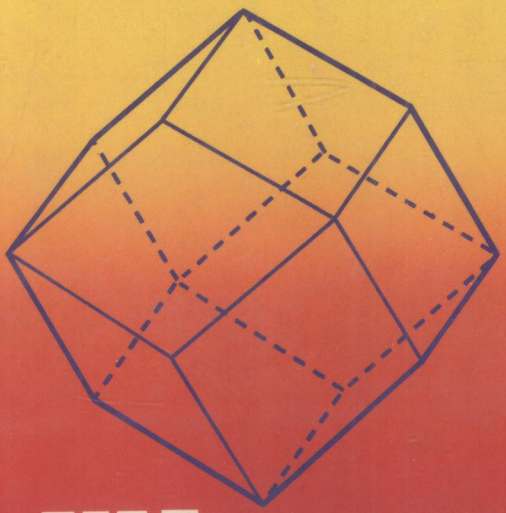
Fifty Subroutines for the Sinclair Spectrum



W. Johnson

Sigma Technical Press

Fifty Subroutines for the Sinclair Spectrum

W. Johnson

Sigma Technical Press

ISBN 0 905104 97 8

Published by:

SIGMA TECHNICAL PRESS,
5 Alton Road,
Wilmslow,
Cheshire,
UK.

Typesetting and Production by:

DESIGNED PUBLICATIONS LTD.
8-10 Trafford Road,
Alderley Edge,
Cheshire.

Distributors:

Europe, Africa:
JOHN WILEY & SONS LIMITED,
Baffins, Lane, Chichester,
West Sussex, England.

Australia New Zealand, South-East Asia
Jacaranda-Wiley Ltd., Jacaranda Press,
JOHN WILEY & SONS INC.,
GPO Box 859, Brisbane,
Queensland 40001, Australia.

Printed and bound in Great Britain by
J. W. Arrowsmith Ltd., Bristol

PREFACE

This collection of subroutines has been compiled for publication from a set of programs developed for the ZX Spectrum. It should be of interest to other Spectrum owners because it contains useful reference routines, as well as routines which are hard to find or tedious to write.

The subroutines are worth studying in detail for the techniques used, and should be of benifit in developing your own programs. How often have you said "I didn't realise you could do it that way",or "I don't remember seeing that in the manual!"

Part of the fun of computing, after the novelty of playing games has worn off, is to develop your own games or useful programs, and make the computer do what you want with the minimum number of instructions and memory requirements. There is great satisfaction in producing an elegant solution to a problem.

The subroutines are written in Spectrum BASIC, but can easily be translated into other dialects of BASIC with a little care and a bit of trial and error.

Every effort has been made to ensure that the subroutines work over the ranges specified, and in an efficient way, but there are no prizes for finding cases where they do not work!

Finally, there is a complete program at the end, for which at least a dozen of the subroutines were originally developed. To fit the program comfortably into a 16K Spectrum some alternative routines are used and the program is a good example of memory saving techniques. The program enables all the basic shapes of cubic crystals to be drawn, by just inserting three numbers which represent the appropriate crystal form.

CONTENTS

14-17 Checking Data Input

14 Data Input (Linear equations with up to eight variables)

15 Data Input (Matrices or arrays)

16 Data Input (Single variable)

17 Data Input (x and y co-ordinates, statistical data etc)

18 Display File

19 Double Size Printing

20 Drawing lines between two points

21 Evaluation of a Determinant

22 Factorial n

23 Heaviside Operator

24-27 Loops

24 Split loop

25 Mixed loop

26 Random loop

27 Multiple choice loop

28 Matrix Inversion

29 Matrix Multiplication

30 Minimum, Maximum, Mean, Median and Mode

31 Name Filter

32 Permutations of three numbers

33 Postwar Inflation '

34 Prime Numbers

35 Printout for Matrix or Determinant

36 Projection

SUBROUTINE LAYOUT

Each subroutine is in two or three parts. The initial part separated from the subroutine itself, will normally be in the main part of the program, and generates the essential information needed by the subroutine.

