PEEK (65)

The Unofficial OSI Users Journal

1819 Bay Ridge Ave., Suite 220
Annapolis, MD 21403

Editor: Al Peabody
Tech Editor: Dick McGuire
Contributing Editor: Corky Kirk
OSIO Editor: Wallace Kendall

Volume I, No. 4
May 15, 1980

#5

Column One

The continued growth and health of PEEK(65) are very exciting to all of us, and bode well for OSI users in general. There are now several OSI users journals or newsletters published around the country, and the exchange of information is very helpful to me as a user.

With growth have come a couple of problems. PEEK(65) is no longer just me and Dick and a couple of kids helping us put it together on a weekend; we are getting unwieldy enough that we will have to start acting more like a real magazine (ugh!) but I hope this will be more than compensated for by the improvement in quality of information we can share and the numbers of people we can share it with.

A couple of policy statements are in order:

> We ordinarily do not publish the full address of letter writers, specifically because we do not want you to write directly to John Doe to solve his problem with his C2. We want you to write to PEEK(65) so that we can share the solution with everybody who has a C2. As usual, there are exceptions, as when you offer something free to everyone who will send a SASE, etc.
>
> Speaking of free stuff and for sale stuff. We will gladly publish, for free, any offering of anything helpful to other OSI users which is also for free. However, if you want to make a few bucks by selling something, help us pay for the cost of printing your offer by making it a classified ad. Deadline for material to be published in the issue which is sent out about the 15th of each month is the first of that same month. Price: $.10/word.

BACK ISSUES are available. Naturally, since they require special handling, hand addressing, first class postage, etc., they are a bit more expensive than $8/12. We started publishing in January, 1980, and the January, February, March and April issues are all available at $1.50 each.

THAT'S JUST THE WAY THINGS ARE, BUNKIE--

By Wallace Kendall, 9002 Dunloggin Rd., Ellicott City, MD 21043

Your program wouldn't run?

That's what was bugging you, Bunkie?

That's why you growled at your wife and yelled at the kids and kicked the cat and slammed the door?

But now it's running okay, so everything's rosy and life is beautiful? That's the way it is, Bunkie?

I hate to say this, Bunkie, but grab a shovel and crawl down here in the trenches with the rest of us. The war isn't over yet.

About those error messages--your computer could find the easy ones and help you fix them. Who's going to help you find the errors the computer can't see? Oh, yes indeed--some are probably there, and if you run into them (you might not), your computer can turn out garbage at 96000 baud. Your program may run for some kinds of data but not for others. Or it may run fine for 100 items but not for 101. Or for positive numbers but not for a stray negative or a zero or a nonprinting character. Or--Oh, well, the list goes on and on.

And Bunkie, when you catch on and list the program to try to find out what went wrong, are you going to remember what the program does and how it does it? Don't be so sure, Bunkie, because very, very few programmers do.

If you want to hear what the author of a university textbook in computer science says, this is from C. William Gear, "Introduction to Computers, Structured Programming, and Applications": "In fact, every large program in existence is almost certainly in error, but if there is a low probability of the occurence of data that causes the error to appear, it is unlikely that the error will ever be discovered."

Dr. Joseph Weizenbaum of MIT, speaking at a Johns Hopkins University seminar on November 1, 1979, shocked at least some of his audience of future professionals when he said (according to our shorthand, which is pretty good but not perfect), "In large corporations and government offices there are today hundreds of long, useful computer programs which nobody understands. These programs are used to process large quantities of valuable and important data, but they were written years ago, and were not well documented. They still run, and they are still in daily use--but nobody can understand them, and nobody even tries to update them."

So don't feel all alone, Bunkie. You may or may not be in good company, but at least you've got company. And if you want to get up in the front rank, remember two ideas: program documentation, which means "Put enough cogent and precise comments in this to make it possible for an utter idiot to understand it by looking at it." And program verification, which means "Fix your program so it won't print beautiful but meaningless garbage even if that utter idiot tries to make it use outlandish data in stupid ways under impossible conditions."

Do it, Bunkie, because the one who tries to use one of your programs might some day be me. Or it might be you.

*** CALL FOR ARTICLES ***

PEEK(65) is the OSI Users Journal. In past issues, we have had several articles donated by readers -- usually the best pieces in each issue. This issue mentions several projects under way around the country which may result in papers eminently qualified for publication in this or any other journal. We encourage you to submit progress reports, final reports, or even frustrated notes on any such projects for publication. We cannot offer money for articles, but we do offer a space for you to sound off, an opportunity to help others, and maybe just a little glory. A couple of requirements will help us out:

NEVER send your only copy of anything in the mail. We may neither of us ever see it again.

