OSI-tems

Adjustable BASIC print routine , abstracted from ROM by George

```
20          JSR        1) OUTPUT VECTOR
4C 6C FF    JMP           Steps 1 to 3
8D 02 02    STA
48          PHA
8A          TXA
48          PHA
98          TYA
48          PHA
AD 02 02    LDA
F0 0B       BEQ
C9 0A       CMP
F0 2B       BEQ
C9 0D       CMP
D0 06       BNE
20          JSR
4C 6D BF    JMP
8D 01 02    STA
20 C2 BF    JSR
EE 00 02    INC
A9 7E       LDA   right margin  20) Print finish (right margin) on D3 Page
CD 00 02    CMP                    steps 15 to 24
30 0E       BMI
4C 6A BF    JMP
20 C2 BF    JSR
A9 61       LDA   left margin   25) Print start (left margin) on D3 Page
8D 00 02    STA                    steps 25 to 28
4C DE BF    JMP
20          JSR
20 C2 BF    JSR
A9 E0       LDA   start scrollarea (E0 = bottom)  30) selected scroll on D3 page
4C 79 BF    JMP
```

ep 31, 79 is the scroll vector
no scroll 6D
horm, 79
selected. 7C

February 1, 1980 Vol II, No I
OSI-tems

BITS
by Salomon Lederman

Here's a useful little routien to map points with x-y coordinates onto a C1-600 video mapped screen. Given x and y, N=54147-32*y+x, where n will be a location on the screen. X must be between zero and 24, and so must y. (0,0) is the origin and is located at the bottom lefthand corner of the screen at location 54147. In using this routine, x and y must be integers.

The reverse routine is also not difficult to apply. Given a location n, on the display, xaand y can be calculated as follows, where n is an integer:

```
X=INT(.5+32*((N-3)/32-INT((N-3)/32)))

Y=1692-INT(N/3)
```

Challenge: from Salomon Lederman

For those who are interested in mathematics I propose the following challenge. Write a program that will calculate and output all the digits of 100 factorial. (100 factorial = 1*2*3*4...98*99*100) The program must be totally in BASIC and must display all of the digits on the screen at one time. The challenge is to make the program run as fast as possible. Memory efficiency is of secondary importance. Note: There are over 200 digits in 100 factorial.

A good program should run in well under five minutes.

Good luck, and have fun!

CONCERNING THE PROGRAMS:

Santa Demo

This is a program written by OSI, and originally appeared in the OSI journal. It is a bit outdated but makes a cute demo, and will become useful in some three hundred odd days.

Please send any articles or programs to:

THOMAS CHENG
26 Madison Street, Apt. 4I
New York, N.Y. 10038

OR: send them in to
ARISTOCRAFT D.M.
314 Fifth Avenue
New York, N.Y. 10001

no money, but it'll get published.....

where n will be a location on the screen. X must be between zero and 24, and so must y. (0,0) is the origin and is located at the bottom lefthand corner of the screen at location 54147. In using this routine, x and y must be integers.

The reverse routine is also not difficult to apply. Given a location n, on the display, xaand y can be calculated as follows, where n is an integer:

```
X=INT(.5+32*((N-3)/32-INT((N-3)/32)))

Y=1692-INT(N/3)
```

Challenge: from Saloman Lederman

For those who are interested in mathematics I propose the following challenge. Write a program that will calculate and output all the digits of 100 factorial. (100 factorial = 1*2*3*4...98*99*100) The program must be totally in BASIC and must display all of the digits on the screen at one time. The challenge is to make the program run as fast as possible. Memory efficiency is of secondary importance. Note: There are over 200 digits in 100 factorial.

A good program should run in well under five minutes.

Good luck, and have fun!

CONCERNING THE PROGRAMS:

Santa Demo

This is a program written by OSI, and originally appeared in the OSI journal. It is a bit outdated but makes a cute demo, and will become useful in some three hundred odd days.

Please send any articles or programs to:

THOMAS CHENG
26 Madison Street, Apt. 4I
New York, N.Y. 10038

OR: send them in to
ARISTOCRAFT D.M.
314 Fifth Avenue
New York, N.Y. 10001

no money, but it'll get published.....

PAGE 5

Special interest program for all OSI C1P owners named Mike. Just type in and run.Submitted by Mike Bassman.

