CHART TOPPER

Our series of articles on the LOGO language has concentrated on developing routines and procedures for handling data and producing turtle graphics displays. Here we use the language to display data in an easily-understandable form, and we present a routine for the construction of barcharts.

Barcharts are a simple graphical method of representing certain types of numerical data. The aim of a barchart is to help the reader to understand at a glance a set of figures without having to examine them in great detail. Business graphics programs are widely available for plotting barcharts, pie charts, and histograms, and these programs are often linked to spreadsheets so that the values calculated in these can be effectively displayed.

We won't be quite so ambitious for the moment, but let's start by looking at how to draw a barchart using LOGO. The version of the program we will discuss is for the Commodore 64 — see 'Logo Flavours' for any changes that are needed for the program to run on other machines.

The process of drawing a barchart splits up into three stages:

a) get the input;
b) find the largest value (this is necessary so that we can scale the columns to fit on the screen);
c) draw out the barchart, scaling as appropriate, and then add labels.

The top-level procedure is called BARCHART. INIT sets the value of a number of constants needed by the program. By collecting them together in one place, modifications are easier to make. COLORS contains the list of colours to be used for the columns — if you do not like our colour scheme, change it! The program as it stands will print a maximum of 15 columns, so you will need only 15 colours at most.

INPUT AND CALCULATION

The necessary input consists of two data items for each bar of the chart: first, the title to be printed at the foot of the bar, and, second, the quantity or value of the bar — effectively, its height. In our input routines we have incorporated some simple 'validation' checks to make sure that the input is sensible. GET.INPUT splits the job of obtaining input into two parts, getting titles for each column and inserting the relevant values. When you've finished inputting the data, type END as the next 'title'.

GET.TITLE gets the title; this will reject a blank entry but will accept any other value. GET.QUANTITY will accept only a number as an input — any other input will cause the program to ask for the data to be entered again. When GET.INPUT has a valid name/number pair, these are added to the end of a list of data items. Items should be entered in the order you wish to see them plotted across the screen, from left to right.

CALCULATE uses GET.MAX to find the largest value and then uses this to establish a scale. A common rule is that the height of a barchart should be about three-quarters of the width.

DRAWING THE CHART

DRAW.CHART calculates the width of each bar of the barchart, draws an axis up the screen, colours in the bars and then labels them.

The width is calculated as a multiple of eight. This is done in order to avoid some problems caused by the way the Commodore 64 displays colours, for in the normal LOGO graphics mode you can't have two colours in the same eight by eight pixel block. DRAW.AXIS draws the line up the screen and marks on it the highest of the data values; this gives us a simple means of estimating the values of the columns.

We must step back from the axis line before printing this number next to the mask indicating the highest value. How far we need to step back so that the number doesn't overprint the line will depend on how many digits there are in the number. This problem is easily solved, for LOGO treats numbers as if they were words, so we can use COUNT to determine the length of the number, and then use this to determine how far to step back.

WRITE prints a message on the graphics screen. This takes three inputs: the x and y step distances for each character and the name to be written. It uses a primitive, STAMPCHAR, which prints a character on the graphics screen at the turtle's position.

DRAW.CHART1 performs the task of drawing the bars. It takes each item in turn, selects the next colour to be used, scales the height and passes the real work over to FILL, which simply goes up and down the bar filling it in. LABEL uses WRITE to write the labels vertically down the screen (very long labels will 'wrap around' and appear at the top of the screen).

PIE CHARTS

Pie charts are another common form of graphical representation. If you want to write a program to draw a pie chart, here's a few hints:

•Data is obtained in exactly the same way as for the barchart.