I am our art department, and I ain't much, so if you need to include a drawing or other art, be sure it is camera-ready, not an offhand sketch. Also, don't use blue ink, since the offset camera is blind to blue.

Really folks, we don't want to be picky, and we certainly do want to publish what you have written!

BASIC OPTIMIZATION

The OSI 8-K Basic is as fast as they come, but you can get a whole lot more from the machine by using a few 'new' ideas. This month we look at overall subroutine structure. Try the following:

1. Load a moderate sized program, 150-200 lines.
2. Insert new lines as follows (we will not SAVE this)

```
    1 GOTO 60000
    2 K=K+1 : RETURN
60000 FOR I=1 TO 1500 : GOSUB 2 :
NEXT I : END
```

Now RUN this program. It should take about 10 seconds. Next, enter the 'patch' as follows:

```
    1 FOR I=1 TO 1500 : GOSUB 60000 :
NEXTI:END
60000 K=K+1: RETURN
```

Behold! The "same" program takes about 30 seconds. The reason is that Basic starts looking for line numbers at the beginning. Therefore, any subroutine called frequently or called from a loop should be located near the start of the program. Put that monster of an initialization routine at the end next time.

LETTERS

ED:

Mark Minasi gave me a simple solution to my problem with a Morse code routine. I needed to scan the keyboard for new letters while outputting messages from a buffer area. The $FEED routine would not work as it interrupted the operation while waiting for a key. A modification of his idea labled "2)" did the trick. I've been wanting output from other people since I purchased my superboard a year ago.

Keith Mund
Cheyenne, WY

ED:

I purchased a C8P-DF in January. Since that time I have done a considerable amount of "playing" to see what the machine can and can't do.

Now for some things that I have discovered. The February issue contained a memory map of 65U which contained a couple of errors. The operation of locations 2972 and 2976 were reversed. POKEing 13 in 2972 wil allow ":" on input. POKEing 2976 with 13 will allow "," on input. The initial contents are listed correctly. Location 11802 (# of floppy drives on system) contains a 4 on my system even though I have only 2 drives. This results in a "Drive Not Ready" error if Device C or D are selected rather than a "Drive Out Of Range" error. Very minor problem.

On the matter of screen clearing routines, with the 2MHZ operation of the C8P, the apparent OSI standard screen clear works rather rapidly. The routine is

```
FOR I=1 TO 32 : PRINT : NEXT I
```

ED:

Do you have an OSIO RESOURCE LIST--a list of people and gear and interests, etc. for computer pen pals?

Do you or anyone know of a black box to do color and high resolutions with an RS232 input and an color CRT output which could be treated like a special output device, given commands down the RS232, would translate them, reformat and do limited vectors for graphics?

Craig Lowband

ED:

Would you advise me if you have a driver routine for a Baudot teletype? (C1P) Thank you.

Jim Arens
Abbotsford, B.C.

ED:

I have a C1P. I modified my 25 x 25 video to 2K RAM of output, and effectively I have the same video memory map as the C2 (540 board) and in fact I now run C2 programs (graphics) with a minor correction to the keyboard polling routine. I have to change that back to the specifications as shown in the C1P manual (The column/row poke and peek routine). This is not too very hard and only takes a minute to do; the rest of the program runs intact. I get a 30 x 50 video display without guard-bands, much improved over the old 25 x 25 display. Since Bruce Showalter of Abilene, Texas queried last issue about this very thing, I recommend he write for the plans to this modification from advertising appearing in most computer magazines, for the plans it costs $8.00. For the parts, possibly under $50.00, depending on your location, and stockpile of your own parts. For those of you who have already done this modification, a question. I have recently added a 610 board, not yet adding my disk drive, but when I do, have you solved running the disc basic on your modified system, since there was a necessity of adding a small program in machine language to correct or modify the ROM BASIC line and cursor and scrolling parameters. There will also have to be a modification for the disc system but in a different manner and with different parameters to correct the DISC BASIC's similar line length/ scrolling/ cursor routines. Perhaps the OSI disc Basic itself has been modified or can be to correct this problem. Perhaps the C2 DISC BASIC will run on the modified C1P - I don't know.