Program on page 6

This program originally appeared in the OSI Small Systems Journal. Modified for the C1P and made to work by Mike Bassman. Control keys are- 1 Move left, 2 Move ri ht, 3 Fire, 1 and 2 Move up

```
?CHR$(30)


OK
LIST

 1 REM DRAWING PROGRAM FOR THE C1P
 2 REM ===========================
 3 REM LETS YOU USE THE VIDEO SCREEN AS A "SKETCH PAD"
 4 REM BY DANNY SCHWARTZ
 10 FORX=1TO26:PRINT:NEXT:POKE54117,32
 20 FORX=546TO571:I=PEEK(X+64490):POKEX,I:NEXT
 30 POKE572,96
 40 POKE11,0:POKE12,253
 50 P=53776:POKEP,95:DEFFNF(P)=95
 60 X=USR(X):I$=CHR$(PEEK(531))
 70 P1=P:Q=PEEK(P)
 80 IFQ=95THENPOKEP,32
 90 IFQ=187THENPOKEP,161
 100 IFI$="W"THENDEFFNF(P)=187
 110 IFI$="B"THENDEFFNF(P)=95
 120 IFI$="S"THENDEFFNF(P)=187+92*(PEEK(P)=161)
 130 IFI$="O"THENDEFFNF(P)=95-92*(PEEK(P)=161)
 140 IFI$="L"THENP=P-1
 150 IFI$="R"THENP=P+1
 160 IFI$="U"THENP=P-32
 170 IFI$="D"THENP=P+32
 180 IFI$="Q"THENP=P-33
 190 IFI$="P"THENP=P-31
 200 IFI$="Z"THENP=P+31
 210 IFI$="/"THENP=P+33
 220 IFI$="C"THENPOKE11,34:POKE12,2:X=USR(X):GOTO40
 230 IFP1<>PTHENPOKEP,FNF(P)
 240 IFP1=PANDI$<>"Y"THENPOKEP,Q
 250 GOTO60
 300 REM DIRECTION COMMANDS
 310 REM ==================
 320 REM L-MOVE LEFT
 330 REM R-MOVE RIGHT
 340 REM U-MOVE UP
 350 REM D-MOVE DOWN
 360 REM Q-MOVE NW
 370 REM P-MOVE NE
 380 REM Z-MOVE SW
 390 REM /-MOVE SE
 400 REM TO USE:
 410 REM     AFTER TYPING "RUN", USE THE DIRECTION KEYS TO MOVE THE
 420 REM CURSOR TO THE DESIRED STARTING POSITION, THEN PRESS THE
 430 REM "W" OR "S" FOR "WHITE" OR "SWITCH" MODES.  PRESS DIRECTION
 440 REM KEYS AGAIN TO DRAW.
 450 REM
 460 REM
```

```
OK
LIST

 1 REM DRAWING PROGRAM FOR THE C1P
 2 REM ===========================
 3 REM LETS YOU USE THE VIDEO SCREEN AS A "SKETCH PAD"
 4 REM BY DANNY SCHWARTZ
 10 FORX=1TO26:PRINT:NEXT:POKE54117,32
 20 FORX=546TO571:I=PEEK(X+64490):POKEX,I:NEXT
 30 POKE572,96
 40 POKE11,0:POKE12,253
 50 P=53776:POKEP,95:DEFFNF(P)=95
 60 X=USR(X):I$=CHR$(PEEK(531))
 70 P1=P:Q=PEEK(P)
 80 IFQ=95THENPOKEP,32
 90 IFQ=187THENPOKEP,161
 100 IFI$="W"THENDEFFNF(P)=187
 110 IFI$="B"THENDEFFNF(P)=95
 120 IFI$="S"THENDEFFNF(P)=187+92*(PEEK(P)=161)
 130 IFI$="O"THENDEFFNF(P)=95-92*(PEEK(P)=161)
 140 IFI$="L"THENP=P-1
 150 IFI$="R"THENP=P+1
 160 IFI$="U"THENP=P-32
 170 IFI$="D"THENP=P+32
 180 IFI$="Q"THENP=P-33
 190 IFI$="P"THENP=P-31
 200 IFI$="Z"THENP=P+31
 210 IFI$="/"THENP=P+33
 220 IFI$="C"THENPOKE11,34:POKE12,2:X=USR(X):GOTO40
 230 IFP1<>PTHENPOKEP,FNF(P)
 240 IFP1=PANDI$<>"Y"THENPOKEP,Q
 250 GOTO60
 300 REM DIRECTION COMMANDS
 310 REM ==================
 320 REM L-MOVE LEFT
 330 REM R-MOVE RIGHT
 340 REM U-MOVE UP
 350 REM D-MOVE DOWN
 360 REM Q-MOVE NW
 370 REM P-MOVE NE
 380 REM Z-MOVE SW
 390 REM /-MOVE SE
 400 REM TO USE:
 410 REM     AFTER TYPING "RUN", USE THE DIRECTION KEYS TO MOVE THE
 420 REM CURSOR TO THE DESIRED STARTING POSITION, THEN PRESS THE
 430 REM "W" OR "S" FOR "WHITE" OR "SWITCH" MODES.  PRESS DIRECTION
 440 REM KEYS AGAIN TO DRAW.
 450 REM
 460 REM
 470 REM MODES:
 480 REM B-BLACK; LEAVES TRAIL OF BLACK, USED TO ERASE AND TO MOVE THE
 490 REM    CURSOR TO A DESIRED INITIAL LOCATION.
 500 REM W-WHITE; LEAVES A TRAIL OF WHITE- THE NORMAL DRAWING MODE.
 510 REM S-SWITCH; TURNS WHITE SQUARES BLACK, BLACK SQUARES WHITE.
 520 REM O-OVER; MOVE CURSOR WITHOUT DRAWING OR ERASING.
 530 REM C-CLEARS THE SCREEN, RETURNS CURSOR TO CENTER AND MODE TO BLAC
 540 REM Y-DELETES THE CURSOR FROM THE SCREEN.
OK
```

```
1 FORK=53248TO54272:POKEK,32:NEXT
2 DIMY(32),Z(32),W(32),V(32),E(27),F(27)
3 I=1:H=4:DIMP(22),O(22)
4 FORX=ITOH:READZ(X):NEXT:FORX=ITOH:READV(X):NEXT
5 FORX=ITOH:READY(X):NEXT:FORX=ITOH:READW(X):NEXT
7 IFH=32THEN11
8 I=I+4:H=H+4:RESTORE:GOTO4
11 GOTO1450
91 FORX=1TO5:READA(X):NEXT:FORX=1TO5:READB(X):NEXT
92 FORX=1TO9:READC(X):NEXT:FORX=1TO5:READD(X):NEXT
93 FORX=1TO27:READE(X):NEXT:FORX=1TO27:READF(X):NEXT
94 FORX=1TO22:READP(X):NEXT:FORX=1TO22:READO(X):NEXT
95 FORK=-1TO-12STEP-1:L=K+53943:H=53596+K
102 FORX=1TO27:POKEH+E(X),F(X):NEXT
105 FORX=1TO5:POKEL-A(X),D(X):NEXT
106 FORX=1TO5:POKEL-A(X)-1,D(X):NEXT
110 FORX=1TO5:POKEL-A(X)+1,32:NEXT
115 FORX=1TO22:POKEP(X)+H,O(X):NEXT
120 FORX=1TO12:POKEL+X,162:POKEL+X+32,188:POKEL+X-96,164:NEXTX
129 POKEL+13,32:POKEL+45,32:POKEL-83,32:L=L-61
150 FORX=1TO5:POKEL+X,B(X):NEXT:POKEL+6,32:L=L+30
182 FORX=1TO9:POKEL+X,C(X):NEXT:POKEL+10,32:IFK=-12THEN1300
1150 FORX=1TO22:POKEP(X)+H,32:NEXT
1200 FORX=1TO27:POKEH+E(X),32:NEXT
1300 NEXTK:RETURN
1450 PRINT:PRINT:PRINT" O.S.I. COMPUTERS SAY..":FORX=1TO21:PRINT:NEXT
1500 FORI=1TO50
1510 Q=(54100-53432)*RND(X)+53432
1515 IFPEEK(Q)<>32THEN1510
1520 POKE54117,32:POKEQ,46:NEXT
1999 FORC=1TO9
2000 FORX=4TO32STEP4
3000 L=53856
3001 L=L-(X+6)
3007 FORI=9TO12:POKEL+I,128:NEXT
3008 POKEL-128,240:POKEL-125,240
3009 POKEL-95,240:POKEL-94,240
3010 POKEL-63,253
3011 POKEL-33,227
3020 POKEL-32,226
3100 FORI=1TO8:POKEL+I,42:NEXT:FORI=2TO8:POKEL+I+32,42:NEXT
3111 POKEL-34,173
4150 R=-(X+6)
4160 X=X/4
4200 POKEY(X)+R,Z(X):POKEY(X)+V(X)+R,W(X)
4201 POKEY(X)+6+R,Z(X):POKEY(X)+V(X)+6+R,W(X)
4202 FORI=1TO101:NEXT
4210 POKEY(X)+R,32:POKEY(X)+V(X)+R,32:POKEY(X)+V(X)+6+R,32
4211 POKEY(X)+6+R,32
4300 X=X*4
4510 POKEL-34,32
4550 POKEL-128,32:POKEL-125,32:POKEL-95,32:POKEL-94,32
4560 POKEL-63,32:POKEL-33,32:POKEL-32,32
4570 FORI=1TO8:POKEL+I,32:POKEL+I+32,32:NEXT
4580 IFC=9ANDX=24THEN4600
4581 NEXTX,C
4590 FORI=1TO9:POKEL+I,128:NEXT
```

