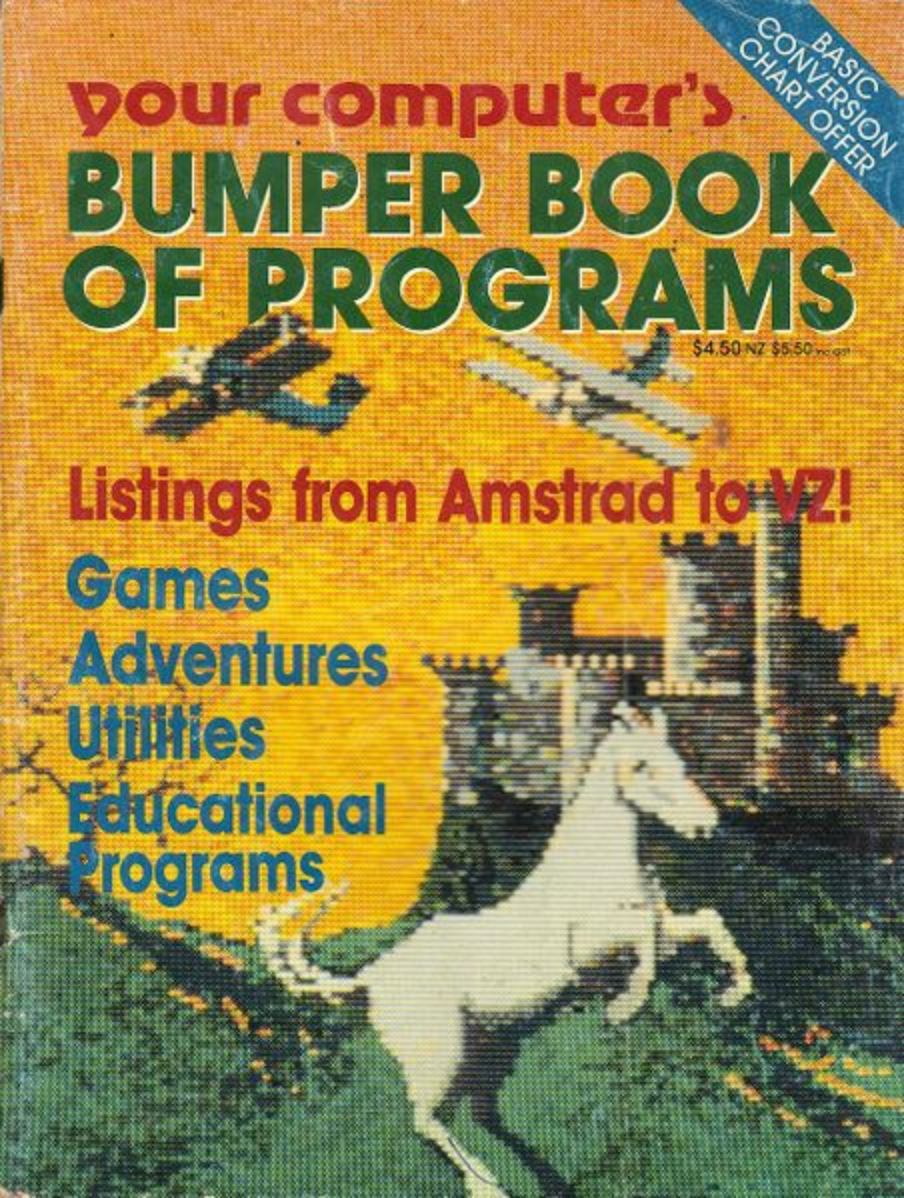
BASIC CONVERSION CHART OFFER

your computer's BUMPER BOOK OF PROGRAMS

$4.50 NZ $5.50 inc GST



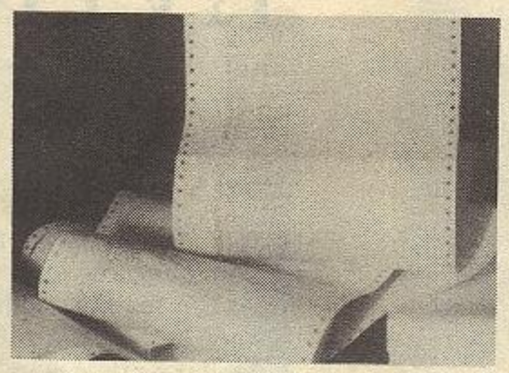
Listings from Amstrad to VZ!

Games
Adventures
Utilities
Educational Programs

CONTENTS

WHAT YOU'LL FIND

You'll find programs galore for just about any purpose in this all new issue of *Your Computer* magazine's ***Bumper Book of Programs***. There are games, games and more games, as well as handy little utilities to help you in your programming and — even one that will give you your biorhthym! We hope you get hours of enjoyment from the programs published. Happy programming!



Editor
Jake Kennedy
Production Editor
Allecia Khartu
Art Director
Pamela Horsnell
Production
Kylie Prats
Office Services
Angela Pagones

BUMPER BOOK OF PROGRAMS was published by The Federal Publishing Co Pty Ltd. Printed by Hannanprint, 140 Bourke Rd, Alexandria 2015. Distributed Nationally by Newsagents Direct Distribution Pty Ltd *Recommended and maximum price only. ISSN 0725-3931.

AMSTRAD

FLASH

Have you ever wanted to have unlimited power at your fingertips? Well, this program is not that good but it does simulate a sort of magical touch.

Every time you touch a key a simulated lightning flash of multicolours scans the screen, almost blinding you. The same happens when you use an operation. Only it lasts longer.

Try (cat) and watch the result. It creates an electrical storm. This is a good program to trick your friends with. You can tell them that your computer is playing up. Then tell them to press a key and watch their reaction. It's quite amusing.

It's also good to test out your sunglasses.

