Springer Code Problem

Introduction

Below is a coding problem that we would like you to solve. Please read through the description and implement a solution for it in any programming language. Candidates for permanent roles should pick a language you're comfortable in, rather than one you think we might like. Candidates for contracts should use one of our required languages.

When submitting your solution, imagine that you've never seen it before and you've been asked to fix something. What would you like to see when you start the job? As part of our interview process, we might ask you to extend the functionality.

Please create a **local repository** using either **git or mercurial** and then commit locally. When you have finished please zip up the whole folder (including .git or .hg folders) and email to us (springer.code@gmail.com) within **7 days**. We will then review it within 7 days as well.

Please do not make either our problem or your solution public, thanks.

Good Luck!

Things we value

- Working software!
- Tests
- A working build
- Small checkins with descriptive comments
- A simple README (maybe talk about trade offs and design decisions)
- Code with good naming and clear intention.
- The problem is small enough that we expect it not to need libraries for the production code. If you want to use a library then please say why in the README.
- Evidence you have thought about errors. If the problem description isn't clear, then make a choice and note it.

Things to expect

- If you get to the next stage we will work with you on your code to change the design and add more features.
- Be prepared to talk about your code and technical choices. We will want to discuss topics such as alternative design decisions and error handling.

The Problem

Description

You're given the task of writing a simple console version of a drawing program. At this time, the functionality of the program is quite limited but this might change in the future. In a nutshell, the program should work as follows:

1. create a new canvas
2. start drawing on the canvas by issuing various commands
3. quit

At the moment, the program should support the following commands:

Command	Description
C w h	Should create a new canvas of width w and height h.
L x1 y1 x2 y2	Should create a new line from (x1,y1) to (x2,y2). Currently only horizontal or vertical lines are supported. Horizontal and vertical lines will be drawn using the 'x' character.
R x1 y1 x2 y2	Should create a new rectangle, whose upper left corner is (x1,y1) and lower right corner is (x2,y2). Horizontal and vertical lines will be drawn using the 'x' character.
B x y c	Should fill the entire area connected to (x,y) with "colour" c. The behaviour of this is the same as that of the "bucket fill" tool in paint programs.
Q	Should quit the program.

Sample I/O

Below is a sample run of the program. User input is prefixed with enter command:.

```
enter command: C 20 4
----------------------
|                    |
|                    |
|                    |
|                    |
----------------------

enter command: L 1 2 6 2
----------------------
```

```
|                     |
|xxxxxx               |
|                     |
|                     |
----------------------

enter command: L 6 3 6 4
----------------------
|                     |
|xxxxxx               |
|     x               |
|     x               |
----------------------

enter command: R 16 1 20 3
----------------------
|               xxxxx|
|xxxxxx         x   x|
|     x         xxxxx|
|     x               |
----------------------

enter command: B 10 3 o
----------------------
|ooooooooooooooxxxxx|
|xxxxxxooooooooox   x|
|     xooooooooxxxxx|
|     xooooooooooooo|
----------------------

enter command: Q
```