Christmas Demo

```
95 FORK=-1TO-12STEP-1:L=K+53943:H=53596+K
102 FORX=1TO27:POKEH+E(X),F(X):NEXT
105 FORX=1TO5:POKEL-A(X),D(X):NEXT
106 FORX=1TO5:POKEL-A(X)-1,D(X):NEXT
110 FORX=1TO5:POKEL-A(X)+1,32:NEXT
115 FORX=1TO22:POKEP(X)+H,O(X):NEXT
120 FORX=1TO12:POKEL+X,162:POKEL+X+32,188:POKEL+X-96,164:NEXTX
129 POKEL+13,32:POKEL+45,32:POKEL-33,32:L=L-61
150 FORX=1TO5:POKEL+X,B(X):NEXT:POKEL+6,32:L=L+30
182 FORX=1TO9:POKEL+X,C(X):NEXT:POKEL+10,32:IFK=-12THEN1300
1150 FORX=1TO22:POKEP(X)+H,32:NEXT
1200 FORX=1TO27:POKEH+E(X),32:NEXT
1300 NEXTK:RETURN
1450 PRINT:PRINT:PRINT" O.S.I. COMPUTERS SAY..":FORX=1TO21:PRINT:NEXT
1500 FORI=1TO50
1510 Q=(54100-53432)*RND(X)+53432
1515 IFPEEK(Q)<>32THEN1510
1520 POKE54117,32:POKEQ,46:NEXT
1999 FORC=1TO9
2000 FORX=4TO32STEP4
3000 L=53856
3001 L=L-(X+6)
3007 FORI=9TO12:POKEL+I,123:NEXT
3008 POKEL-128,240:POKEL-125,240
3009 POKEL-95,240:POKEL-94,240
3010 POKEL-63,253
3011 POKEL-33,227
3020 POKEL-32,226
3100 FORI=1TO8:POKEL+I,42:NEXT:FORI=2TO8:POKEL+I+32,42:NEXT
3111 POKEL-34,173
4150 R=-(X+6)
4160 X=X/4
4200 POKEY(X)+R,Z(X):POKEY(X)+U(X)+R,W(X)
4201 POKEY(X)+6+R,Z(X):POKEY(X)+U(X)+6+R,W(X)
4202 FORI=1TO101:NEXT
4210 POKEY(X)+R,32:POKEY(X)+U(X)+R,32:POKEY(X)+U(X)+6+R,32
4211 POKEY(X)+6+R,32
4300 X=X*4
4510 POKEL-34,32
4550 POKEL-128,32:POKEL-125,32:POKEL-95,32:POKEL-94,32
4560 POKEL-63,32:POKEL-33,32:POKEL-32,32
4570 FORI=1TO8:POKEL+I,32:POKEL+I+32,32:NEXT
4580 IFC=9ANDX=24THEN4600
4581 NEXTX,C
4600 FORI=1TO9:POKEL+I,128:NEXT
5000 GOSUB91
7000 DATA 189,143,143,190,31,32,33,33
7001 DATA 53921,53921,53922,53923
7002 DATA 189,143,190,190
8000 DATA 64,128,97,65,32,77,69,82,82,89,67,72,82,73,83,84,77,65,83
8001 DATA 32,64,64,64,64
9000 DATA 0,-31,-30,1,32,33,2,3,64,65,34,66,98,63,95,31,128,160
9001 DATA 96,130,162,226,194,127,158,190,222
9002 DATA 189,189,190,143,155,155,135,226,168,41,136,136,136,189,135
9003 DATA 143,161,175,205,200,139,139,139,196,201,140,140
9005 DATA 67,68,69,70,104,71,136,168,200,231,230
9006 DATA 100,101,131,133,134,163,164,166,196,198,228
9010 DATA 196,195,135,197,190,198,143,143,189,195,196
9011 DATA 5,6,0,9,10,181,182,17,248,14,1
10000 FORI=1TO5000:NEXT
10001 CLEAR:GOTO2
```

```
LIST

 10 FORK=53248TO54272:POKEK,32:NEXT
 20 GOTO150
 30 FORL=ATOBSTEPC
 40 POKEL,240:POKEL+C,D
 50 FORQ=1TO150:NEXTQ
 60 POKEL,32:POKEL+C,32
 70 NEXTL
 80 POKEL,D
 90 FORL=BTOASTEPC*-1
 100 POKEL,240
 110 FORQ=1TO100:NEXTQ
 120 POKEL,32
 130 NEXTL:RETURN
 150 FORK=1TO11
 160 READA,B,C,D
 170 GOSUB30
 180 NEXTK
 190 GOTO310
 200 DATA53603,53608,1,143
 210 DATA54154,53642,-32,191
 220 DATA53627,53612,-1,136
 230 DATA54157,53646,-32,135
 240 DATA53627,53614,-1,139
 250 DATA53389,53549,32,128
 260 DATA54159,53616,-32,147
 270 DATA53627,53617,-1,60
 280 DATA53394,53554,32,210
 290 DATA53627,53619,-1,210
 300 DATA54162,53682,-32,135
 310 POKE53544,210:POKE53672,209:POKE53555,207:POKE53533,203
 320 FORK=53545TO53554:POKEK,135:POKEK+32*4,123:NEXTK
 330 POKE53576,136:POKE53576+32,136:POKE53537,143:POKE53587+32,143
 340 A=53635:B=53639:C=1:D=135:GOSUB 30