P. White
Elizabeth Park SA

```
100 REM ############## FLASH ###########
####
110 REM # Stephen White invention # Jan'
87 #
120 ' Causes colours to flash on the scr
een
130 ' whenever something is printed.
140 '
150 MEMORY &9FFF
160 DATA 245,229,213,197,33,208,189,17,3
6,160,1,3,0
170 DATA 237,176,33,39,160,17,208,189,1,
3,0,237,176,193
180 DATA 209,225,241,201,245,205,37,189,
241,195,91,18,195,31,160
190 '
200 FOR i=&A000 TO &A000+41
210  READ d: POKE i,d
220 NEXT i
230 CALL &A000
240 '
250 END
```

BOUNCE

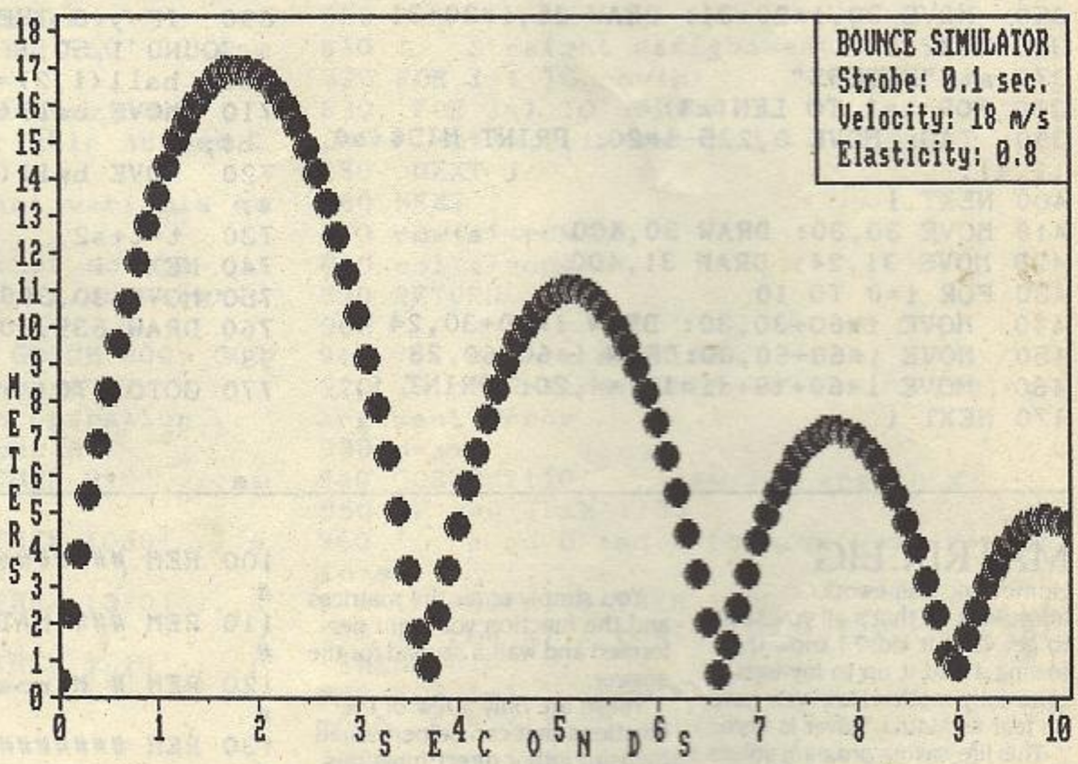
This is an excellent program for physics enthusiasts. Remember the days in school when you tried the experiment with the strobe light, the golf ball and the camera, to see the effect of distance against time? Well, you can forget that primitive way of deriving a result. This new Bounce simulator does it all for you — plus more.

You firstly specify the initial velocity of a ball thrown straight up in the air, measured in metres per second. Then you put in the co-efficient of elasticity for the ball, which is a measure of the bounciness. For realism you have to put in a decimal fraction under 1. This is because no ball can bounce higher than the place from where it was dropped. If you put in 1 this will make it bounce just as high as where it was dropped from. You can experiment with numbers larger than 1 to get some interesting results, but you can't rely on these results as it's a contradiction of the law of kinetics. The next input is the time increment for the strobe. This is the space of time between each simulated flash on the ball.

When the information has been entered, the screen is drawn up, comprising a graph of distance versus time filling the screen, with a box in the top right hand corner showing your input. The information is then simulated into graphics giving you a pictorial view of what you want.

This program saves a lot of time and setting up, not to mention film for all the blunders that are destined to occur.

M. Kostecki
Elizabeth Park SA




```
100 REM  ##### BOUNCE SIMULATOR #####
110 REM # Miroslav Kostecki # Dec'86 #
120 '
130 MODE 2: INK 1,0: INK 0,13: BORDER 13
140 PRINT:PRINT:PRINT: ZONE 5
150 PRINT ,,,,"BOUNCE SIMULATOR": PRINT:
PRINT
160 PRINT ," This simulation lets you sp
ecify the initial velocity"
```

```
170 PRINT ,"of a ball thrown straight up
, and the coefficient of"
180 PRINT ,"elasticity of the ball.  Ple
ase use a decimal fraction"
190 PRINT ,"coefficiency (less then 1)."
200 PRINT
210 PRINT ," You also specify the time i
ncrement to be used in"
220 PRINT ,"'strobing' the ball's flight
 (try .1 initally)."
230 PRINT
240 INPUT "    Time increment (sec) ";S2
250 PRINT
260 INPUT "    Velocity (m/s) ";v
270 PRINT
280 INPUT "    Elasticity Coefficient ";
c
290 CLS
300 SYMBOL AFTER 230
310 SYMBOL 230,&X101,&X101111,&X1011111,
&X10111111,&X10111111,&X1011111,&X101111
,&X101
320 SYMBOL 231,&X11100000,&X11111100,&X1
1111110,255,255,&X11111110,&X11111100,&X
11100000
330 FOR i=0 TO 20
340  TAG: MOVE -3-(i<10)*8,i*20+36: PRIN
T i;
350  MOVE 30,i*20+31: DRAW 25,i*20+31
360 NEXT i
370 a$="METERS"
380 FOR i=1 TO LEN(a$)
390  TAG: MOVE 0,225-i*20: PRINT MID$(a$
,i,1);
400 NEXT i
410 MOVE 30,30: DRAW 30,400
420 MOVE 31,24: DRAW 31,400
430 FOR i=0 TO 10
440  MOVE i*60+30,30: DRAW i*60+30,24
450  MOVE i*60+60,30:DRAW i*60+60,28
460  MOVE i*60+19+(i=10)*4,20: PRINT i;
470 NEXT i
480 MOVE 30,30: DRAW 640,30
490 a$="SECONDS"
500 FOR i=1 TO LEN(a$)
510  TAG: MOVE i*30+192,12: PRINT MID$(a
$,i,1);
520 NEXT i
530 MOVE 30,398: DRAW 639,398
540 DRAW 639,30: MOVE 638,30: DRAW 638,3
98
550 MOVE 500,390: PRINT "BOUNCE SIMULATO
R";
560 MOVE 500,370: PRINT "Strobe:";s2;"se
c.";
570 MOVE 500,350: PRINT "Velocity:";v;"m
/s";
580 MOVE 500,330: PRINT "Elasticity:";c;
590 MOVE 486,398: DRAW 486,308: DRAW 639
,308
600 MOVE 487,398: DRAW 487,308
610 ORIGIN 0,0,32,640,396,32
620 GRAPHICS PEN 1,1
630 num=10/s2: t=0: b$=CHR$(230)+CHR$(23
1)
640 DIM ball(num+num/20,2)
650 FOR i=0 TO num+num/20
660  ball(i,1)=22+i*600/num
670  y=v*t-4.9*t*t
680  IF v<0.000001 THEN 750
690  IF y<0 THEN t2=v/4.9: t=t-t2: v=v*c
: SOUND 1,50,1: GOTO 670
700  ball(i,2)=y*20+47
710  MOVE ball(i,1)-1,ball(i,2),0: PRINT
 b$;
720  MOVE ball(i,1),ball(i,2),1: PRINT b
$;
730  t=t+s2
740 NEXT i
750 MOVE 30,398: DRAW 639,398
760 DRAW 639,30: MOVE 638,30: DRAW 638,3
98
770 GOTO 770
```

