Developer Exercise

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

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
--------------------
|oooooooooooooxxxxx|
|xxxxxxooooooox   x|
|     xooooooooxxxxx|
|     xooooooooooooo|
--------------------

enter command: Q
```

Requirements

1. Develop the console drawing program as described above, using any one of the following programming languages:
   a. Python
   b. JavaScript

      c. TypeScript
      d. Java
2. Include any unit tests required to prove that it works correctly
3. Upload the completed code, including tests and a README.md file to describe how it should be used, to a **private repository** on GitHub, which only you have access to.
4. In the private repository you have uploaded the code to, invite the following user as a collaborator: grahamdaley

Submissions will be graded based on quality of the code (readability, structure, naming conventions, efficiency, etc), tests and documentation.