Finally - OSI when selling the 610 board separately, does not have the real time clock "jumpered" in the active state. The instructions only state simply, that you need to wire the clock to the PIA. Even Howard Sams new technical manual and diagrams on the 600 and 610 boards do not specifically show the complete connec-tions, as there are several timing options. I need to know where in the decimal location/address for the C1P I can peek/poke this clock, and mainly I need to know exactly what pins of what clock chip need to be wired to what pins of the PIA chip.

James T. Hadfield
Anchorage, Alaska

ED:

Please accept the following routines which I have found useful in some graphic games programs.

1. To plot the point X,Y with CHR$(Z)
   POKE 53248 + X + 64*Y,Z

This assumes that (0,0) is top left of screen.

POKE 55232 + X - 64*Y,Z

This puts (0,0) at the normal bottom left of screen.

2. To print a string across the screen starting at a given location.
   String to be printed is B$
   Address at which message to start = B
   For Y=1 to LEN(B$):POKE B + Y,ASC(MID$(B$,Y,1):NEXT

3. To print a string;B$;down the screen starting at a given location; B.
   For Y=1 TO LEN(B$):POKE B + 64*(Y-10),ASC(MID$(B$,Y,1)):NEXT.

R.E. Bonser
London, England

I generally place this as a subroutine in all my programs, and then GOSUB to clear the screen.

On the subject of operating speed, I took part in the following scenario. The process computer group of the company I work for was investigating the use of small computers to do "busy work" in some of our operating departments. OSI came in fourth on a list of ten. However, I was told that speed was one of the criteria for selection. The following program was used for a benchmark:

```
10 J = 0
20 FOR I = 1 TO 10000
30 J = J + 1
40 NEXT I
50 PRINT J
```

The top two contenders (?) had the following operating times:

85 seconds	TRS-80	(BASIC)
43 seconds	CROMEMCO	(BASIC)

My C8P ran the program in 20 seconds using either 65U or 65D. The process computer people were amazed. I wasn't. I had told them that I could run about four times as fast as a TRS-80.

In a recent issue of PEEK(65), Larry Kiner had some questions about program compatibility. I would highly recommend "The Basic Handbook" by David A Lein. To answer Larry's specific questions,

1) CLS is a "clear screen" statement
2) ELSE does not work in OSI Basic. However you can get the same effect by proper use of IF-THENs and GOTOs.
3) OUT 255,4 sends the binary equivalent of the number 4 to port number 255. We use POKEs.
4) BREAK is not a space. It halts operation. We can use a stop.

John R. Soice
Charleston, WV

ED:

It seems your letter section is becoming the real backbone of PEEK(65). I'm sure there are more questions than answers but I hope people write in the answers to PEEK(65) and not the individual as many questions could have just as easily been written by me.

On this note, I will buy a year of PEEK(65) to anyone who helps me the most with these questions.

--What are the addresses and where are the I/O locations on the 600 board.
--Does anyone know what the values of the resistors are on the keyboard noise circuit.
--How do I interface the 600 to an S-100 system.
--And likewise how do I interface to a 500 board.

To anyone who can give me specifics on the S-100 and 500 board interface I will send PEEK(65) and enroll him in OSIO. Please send information direct to me and I will compile the info. and send it to PEEK(65) for other readers.

Brian Fearnow
RR 1, Box 169
Forest, Indiana

ED:

Tell your readers that I hope to publish (probably in MICRO or COMPUTE) complete memory maps, machine programmers guides, and users descriptions of OS65D. After these articles have appeared, then I will make my annotated listings available to anyone wishing them. Readers can send a self-addressed stamped postcard to me which I will return as soon as the listings are available informing them of this fact and how they may obtain a copy.

Now more to the point. If hobbyists continue to copy and distribute copyrighted programs, then they can expect the good professional software authors to move on to bigger things. Small programs are always available for free from hobbyists. My policy is to copy only non-copyrighted material and always to leave the author's name in the listing.

Over the next few months, I will put together (on a low priority basis) a bibliography of assembly language utilities which should enable anyone learning assembly language to put together a good monitor for next to nothing. I'll send along the list if and when it is complete. In this same vein, the OSI Extended Monitor and Assembler are quite good and quite competitively priced, even though it takes an expert to understand the documentation.

Tom Berger
10670 Hollywood Blvd.
Coon Rapids, MI 55433

ED:

I would like to apprize you of my article in June 79 'Micro'. It tells of speeding up my computer clock and tape baud rate both by 2 and of reversing my video. The white-up display is preferred by everyone and the 600 baud tape has extremely good reliability.