MATRIX.EIG

Homework, homework, homework . . . that's all you seem to get, isn't it kids? I know the feeling. I had it up to my ears, especially maths. However, have no fear as Matrix Solver is here.

This life-saving program solves any matrix operation you may ask with the greatest of ease, which is a bit different from the long agonizing minutes on one question done manually. The functions are as follows:

a) Matrix multiplication.
b) Scalar multiplication.
c) Matrix addition.
d) Matrix subtraction.
e) Row reduction.
f) Determinant finding.

You simply enter the matrices and the function you want performed and wait a second for the answer.

These are only a few of the functions that can be performed. By using other direct functions you can come to the answer. For example, to subtract you would multiply the matrix by negative one and add the other matrix.

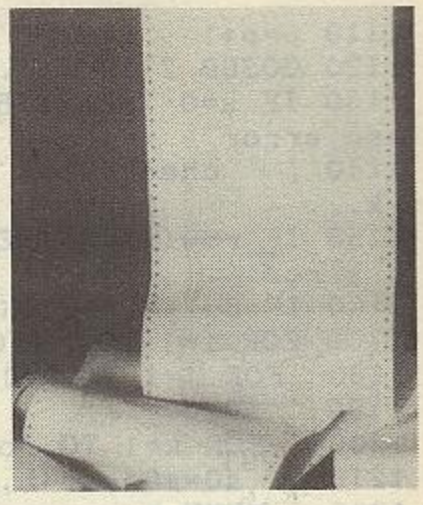
The good thing about this program is that you don't have to know anything about matrices. Nothing at all! That is, of course, except for the question. The program is not only great for kids but also the teachers as well. You teachers out there, don't you

```
100 REM ##############################
#
110 REM ### MATRIX ARITHMETIC PACKAGE #
#
120 REM # Miroslav Kostecki ## 18.12.86
#
130 REM ##############################
#
140 '
150 varnum=10' ...Maximum number of var
ables
160 size=10' ...Maximum number of Rows
r Columns per Matrix
170 DIM mat(varnum,size,size),temp(size
size),row(varnum),col(varnum)
180 CLS: PRINT
190 PRINT "                COMMANDS"
200 PRINT
```

hate the arduous task of working out the right matrices to an answer for a test or exam? This work can be halved by the matrix package. Using the functions you can enter the desired answer and the computer will manipulate it for you.

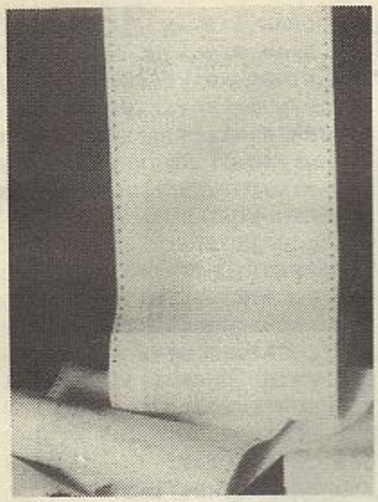
This program is easy to expand on if your requirements are not met, and if you don't have an Amstrad, no need to panic, as Matrix will work on most Basic computers.

