.

SIX TOOLS TO HELP YOU WRITE YOUR OWN FAST ACTION ARCADE-STYLE GAMES

DECODER — Decodes programs written in machine language (like game cartridges, utility cartridges, and even the computer's own internal operating programs). Produces a program in an English-like language (Assembler) which can be studied to figure out how they did it. The programs created with the decoder can be customized with the **EDITOR** AND INCORPORATED INTO YOUR OWN NEW GAME PROGRAM. The **ASSEMBLER** turns your programs created with the Decoder and the Editor back into machine language and puts them out to tape or disk so the **LOADER** can load them into the computer's memory to be tested and RUN. The **MONITOR** assists you in debugging your new game program by allowing you to run it a step at a time and making modifications if you need to. The **INSTRUCTION GUIDE** is written so that even a beginner can learn the skills needed to become a pro!!!

$49.95 plus $2.00 p&h buys the kit that could make you rich. Why wait?

Send check, M.O., VISA/MC ($2.00 s.c., please include expiration date), or specify COD (add $3.00) to:

French Silk smooth ware

P.O. Box 207, Cannon Falls, MN 55009

507-263-4821

When I first performed this experiment, my impulse was to consider the problem solved and go on with my programming. What a variety of interesting possibilities I would have missed!

Go to your computer and follow this easy exploration of INPUT#. It will be worth your time.

First, enter the following test program. Use the line numbers as shown because we will be adding to the program.

```
10 OPEN 1,0
20 :
30 INPUT#1,A
70 PRINT A
80 FOR J=1 TO 1500: NEXT J
90 GOTO 20
```

Line 80 has no direct bearing on the problem. It gives you a way to halt the program by providing time to press the STOP key after an input.

When the program is run, notice that the customary question mark is not printed. If we want the question mark we must include it in the input prompting. But we are also free to leave it out for inputs where it is not appropriate.

Next, enter a few numbers and notice they are printed back on the same line as the input even though we made no special provision for that. It seems we will have to print a "return" after inputting to get back to the left margin.

Now press RETURN without any data, and the computer responds as though you had entered a zero. Here is a feature we have wished for. But don't get too excited yet, because when we enter non-numeric characters the computer responds:

?FILE DATA ERROR IN 30
READY.

and we see that we can't solve our problem with a numeric variable input. Why a "file data error"? Remember, the computer thinks it is getting data file inputs from an external device.

Now let's make some changes to our test program to correct the defects we have seen. Unlike INPUT, the INPUT# command has no provisions for built-in prompting, so we must provide it in a separate PRINT command (line 20).

```
20 PRINT "ENTER DATA ? "
30 INPUT#1,A$: PRINT
70 PRINT A$
```

Running this now reveals that we won't stop the program during the input process either with a bad input or with the STOP key. Also notice that a simple RETURN is interpreted as a null (nothing there) string. We can really make use of this feature. (You can prove that the string is really null by manually creating a field of characters, a line of reversed spaces, for example, where the printing will be done. Observe that they are not altered during the execution of line 70.)

Several Handy Features

Of course, we neglected the semicolon at the end of line 20 to force the input to follow its prompt, so let's put it in and re-test.

```
20 PRINT "ENTER DATA ? ";
```

Careful now – something important has changed. A simple RETURN is no longer treated as a null string. It is just ignored. Another feature we can use; no more tests for null inputs – they simply can't happen with this program structure. And so we have RETURN = "ignored" or RETURN = "null" by including or deleting a single semicolon after the prompting message. Very handy.

Let's add lines 40-60 and change 70 to complete our test program for INPUT#. Now it will accept only numeric inputs. The full test program is:

```
10 OPEN 1,0
20 PRINT "ENTER DATA ? ";
30 INPUT#1,A$: PRINT
40 IF A$="0" THEN 60
50 IF VAL(A$)=0 THEN 20
60 B=VAL(A$)
70 PRINT A$;B
80 FOR J=1 TO 1500: NEXT J
90 GOTO 20
```

This structure rejects non-numeric inputs and (because of the semicolon in line 20) will not accept null inputs.

If you want a simple RETURN to be accepted as a zero, delete the semicolon in line 20 and change line 40 to read:

```
40 IF A$="0" OR A$="" THEN 60
```

So we have solved our problem quite nicely for inputs and come up with several very useful options in the process.

Let's make a general observation on the lack of built-in prompting with INPUT#. Neither the double question mark requesting additional inputs nor the "extra ignored" warning is printed. This suggests that you should keep the structure of your input commands as simple as possible, because the operator will not get the usual warnings when he enters incorrect data patterns.

