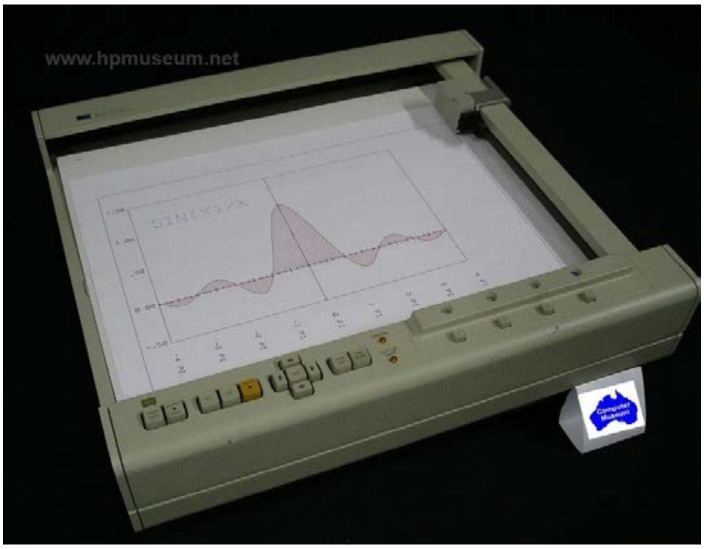
15 HP-GL plotter controller

14 points

Special Award

Introduction

With the launch of the Hewlett Packard 9872A flatbed plotter in 1977, HP began naming its 2-letter mnemonic graphics language as HP-GL (Hewlett Packard Graphics Language). Nowadays, modern printers still implementing HP-GL.



HP-GL is a command set embedded in the ROM of pen plotters to help reduce the work required by applications programmers to create plotted output. HP-GL uses two-letter mnemonics as instructions for drawing lines, circles, text, and simple symbols.

Program specification

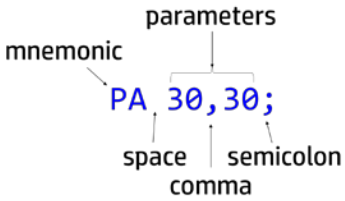
Your program should read an list of simplified HP-GL commands, interpret the commands, and translate them to simple instructions to control an X-Y table.
These output instructions are the movements on the X and Y axis, and a command to raise or lower the pen.

Input: A list of HP-GL commands (one per line), ending with a #
Output: The translated list of commands to the X-Y table (one per line), ending with a #

HP-GL reference guide (reduced)

For simplification, we'll use a reduced set of instructions.
The whole reference guide is not needed for this exercise, but in case of interest, you can find it at:
www.hpmuseum.net/document.php?catfile=213

HP-GL commands have four components: a mnemonic, parameters, separators, and a terminator. Refer to the following illustration of a typical HP-GL command and the description of its components.



Mnemonic--The two-character mnemonic defines the command's function. The mnemonic will be uppercase.
Parameters are used in certain instructions to tell the device to complete the instruction in a particular way.
Separators are used to separate one parameter from the next. They will be a comma.
Terminator separates one instruction from the next. They will be a semicolon.

For this exercise, we'll only consider the next commands:

CO	Comment
PD	Pen Down
PU	Pen Up
PA	Plot Absolute

CO, Comment

CO *["c...c"];*

Allows comments to be inserted within an HP-GL instruction sequence. HP-GL comments are ignored by the device.

PA, Plot Absolute

PA *[X,Y [,X,Y [...]]] ;*

Establishes absolute plotting (vs relative plotting) and moves the pen to the specified coordinates.

- No parameters: Establishes absolute plotting for subsequent instructions.
- X,Y coordinates: Specify the absolute location to which the pen moves. When you include more than one coordinate pair, the pen moves to each point in the order given, using the current pen up/down status. If the pen is up, PA moves the pen to the point, if the pen is down, PA draws a line to the point.

PD, Pen Down

PD *[X,Y [,X,Y [...]]] ;*

This instruction lowers the pen to draw lines on the page.