M. Kostecki & P. Vermeer
Elizabeth Park SA



```
210 PRINT "          a -Print a matrix."
220 PRINT "         a= -Entry."
230 PRINT "       a=b -Assignment betwe
en variables."
240 PRINT "     a=b*c -Matrix multiplic
ation."
250 PRINT " a=b.<real> -Scalar multiplic
ation."
260 PRINT "     a=b+c -Matrix addition.
"
270 PRINT "      a=b: -Row reduction."
280 PRINT
290 '     Get command and do as asked...
300 INPUT " #",in$
310 IF in$="quit" THEN END
320 IF in$="help" THEN 180' ...show oper
ations
330 p=1' ...Position in string
340 GOSUB 2110' ...Get 1st variable
350 IF x=0 THEN 1750' ...not variable er
ror
360 a=x
370 IF p-1= LEN(in$) THEN 700' ...show a
 matrix
380 GOSUB 2040' ...skip spaces
390 IF MID$(in$,p,1)<> "=" THEN 1770
400 IF p=LEN(in$) THEN 560' ...get a mat
rix
410 GOSUB 2040' ...skip spaces
420 IF MID$(in$,p,1)="" THEN 560' ...spa
ces after = so get matrix
430 p=p+1' ...point to next variable
440 GOSUB 2110' ...get variable number i
n x
450 IF x=0 THEN 1750' ...not variable er
ror
460 b=x' ...first argument of source
470 IF row(b)=0 THEN 1830' ...unassigned
 error
480 IF p-1= LEN(in$) THEN GOSUB 800: GOT
O 290' ...straight assighnment
490 '            Must be an operation...
500 GOSUB 2040' ...skip spaces
510 IF MID$(in$,p,1)="+" THEN 910' ...ma
trix addition
520 IF MID$(in$,p,1)="*" THEN 1080' ...m
atrix multiplication
530 IF MID$(in$,p,1)="." THEN 1370' ...s
calar multiplication
540 IF MID$(in$,p,1)=":" THEN 1510' ...r
ow reduction
550 GOTO 1860
560 '
570 '   Get a matrix...
580 INPUT "Number of rows ";row(a)
590 IF row(a)<1 OR row(a)>size THEN 580
600 INPUT "Number of columns ";col(a)
610 IF col(a)<1 OR col(a)>size THEN 600
620 '   get array...
630 FOR i=1 TO row(a)
640  FOR j=1 TO col(a)
650   INPUT ;mat(a,j,i): PRINT,
660  NEXT j
670  PRINT
680 NEXT i
690 GOTO 290
700 '
710 '   Print a matrix...
720 IF row(a)=0 THEN 1830' ...unassigned
 error
730 FOR i=1 TO row(a)
740  FOR j=1 TO col(a)
750   PRINT mat(a,j,i),
760  NEXT j
770  PRINT
780 NEXT i
790 GOTO 290
800 '
810 '   Straight assighnment...
820 FOR i=1 TO row(b)
830  FOR j=1 TO col(b)
840   mat(a,j,i)=mat(b,j,i)
850  NEXT j
860 NEXT i
870 row(a)=row(b)
880 col(a)=col(b)
890 RETURN
900 '
910 '   Matrix addition
920 IF p=LEN(in$) THEN 1790' ...missing
argument error
930 p=p+1
940 GOSUB 2110' ...get 3rd arg in x
950 IF x=0 THEN 1750
960 '   need b and x to have same diment
ions
970 IF row(x)=0 THEN 1830
980 IF row(x)<> row(b) OR col(x)<>col(b)
 THEN 1810
990 FOR i=1 TO row(b)
1000  FOR j=1 TO col(b)
1010   mat(a,j,i)=mat(b,j,i)+mat(x,j,i)+
1-1
1020  NEXT j
1030 NEXT i
1040 row(a)=row(b)
1050 col(a)=col(b)
1060 GOTO 290
1070 '
1080 '   Matrix multiplication...
1090 '   check for third argument
1100 IF p=LEN(in$) THEN 1790' ...missing
 arg. error
```

```
1110 p=p+1
1120 GOSUB 2110' ...get third arg in x
1130 IF x=0 THEN 1750' ...bad variable n
ame error
1140 '   check that rows/columns will wo
rk...
1150 IF row(x)=0 THEN 1830' ...unassigne
d error
1160 IF col(b)<>row(x) THEN 1880
1170 FOR i=1 TO row(b)
1180  FOR j=1 TO col(x)
1190   s0=0
1200   FOR k=1 TO row(x)
1210    s0=s0+mat(b,k,i)*mat(x,j,k)
1220   NEXT k
1230   temp(j,i)=s0
1240  NEXT j
1250 NEXT i
1260 '   assign result to destination a.
..
1270 FOR i=1 TO row(b)
1280  FOR j=1 TO col(x)
1290   mat(a,j,i)=temp(j,i)+1-1
1300  NEXT j
1310 NEXT i
1320 '   assign correct dimentions to a.
..
1330 row(a)=row(b)
1340 col(a)=col(x)
1350 GOTO 290
1360 '
1370 '   Scalar multiplication
1380 IF p=LEN(in$) THEN 1790
1390 p=p+1
1400 n0$=RIGHT$(in$,LEN(in$)-p+1)
1410 n1=VAL(n0$)
1420 FOR i=1 TO row(b)
1430  FOR j=1 TO col(b)
1440   mat(a,j,i)=n1*mat(b,j,i)
1450  NEXT j
1460 NEXT i
1470 row(a)=row(b)
1480 col(a)=col(b)
1490 GOTO 290
1500 '
1510 '   Row reduction...
1520 IF row(b)<2 OR col(b)<2 THEN 1900'
...too small error
1530 IF row(b)>=col(b) THEN 1920' ...Err
or ***
1540 GOSUB 800
1550 FOR i=1 TO row(a)
1560  y1=mat(a,i,i): IF y1=0 THEN 1650
1570  FOR j=1 TO row(a)
1580   IF j=i THEN 1640
1590   z1=mat(a,i,j)
1600   FOR k=1 TO col(a)
1610    mat(a,k,j)=mat(a,k,j)-mat(a,k,i)
*z1/y1
1620   NEXT k
1630   mat(a,i,j)=0
1640  NEXT j
1650 NEXT i
1660 FOR i=1 TO row(a)
1670  IF ABS(mat(a,i,i))<0.0001 THEN 172
0
1680  FOR j=row(a)+1 TO col(a)
1690   mat(a,j,i)=mat(a,j,i)/mat(a,i,i)
1700  NEXT j
1710  mat(a,i,i)=1
1720 NEXT i
1730 GOTO 290
1740 '
1750 PRINT "  ";MID$(in$,p,1);" is not a
 variable"
1760 GOTO 1950
1770 PRINT "  Missing equals..."
1780 GOTO 1950
1790 PRINT "  Missing argument"
1800 GOTO 1950
1810 PRINT "  Cannot add matricies of di
fferent dimentions"
1820 GOTO 290
1830 PRINT "  Unassigned variable"
1840 p=p-1
1850 GOTO 1950
1860 PRINT "  Not an operation"
1870 GOTO 1950
1880 PRINT "  Cannot compose due to mism
atched codomain and domain"
1890 GOTO 1930
1900 PRINT "  Too small to reduce"
1910 GOTO 1930
1920 PRINT "  Too many rows"
1930 PRINT CHR$(7);:GOTO 290
1940 '
1950 PRINT "  ";CHR$(7);in$
1960 PRINT "  ";
1970 IF p=1 THEN 2010
1980 FOR i=1 TO p-1
1990  PRINT " ";
2000 NEXT i
2010 PRINT "^"
2020 GOTO 290
2030 '
2040 '   Subroutene to skip spaces...
2050 IF p=LEN(in$) THEN 2100
2060 '   OK to see if there are some spa
ces
2070 IF MID$(in$,p,1)<>" " THEN 2100' ..
.not a space
2080 p=p+1
2090 GOTO 2070
2100 RETURN
2110 '
2120 ' ...Get a variable name in x from
in$,
2130 '   returning 0 if error
2140 '   uses in$,p,x
2150 GOSUB 2040' ...skip spaces
2160 m$=MID$(in$,p,1): IF m$<"a" OR m$>"
z" THEN x=0: GOTO 2200
2170 x=ASC(m$)-ASC("a")+1
2180 IF x>varnum THEN x=0: GOTO 2200
2190 p=p+1
2200 RETURN
```



