Data Structures and Algorithms Lab

Lab 10 **Marks 05**

Instructions

Work on this lab individually. You can use your books, notes, handouts etc. but you are not allowed to borrow anything from your peer student.

Marking Criteria

Show your work to the instructor before leaving the lab to get some or full credit.

What you have to do

A computer graphics image is composed of **rectangular points or pixels** on the computer screen. In a **black-and-white** picture, we can use **0** to represent **white** and **1** to represent **black**. We can store a representation of the picture in a **2-D** array of **Boolean** values.

Two **black** pixels are part of the same object if we can get from one to the other with **horizontal or vertical** moves. For example, the following **2-D** array contains **3** objects:

1	1	0	1	0
1	0	1	1	0
1	0	1	1	0
0	1	1	0	0
0	0	0	1	1

Given the coordinates of a **black pixel**, design and implement a **recursive function** to **erase (or white-out)** the object to which the pixel belongs. The prototype of your function should be:

```
void eraseObject (int** ar, int r, int c, int i, int j)
```

where, **ar** is the integer array containing the pixels of the picture, **r** and **c** are the dimensions of **pic** array, and **i** and **j** are coordinates of a pixel in the given picture.

For example, if we make the function call `eraseObject (pic, 5, 5, 2, 3)` on the above picture, then the whole object containing the **black pixel** at `index (2, 3)` should be **erased** i.e., the resulting pic array should look like:

1	1	0	**0**	0
1	0	**0**	**0**	0
1	0	**0**	**0**	0
0	**0**	**0**	0	0
0	0	0	1	1

You have to complete the implementation of **eraseObject** function exist in a **Source.cpp** file provided in this lab's folder that will read an **image** data from a file **input.txt** and perform the above-mentioned task. The input file is in the following format:

Line 1: two numbers separated by space indicates the **size** of the **image** in **row** and **column** format.

Line 2: two numbers separated by space indicates the **index** of a **black** pixel in **row** and **column** format.

Line 3: The **image** data started. Each pixel position is separated with a space.

Input.txt

```
5 5
2 3
1 1 0 1 0
1 0 1 1 0
1 0 1 1 0
0 1 1 0 0
0 0 0 1 1
```

Output

```
1 1 0 0 0
1 0 0 0 0
1 0 0 0 0
0 0 0 0 0
0 0 0 1 1
```

☺ ☺ ☺ **BEST OF LUCK** ☺ ☺ ☺