- No parameters: Lower the pen.
- X,Y coordinates: Lower the pen and draws to the points specified. You can specify as many X,Y coordinate pairs as you want. When you include more than one coordinate pair, the device draws to each point in the order given.

Example

```
PA    10,10;                     Begin absolute plotting from coordinate (10,10).
PD    2500,10,10,1500,10,10;   Set the Pen Down and draw lines between the specified points.
```

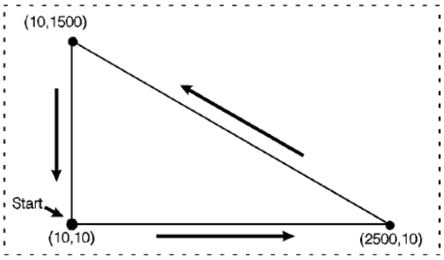


Figure 119. Using the PD (Pen Down) Instruction

PU, Pen Up

PU *[X,Y [,X,Y [...]]] ;*

This instruction raises the pen to prevent drawing lines on the page.

- No parameters: Raises the pen.
- X,Y coordinates: Raises the pen and moves to the points specified. You can specify as many X,Y coordinate pairs as you want. When you include more than one coordinate pair, the device moves to each point in the order given.

X-Y table reference guide

Our X-Y table has two steeper motors, for the X and Y axis, and one actuator to lower/raise the pen. Your program must translate from the HP-GL commands, to these ones.
The set of instructions that takes is very simple:

VECTOR GROUP	
MA	Move absolute
PD	Pen Down
PU	Pen Up

MA, Move Absolute

Purpose

MA *X,Y ;*

Each instruction has one, and only one, pair of coordinates.
This instruction moves the pen to the X,Y coordinates using the current pen up/down status. If the pen in up, moves the pen to the point, if the pen is down, draws a line to the point

PD, Pen Down

PD *;*

Lowers the device's pen.

PU, Pen Up

PU *;*

Raises the device's pen.

Example

The input is:

```
CO “Triangle”;
PU;
PU;
PA 10,10;
PD 2500,10,10,1500,10,10;
#
```

The output is the list of commands to be send to the X-Y table:

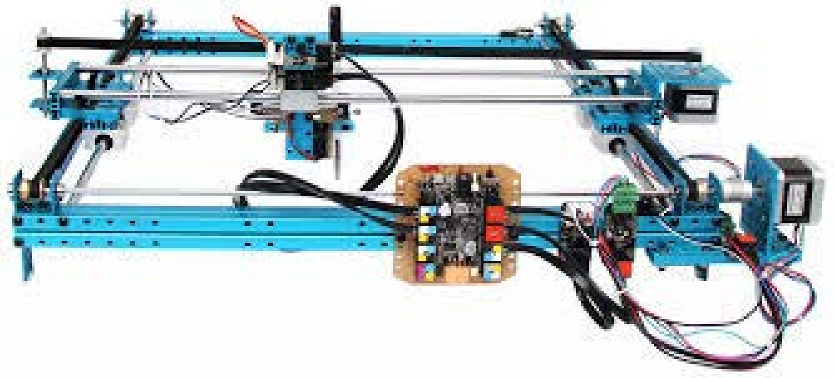
```
PU;
PU;
MA 10,10;
PD;
MA 2500,10;
MA 10,1500;
MA 10,10;
#
```

Surprise output....!!!!

Be aware!!! If you solve this exercise, you will see the connection of your PC's virtual world, with the physical one where we live ;-)

Programming is not just a serial of commands that nerds use to display some game hero in your PC screen: as in old HP's plotters, nowadays, programming is the core of many domestic to industrial machines; the bridge of many devices with our environment.
From the controller of one of the blinds at your home (with end of travel, obstacle detection, timer programming...), to the ABS algorithm of your car, there are many applications where you can find software modules, interacting with physical components.

Each time a team submits a correct solution to the problem, your fresh written code is going to be used to send a plot to the X-Y table with your team id and the order of submittal. Take this print back at home with you !



Do you want to have a closer look to the X-Y table? the code inside? send your own HP-GL file?....feel free to come to the table after the competition.