 350 A=53659:B=53683-32+1:C=-1:D=143:GOSUB30
 360 D$=" ARENT,COMPUTERS WONDERFUL"
 370 FORK=1TOLEN(D$):POKEK+53327,ASC(MID$(D$,K,1)):NEXTK
 380 FORK=1TO5000:NEXT
 390 FORK=1TOLEN(D$):POKEK+53827,32:NEXTK
 400 RUN
OK
```

```
OK
LIST

 1 FORK=53248TO54272:POKEK,32:NEXT
 3 KE=57088:POKE530,1
 4 POKEKE,64:PRINT"INPUT A RANDOM NUMBER"
 5 INPUTZ1:FORX=53248TO54272:POKEX,32:NEXT
 6 POKE54117,32
 10 FORX=53355TO54219STEP32
 20 POKEX,147:POKEX+5,146
 30 NEXT
 35 DP=53996:POKEDP,0
 50 RP=INT(RND(Z1)*4+1)+53355
 51 FORY=1TO30:POKERP-RC,32
 52 RP=RP+INT(RND(Z1)*2+1)*32:POKERP,1
 53 FORX=1TOZ:NEXT:Z=Z*.8788
 54 IFRP=DPTHEN60
 58 RC=0:GOTO80
 60 FORX=1TO150:POKERP,X:NEXT
 61 POKERP,0:Z=75:POKERP-32,32
 62 FORX=1TO500:NEXT
 65 GOTO50
 80 POKEKE,64:G=PEEK(KE)
 81 IFG=126THEND=-1
 82 IFG=190THEND=1
 83 IFG=62THEND=-32
 84 IFG=254THEND=32
 85 IFG=222THEN10000
 92 IFPEEK(DP+D)=32THENDP=DP+D:POKEDP-D,32
 93 IFDP<53355THENDP=DP+27*32
 94 IFDP>54224THENDP=DP-27*32
 95 POKEDP,0:D=0
 100 FI=RND(Z1)*5
 110 IFFI>4THEN150
 120 GOTO490
 150 FORX=1TO24:POKERP+32*X,139
 151 IFRP+32*X=DPTHEN190
 160 NEXTX:FORX=1TO24:POKERP+32*X,32:NEXT
 170 GOTO490
 190 FORY=1TOX:POKERP+Y*32,32:NEXT
 200 FORX=1TO150:POKEDP,X:NEXT:POKEDP,32
 210 GOTO12000
 490 IFI>0THEN700
 500 H=INT(RND(Z1)*23)+53363
 550 H(1)=INT(RND(Z1)*3+13)
 600 I=1
 700 POKEH,32:H=H+64
 710 POKEH,H(1)
 720 IFH>54233THENPOKEH,32:I=0
 800 Q=INT(RND(Z1)*6-6)
 801 IFQ=5THENRC=1
 802 IFQ=-5THENRC=-1
 820 RP=RP+RC
 850 IFPEEK(RP)<>32THENRP=RP-RC:RC=0
 860 IFRC<>0THENPOKERP-RC,32
 900 NEXT
```

```
6 POKE54117,32
10 FORX=53355TO54219STEP32
20 POKEX,147:POKEX+5,146
30 NEXT
35 DP=53996:POKEDP,0
50 RP=INT(RND(Z1)*4+1)+53355
51 FORY=1TO30:POKERP-RC,32
52 RP=RP+INT(RND(Z1)*2+1)*32:POKERP,1
53 FORX=1TOZ:NEXT:Z=Z*.8788
54 IFRP=DPTHEN60
58 RC=0:GOTO80
60 FORX=1TO150:POKERP,X:NEXT
61 POKERP,0:Z=75:POKERP-32,32
62 FORX=1TO500:NEXT
65 GOTO50
80 POKEKE,64:G=PEEK(KE)
81 IFG=126THEND=-1
82 IFG=190THEND=1
83 IFG=62THEND=-32
84 IFG=254THEND=32
85 IFG=222THEN10000
92 IFPEEK(DP+D)=32THENDP=DP+D:POKEDP-D,32
93 IFDP<53355THENDP=DP+27*32
94 IFDP>54224THENDP=DP-27*32
95 POKEDP,0:D=0
100 FI=RND(Z1)*5
110 IFFI>4THEN150
120 GOTO490
150 FORX=1TO24:POKERP+32*X,139
151 IFRP+32*X=DPTHEN190
160 NEXTX:FORX=1TO24:POKERP+32*X,32:NEXT
170 GOTO490
190 FORY=1TOX:POKERP+Y*32,32:NEXT
200 FORX=1TO150:POKEDP,X:NEXT:POKEDP,32
210 GOTO12000
490 IFI>0THEN700
500 H=INT(RND(Z1)*23)+53363
550 H(1)=INT(RND(Z1)*3+13)
600 I=1
700 POKEH,32:H=H+64
710 POKEH,H(1)
720 IFH>54233THENPOKEH,32:I=0
800 Q=INT(RND(Z1)*6-6)
801 IFQ=5THENRC=1
802 IFQ=-5THENRC=-1
820 RP=RP+RC
850 IFPEEK(RP)<>32THENRP=RP-RC:RC=0
860 IFRC<>0THENPOKERP-RC,32
900 NEXT
1000 POKERP,32:GOTO50
10000 FORX=1TO30:POKEDP-32*X,140
10010 IFDP-32*X=RPTHENFORY=1TOX:POKEDP-32*Y,32:NEXT:FORY=1TO150
11020 IFDP-32*X=RPTHENPOKERP,Y:NEXT:POKERP,32:KI=KI+1:GOTO50
11030 NEXTX:FORX=1TO30:POKEDP-32*X,32:NEXTX
11040 GOTO93
12000 FORK=53248TO54272:POKEK,32:NEXT
12010 PRINT"YOU LOSE AFTER":PRINT"KILLING"KI"CARS"
12020 INPUT"TRY AGAIN";Y$:IFLEFT$(Y$,1)="Y"THENRUN
OK
```

