Michael Dandrea

CSCI 207

Dr Jones

10/16/2023

This assignment was an extension of our previous assignment. The goal was to create four new driver programs.

The first one is a program to multiply two matrices that are gotten from a command line argument, then print it to a new file, which is the output. This is the driver program:

A screen shot of a computer program

Description automatically generated

This was pretty easy to write, as it is set up pretty much the same as the program to add two matrices.

The next program was to checksum the rows of a matrix. It adds all the values in a row and puts that value at the end of the row in a new column. This is the driver program:

A screenshot of a computer program

Description automatically generated

This one, along with the rest of the driver programs were relatively easy write. The hardest part for this one was figuring out how to call the allocate2d and write\_to\_file functions with the correct matrix dimensions.

The third program was to checksum the columns instead of the rows. It adds all of the values in a column, then puts that value in a new row in the same column. Here is the code:

A screen shot of a computer program

Description automatically generated

Again, the hardest part was figuring out what dimensions the new matrices needed to be, which was not really that hard.

The final driver program was one to do both the row checksums and column checksums at the same time. Here is the code:

A screenshot of a computer program

Description automatically generated

This one actually gave me a bit of trouble, but it was in the matrix\_checksum function itself, not the driver program.

All of the new functions were written in the utilities.c file, with their prototypes added to utilities.h. These are all the functions I needed:

A screenshot of a computer program

Description automatically generated

The row and column checksum functions were fairly simple. All I did was set the second matrix (B) equal to the first (A) then for each row, add all the values and put the sum in the final column for row checksum, and add all the values in a column and put the sum in the final row for column checksum.

The matrix\_checksum function actually gave me a bit of trouble, even though it looks pretty simple. For a while, the resulting matrix would only be zeros. Honestly, I still do not know why this happened, but when I changed the order the functions were called, (to the order shown in the screenshot) only the final column were zeros. From there, I realized that I needed to pass cols + 1 to col\_checksum because it was not going through the entire matrix to do the checksum.

The mul\_matrix function gave me a little bit of trouble as well. Originally, I was not passing acols to the function and the acols in the innermost for loop was bcols. This caused my program to give me a segmentation fault. I forgot the resulting matrix would have a number of rows equal to the number of columns in the first matrix, and bcols was larger than that number, causing the segmentation fault.

Here is a screenshot showing compilation on my macbook with no errors:

A screen shot of a computer

Description automatically generated

Here is a screenshot showing compilation on the ci:

A screenshot of a computer program

Description automatically generated

Here are screenshots of all the files passing valgrind:

Column checksum:  
A computer screen with white text

Description automatically generated

Row Checksum:

A computer screen with white text

Description automatically generated

Matrix (data) checksum:

A computer screen with white text

Description automatically generated

Matrix multiplication:  
A screenshot of a computer program

Description automatically generated

The only thing I couldn’t figure out was how to print a space between each number.