At last, as promised, here is the one-line, no-accidental-exit, solution to the long-standing input problem.

```
10 OPEN1,0:PRINT"PROMPT ? ";:INPUT
   #1,A$:PRINT:CLOSE 1
```

Obviously, the OPEN and CLOSE commands could span a group of inputs or even the entire program. You can safely leave the "file" open while doing other things.

As a bonus, you can accept null inputs by deleting the semicolon in the first print command. This allows you to convert the nulls to zeros for numeric applications.

Finally, I think you will find it interesting to watch your experienced friends try to terminate a program which uses the INPUT# technique. They tend to get very frustrated, which suggests that maybe you shouldn't build escape-proof programs unless they are really needed. ©

Atari Autonumber

Barry M. Bernstein, Willowdale, Ontario

This provides quite a convenience when using Atari BASIC. AUTONUM adds automatic line numbering to Atari BASIC. It is used in much the same way as the Atari ASSEMBLER/EDITOR NUM function is used.

The program creates a cassette boot program to be loaded in when you turn the computer on with the BASIC cartridge in place. Once in, it can be called at any time and can quite easily be disengaged or reactivated.

Type in the BASIC program, being especially careful to get the DATA statements correct, and then execute the following statement in direct mode:

A=USR(12288)

You will hear two beeps, signalling you to press PLAY and RECORD on the 410 (with a tape in place) and then pressing the RETURN key. You have just made a boot tape.

To use BASIC AUTONUM, load the boot tape in the 410 tape player, make sure the BASIC cartridge is in place, press the START console switch and turn the computer on. When it beeps, press PLAY on the 410 and then RETURN. The AUTONUM program is now in memory. To activate it execute the following in direct mode:

A=USR(1550,a,b)

where a is the line number to begin at and b is the step size. If b is omitted then a is the step size and it will begin where it left off. If both a and b are omitted then it will begin numbering where it left off (ten to begin with) and increment by tens. You may have to press RETURN twice to activate AUTONUM. To disengage the automatic numbering simply press RETURN twice in a row. It is reactivated by repeating the above procedure.

Though BASIC AUTONUM may take up to half an hour to type in, you will find that it is well worth the effort for the great convenience that it offers, especially when typing in large programs.

```
10 DIM A$(100),B$(2),H$(23):H$="{,}
   {A}{B}{C}{D}{E}{F}{G}{H}{I}!!!!!!!
   {J}{K}{L}{M}{N}{O}":REM ALL CHARAC
   TERS IN BRACKETS ARE CONTROL CHARA
   CTERS
20 MEM=1536:M=-1
30 READ A$
40 FOR I=0 TO 49:B$=A$(I*2+1,I*2+2)
50 IF B$="YY" THEN RESTORE 700:MEM=12
   288:M=-1:GOTO 30
60 IF B$="ZZ" THEN ? "ALL DONE":END
70 N=0:FOR J=1 TO 2:N=N*16+ASC(H$(ASC
   (B$(J))-47)):NEXT J:M=M+1:POKE MEM
   +M,N
80 NEXT I
90 GOTO 30
100 DATA 000200060D06A93C8D02D3186060
    68C900D00BA90085CBA91085CC4C4F068
    5CFC901F00B20D306A5D585CDA5D485CE
    20D306
200 DATA A5D585CBA5D485CCA5CFC902D00F
    F838A5CEE5CC85CEA5CDE5CB85CDD8A98
    08510A9628D0802A9068D0902A9C08510
    60A5CF
300 DATA C9FFF011AD09D2C90CF0034CBEFF
    A9FF85CF4CBEFFAD09D2C90CD015A9808
    510A9BE8D0802A9FF8D0902A9C085104C
    BEFFA9
400 DATA 0085CFF818A5CE65CC85CEA5CD65
    CB85CDD820BD06A5CD20CA06A5CE20BD0
    6A5CE20CA064CBEFF29F04A4A4A4A1869
    3020A4
500 DATA F660290F18693020A4F6606885CC
    6885CB6885D56885D420AAD9A5D4C941F
    00BA5D585D4A90085D54CF706A5D685D4
    A5CB48
600 DATA A5CC4860YY
700 DATA A210A9039D4203A9089D4A03A908
    9D4B03A94A9D4403A9309D4503205 6E43
    028A90B9D4203A9009D4403A9069D4503
    A9009D
800 DATA 4803A9019D49032056E4300AA90C
    9D42032056E43000686043 3A9BZZ
```

©

NEW FOR ATARI

***********NECESSITIES***********

DISK COMMANDER II - Just save this program on your BASIC disks and it will autoboot and automatically list all programs from the disk into your screen. Simply run any **BASIC** or **Machine Language** program by typing a single number.
Requires 16K, Disk Only 29.95

