Michael Dandrea

C08 Writeup

This project was an extension of previous assignment. We had to update detect\_data\_2d to identify which bit position the corruption occurs at and add a new driver program called correct\_data\_2d that will identify and fix the corruption.

In order for my detect function to find the bit where the corruption occurs, I added the following:A computer screen with text

Description automatically generated

I a corruption is detected, it finds what the correct number should be and loops through all possible bits, XOR-ing the correct number and seeing if it matches what is in the corrupted index of the matrix. If they are equal, the loop index is the bit that has been corrupted. I also made my detect function return a struct that includes the row index, column index and bit position of the corrupted data:

A black screen with white text

Description automatically generated

This is useful because I call the detect function in my correct function:­­­­

A screen shot of a computer program

Description automatically generated

If correctRow and correctCol are -1 there is no corruption to fix, so it just returns, as calling detect will print that no corruption is present. If not, the second matrix is set to be equal to the first matrix (so I can print them if needed) and finally, the corruption is fixed.

Here are screenshots:

First we make sure there is no corruption:  
A black screen with white text

Description automatically generated

Next we corrupt a bit:  
A computer screen with white text

Description automatically generated

Make sure the corruption happened:

A screen shot of a computer

Description automatically generated

Finally, we fix the corruption. I learned of a command on mac called ‘leaks’ that does the same thing as valgrind on the ci, so I have included the correct\_data\_2d passing leaks in this screenshot:  
A screenshot of a computer

Description automatically generated

As you can see, there are no errors, warnings, or memory leaks.

Just in case, here it is with valgrind as well (without -p to show it works):  
A screenshot of a computer error

Description automatically generated