

CS1083 Assignment #8 – Winter 2021

Due: Friday, March 19th before 4:00pm (Atlantic), submitted in the Assignment 8 dropbox in Desire2Learn. (Read the submission instructions at the end of this document carefully).

The purpose of this assignment is to allow you to gain more practice with recursion.

This assignment is to be done individually. If you have questions, direct them to a tutor/assistant during a help session in the "Faculty of Computer Science Student Success Centre" team or to your course instructor.

Room Painting

Consider a 2D space divided into regions by "walls" (like on the plan for a house with many rooms; all doors are closed and therefore look like walls). This 2D space is represented by a rectangular array of characters, with zeros (i.e. '0') marking the open spaces (floors of rooms, porches, etc.), and ones (i.e. '1') marking the "walls". Open region locations connect up/down and left/right, but not on the diagonal. The objective of this project is to colour (paint) the floors. In each open region you may find one of the open spaces marked with a character indicating a color, 'R' for red, 'B' for blue, 'G' for green, etc. In general, a symbol other than '0' or '1' indicates a colour and represents the original starting location for the colouring process. (You can imagine that this is the location of the paint bucket and you start painting the floor at this location. If no paint bucket exists in a room, that room's floor will remain unpainted.)

Write a Java program that reads the array dimensions and then the array representing a 2D space divided into regions. The program will then colour all open space locations adjacent to any symbol representing a colour using this colour. When colouring a region, start from the original location of the paint bucket and then colour the locations adjacent to it; then you will repeat this process (i.e. colour the adjacent locations then locations adjacent to already coloured locations and so on). Obviously, colouring with a particular colour stops at the walls. The program's output should include the original array, followed by the array with every open space location in each region coloured in the corresponding colour.

Design your program using 2 classes described here:

1. Painter.java

- Constructor: passed the filename, it creates a scanner object, reads the array dimensions and then the 2D array. The 2D array is stored as an instance variable.
- `paint()`: public method that iteratively finds the locations of the paint cans in the array and calls the recursive paint method to paint the floor of the room.
- `paint (int row, int col, char colour)`: private method that **recursively** paints the floor of the room.
- You may also choose to create other private helper methods.
- Override the toString method so it prints the contents of the array.

2. PaintDriver.java

- Creates an instance of the Painter using the name of the file passed in on the command line.
- Prints the array before painting the rooms.
- Paints the rooms in the array by calling the paint method.
- Prints the array after painting the rooms.

You may assume that there will be at most one paint bucket per room. (That is, no more than one letter character will be present within an open space.)

Note: You must use recursion to colour (paint) the 2D space. However, you may use loops (if you wish) for other things (e.g. reading in the data, displaying the original (unpainted) array, and displaying the final (painted) array.)

Testing:

Create 3 test cases. Think about different scenarios when making up your test cases. The test cases should include different floor plan designs and sizes, and must provide good coverage. Prepare one text file containing the input data for each test case. Include appropriate labels when printing your results and ensure your printout is well formatted so it can be easily read.

Sample input:

```
8
7
1000000
11111G0
00B0101
1111111
101R101
1010101
0001111
0B010G0
```

The array after painting:

```
1GGGGGG
11111GG
BBBB1G1
1111111
1B1R101
1B1R101
BBB1111
BBB1GGG
```

Note: Code must be appropriately commented (including Javadoc comments).

For this assignment, only an electronic submission is required.

Your electronic submission (submitted via Desire2Learn) will consist of two files:

- i. a written report. This should begin with a title page that includes: the course (CS 1083), your section (FR01B, FR02B, FR03B), the assignment number, your full name, and your UNB student number. That should be followed by two sections, with each part clearly identified with a section heading. Include:
 - the source code for `Painter.java` and `PaintDriver.java`
 - the sample input and output for each test case

This written report should be prepared using a word processor; we recommend using Microsoft Word (i.e. create a .docx file for your report). Copy & paste your java source code, input & required output into the report document. Add appropriate headings for each part. Fix up the formatting where necessary, adjusting line breaks & page breaks to ensure that your document is easy to read. Use a monospaced font for your code to maintain proper indentation.) Once the report is complete and you've checked it all over, save the .docx file for your own records, and then **save a second copy in pdf format for submission**. (Note: Be sure to open that file in a pdf viewer to verify that the pdf was generated correctly.) **The SINGLE pdf file containing your report will be submitted to the appropriate assignment drop box on Desire2Learn.** (It is important that you submit a pdf file and NOT the original Word document. This pdf will allow the marker to write comments directly on your work to give you better feedback.)

Note: name this report as follows: **YourName_CS1083_As8_Report.pdf**

- ii. an archive file (**.zip**) that contains all your Java source code for this assignment. Make sure that your archive **includes all .java files and input files** (in case the marker wishes to compile & run your code to test it). You should not include the

report document or the .class files in your archive. This archive should be submitted as a **single file** to the appropriate drop box on Desire2Learn.

Note: name this archive file as follows: **YourName_CS1083_As8_Archive.zip**