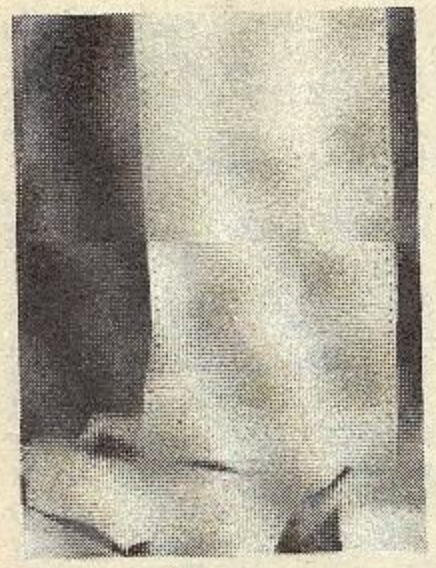
BRICKS

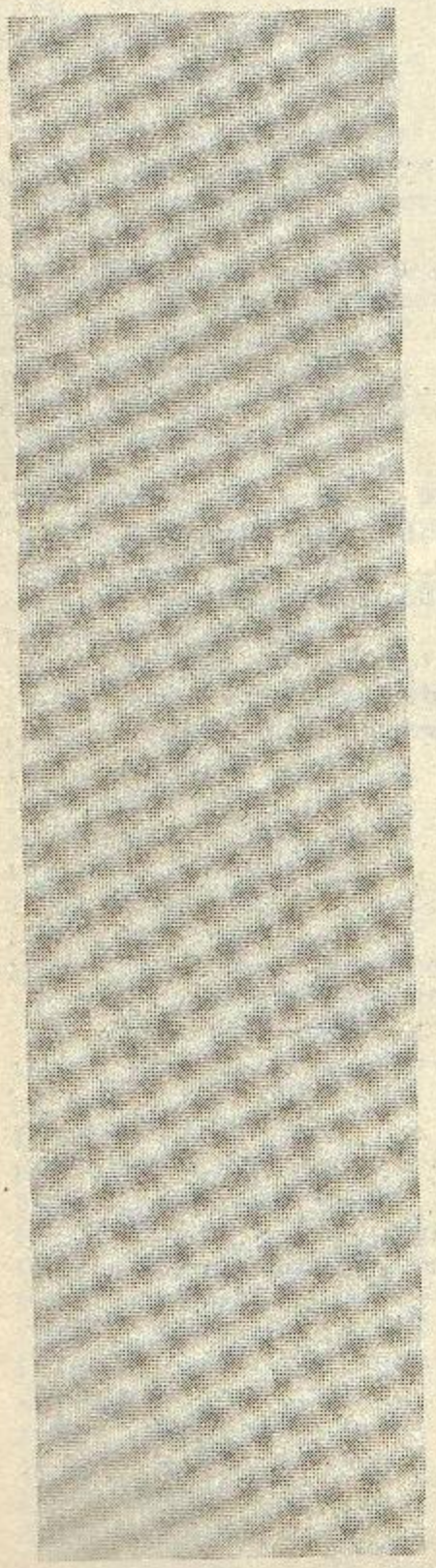
Bricks is a game where you must knock down a row of bricks with a 'ball'. You use the joystick or keyboard to control the 'bat' to hit the 'ball'.

Pressing 'Q' moves the bat up while pressing 'A' moves the bat down. 'Y' or pressing the fire button restarts the game.