However, to run the subroutine as a self-contained program, or to test it when entered in the computer, a simple input routine is given.

The main body of instruction is the subroutine itself. It is written without line numbers so that it can be entered to suit the main program. Jump destinations are indicated by letters. Where an instruction takes more than one line of typing, the second and subsequent lines are indented though it is usually quite obvious where a new line number is required. Most lines contain several instructions to save space.

Finally, an output routine is appended to the subroutine to output the end result of the subroutine, either to check it on entering it in the computer or, if the subroutine is used as a free-standing routine to give or display the result.

R is used for the main subroutine RETURN and for the beginning of the final section.

Care should be taken that the variables used in the subroutines (mainly i,j,k,p,q,n,t, and z for single letter variables), do not conflict with the variables used in the main program.

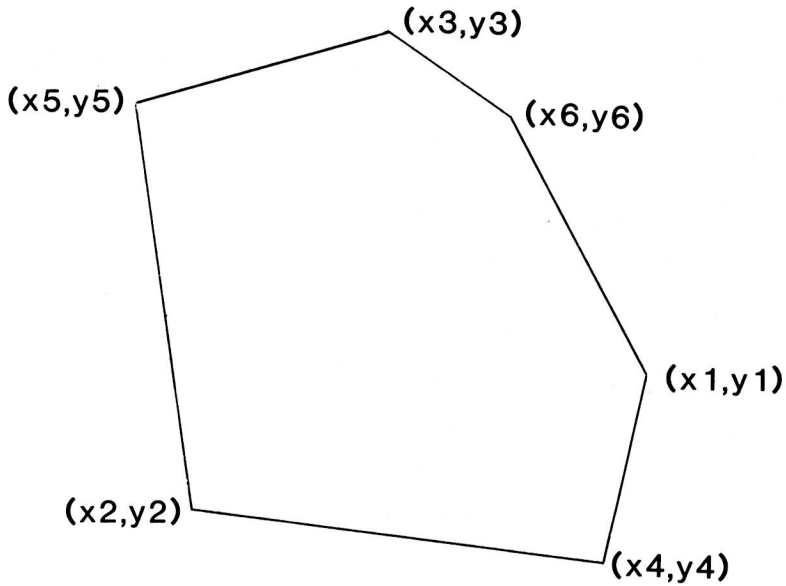
A brief explanation of how the subroutine works is given for each one.

1 ANGLESORT

This subroutine puts a list of coordinates into angular order relative to their centre.

The first eight lines find the highest and lowest values of x and y and the point xm, ym is chosen to be halfway in between . The angle between the horizontal and the line joining this point to each of the original ones, is calculated and inspected to see which quadrant it is in. This is because a tangent is positive in the first and third quadrants, and negative in the other two . The angle is adjusted by adding PI or 2 * PI as necessary and stored in B(n). A bubblesort subroutine, B, is used to put the angles in order.

```
    INPUT "Number of pairs of readings ";n: DIM A(2,n):FOR p = 1 TO n : INPUT
    "x";(p);"= ";A(1,p),"y";(p);"= "; A(2,p)
    NEXT p
    or use Data Input routine to establish n and A(2,n)
    LET x max = 0 : LET y max = 0
    LET x min = A(1,1) : LET y min = A(2,1)
    FOR p=1 to n
    IF A(1,p)>=x max THEN LET x max = A(1,p)
    IF A(2,p)>=y max THEN LET y max = A(2,p)
    IF A(1,p)< x min THEN LET x min = A(1,p)
    IF A(2,p) <y min THEN LET y min = A(2,p)
    NEXT p
    LET xm = (x max + x min)/2: LET ym = (y max + y min)/2
    DIM B(n) : FOR p=1 TO n
    LET z = ATN ((A(2,p)-ym)/(A(1,p)-xm))
    IF A(1,p)>= xm AND A(2,p)>= ym THEN LET B(p) = z
    IF A(1,p)<xm AND A(2,p)> = ym OR A(1,p)<xm AND A(2,p)<ym THEN
    LET B(p) = PI + z
    IF A(1,p)>= xm AND A(2,p)<ym THEN LET B(p) = 2*PI+z
    NEXT p
B   LET q=0 : FOR p=1 TO n-1
    IF B(p+1)<B(p) THEN GO SUB A : LET q=q+1
    NEXT p: IF q<>0 THEN GO TO B
    GO TO R
A   LET z = B(p+1): LET B(p+1)=B(p): LET B(p)=z
    LET z = A(1,p+1): LET A(1,p+1) = A(1,p): LET A(1,p)=z
    LET z = A(2,p+1): LET A(2,p+1) = A(2,p): LET A(2,p)=z
    RETURN
R   RETURN
R   PLOT A(1,1),A(2,1):REM IF 0<=x min< =255,0<=y min<= 175 etc
    FOR p=1 TO n-1
    DRAW A(1,p+1)-A(1,p),A(2,p+1)-A(2,p) : NEXT p
    DRAW A(1,1)-A(1,n),A(2,1)-A(2,n)
```