SOFTWARE REVIEWS

submitted by:Mike Bassman (1 to 5 stars ratings)

C1 cursor control Aardvark Technical Services $9.95
****

One of the best utilities for the C1P available today. This program provides a true backspace,a new cursor,and most improtantly, mid-line editing,so that a line of code does not have to be typed over if there is a mistake in it.It has a machine-language screen clear in it,but this has some potentially dangerous bug in it,so don't use this.Overall,an excellent package.

Here is a list of total losers to be avoided like the plague:
0 *

Bomber	Ohio Scientific	$6.00
Fighter Pilot	Aardvark Technical Services	$6.00
Destroyer	Ohio Scientific	$6.00
Torpedo	"	"
New York Taxi	"	"

Alien Intruders Aardvark Technical Services $5.00
***1/2*

This is a replica of the incredible arcade game,Space Invaders. Since we do not have a high-resolution graphics board and BASIC is comparatively slow (compared to assembly,that is),this is not like the original,but it still has all the fun and excitement.And the (low) price shouldnt deter anyone from picking up a copy,either.

Assembler/Editor Ohio Scientific $55.00
**

This program,unfortunately,is virtually necessary if you intend to write in assembly language.The program works and provides the most necessary features,but the documentation is terrible and it takes up over 5K of users memory.Not to mention incredibly expemsive.

Machine Language Monitor Ohio Scientific $15.00
****

Very good program for all of you people working in machine language.A host of commands for a somewhat reasonable price.Main features are:Data move (with or without 6502 renumberer)
String and numerical search.
Disassembler
Data dump
Many more... Altogether a good buy.

Space Wars Procom Software $5.00
****1/2*

Accurate simulation of the arcade game.Beautiful graphics and real-time action make it great.Run out and buy it !

Superutility Aardvark Technical Services $9.95
**1/2*

Includes search,variable table maker and renumber. Search works,variable table maker works but is useless,and renumber, the major program,works when and only when it wants to.Besides it's expensive and most of the routines can be obtained free from other sources(like magazines,ositems,etc)

10 Tank Blitz Aardvark Technical Services $9.95

mid-line editing,so that a line of code does not have to be typed over if there is a mistake in it.It has a machine-language screen clear in it,but this has some potentially dangerous bug in it,so don't use this.Overall,an excellent package.

Here is a list of total losers to be avoided like the plague:
0 *

Bomber	Ohio Scientific	$6.00
Fighter Pilot	Aardvark Technical Services	$6.00
Destroyer	Chio Scientific	$6.00
Torpedo	"	"
New York Taxi	"	"

Alien Intruders Aardvark Technical Services $5.00
***1/2*

This is a replica of the incredible arcade game,Space Invaders. Since we do not have a high-resolution graphics board and BASIC is comparatively slow (compared to assembly,that is),this is not like the original,but it still has all the fun and excitement.And the (low) price shouldnt deter anyone from picking up a copy,either.

Assembler/Editor Ohio Scientific $55.00
**

This program,unfortunately,is virtually necessary if you intend to write in assembly language.The program works and provides the most necessary features,but the documentation is terrible and it takes up over 5K of users memory.Not to mention incredibly expemsive.

Machine Language Monitor Ohio Scientific $15.00
****

Very good program for all of you people working in machine language.A host of commands for a somewhat reasonable price.Main features are:Data move (with or without 6502 renumberer)
String and numerical search.
Disassembler
Data dump
Many more... Altogether a good buy.

Space Wars Procom Software $5.00
****1/2*

Accurate simulation of the arcade game.Beautiful graphics and real-time action make it great.Run out and buy it !

Superutility Aardvark Technical Services $9.95
**1/2*

Includes search,variable table maker and renumber. Search works,variable table maker works but is useless,and renumber, the major program,works when and only when it wants to.Besides it's expensive and most of the routines can be obtained free from other sources(like magazines,ositems,etc)

10 Tank Blitz Aardvark Technical Services $9.95
***1/2*

Each of two players has a fleet of 5 tanks with which to destroy his tanks,and ultimately,his bunker.Your weapons are front,side and guided missiles.Can be adjusted to be more of a video-based or more of a strategy game.But somewhat expensive and not many frills.