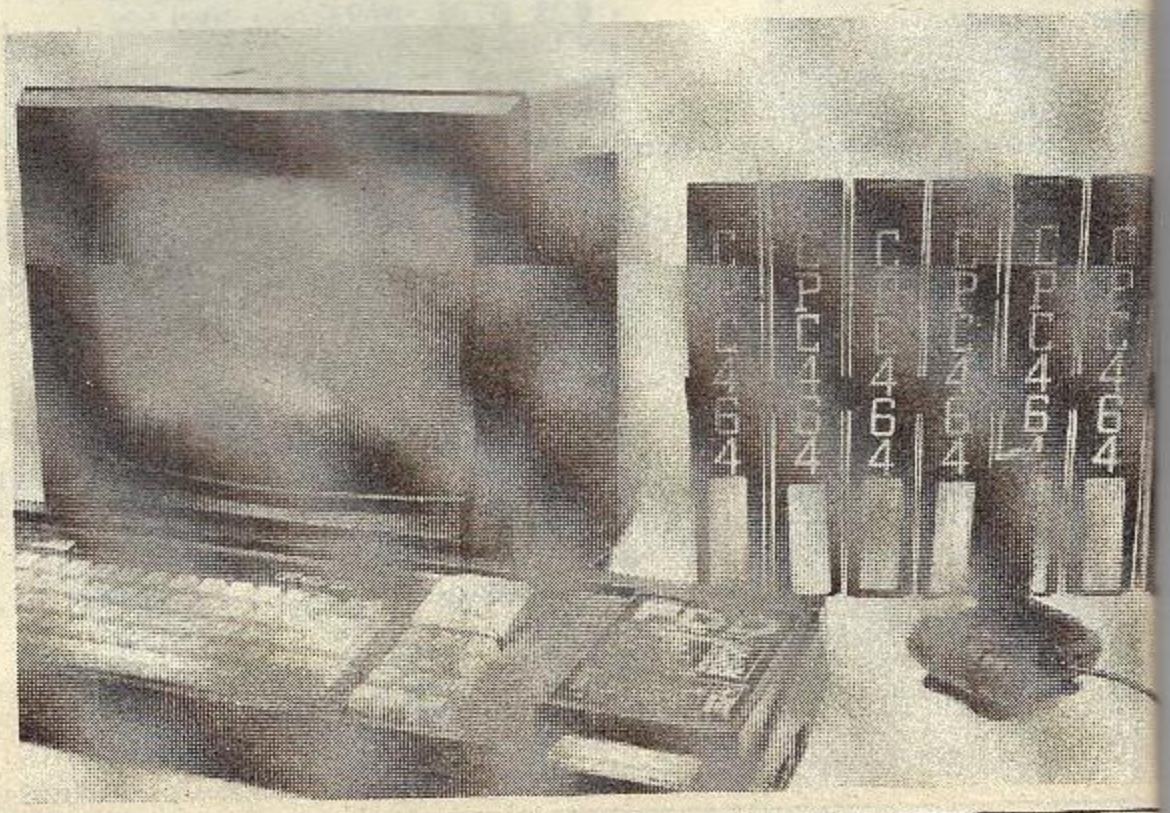
D. O'Connor
Mt Gambier SA

```
10 '
20 '
30 '
40 '
50 '
60 '
70 '
80 '
90 '
100 DIM s(6,22)
110 GOSUB 990
120 x=5
130 y=INT(RND(TIME)*19)+3
140 d1=1
150 d2=INT(RND(TIME)*2)+1
160 IF d2=2 THEN d2=-1
170 '
180 '  move ball
190 '
200 IF y>21 THEN d2=-1
210 IF x>18 THEN d1=-1
220 IF y<3 THEN d2=1
230 IF x<2 THEN 850
240 LOCATE x,y
250 PEN 0
260 PRINT CHR$(143);
270 x=x+d1
280 y=y+d2
290 IF x=3 AND (y=y1 OR y=y1+1 OR y=y1+2) THEN d1=1:SOUND 7,4000,5,15,0,0,5
300 IF y=y1+1 THEN y=y-1:GOTO 320
310 IF y=y1 THEN y=y+1
320 LOCATE x,y
330 PEN 5
340 PRINT CHR$(143);
350 IF x<14 THEN 590
360 '
370 '  hit bricks
380 '
390 IF s(x-13,y)=1 THEN 200
400 SOUND 3,4000,10,12,0,0,10
410 sc=sc+1
420 LOCATE #1,12,1
430 PRINT#1,sc
440 s(x-13,y)=1
450 IF d1=-1 THEN d1=1 ELSE d1=-1
460 IF sc/126=INT(SC/126) THEN 110
470 IF sc/63<>INT(SC/63) THEN 550
480 LOCATE x,y
490 PEN 0
500 PRINT CHR$(143);
510 FOR x=1 TO 300:NEXT
520 x=5
530 d1=1
540 IF y>15 THEN y=y-1 ELSE y=y+1
550 GOTO 200
560 '
570 '  move bat
580 '
590 j=JOY(0)
600 j$=BIN$(j,5)
610 i$=UPPER$(INKEY$)
620 IF i$="Q" OR MID$(j$,5,1)="1" THEN y1=y1-1:IF y1<2 THEN y1=2
630 IF i$="A" OR MID$(j$,4,1)="1" THEN y1=y1+1:IF y1>20 THEN y1=20
640 y2=4
650 IF x1=y1 THEN x2=4 ELSE x2=0
660 LOCATE 2,x1
670 PEN x2
680 PRINT CHR$(143)
690 LOCATE 2,x1+1
700 PRINT CHR$(143)
710 LOCATE 2,x1+2
720 PRINT CHR$(143)
```





```
730 LOCATE 2,y1
740 PEN y2
750 PRINT CHR$(143)
760 LOCATE 2,y1+1
770 PRINT CHR$(143)
780 LOCATE 2,y1+2
790 PRINT CHR$(143)
800 x1=y1
810 GOTO 200
820 '
830 '  game over
840 '
850 FOR i=1 TO 5
860 SOUND 3,4000,10,12,0,0,10
870 NEXT
880 PEN #1,3
890 LOCATE #1,4,2
900 PRINT#1,"Play again ?";
910 CLEAR INPUT
920 i$=UPPER$(INKEY$)
930 IF i$="" THEN 920
940 IF i$="N" THEN MODE 1:PEN 1:END
950 IF i$="Y" OR i$="X" THEN RUN ELSE 920
960 '
970 '  draw board
980 '
990 MODE 0
1000 INK 0,0
1010 INK 1,6
1020 INK 2,15
1030 INK 3,10
1040 INK 4,9
1050 INK 5,4
1060 INK 6,7
1070 PAPER 0
1080 BORDER 0
1090 GRAPHICS PEN 1
1100 PEN #1,5
1110 FOR X=1 TO 5
1120 FOR Y=1 TO 22
1130 S(X,Y)=0
1140 NEXT:NEXT
1150 PLOT 8,46
1160 DRAW 8,384
1170 DRAW 610,384
1180 DRAW 610,46
1190 DRAW 8,46
1200 WINDOW #1,1,20,24,25
1210 WINDOW #0,1,20,1,23
1220 PRINT#1,"      Score:"s
1230 y1=10
1240 x1=y1
1250 PEN 4
1260 LOCATE 2,y1
1270 PRINT CHR$(143)
1280 LOCATE 2,y1+1
1290 PRINT CHR$(143)
1300 LOCATE 2,y1+2
1310 PRINT CHR$(143)
1320 FOR i=2 TO 22 STEP 2
1330 PEN 2
1340 LOCATE 14,i
1350 PRINT CHR$(143)
1360 LOCATE 16,i
1370 PRINT CHR$(143)
1380 LOCATE 18,i
1390 PRINT CHR$(143)
1400 IF i=22 THEN 1470
1410 LOCATE 15,i+1
1420 PRINT CHR$(143)
1430 LOCATE 17,i+1
1440 PRINT CHR$(143)
1450 LOCATE 19,i+1
1460 PRINT CHR$(143)
1470 PEN 3
1480 LOCATE 15,i
1490 PRINT CHR$(143)
1500 LOCATE 17,i
1510 PRINT CHR$(143)
1520 LOCATE 17,i
1530 PRINT CHR$(143)
1540 IF i=22 THEN 1610
1550 LOCATE 14,i+1
1560 PRINT CHR$(143)
1570 LOCATE 16,i+1
1580 PRINT CHR$(143)
1590 LOCATE 18,i+1
1600 PRINT CHR$(143)
1610 NEXT
1620 RETURN
```



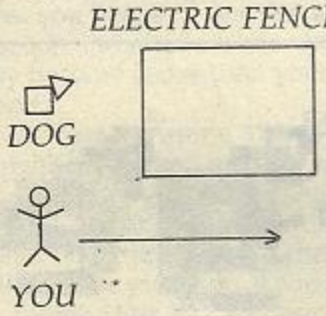
CHASE

This is quite an entertaining game which will pass the time quite quickly. Five wild dogs have been eating the chickens from your farm and you haven't been able to shoot them yet.