I feel that a major shortcoming in the OSI monitor is the lack of decent editing; not to mention trace; find; dump and other niceties (see 'Compute' Jan. 80 pp.4-5). Perhaps a better monitor could be loaded in high memory although I think that replacing the ROM is the only satisfactory solution. It certainly would be simple to move the ACIA addresses down from FC00 to some agreed on location such as DC00 and then use FC00-FFFF for a IK monitor or F800-FFFF if needed. I would be very interested in joining a cooperative effort to obtain replacement ROMs with repairs to GC and RND() and containing a new monitor.

I have a disassemble program that prints address in decimal and hex; instruction; mnemonic and decimal address. It converts relative (branch) addresses to absolute. With it I found the typo in the GC before trying to run it. The correct lines are:

```
100 IFI<>A+59THEN120
110 M=K+141:GOTO230
```

Also there might be a point in those of us who are radio amateurs to identify our callsigns--mine is AA6K.

James L. Cass
Northridge, California

ED.

There seems to be a misconception of the string-handling problem in Microsoft Basic, judging from the note in V1,#2 of your journal. The following program illustrates it simply:

```
10 D$(1)="DUMMY"
20 A$="MARY HAD A LITTLE LAMB"
30 PRINT A$
40 L$=LEFT$(A$,1)
50 R$=RIGHT$(A$,LEN(A$)-1)
60 A$=R$+L$
70 GOTO 30
```

As lines 40-60 operate, new copies of the revised A$ propagate down through from high memory until they reach the top of program store. Then the system seems to stop for garbage collection. Apparently all of memory is scanned repeatedly during this process and it may or may not terminate, depending on some factor of divisibility of memory space.

line 10 is deleted, the garbage collection is instant and successful. Thus it may be said that the problem comes from extensive string manipulation when any subscripted strings exist. The divisibility referred to appears to involve either string or program length, and varies with multiples of three.

Instructions for the Apple, which also uses (modified) Microsoft Basic, suggest that

```
65 X=FRE(9)
```

will distribute the garbage collection job at each pass and negate the problem, but this doesn't work on the OSI. In fact, that statement itself will cause the problem if your program contains any subscripted strings.

It should be pointed out that, by re-structuring your program, you can sometimes do a given job without subscripted strings, and avoid the problem.

J.E. Lucas
Canton, OH

AD$

C1P - SUPERBOARD II USERS - Forming software exchange club - Send SASE for details.

Bruce Mitchell, 1441 Clermont Dr., Durham, NC 27713

SOFTWARE FOR C1P AND SBII ON CASSETTE-FORT-8K-Simulation of a fort under attack where you are in command of the fort - $7.95. MASTERMIND - Try to guess the 4 letter code. With two levels of difficulty - $6.95. Instructions and listing included. Write:

Ed Kuzemchak, RD#2, Box 163, Clymer, PA 15728

FULL-HALF DUPLEX MODEM PROGRAM For C4 and C8-MF operating under OS65D. Your OSI behaves just like a terminal. Makes The Source and Micro-Net possibilities for the OSI user. Or, dial up any large time-share computer and down load or save programs. Disks are $35.00 for 5" and $40.00 for 8".

Send check or money order to: Kentucky Video, 2712 Garrett Dr., Bowling Green, KY 42101

ALL ABOUT OSI BASIC IN ROM - a 59 page reference manual (not a tutorial). Better organized than OSI's with errors fixed, omissions restored & subtle points explained. Much about the machine language implementation of BASIC. Memory maps of pages $00, 02, FE, FF & A000 - BFFF. $8.95. Dealer inquiries invited. Edward Carlson, 3872 Raleigh Dr., Okemos, MI 48864

ED:

We are working under OS 65 U on a double sided floppy disk and we have very often a failure on the second drive.

It appears: Error 17--Framing error!

Many OSI users in Europe have the same problem. Could you help us to remedy it?

Jacques Laborie
Versailles, France

Mr. Laborie:

Error 17 is a dangerous error, because it can propagate through RAM onto other disks. This is why if you get it once, you may see it start coming up again again.

To prevent this: If you get a disk read error 17, remove your disks and turn off your computer for 10 seconds. Then turn it back on and try again. If the error 17 repeats, do not use that disk any more!