A LIST OF THE SPECIAL MEMORY LOCATIONS IN THE C1-P
AND THEIR FUNCTIONS
BY DANNY SCHWARTZ

****

HEX 000B-000C
DECIMAL 11-12
FUNCTION: USE DISPATCHER. WHEN A STATEMENT CONTAINING THE KEYWORD "USR" IS ENCOUNTERED, THE MICROPROCESSOR JUMPS, AS A SUBROUTINE, TO THE MEMORY LOCATION HELD IN THESE TWO BYTES (TO PEEK(11)+256*PEEK(12)) AND EXECUTES THE MACHINE LANGUAGE PROGRAM FOUND THERE, UNTIL IT ENCOUNTERS AN OPCODE OF 60(HEX)(RTS) , AT WHICH POINT IT RETURNS TO BASIC.

****

HEX 000E
DECIMAL 15
TERMINAL WIDTH. WHATEVER YOU ANSWER TO THE PROMPT "TERMINAL WIDTH" IS STORED IN THIS LOCATION, AND THE WIDTH CAN BE CHANGED BY POKING A DIFFERENT VALUE IN THIS LOCATION. MAXIMUM IS 255. ALWAYS SET TERMINAL WIDTH TO 72 OR ABOVE WHEN SAVING A PROGRAM, OTHERWISE EXTRA CARRIAGE RETURN CHARACTERS WILL BE RECORDED ON THE TAPE, WHICH WILL CAUSE ERRORS WHEN LOADING BACK THE PROGRAM.

******

HEX 0079-007A
DECIMAL 121-122
POINTS TO THE FIRST BYTE OF BASIC PROGRAM. NORMALLY 301(HEX). I.E. 0079=01, 007A=30

****

HEX 007B-007C
DECIMAL 123-124
POINTS TO THE FIRST BYTE AFTER THE BASIC PROGRAM. START OF BASIC VARIABLE TABLE.

****

HEX 007D-007C
DECIMAL 125-126
POINTS TO THE FIRST BYTE AFTER THE SUBSCRIPTED VARIABLE TABLE.

****

HEX 0081-0082
DECIMAL 129-120
STRING WORKSPACE POINTER POINTS TO A LOCATION IN HIGH RAM, IMMEDIATELY BELOW WHICH "CONCATENATED" STRINGS AND STRING FUNCTIONS WILL BE ASSEMBLED. STARTS OUT THE SAME AS LOCATIONS 133-134, BUT IS DECREMEMTED EACH TIME A STRING FUNCTION (MID$, LEFT$, ETC.) OR CONCATENATED STRINGS (SUCH AS A$=A$+B$)IS FORMED, BY AN AMOUNT EQUAL TO THE LENGTH OF THE STRING FORMED.
IF A LONG PROGRAM WHICH MAKES EXTENSIVE USE OF THE STRING FUNCTIONS CAUSES THE VALUE OF THIS POINTER TO BE LESS THAN THAT OF LOCATIONS(127-128) (CAUSES STRING WORKSPACE TO CONFLICT WITH VARIABLE TABLE).
THE SYSTEM WILL CRASH THE NEXT TIME A STRING FUNCTION IS CALLED!!

****

DECIMAL 531
HOLDS ASCII VALUE OF LAST CHARACTER INPUT FROM THE KEYBOARD.

****

HEX 0085-0086
DECIMAL 133-134
MEMORY SIZE. YOUR ANSWER TO "MEMORY SIZE?", OR THE RESULT OF THE COMPUTERS MEMORY TEST, IS STORED HERE IN 16 BIT, LOW BYTE FIRST FORM. ONLY LOCATIONS LESS THAN PEEK(133)+PEEK(134)*256 WILL BE USED BY BASIC.

****

HEX F000-F001
DECIMAL 61440-61441
CASSETTE PORT (SERIAL PORT 1)
POKE(ING) A VALUE INTO 61441 TRANSMITS THE CORRESPONDING ASCII CHARACTER TO THE CASSETTE PORT. PEEK(61441) IS THE ASCII CHARACTER CURRENTLY BEING RECEIVED FROM THE PORT.
LOCATION 61440 IS A "CHARACTER READY" INDICATER. IF PEEK(61440)AND 1 THEN A NEW CHARACTER IS READY TO BE RECEIVED.

****

HEX DF00
DECIMAL 57088
POLLED KEYBOARD.

****

HEX FD00-FDFF
DECIMAL (POKE 11,0 : POKE 12, 253): BEFORE X=USR(X)

CAN BE CHANGED BY POKING A DIFFERENT VALUE IN THIS LOCATION. MAXIMUM IS 255. ALWAYS SET TERMINAL WIDTH TO 72 OR ABOVE WHEN SAVING A PROGRAM, OTHERWISE EXTRA CARRIAGE RETURN CHARACTERS WILL BE RECORDED ON THE TAPE, WHICH WILL CAUSE ERRORS WHEN LOADING BACK THE PROGRAM.

******

HEX 0079-007A

DECIMAL 121-122

POINTS TO THE FIRST BYTE OF BASIC PROGRAM. NORMALLY 301(HEX). I.E. 0079=01, 007A=30

****

HEX 007B-007C

DECIMAL 123-124

POINTS TO THE FIRST BYTE AFTER THE BASIC PROGRAM. START OF BASIC VARIABLE TABLE.

****

HEX 007D-007C

DECIMAL 125-126

POINTS TO THE FIRST BYTE AFTER THE SUBSCRIPTED VARIABLE TABLE.

****

HEX 0081-0082

DECIMAL 129-120

STRING WORKSPACE POINTER POINTS TO A LOCATION IN HIGH RAM, IMMEDIATELY BELOW WHICH "CONCATENATED" STRINGS AND STRING FUNCTIONS WILL BE ASSEMBLED. STARTS OUT THE SAME AS LOCATIONS 133-134, BUT IS DECREMEMTED EACH TIME A STRING FUNCTION (MID$, LEFT$, ETC.) OR CONCATENATED STRINGS (SUCH AS A$=A$+B$)IS FORMED, BY AN AMOUNT EQUAL TO THE LENGTH OF THE STRING FORMED.