So you've set a trap for them. You did this by erecting an electric fence of death-high amperage, around a field, hoping to capture them in there. However, as you turn to walk out the gate and wait for the dogs, they walk in and the wind blows the gate shut... You're trapped in with them.

Fortunately, the dogs haven't eaten for a while and are sick from starvation. This means that they can't run very fast. The fastest they can run is equal to your speed and they can't suddenly stop. They have to slow down for quite a few meters before coming to a halt.

You have no weapons so the only way you can kill them is to somehow make them run into the fence. This is done as the diagram shows:



The dogs, when next to you, move in the same direction as you so as to create the least distance between you and them. Therefore, you can run them into the fence blocks.

There is a neat setup for the screen with the field in the middle and just to the left of it, a map of direction keys. These keys are the number function keys, or if you want more of a challenge, try the normal numbers.

If you answer 'no' to having another game at the end of the program you get a summary of the whole set of games you played. The summary includes how many games you won, how many games the computer won, your average and the computers average.

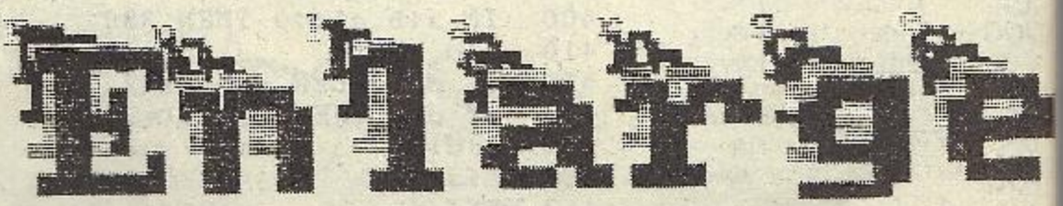
Experimenting may give you interesting results. For instance, change the amount of dogs from five to ten and then try to play. That's what I call challenging!

P. Vermeer
Elizabeth Downs SA

```
100 REM    ### CHASE ###
110 REM Paul Vermeer. Jan'86
120 '
130 DIM a(10,20),e(21),f(21)
140 MODE 1: INK 0,13: INK 1,3
150 ENV 1,=9,2000: ENT -1,6,3,1
160 ENV 2,127,0,0,127,0,0,127,0,0,127,0,
0,127,0,0
170 ENV 3,=9,9000
180 SOUND 1,1000,0,12,2
190 SOUND 2,900,0,12,2
200 LOCATE 1,6
210 PRINT " MOVES": PRINT
220 PRINT " 7 8 9": PRINT
230 PRINT " 4 "+CHR$(248)+" 6": PRINT
240 PRINT " 1 2 3"
250 WINDOW 8,40,1,25
260 PRINT: PRINT TAB(8);"CHASE"
270 REM set up the game
280 g=0:z7=0
290 FOR b=1 TO 10
300  FOR c=1 TO 20
310   a(b,c)=0
320   IF b=1 OR b=10 THEN a(b,c)=1
330   IF c=1 OR c=20 THEN a(b,c)=1
340  NEXT c
350 NEXT b
360 '
370 FOR d=1 TO 21
380  b=INT(RND*8)+2
390  c=INT(RND*18)+2
400  IF a(b,c)<>0 THEN 380
410  a(b,c)=1
420  IF d<6 THEN a(b,c)=2
430  IF d=6 THEN a(b,c)=3
440  e(d)=b
450  f(d)=c
460 NEXT d
470 '
480 REM print pattern
490 p$(0)=" ": p$(1)=CHR$(207)
500 p$(2)=CHR$(181): p$(3)=CHR$(248)
510 LOCATE 1,5: FOR b=1 TO 10
520  a$="":FOR c=1 TO 20
530 a$=a$+p$(a(b,c)): NEXT c
540  PRINT a$
550 NEXT b
560 '
570 REM make move
580 b=e(6)
590 c=f(6)
600 a(b,c)=0
610 y=VAL(INKEY$+" "): IF y=0 THEN y=5
620 ON y GOTO 650,650,650,660,720,660,63
0,630,630
630 b=b-1
640 GOTO 660
650 b=b+1
660 ON y GOTO 670,720,690,670,720,690,67
0,720,690
670 c=c-1
680 GOTO 720
690 c=c+1
700 '
```

```
710 REM Calculate the results
720 IF a(b,c)=1 THEN 1010
730 IF a(b,c)=2 THEN 1040
740 a(b,c)=3
750 e(6)=b
760 f(6)=c
770 FOR d=1 TO 5
780 IF a(e(d),f(d))<>2 THEN 940
790 a(e(d),f(d))=0
800 IF e(d)>=b THEN 820
810 e(d)=e(d)+1
820 IF e(d)<b THEN 840
830 e(d)=e(d)-1
840 IF f(d)>=c THEN 860
850 f(d)=f(d)+1
860 IF f(d)<=c THEN 880
870 f(d)=f(d)-1
880 IF a(e(d), f(d))=3 THEN 1040
890 IF a(e(d), f(d))=0 THEN 910
900 g=g+1
910 IF a(e(d), f(d))<>0 THEN 930
920 a(e(d),f(d))=2
930 IF g=5 THEN 1060
940 NEXT d
950 '
960 GOTO 510
1000 GOTO 1070
1010 GOSUB 1170: PRINT: PRINT "You touch
ed the fence."
1020 z9=z9+1
1030 GOTO 1070
1040 GOSUB 1170: PRINT: PRINT "You have
been destroyed."
1045 z9=z9+1
1050 GOTO 1070
1060 PRINT: PRINT "***YOU DESTROYED THE
ENEMY***"
1065 z8=z8+1
1070 PRINT "Want to play again (y or n)"
;
1080 INPUT y$
1090 IF y$="y" THEN 140
1100 PRINT "Computer won:";z9;"", "the h
uman won:";z8
1110 PRINT "Computers average:";z9/(z9+z
8): PRINT"The humans average:";z8/(z9+z8
)
1120 PRINT
1130 PRINT "Hope you dont feel fenced in
."
1140 PRINT "Try again sometime."
1150 WINDOW 1,40,1,25
1160 END
1170 SOUND 135,100,0,13,3,1,20
1180 RETURN
```

CHARACTER ENLARGER



Don't you get annoyed when you sit down to draw up graphics for enlarged characters, knowing its going to take you at least three hours, and a lot more if you have a lot of characters? Why, if you wanted to do the graphics for the entire alphabet — only 5 times larger, it would take about 15 hours.

