

Instructions

- Work in this lab individually. Follow the best coding practices and include comments to explain the logic where necessary.
- You can use your books, notes, handouts, etc. but you are not allowed to borrow anything from your peer student.
- **Do not use any AI tool for help; doing so will be considered cheating and may result in lab cancellation and possible disciplinary action.**
- Test your program thoroughly with various inputs to ensure proper functionality and error handling.
- Show your work to the instructor before leaving the lab to get some or full credit.

Erasure of Connected Objects in a 2D Image

A computer graphics image is composed of rectangular points or pixels on the computer screen. In a black-and-white image, we use **0** to represent white and **1** to represent black. The image can be stored as a 2-D array of integers.

Two black pixels are part of the same object if they are connected either **horizontally** or **vertically**. For example, the following 2-D array contains **3 distinct 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
```

Design and implement a **recursive function** to erase (or “white-out”) the object that contains a given black pixel. When a black pixel is erased, all black pixels connected to it (either horizontally or vertically) must also be erased. This means setting the corresponding entries in the 2-D array to **0**.

The prototype of the function is:

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

Where:

- **ar** is the 2-D array representing the image.
- **r** and **c** are the number of rows and columns in the image.
- **i** and **j** are the coordinates of the black pixel to erase.

Example

For the above image, if the function call is:

```
eraseObject(pic, 5, 5, 2, 3);
```

The resulting 2-D 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
```

Instructions

1. Complete the implementation of the **eraseObject** function in the provided **Source.cpp** file.
2. The program should:
 - Read the image size, pixel coordinates, and image data from **input.txt**.
 - Process the image using the **recursive function** to erase the object containing the given pixel.
 - Display the updated image on the console.

Input Format

The input is read from a file named **input.txt**, formatted as follows:

- **Line 1:** Two integers separated by a space, representing the dimensions of the image (**rows** and **columns**).
- **Line 2:** Two integers separated by a space, representing the coordinates (**row** and **column**) of the black pixel to erase.
- **Line 3 and onwards:** The image data, with each pixel represented as a **0** or **1**, separated by spaces.

Sample (input.txt):

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

Output

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