Anglesort

2 ANNUITIES CERTAIN

These subroutines work out the annuity certain tables £ $A\overline{n}$ for different annual interest rates and for annual or monthly payments.

Financial transactions based on compound interest have the geometric series underlying them. If i is the interest rate, then £1 will become £$(1+i)^n$ in n years' time so that, turning it over, £$1/(1+i)^n$ will become £1 in n years' time. The annuity certain is the sum of the present values $1/(1+i)^n$ for each of the years to come. Hence

$A\overline{n}= v + v^2 + v^3 + \ldots\ldots v^n$

where $v=1/(1+i)$

This series equals $(1-v^n)/i$ which is used to calculate the table shown.

(a) Annual payments

```
INPUT "Annual interest rate as %  ";i : DIM A (30)

FOR n=1 TO 30 : LET A(n) = (1-(100/(100+i))↑n)/i*100: NEXT n

FOR n=1 TO 30 : PRINT TAB 5;n; TAB 10;A(n) : NEXT n
```

(b) Monthly Payments

```
INPUT "Annual interest rate as% ";i: LET i=i/12

DIM A(30) : FOR n=1 TO 30 : LET A(n) =
                    (1-(100/(100+i))↑(n*12))/i*100 : NEXT n

FOR n=1 TO 30 : PRINT TAB 5;n;TAB 10;A(n): NEXT n
```

The annuity certain is the initial sum which at x% per annum yields £1 per annum over n years

So for example, if you have £25,000 to invest at 8%pa and want to draw 15 equal annual instalments, then the annuity certain for 15 years at 8% is £8.5595 and £25,000/8.5595=£2,920.74 is the annual instalment. After 15 years the money is all used up, as you have been drawing capital and interest.

The reverse example is paying off loans , e.g. a mortgage, so that the monthly payments are constant, i.e. initially you pay mainly interest charges, but gradually pay more capital back.

To borrow £20,000 over 25 years gives, at say 9%, an annuity certain calculated on a monthly basis of £119.16162 making the monthly repayments £20,000/119.16162 = £167.84

Monthly Repayments

9%

Years	Annuity Certain
10	78.94169
11	83.60642
12	87.87109
13	91.77002
14	95.33457
15	98.59341
16	101.57277
17	104.29661
18	106.78686
19	109.06353
20	111.14495
21	113.04787
22	114.78759
23	116.37811
24	117.83222
25	119.16162
26	120.37701
27	121.48817
28	122.50404
29	123.43278
30	124.28187

Annual Instalments

8%

Years	Annuity Certain
5	3.99271
6	4.62288
7	5.20637
8	5.74664
9	6.24689
10	6.71008
11	7.13896
12	7.53608
13	7.90378
14	8.24424
15	8.55948
16	8.85137
17	9.12164
18	9.37189
19	9.60360
20	9.81815