Homework 4 - Design Algorithm

My process was mostly trial and errors with many trials that did not benefit final algorithm, but I did learn a lot in the programming for this lab. I started off writing pseudo-code of how I was planning on iterating through the stored data. This algorithm I kept throughout, but it's implementation is what I struggled with.

I knew that the data structure that was going to be used would have to be something that was easily resized, preferably dynamic, so I decided to store all of the input into a vector. My main issues arose when I was trying to organize the data into a square grid, the way the input file was formatted. I finally got it to to store successfully in a 2-dimensional vector (matrix), or a vector of vectors. This made each element easily accessible and easy to locate.

I did all of the "scanning" of the image file in a function where both *passes* are executed, and the result matrix is printed to the screen and into an output file (both in separate functions). Here, is where I had to implement that algorithm to scan, check, and manipulate the image data. This data is the what was received by the input file, which was only the characters that were numbers (isdigit) then converted into integers to make comparing and labeling more efficient.

For the first pass, the most significant bits of the data were the elements of my matrix that were "1" which would be compared to the pixels that were directly to the left and on top of the current pixel. Of the three possible cases, two of them had one of the comparing pixels to either be a "0" or non-existent, which automatically makes the first row and column subject to on of these two cases. My code reflects this by have if-statements that separate these and only have the pixel that exists to be checked. The labels were just integers, which when used were increment to reflect data clusters.

The second pass simple just connected the labels to form the objects by changing the larger label to smaller, related label. This produced the final product, where the largest label represented the number of objects in the image and then iterated through once more to count the size of these images, or the number of times label appears.