IF A LONG PROGRAM WHICH MAKES EXTENSIVE USE OF THE STRING FUNCTIONS CAUSES THE VALUE OF THIS POINTER TO BE LESS THAN THAT OF LOCATIONS(127-128) (CAUSES STRING WORKSPACE TO CONFLICT WITH VARIABLE TABLE).

THE SYSTEM WILL CRASH THE NEXT TIME A STRING FUNCTION IS CALLED!!

****

DECIMAL 531

HOLDS ASCII VALUE OF LAST CHARACTER INPUT FROM THE KEYBOARD.

****

HEX 0085-0086

DECIMAL 133-134

MEMORY SIZE. YOUR ANSWER TO "MEMORY SIZE?", OR THE RESULT OF THE COMPUTERS MEMORY TEST, IS STORED HERE IN 16 BIT, LOW BYTE FIRST FORM. ONLY LOCATIONS LESS THAN PEEK(133)+PEEK(134)*256 WILL BE USED BY BASIC.

****

HEX F000-F001

DECIMAL 61440-61441

CASSETTE PORT (SERIAL PORT 1)

POKE(ING) A VALUE INTO 61441 TRANSMITS THE CORRESPONDING ASCII CHARACTER TO THE CASSETTE PORT. PEEK(61441) IS THE ASCII CHARACTER CURRENTLY BEING RECEIVED FROM THE PORT.

LOCATION 61440 IS A "CHARACTER READY" INDICATER. IF PEEK(61440)AND 1 THEN A NEW CHARACTER IS READY TO BE RECEIVED.

****

HEX DF00

DECIMAL 57088

POLLED KEYBOARD.

****

HEX FD00-FDFF

DECIMAL (POKE 11,0 : POKE 12, 253), BEFORE X=USR(X)

POLLED KEYBOARD. SCANNING ROUTINE IN ROM. CALLING FD00 VIA THE USR FUNCTION CAUSES A SINGLE CHARACTER TO BE INPUT FROM THE KEYBOARD AND STORED IN LOCATION 531(DECIMAL).

*****

****

FOR THOSE OF YOU OUT THERE WHO ARE TOO CHEAP TO BUY A TI PROGRAMMER AND CANNOT CONVERT HEX TO DECIMAL IN THEIR HEADS, HERE IS A QUICK HEX TO DECIMAL PROGRAM, GRATIS FROM ME.

```
10 DIMB$(15):FORK=0TO15:READB$(K):NEXT
20DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
30 A=0:INPUT"HEX #";A$:FORL=LEN(A$)TO1STEP-1:FORK=0TO15
40IFMID$(A$,L,1)=B$(K)THENA=A+(16LEN(A$)-L)*K)
50 NEXTK:NEXTL:?:?"THE HEX # IS";A:?:?.GOTO30
```

****

```
5 FORX=546TO571:I=PEEK(X+64490):POKEX,I:NEXT:POKE572,96:X=USR(X)
10 FORK=53576TO53590:POKEK,135:POKEK+32*10,129:NEXTK
20 FORK=53607TO53863STEP32:POKEK,136:POKEK+15,143:NEXTK
30 POKE53575,210:POKE53591,207:POKE53895,209:POKE53911,209
40 FORG=53640TO53832STEP64:FORK=GTOG+14:POKEK,132:NEXTK:NEXTG
50 L=53736:FORK=1TO5:R(K)=53558+K*2*32:NEXTK
60 POKEL,18:FORK=1TO5:POKER(K),3:NEXTK:P=57088:POKE530,1:POKEP,127
70 REM***MIKE BASSMAN***
80 IFPEEK(P)<>127ANDPEEK(P)<>125THEN120
90 IFPEEK(L-32)=135THEN500
100 IFPEEK(L-64)<>32THEN600
110 POKEL,32:L=L-64:POKEL,18:GOTO500
120 IFPEEK(P)<>191ANDPEEK(P)<>189THEN160
130 IFPEEK(L+32)=129THEN500
140 IFPEEK(L+64)<>32THEN600
150 POKEL,32:L=L+64:POKEL,18:GOTO500
160 IFPEEK(P)<>253THEN500
170 IFPEEK(L+1)<>143THEN200
180 POKEL,32:L=L-14:IFPEEK(L)<>32THEN600
190 POKEL,18:GOTO500
200 IFPEEK(L+1)=3THEN600
210 POKEL,32:L=L+1:POKEL,18:S=S+1
215 S$=STR$(S)
220 FORK=2TOLEN(S$):POKE53515+K,ASC(MID$(S$,K,1)):NEXTK
500 A=INT(RND(1)*5)+1:IFPEEK(R(A)-1)=19THEN600
510 IFPEEK(R(A)-1)=136THENR(A)=R(A)+14:POKER(A)-14,32:GOTO550
520 POKER(A),32:R(A)=R(A)-1
550 POKER(A),3:GOTO70
600 PRINT"CRASH!!":FORK=1TO1000:NEXTK:RUN
```

RACER

1 - UP
2 - DOWN
7 - FORWARD

WRITTEN BY MIKE BASSMAN

```
OK
OK
LIST

1 DATA53000,54500,1,32
2 DATA53382,53403,1,161
3 DATA54150,54171,1,161
4 DATA53382,54150,32,161
5 DATA53403,54171,32,161
6 FORJ=0TO4
7 DATA-32,-31,1,33,32,31,-1,-33:
8 READA,B,C,D
10 FORX=ATOBSTEPC:POKEX,D:NEXTX
12 NEXTJ
15 FORN=1TO8:READA(N):NEXTN
20 POKE530,1
30 POKE53810,248
40 POKE53420,70
43 T=53810:K=1
50 FORJ=53379TO54171:IFPEEK(J)<>70THEN200
60 R=INT(RND(96)*8)+1
65 IFR>5THEN210
70 IFPEEK(J+A(R))=161ORPEEK(J+A(R))>247THEN50
80 POKEJ+A(R),70
90 NEXTJ
100 GOTO210
200 GOTO90
210 POKE57088,127
220 IFPEEK(57088)=127THEN300
230 IFPEEK(57088)=191THEN400
240 IFPEEK(57088)=223THEN500
250 IFPEEK(57088)<>239THEN50
```

L A TANK AND THE OBJECT OF THE GAME IS TO PUT OUT THE FIRE ON THE SCREEN THAT
EPRESENTS A UNIT OF THE FLAME.