BASIC COMMANDER - An all machine language program which occupies only 4K of RAM and is co-resident with your BASIC program. It is an indispensable tool for every ATARI BASIC programmer. **Basic Commander** allows single key access to DOS functions, BASIC file manipulation commands such as LOAD, LIST, ENTER, RUN, SAVE and more! Never need to access DOS again. **RENUMBER** allows you to renumber all Basic lines and all references instantaniously! In addition, **BLOCKDELETE** allows deletion of *any range* of lines, eliminating computer lock-up. **AUTONUMBER** automatically provides line numbers for your BASIC program, increasing program entry speed from 25 to 75%. If we've omitted your favorite commands, **Basic Commander** even provides **3 PROGRAMMABLE KEYS!**

THE MOST POWERFUL PROGRAMMING AID AVAILABLE FOR THE ATARI BASIC PROGRAMMER
Requires 16K, Disk Only $34.95

RAM TEST II - The fastest and most thorough memory test available for the ATARI has now been further improved! Tests not only all locations, but also tests the memory addressing system. This all machine language program takes 4 min. to test 48K. It's the only program that tests the cartridge area of RAM. Good for new 400/800 computer owners, for testing new RAM boards and for use in computer stores to test for any bad memory locations. Bad memory locations are pinpointed so repair is as simple as replacing a chip!
Requires 8K, Disk or Cassette $29.95

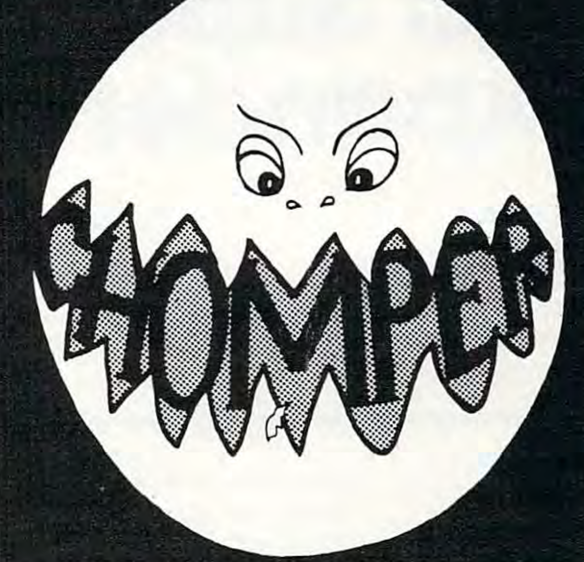
**********BUSINESS/HOME***********

MAIL LIST - Extremely fast BASIC and machine language program. Each data disk holds over 500 files. Sort on any of 6 fields at machine language speed or search on any fragment of a field! Use any size lables or envelopes.
Requires 40K, Disk Only $39.95

************TUTORIALS************

ASTEROID MINERS - This 50 page book and program provide for a unique intermediate-to-advanced tutorial. A 32K BASIC game utilizing over 25 players in player-missile graphics, machine language subroutines, a redefined character set, multiprocessing utilizing the vertical blank interrupt interval, and much more! The 50 page book included with the program documents each part of the entire program and contains the fully documented source code for both the BASIC and assembly language parts of the program. Use these routines in your own programs. These examples make it easy!
Requires 32K, Disk or Cassette $34.95

ATARI is a registered trademark of ATARI, INC. N.J. Residents add 5% sales tax.



ALL MACHINE LANGUAGE ARCADE GAME WITH INTELLIGENT MONSTERS!

This arcade style game is sure to become an ATARI classic. Chomper requires 16K RAM, 1 joystick and nerves of steel.
Available on Disk or Cassette $29.95

Available at your favorite computer store or
Send a check or money order directly to:

MMG MICRO SOFTWARE
P.O. BOX 131 • MARLBORO, NJ 07746
or call **(201) 431-3472**
for MasterCard, Visa or COD deliveries

You've invested a lot of time and money into your computer . . .
It's time that investment paid off!

THE COLOR ACCOUNTANT

The Programmer's Institute introduces **THE COLOR ACCOUNTANT**, the only complete personal financial package specifically designed for the Atari 400/800 and VIC-20 computers. This unique package includes:

1. **Complete Checkbook Maintenance**
2. **Chart of Accounts Maintenance**
3. **Income/Expense Statement**
4. **Net Worth Statement**
5. **Payments/Appointments Calendar**
6. **Color Graph Design Package (graphs any files)**
7. **Check Search**
8. **Home Budget Analysis**
9. **Decision Maker**
10. **Mailing List**

After the initial setup, **THE COLOR ACCOUNTANT** requires less than an hour of data input each month.

The checkbook maintenance program is the key to the entire package. Once your checkbook is balanced, the checkbook summary file will automatically update the home budget analysis, net worth, and income/expense statements. You can then graph any file, record bills and appointments, make decisions, print a mailing list, and analyze various accounts.

