Springer Code Problem

Below is a coding problem that we would like you to solve. Please read through the description carefully and implement a solution for it. You're free to write your solution in any programming language. We don't want you to over-engineer the solution but be prepared to extend the functionality in the next step of the interview process. Finally, we ask you to submit a solution that you'd be happy to go live with and works "out of the box".

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 good comments
- A simple read me (maybe talk about trade offs and design decisions)
- Simple code (but not necessarily easy!)
- The less libraries the better, we want to see **your** code but if you want to use X then say why in your readme.
- We like functional constructs but also value good domain names and modelling
- Evidence you have thought about errors (either in code or the readme)

Things to expect

- If you get to the next stage we will pair on your code
- We will add some more features
- Maybe refactor some things
- Be prepared to talk about your code and/or language choice
- Understand alternatives of design decisions
- Discussions around input and error handling

Enough talk, 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
```