Instructor Name: Siddhartha Nath
Email: snath@scu.edu

Santa Clara University
School of Engineering
Computer Science and Engineering

Design and Analysis of Algorithms (COEN 279)
Programming Assignment 1 – 100 points

Note: Please make sure your program is in C++ or Python only.

Assignment 1 (100 points)**.** *The "paint fill" or "floor fill" algorithm is implemented in many image editing programs. Given a screen (represented by a two-dimensional array of colors), a point, and a new color, the algorithm fills in the surrounding area until the color changes from the original color. The 2D array is like a graph where the nodes are each cell (or, pixel)* (i, j) *and edges are between adjacent cell / pixel. For example,* $(i+1, j)$*,* $(i-1, j)$*,* $(i, j+1)$ *and* $(i, j-1)$*.*

You will implement this algorithm in a program to fill connected, similarly colored areas with a different color. Your algorithm takes as input four parameters: a 2D array, a start node, a target color, and a replacement color. The 2D array's elements are colors whose values are ('R', 'G', 'B', 'Y', 'W', 'g', 'X'), which refer to (red, green, blue, yellow, white, grey and black) colors.

Figure 1(a) shows example of a 9×8 *array with various cell colors, and Figure 1(b) shows its corresponding 2D array, which is what your program will take as input.*

Assume that array indices start from 0, and the top-left cell is $(0, 0)$*.*

If the start node is $(1, 6)$*, target color is 'B' (i.e., black) and the replacement color is 'g' (i.e., grey), the algorithm looks for all nodes in the 2D array that are connected to the start node by a path of the target color and changes them to the replacement color. Figure 2(a) shows output of the above input as a* 9×8 *array with black cell colors modified to grey. Figure 2(b) is what your program should output for this input. You should also print a list of cell locations* (i, j) *in row major order and the total number of cells modified. For this input,*
List of cell locations modified:
$(6, 1), (7, 1), (8, 1),$
$(6, 2), (7, 2), (8, 2),$
$(7, 3), (8, 3),$
$(6, 4), (7, 4), (8, 4),$
$(6, 5), (7, 5), (8, 5),$
$(2, 6), (7, 6), (8, 6),$
$(2, 7), (3, 7), (4, 7), (5, 7), (6, 7), (7, 7), (8, 7)$

Number of cells modified: 24

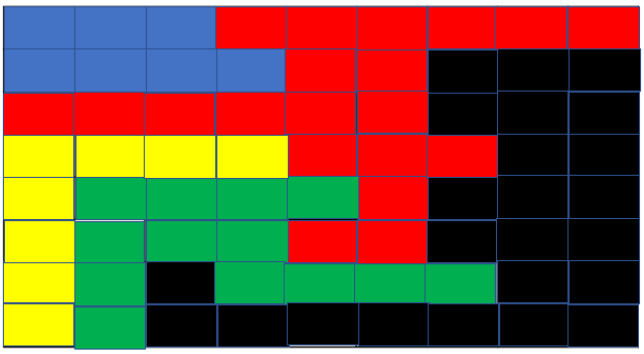
Your submission package (.zip, *.tar.gz)* ***must*** *contain the following to receive full credit.*

(1) Python or gcc/g++ version used.

(2) C++ code and binary or Python code with a requirements.txt that has all package dependencies.

(3) Testcase(s) on which you validated your program. Each testcase must be a testcase-i.txt file where $i = 1, 2, ..., n$ *are indices to your testcase files.*

(4) Please make sure you handle corner cases, and gracefully error out when you are provided with incorrect inputs, etc. Program crashes or errors will be penalized.

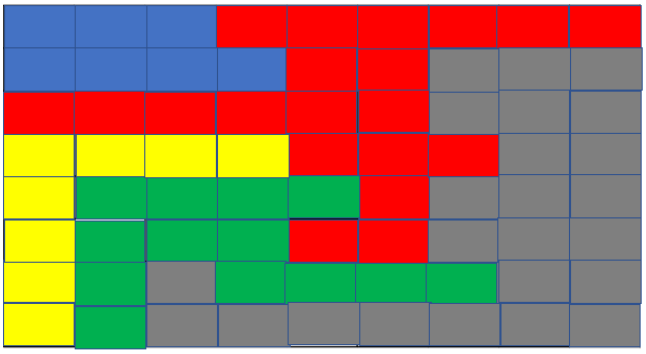


(a)

‘B’	‘B’	‘B’	‘R’	‘R’	‘R’	‘R’	‘R’	‘R’
‘B’	‘B’	‘B’	‘B’	‘R’	‘R’	‘X’	‘X’	‘X’
‘R’	‘R’	‘R’	‘R’	‘R’	‘R’	‘X’	‘X’	‘X’
‘Y’	‘Y’	‘Y’	‘Y’	‘R’	‘R’	‘R’	‘X’	‘X’
‘Y’	‘G’	‘G’	‘G’	‘G’	‘R’	‘X’	‘X’	‘X’
‘Y’	‘G’	‘G’	‘G’	‘R’	‘R’	‘X’	‘X’	‘X’
‘Y’	‘G’	‘X’	‘G’	‘G’	‘G’	‘G’	‘X’	‘X’
‘Y’	‘G’	‘X’	‘X’	‘X’	‘X’	‘X’	‘X’	‘X’

(b)

Figure 1: Example of an input 9×8 array. (a) Cell colors. (b) Input array to program.



(a)

‘B’	‘B’	‘B’	‘R’	‘R’	‘R’	‘R’	‘R’	‘R’
‘B’	‘B’	‘B’	‘B’	‘R’	‘R’	‘g’	‘g’	‘g’
‘R’	‘R’	‘R’	‘R’	‘R’	‘R’	‘g’	‘g’	‘g’
‘Y’	‘Y’	‘Y’	‘Y’	‘R’	‘R’	‘R’	‘g’	‘g’
‘Y’	‘G’	‘G’	‘G’	‘G’	‘R’	‘g’	‘g’	‘g’
‘Y’	‘G’	‘G’	‘G’	‘R’	‘R’	‘g’	‘g’	‘g’
‘Y’	‘G’	‘g’	‘G’	‘G’	‘G’	‘G’	‘g’	‘g’
‘Y’	‘G’	‘g’	‘g’	‘g’	‘g’	‘g’	‘g’	‘g’

(b)

Figure 2: Example of output 9×8 array. (a) Cell colors. (b) Output array from program.