All programs are menu-driven and allow add/change/delete. Files and statements can be listed to screen or printer, and saved to casette or diskette. **THE COLOR ACCOUNTANT** also comes with 60 pages of documentation that leads you step-by-step through the entire package. The Atari 400/800 requires 24K cassette and 32K diskette; the VIC-20 requires 13K. **($74.95 cassette, $79.95 diskette).**

Add $3 for postage and handling.

THE TAX HANDLER

The perfect supplement to **THE COLOR ACCOUNTANT**, *The Tax Handler* includes:

1. **Complete From 1040**
2. **Schedule A (Itemized Deductions)**
3. **Schedule G (Income Averaging)**
4. **Schedule B**

This year let *The Tax Handler* prepare *your* taxes **($34.95 cassette, $39.95 diskette).**

See your local dealer or order direct:

THE PROGRAMMER'S INSTITUTE
a division of **FUTUREHOUSE**
P.O. BOX 3191, DEPT. C
CHAPEL HILL, NC 27514

1-800-334-SOFT
Mon-Fri 10-6; Sat 11-3

VIC Super Expander Graphics

Tim Parker, Kanata, Ontario

Want to see some stunning graphics on your VIC? Type in these short programs and you might be surprised to see what's possible with the Super Expander cartridge.

The VIC-1211A Super Expander is a plug-in cartridge for the VIC-20 that provides several extra features to the graphics and sound abilities. It also adds an extra three kilobytes of memory, giving a power-up of 6519 bytes free. (The missing bytes are used by the expander.)

Program 1 is a short routine that draws a grid on the screen, then selectively erases parts. This is done by drawing vertical and horizontal lines in a character color, then redrawing at a random interval with the screen's color. When RUN for several cycles, the patterns produced can be quite complex. Changing the color of the character in line 20 and the STEP interval in lines 100, 200, 300 and 400 can alter the complexity and appearance.

A variation on this program is to draw the lines on the graphics display diagonally, as Program 2 does. Here, lines 100-220 draw a circular pattern, skipping dots at intervals set by the STEP command. Then, lines 300-420 redraw at intervals in the screen color. The effects are produced as a consequence of the 1024x1024 graphics screen being shortened to 160x160. As the coordinates are altered, some dots will lie on either side of the line. As a result, some dots that are on are turned off, and vice versa.

A long routine to accomplish the above could be arranged using the RDOT(x,y) command, to see if a dot is on, then reverse it. Needless to say, this is an extremely time-consuming task, even in machine language.

Program 3 provides a pattern familiar to most people, although here it is generated in four corners. To see the pattern by itself, leave out lines 110-130. The simple routine here can be enhanced by adding circles concentrically in the center, or by repeating sections in the screen color, as above.

Program 4 draws rectangles on the screen concentrically and is then repeated to color in some areas. Again, when this is elaborated, it can have the effect of a moiré pattern, almost achieving movement of its own.

An alternate method of obtaining the concentric rectangles of Program 4 requires drawing squares with multiple TO's in the DRAW statement (Program 5). Repeating the pattern without a screen-clear command (SCNCLR) produces overlapping bands in the pattern. The pattern can be inverted (i.e., have the rectangles drawn from the outside in) by rewriting lines 100-140 to step down, instead of up. Naturally, concentric circles can be done the same way, by changing line 120 to read:

```
120 CIRCLE1,511,511,X,X
```

This actually produces ellipses, as the axes are not of equal length. This can be changed to produce true circles by adding a constant parameter to the X-axis value.

These programs are by no means as sophisticated as can be achieved with the Super Expander, but they do fill the need for a basic subroutine library on which to base future graphics displays. Combining these with PAINT commands can produce some interesting effects. The Super Expander cartridge's graphics abilities are limited only by the resolution of the graphics screen.

Possible future work for examination of the commands available includes drawing Archimedes' spiral, a herringbone-grid of diagonals, and changing to multicolor graphics to build up a quilt-like display.

Program 1.

```
10 GRAPHIC 2
20 REGION 5
50 DEFFNA(X)=INT(RND(1)*X)+1
100 FOR X=1 TO 1023 STEP FNA(40)+10
110 DRAW1,X,0TOX,1023
120 NEXT
200 FOR Y=1 TO 1023 STEP FNA(40)+10
210 DRAW1,0,YTO1023,Y
220 NEXT
300 FOR X=1 TO 1023 STEP FNA(40)+20
310 DRAW0,X,0TOX,1023
320 NEXT
400 FOR Y=1 TO 0123 STEP FNA(40)+20
410 DRAW0,0,YTO1023,Y
420 NEXT
500 GOTO 100
```