R TANK IN A COUNTER-CLOCKWISE DIRECTION
NK IN A CLOCKWISE FASHION.
TANK TO MOVE FORWARD.

(2) BOMB THAT WILL MOVE IN THE DIRECTION YOUR TANK IS FACING. IT WILL WIPE
ILL KEEP MOVING UNTIL IT HITS A WALL, IN WHICH CASE, IT WILL DISAPPEAR.
HIS PROGRAM AS A SIMPLE EXERCISE. SEVERAL THINGS ARE MISSING. FIRST OF ALL IS
FT TO GIVE YOU A SCORE. ALSO, THE PROGRAM DOES NOT CHECK TO SEE IF THE FIRE
E CHALLENGE WOULD BE TO PUT OUT THE FIREH IN AS LITTLE TIME AS POSSIBLE AND
CTION. OF COURSE ONE COULD ADD THESE FEATURES IF ONE WANTED TO. ALL IN ALL

```
 160 IFPEEK(P)<>253THEN500
 170 IFPEEK(L+1)<>143THEN200
 180 POKEL,32:L=L-14:IFPEEK(L)<>32THEN600
 190 POKEL,18:GOTO500
 200 IFPEEK(L+1)=3THEN600
 210 POKEL,32:L=L+1:POKEL,18:S=S+1
 215 S$=STR$(S)
 220 FORK=2TOLEN(S$):POKE53515+K,ASC(MID$(S$,K,1)):NEXTK
 500 A=INT(RND(1)*5)+1:IFPEEK(R(A)-1)=19THEN500
 510 IFPEEK(R(A)-1)=136THENR(A)=R(A)+14:POKER(A)-14,32:GOTO550
 520 POKER(A),32:R(A)=R(A)-1
 550 POKER(A),3:GOTO70
 600 PRINT"CRASH!!":FORK=1TO1000:NEXTK:RUN
 OK
OK
LIST

 1 DATA53000,54500,1,32
 2 DATA53382,53403,1,161
 3 DATA54150,54171,1,161
 4 DATA53382,54150,32,161
 5 DATA53403,54171,32,161
 6 FORJ=0TO4
 7 DATA-32,-31,1,33,32,31,-1,-33:
 8 READA,B,C,D
 10 FORX=ATOBSTEPC:POKEX,D:NEXTX
 12 NEXTJ
 15 FORN=1TO8:READA(N):NEXTN
 20 POKE530,1
 30 POKE53810,248
 40 POKE53420,70
 43 T=53810:K=1
 50 FORJ=53379TO54171:IFPEEK(J)<>70THEN200
 60 R=INT(RND(96)*8)+1
 65 IFR>5THEN210
 70 IFPEEK(J+A(R))=161ORPEEK(J+A(R))>247THEN50
 80 POKEJ+A(R),70
 90 NEXTJ
 100 GOTO210
 200 GOTO90
 210 POKE57088,127
 220 IFPEEK(57088)=127THEN300
 230 IFPEEK(57088)=191THEN400
 240 IFPEEK(57088)=223THEN500
 250 IFPEEK(57088)<>239THEN50
 260 GOTO530
 300 K=K-1:IFK=0THENK=8
 310 POKET,247+K
 320 GOTO50
 400 K=K+1:IFK=9THENK=1
 410 POKET,247+K
 420 GOTO50
 500 IFPEEK(T+A(K))<>32THEN50
 505 POKET,32
 510 T=T+A(K):POKET,247+K
 520 GOTO50
 530 M=A(K)
 540 IFPEEK(T+M)=161THEN600
 550 IFPEEK(T+M-A(K))<247THENPOKET+M-A(K),32
 555 POKET+M,232:M=M+A(K):GOTO540
 600 IFPEEK(T+M-A(K))>246THEN50
 610 POKET+M-A(K),32:GOTO50
```

FIRE GAME BY SOL
IN THE GAME OF FIRE, YOU CONTROL A TANK, AND THE OBJECT OF THE GAME IS TO PUT OUT THE FIRE ON THE SCREEN THAT IS SPREADING. THE LETTER "F" REPRESENTS A UNIT OF THE FLAME.
YOU HAVE THREE COMMANDS.
1-SETTING THE 1 KEY ROTATES YOUR TANK IN A COUNTER-CLOCKWISE DIRECTION
2-THE 2 KEY WILL ROTATE YOUR TANK IN A CLOCKWISE FASHION.
3-THIS COMMAND WILL CAUSE YOUR TANK TO MOVE FORWARD.
4-THIS COMMAND WILL LAUNCH A CO(2) BOMB THAT WILL MOVE IN THE DIRECTION YOUR TANK IS FACING. IT WILL WIPE OUT ALL FLAME IN ITS PATH, AND WILL KEEP MOVING UNTIL IT HITS A WALL, IN WHICH CASE, IT WILL DISAPPEAR.
NOTE - I (S. LEDERMAN) WROTE THIS PROGRAM AS A SIMPLE EXERCISE. SEVERAL THINGS ARE MISSING. FIRST OF ALL IS THAT THIS PROGRAM MAKES NO ATTEMPT TO GIVE YOU A SCORE. ALSO, THE PROGRAM DOES NOT CHECK TO SEE IF THE FIRE IS TOTALLY EXTINGUISHED. THUS THE CHALLENGE WOULD BE TO PUT OUT THE FIREH IN AS LITTLE TIME AS POSSIBLE AND THE ONLY RATING IS YOUR SATISFACTION. OF COURSE ONE COULD ADD THESE FEATURES IF ONE WANTED TO. ALL IN ALL THE PROGRAM WAS A NICE DEMO.
HAVE FUN!

WRITTEN BY MIKE BASSMAN