Turn off the computer for 10 seconds and use your backup disk. Of course, you should copy disks frequently so your backup is always almost up to date.

Once you get your disks purged of all error 17's, you should see it very rarely. Unfortunately, it will be more frequent on double-sided drives than single-sided drives, since they handle the disks more roughly.

Al

OS-65U LEVEL 3 STUFF

PEEK(10317) is your user No.

Error 45 on CD-23: Level 3 can't access past 275967. Add the following line to the program "LEVEL3" to fix it:

944 DATA 57259,169,01: REM SEEK TIME OUT

We have received two books on OSI computers during the past month, both worth writing about.

The first is OSI BASIC IN ROM by Edward Carlson, advertised elsewhere in this issue.

Carlson is quite specific in his Introduction, pointing out that the book is all about OSI Microsoft Basic version 1.0, Rev. 3.2. The book is well written, starting off, as it were, with the basics, and carrying interested readers through Basic statements, the operation of the Interpreter, USR(X) and other exotica such as floating point and two's complement arithmetic. Although no previous knowledge of Basic is assumed, Carlson goes far enough in his explanations, memory maps and bug fixes to hold the interest of advanced programmers as well as novices. In fact, I intend to re-read this book periodically for a while, and I'm sure I will learn a new trick or two each time through.

The second interesting book is THE FIRST BOOK OF OSI, as told by Jim Williams and George Dorner, available from AARDVARK, 1690 Bolton, Walled Lake, MI 48088. If possible, this book is even more entertaining than Ed Carlson's book. It starts off with Chapter 0, and most communications are in hex, "...as God and MOS Technology intended." Less space is given to Basic in ROM, since the book has a wider scope, but some neat flowcharts and memory dumps explain in great detail how Basic stores and executes the data files we like to think of as programs. A useful bibliography and index are welcome features.

All in all, we heartily approve of both these books: we like their attitude of helpfulness and welcoming criticism and comment, the information they contain is useful and, though somewhat repetitive, there is sufficient unique material in each book that they don't compete. The complete OSI library should include both.

Read any good books lately? Reader book reviews are always welcome!

THE EXTERMINATOR

By Jim Sanders

This column is devoted to correction of the bugs in the standard utilities that come with the OS-65U disk. We will only cover a part of one utility per issue due to space limitations, so if you would like your favorite bug addressed, write and let us know which one. I have offered to correct the code on the utilities to OSI and even sent them a sample printout from RESEQ-5 showing US errors but didn't get a reply. So, you may be the only one on your block to have working utilities! Save these changes, since the old bugs and some new ones appear in every release.

This month we look at FPRINT and fix the random file print feature. The release is OSU-65 V 1.2 (March 6, 1980) Level 1. For earlier releases, find the problem area and make the same changes...it has never worked, on any release I know of.

After line 540....INPUT"FINISHING RECORD NUMBER"
INSERT: 545 IF FR=0 THEN FR=9999999: GOTO 560
BUG: The next line says "if FR less than 0 or less than SR..."
.....
From lines 750 to 870 change all "RØ" to "RO"
BUG: They input RO and then use RØ.
.....
After line 780 PRINT" FIELD","CONTENT"
790 Z=1
CHANGE: 800 Y=X * RL + RO
BUG: They did not kick the index to the next record.
.....
Line 50000 should appear and be:
50000 GOTO 990
Where line 990 is "EQ=PEEK(10226)" to detect EOF.
BUG: Their new resequencer trashes line 50000 and they don't replace it.
.....

To make these changes, you must load FPRINT with the password "PASS" and then save it with the same password. Next month, more bugs in the rest of FPRNT.

THE 8K BASIC "WAIT" INSTRUCTION
An Example
by Jim Sanders

The Basic program below uses the OSI 8K Basic "wait" command to obtain characters from a serial console keyboard. This example program illustrates the apparent echo of characters which are different from the ones which were typed.

The wait command is of the form: WAIT I,J,K and when interpreted the processor peeks at location 'I', performs an Exclusive-Or with the value of 'K', AND's the result with 'J', and continues to repeat this until a non-zero result is obtained. The program then continues.

For those of you who care, the EXOR simply takes the 8 bits of the word in memory and compares them with the 8-bits of 'K'. If they are both the same bit, the result is zero, otherwise the result is a one. For example, (5 EXOR 7) gives 2. The AND operation says if both bits are one, the result is one, otherwise zero.

