Status of code: the code is fully implemented and completely working. I do not know of any bugs in my code.

Testing: I tested my code by running it on certain input files, and then manually computing what the output should be by drawing the islands and lakes and counting their sizes (essentially doing what the program should do) and then comparing my result to that of the code to see if it is correct or if there are bugs. I also used print statements such as “number of islands”, “total size of islands”, and other such print statements, so that I was dead sure I knew what it was printing. Afterwards I deleted those print statements so that the output could conform to the expected output of the assignment.

Program: Maybe just to discuss the general implementation, so for part 2a i first created the singleton clusters. But also I had in each position stored x and y coordinates so that later when all the singleton clusters are created I can iterate over them, see if the value (1 or 0) is the same as the left node or node directly above. I did not check right or down to avoid double checking on the same node. And if equal I merged them, and the end of this process on the entire grid resulted in the land and water clusters where I could print the required info. For part 2b, I did all the same stuff and then a lake check. For the lake check, I went to every cluster of water and kinda did an iterative thing, where I checked if the node above is water. And then the node above. All the way until either it is the top of the grid or I hit land. If I am at the edge of the grid, this water body is not a lake. If the node above is land, I checked the node at top left corner and top right corner. If any of these corner connections were water, it is also not a lake. If both are land, it can be a lake. If i am able to do this verification in all 4 directions for each node of the body of water, i then confirm it is indeed a lake and i get its associated land (the land inside of which is the lake) and add its area to the area of that land, and then do all the necessary work of gathering the amount of islands and lakes and printing the necessary info.

Resources: I designed the algorithms on my own, but when VSCode autocomplete suggested code, I would use the tab button to accept the suggestion while rereading it afterwards and modifying as needed to match my original algorithm design. So you could say the code autocomplete was a resource used. Also, at times where I was confused as to why the desired output was not printed when I tested my code or when Java threw an error, I would sometimes show it to GitHub Copilot to ask it why my code was doing so and where I went wrong. And after the AI would point out my mistakes, I would redesign the algorithms as needed to attain the correct output. So I also had GitHub Copilot used in this manner, to just point out flaws but not write the code for me. For example, in a previous nonfunctional version of the code, I was modifying a list while iterating over it, and so java threw an error but i didn’t understand why my stuff was wrong. So I took it to GitHub Copilot who explained to me that I am not allowed to modify it while iterating over it, and then I was able to fix the code.