Character Enlarger eradicates the need for this wasting of precious time, by doing it all itself in the space of a few seconds. You simply type in the letter, location and size (relative to the normal size), and it draws it up in which ever colour you specify.

In the adjoining program, the subroutine is the part which does the actual enlarging of the characters and the rest of the program is a demonstration. Therefore, you only need to type the subroutine into your programs.

The demonstration is very striking to look at. You type in the sentence you want and the size and sit back watching the program go to work in front of your eyes.

M. Kostecki & P. Vermeer
Elizabeth Park SA

```
10 ' ### Character Enlarger ###
20 ' Miroslav Kostecki  18.12.86
30 '
40 DEFINT a-z
50 DIM a(8,8)' to store a character
60 INPUT "How many times larger do you w
ant your sentance"; size
70 INPUT "What is your sentance"; sentan
ce$
80 '
90 MODE 1
100 INK 1,0: INK 2,6: INK 3,2
110 dots=16*size
120 FOR size=1 TO size
130 inc=size-INT(size/3)*3+1
140 jj=1
150 FOR dn=300-size*4 TO dots STEP -dots
160  FOR ac=size*4 TO 639-dots STEP dots
170   a$=MID$(sentance$,jj,1)
180   jj=jj+1: IF jj-2=LEN(sentance$) TH
EN 210
190   GOSUB 1000' put enlarged character
200 NEXT ac,dn
210 NEXT size
220 '
```

```
230 GOTO 230
240 '
1000 'Enlarger subroutene
1010 'takes variables ...size = times en
larged
1020 '                    ...ac = across ,
 dn = down position
1030 '                   ...inc = ink numb
er
1040 '                   ...a$ = characte
r
1050 '
1060 GRAPHICS PEN inc
1070 FOR x=0 TO 7
1080  FOR y=0 TO 7
1090   a(x,y)=TEST( ac+x*2 ,dn-y*2)
1100 NEXT y,x
1110 '
1120 TAG: MOVE ac,dn: PRINT a$;
1130 '
1140 FOR x=0 TO 7
1150 FOR y=0 TO 7
1160  t=TEST( ac+x*2 ,dn-y*2)
1170  PLOT  ac+x*2 ,dn-y*2,a(x,y)
1180  a(x,y)=t
1190 NEXT y,x
1200 '
1210 FOR x=0 TO 7
1220 FOR y=0 TO 7
1230  IF a(x,y)=0 THEN 1280
1240  FOR xx=0 TO size*2-2 STEP 2
1250   PLOT ac+x*2*size+xx,dn-y*2*size,a
(x,y)
1260   DRAW ac+x*2*size+xx,dn-y*2*size-(
size*2-2)
1270  NEXT xx
1280 NEXT y,x
1290 '
1300 RETURN
```

SPEED READER

This . . . uh . . . sentence says . . . um . . . this . . . is a . . . great . . . um . . . program — having trouble like this reading? No need to fear — this program gives you heaps of practice at reading, and after a while you may become faster than your friends.

Don't get me wrong. This is not a highly acclaimed medical program. It's not guaranteed. All I know is that it's improved the speed of my reading and that it will improve yours too, if you use it properly. Just half an hour every day should do the trick.

Now you probably don't even know what I'm babbling on about because I haven't even talked about what the program does yet. Don't worry, we'll now rectify that situation.

The screen is set up as in the adjoining picture. The sentence of your choice moves from the right hand side of the box to the left very quickly. You're then asked to type in what that sentence is. If you're wrong, the sentence repeats the sequence but at a slower pace. This routine continues until you get it right. When you do get it right, the computer tells you how many 'specks' it took you. Specks are a measurement used by me, the programmer.

The best thing to do is get someone else to type in a sentence for you without your seeing. This means that you don't know what it is and you can't cheat. Or, by slightly modifying the program you can make it easier. By adding some data in the form of sentences the computer can randomly pick out a sentence to test you on.

Anyway, I hope you succeed as reading can be the greatest enjoyment in the world.

Paul Vermeer
Elizabeth Park SA

```
100 REM ##### SPEED READER #####
110 REM # Paul Vermeer. Jan'87 #
120 '
130 RANDOMIZE TIME/255 MOD 999
140 MODE 2
150 WINDOW 20,80,1,25
160 LOCATE 9,3: PRINT "SPEED READER"
170 LOCATE 9,4: PRINT STRING$(12,208)
190 INK 0,13: INK 1,0
200 a$="This is a sentence for your read
ing practice!!"
210 b$=STRING$(10," ")
220 '
230 'Random delay
240 MOVE 0,0,s: DRAW 639,0: DRAW 639,398
250 DRAW 0,398: DRAW 0,0: s=s MOD 2+1
260 IF RND<0.98 THEN 240
270 '
280 'border
290 LOCATE 8,9: PRINT CHR$(150)+STRING$(
12,154)+CHR$(156)
300 LOCATE 8,10: PRINT CHR$(149)+STRING$
(12," ")+CHR$(149)
310 LOCATE 8,11: PRINT CHR$(147)+STRING$
(12,154)+CHR$(153)
320 c$=b$+a$+b$
330 GOSUB 490
340 '
350 'scan the sentance across the window
360 FOR i=1 TO LEN(a$)+13 STEP 3+(sp>10)
+(sp>5)
370 LOCATE 10,10: PRINT MID$(c$,i,10);
380 FOR j=1 TO sp: NEXT j'delay
390 NEXT i
400 '
410 PRINT: PRINT: PRINT
420 INPUT "   What did it say";an$
430 IF an$<>a$ THEN PRINT: PRINT "   NO
Try again...a little slower this time.":
 sp=sp*1.1+2: FOR i=1 TO 1000: NEXT i: G
OTO 130
440 PRINT "   THATS RIGHT"+STRING$(30,"
")
450 PRINT "   Your speed was ";sp;" spec
ks!"
460 PRINT "   Next time I expect you to
do it in less specks!"
470 END
480 '
490 ENV 1,=9,2000: ENT -1,6,3,1
500 ENV 2,127,0,0,127,0,0,127,0,0,127,0,
0,127,0,0
510 ENV 3,=9,9000
520 SOUND 135,100,0,13,3,1,20
530 RETURN
```