In the example below we wish to stop the program until a key is pressed. In OSU, the serial keyboard character is found at 64513. The address before that is a flag bit which is zero until a key is pressed, and then changes to a one. This example tests location 64512 until it becomes a one, then peeks the character at the next address. The characters are saved in A$ until a return (ASCII 13 or 141 is seen. Each time a key is pressed another character from the "dummy echo" is printed.

For the big finish, when the return is pressed, anything leftover in the dummy string is printed, and the actual characters are compared with a password.

```
10 NU$="BENJAMIN HORSEFRATZ"          :REM DUMMY 'ECHO' LONGER THAN PW$
20 A=0 : PW$="SECRET"                 :REM INITIALIZE 'TRIES' AND PW$
30 KE=64513                           :REM SERIAL KEYBOARD UART ADDR.
40 IF A>2 GOTO 200                    :REM START OF LOOP - TEST TRIES
50 PRINT                              :REM UP TO THREE TIMES.
60 PRINT"ENTER PASSWORD: ";
70 A$=""                              :REM CLEAR PLACE FOR INPUT
80 FOR I = 1 TO LEN(NU$)              :REM LOOP UNTIL MAX DUMMY ECHO'S
90 WAIT KE-1, 1, 0                    :REM WAIT FOR A KEY TO BE HIT
100 B=PEEK(KE)                        :REM GET THE KEY'S ASCII VALUE
110 IF B=13 OR B=141 GOTO 160         :REM TEST FOR 'RETURN' KEY
120 A$=A$+CHR$(B)                     :REM ADD THIS KEY TO THE LIST
130 PRINT MID$(NU$,I,1);              :REM 'ECHO' FROM THE DUMMY STRING
140 NEXT I: A=A+1: GOTO 40            :    DUMMY COMPLETED! TRY AGAIN...
150 -                                 :REM GOT A RETURN - TIDY UP THE LOOP
160 IF A$=PW$ GOTO 180                :REM IF NOT CORRECT AND PERMIT
170 A=A+1:I=LEN(NU$):NEXTI:GOTO40     :    WRONG ANSWER- TIDY LOOP AND GO
180 PRINT MID$(NU$,I)                 :REM FINISH THE DUMMY STRING
190 PRINT:PRINT"CORRECT!":END
200 PRINT:PRINT"WRONG!":  END
```

Editor's Note: This is not only an example of the "WAIT" instruction. It is also an example of excellent, self- documenting code, the kind Wallace Kendall speaks of in his article in this issue. Notice the use and non-use of REM's. Lines 140 and 170 seem to have syntax errors, but since they both contain non- conditional GOTO's, BASIC never attempts to interpret the REMless comments! If we would all try to write code which approximates this in clarity, I might be able to read programs I wrote two years ago.

A1

OSI OSI

SOFTWARE FOR OHIO SCIENTIFIC

Over 50 programs for C1, C2, C4 & Superboard, on tape and disk. All come with listings and compete documentation.

GAMES - 4K - Tape

CHESS FOR OSI - specify system $19.95

STARFIGHTER 5.95
Real time space war.

SEAWOLFE 5.95
Floating mines, three target ships, etc.

LUNAR LANDER 5.95
With full graphics

TEN TANK BLITZ 9.95
A sophisticated real time tank game.

8K GAMES

BACKGAMMON 9.95

BLACKJACK 6.95
Plays all Vegas rules

Add $1.00 each for Color/Sound

UTILITIES

C1P CURSOR CONTROL $9.95
gives real backspace, one key screen clear, and **midline editing**

RENUMBERER 5.95

SUPERUTILITY 12.95
Has Renumberer, Variable table maker and Search

BUSINESS

SMALL BUSINESS ANALYSIS 15.95
Does profit and loss, quick ratio, breakeven analysis and more. 13 pages of documentation.

STOCK PORTFOLIO 6.95
Keeps track of your investments

Alien Invaders $6.95

Our $1.00 catalog has free game and utility listings, programming hints and a lot of PEEKs and POKEs and other stuff that OSI forgot to mention - and a **lot** more programs for sale.

DISKS 5" COLOR/SOUND $29.95
DISK 1. STARFIGHTER, ROBO-TANK, SEA WOLFE, BOMBER, TEN TANK BLITZ
DISK 2 BREAKTHROUGH, LUNAR LANDER, ALIEN INVADER, KILL-ERROBOTS, SLASHBALL

AARDVARK

1690 Bolton, Walled Lake, Michigan 48088 • (313) 624-6316