

Programming Assignment 2

The Ink Blob on The Picture

Max Points 100

Due: 2/18/2022 at 11:59pm

Background Story of this Assignment

We have just learned about recursion. Super Cool! This programming assignment will allow you to apply your recursion skills in this assignment.

Have fun and start early! Make sure to see the TAs and ULAs for help!

Assignment Details

You are given a super duper gigantic picture for your birthday, however you notice some ink blobs on the picture. You call a service that can remove blobs from pictures, however the company needs to know how many blobs total there are in order to properly charge the customer (sadly it is not free). The company doesn't care about the size of each blob, they are interested in just the total. In order to determine the number of blobs, you take the picture and place it on a grid system. Each cell in the picture is a pixel of the picture. If there is a blob in the pixel, then the value 1 is assigned to it. Otherwise 0 is assigned which indicates the respective pixel does not contain a blob. Here are some samples of pictures and the number of blobs they contain. The blobs are colored blue.

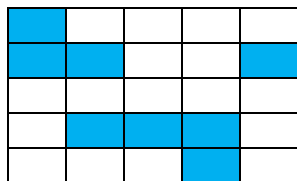


Figure 1: This picture contains 3 blobs.

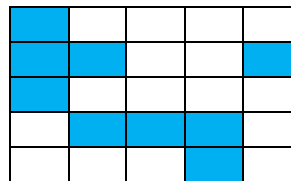


Figure 2: This picture contains 2 blobs.

The Provided Skeleton File

You were provided with a skeleton C file that has the main function and function prototype. This section will discuss the lines of code provided for you in the main function to assist you with understanding how the code will execute.

- Lines 6 – 7 are preprocessor directive statements.
- Line 16 – 21 sets the seed for the random generator. It has the same role from the previous programming assignment.
- Lines 23 – 26 will ask for the size of the picture. It is assumed that our picture is always a square. This means the length and width are the same.
- Lines 28 – 31 allocates a 2D dynamic integer array

- Lines 33 – 50 shows a loop that will run 5 scenarios and display the result to the terminal window.
 - Lines 37 – 39 sets up the scenario for a random picture with blobs to generate.
 - Lines 44 – 47 will check for blobs in each scenario by calling a user defined function `blobDetect` (you will have to implement this for the assignment).
- Lines 53 – 56 frees the heap.

The Function Prototype

You are going to implement 1 user defined function. In the provided main function, you were provided the prototype. This section will describe what it is expected from the definition.

```
int blobDetect(int ** picture, int x, int y, int limit);
```

The `blobDetect` function will search the picture for a blob in the respective x and y coordinate provided in the call. The function takes 4 arguments. The first argument is the 2D dynamic integer array. The second and third arguments represent the coordinate of the grid. The fourth argument takes the length of the grid in terms of length and width. The function will determine if the pixel on the grid is a blob or not. If it is a blob, then will need to determine how big the blob. You will have to consider the neighbors and will also have to make sure your solution does not go out of bounds. The function returns the number of pixels that contains a blob (summation). If no blob is detected in the pixel, then return 0. You are required to use recursion in solving this problem. Here are some sample scenarios.

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

There are 4 blob(s) in the picture.

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

There are 2 blob(s) in the picture.

Requirements

Your program must follow these requirements.

- You must use recursion in this assignment. If recursion is not use, then a **score of 0** will be given as the grade in this assignment. Even if the assignment is attempted and correct with output, you will still receive a 0 for not using recursion as requested.
- The output must match exactly (this includes case sensitivity, white space, and even new lines). Any differences in the output will cause the grader script to say the output is not correct. Test with the script provided in order to receive potential full credit. Points will be deducted!
- Do not change ANY content of the skeleton that was provided for you. Your code will be tested through a script that relies on this main function flow along with the header file being imported. Any changes to this will result in your program not working fully which will lead to point deductions that will not be fixed!

- Name your C file `program2_lastname_firstname.c` where `lastname` and `firstname` is your last and first name respectively. Please make sure it matches the spelling exactly how it is registered in Webcourses. Points will be deducted if the file is not named correctly.

Tips in Being Successful

Here are some tips and tricks that will help you with this assignment and make the experience enjoyable.

- Do not try to write out all the code and build it at the end to find syntax errors. For each new line of code written (my rule of thumb is 2-3 lines), build it to see if it compiles successfully. **It will go a long way!**
- After any successful build, run the code to see what happens and what current state you are at with the program writing so you know what to do next! If the program performs what you expected, you can then move onto the next step of the code writing. If you try to write everything at once and build it successfully to find out it doesn't work properly, you will get frustrated trying find out the logical error in your code! **Remember, logical errors are the hardest to fix and identify in a program!**
- Start the assignment early! Do not wait last minute (the day of) to begin the assignment.
- Ask questions! It's ok to ask questions. If there are any clarifications needed, please ask TAs/ULAs and the Instructor! We are here to help!!! You can also utilize the discussion board on Webcourses to share a general question about the program as long as it doesn't violate the academic dishonesty policy.