THIS program produces a graph of a function (expressible in
the form y = f(x) using standard BASIC notation) for a range
of x values. It is self-scaling once it has been given the range of x,
finding the lowest and highest y values. It also draws the lines
x=0, and y=0, when they fall within the x or y values.

METHOD
On the UK101, reasonable resolution can be obtained using
the graphics characters. The screen is 46 by 16 chars, but the
graphics include 8 horizontal bars.

These bars can increase the resolution to 46 by 128 allowing a
reasonable curve to be drawn.

The first problem encountered was how to change the function
of x every time the program is run. The simplest answer is
this:

100 PRINT " Please type : 
110 PRINT " 5000 Y=f(x) "
120 PRINT " GOTO 1234 "
130 STOP
1234 program

This is not ideal. The best answer is to make the program alter itself. Microsoft BASIC is memory efficient for the reason
that the commands are abbreviated by the use of tokens.

What does a Basic line look like in memory? Consider this line:

10 Y=X

If the relevant section of memory is examined, the line is
stored as follows:

14 3 10 0 89 171 88 0

The “14 3” in the first and second byte means the next
BASIC line is stored at memory location 14 + 3*256 (=782
decimal). The “10 0” in the next two bytes indicates that this is
BASIC line number 10 + 0*256 (=10 decimal). 89 is the
ASCII code for Y, and 88 for X. So somehow 171 means “=”,
and 0 means the end of the line.

So far:

<table>
<thead>
<tr>
<th>14</th>
<th>3</th>
<th>10</th>
<th>0</th>
<th>89</th>
<th>171</th>
<th>88</th>
<th>0</th>
</tr>
</thead>
</table>

782 line 10 Y = X END of line

A full list of tokens is given in Table 2 only those underlined
are useful for the function of x.

### TABLE 2: Tokens. Those underlined are used

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>128 END</td>
<td>151 PRINT</td>
<td>174 INT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>129 FOR</td>
<td>152 CONT</td>
<td>175 ABS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>130 NEXT</td>
<td>153 LIST</td>
<td>176 USR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>131 DATA</td>
<td>154 CLEAR</td>
<td>177 FRE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>132 INPUT</td>
<td>155 NEW</td>
<td>178 POS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>133 DIM</td>
<td>156 TAB</td>
<td>179 SQR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134 READ</td>
<td>157 TO</td>
<td>180 RND</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>135 LET</td>
<td>158 FN</td>
<td>181 LOG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>136 GOTO</td>
<td>159 SPC</td>
<td>182 EXP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>137 RUN</td>
<td>160 THEN</td>
<td>183 COS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>138 IF</td>
<td>161 NOT</td>
<td>184 SIN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>139 RESTORE</td>
<td>162 STEP</td>
<td>185 TAN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>140 GOSUB</td>
<td>163 +</td>
<td>186 ATN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>141 RETURN</td>
<td>164 —</td>
<td>187 PEEK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>142 REM</td>
<td>165 *</td>
<td>188 LEN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>143 STOP</td>
<td>166 /</td>
<td>189 STR$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>144 ON</td>
<td>167 ↑</td>
<td>190 VAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>145 NULL</td>
<td>168 AND</td>
<td>191 ASC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>146 WAIT</td>
<td>169 OR</td>
<td>192 CHR$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>147 LOAD</td>
<td>170 =</td>
<td>193 LEFT$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>148 SAVE</td>
<td>171 &gt;</td>
<td>194 RIGHTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>149 DEF</td>
<td>172 &lt;</td>
<td>195 MID$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>150 POKE</td>
<td>173 SGN</td>
<td>197 to 211 BASIC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Thus if we input the function of x, we can find the suitable line
in memory and poke into it the function. The line which we look
for is line 5000:

5000 5000 Y=f(x) REM 5010 RETURN

The 5 symbols are looked for in memory. When they are
found the function y=f(x) is poked in (the function must be in
standard BASIC notation). The most convenient way to end a
line is to make it a multi-line statement, so the colon and REM
are both POKEd in at the end of the line. We now have the
function of our graph in the form:

5000 y=f(x) : REM 5010 RETURN

Values of the range of x are then inputted, and the highest and
lowest values of y are found for scaling purposes. The graph can
then be plotted using the graphic characters.

SOFTWARE IDEAL FOR MATHEMATICS
DEMONSTRATION.

Left: Graph of \( y = 1/x^2 \). Note that the
function must be keyed into the computer as if a BASIC statement in
itself.

Right: A more complex, cyclic func-

Some suggested graphs
1) \( y=\text{ABS}(x) \) for \( x=-1 \) to \( 1 \)
2) \( y=-\text{ABS}(x) \) for \( x=-1 \) to \( 1 \)
3) \( y=\sin(x/2) + \sin(2\pi x) \) for \( x=0 \) to \( 2\pi \)
4) \( y=1+(x^2) \) for \( x=1 \) to \( 3 \)
850 A$=STR$(HI); A$=A$+26A$=LEN(A$); GOSUB6000
860 POKEx59,11; POKE527088,0
870 IPEEK(527088)=5247H 890
880 POKE330,0; FORA=10; POKE194; NEXT
890 RUN
1000 2=424255; S1=INT(1(Y-TL(1)))*7+5
1010 POKE56; INT(56+126+11; NEXT
2000 A=53262A=54A=54(A=56)+6(A=56)+6(A=56)
2010 POKE4,46
2020 FORA=10 TO 15
2030 POKE+(A$)+46,143
2040 NEXT; GOTO740
3000 A=54220
3010 A=1; INT(A$)+S1/(Y-TL)+5)*64
3020 POKE4,48
3030 A=+2
3040 POKE=6; GOTO65
3050 IPEEK(A$)=143; NEXT; POKEA+B,208; GOTO3070
3060 POKE4,128
3070 NEXT; GOTO150
5000 """"
5010 RETURN
6000 FORA=10 TO LEN(A$)
6010 POKEA=A$; ASC(MID$(A$); a; 1))
6020 NEXT; RETURN
OK

LINES

100 to 160 Inputs function
170 to 540 Finds line 5000 in memory and poker the function of x in
550 to 650 Inputs the low and high values of x and from these calculates the maximum and minimum y values
570 to 850 Plots graph
860 to 890 Waits for any key to be pressed and then returns line 5000 to its original format
1000 to 1050 Plots the horizontal bar
2000 to 2040 Plots the line X=0 [if applicable]
3000 to 3070 Plots the line Y=0 [if applicable]
5000 to 5010 Function of graph in form Y=f(X)
6000 to 6010 Pokes the contents of A$ into a part of the screen pointed to by the contents of A

IMPORTANT VARIABLES

HI highest value of X
LO lowest value of X
LP position of line 5000 in memory
YH highest value of Y
YL lowest value of Y

NOTES
1) line 5000 should be retyper if the program is stopped halfway through
2) the function of X must be in standard Basic notation. Any error will mean that you will have to retype line 5000, and re-run the program.
3) all angles are in radians.
4) do not try to plot infinity [eg tan of a half of pi]