What Life Taught Me

Alexander Ragland #1858717

February 10, 2023

1 Modulus isn't just a funny name

Before this assignment, I had only some experience with the modulus operator. I never used it in any meaningful way though. To be fair, modulus is fairly uncommon in daily life, but here it proved extremely useful. When my program checks for cells whose indices are outside of the given matrix, it needs a way to check the other side of the matrix, provided the matrix is toroidal, or *donut-like*. Using modulus between the *too large* index and the number of rows (or columns, depending on the index) allowed me to get the index of the cell on the opposite side of the matrix.

2 Do not be afraid of temporary variables

I used to have an irrational aversion to temporary variables. After this assignment, I can say with full confidence that my fear has been extinguished. For one, having a temporary variable makes swapping pointers easy. Not only that, but I also had a fantastic little bug in my program when calling *uv_census* on a *donut* matrix. See, anytime my loop would look at a cell outside of the matrix, it should've looked at the cell on the opposite side of the matrix. And it did, however, I wanted to use as few variables as possible so I just updated the indices I was iterating over with the far cell's index. This had the unfortunate side effect of making me endlessly loop over the matrix, resetting the indices each time I looked at an "out of bounds" cell.

3 Unsigned vs signed integers is a NON-trivial distinction

Another amazing discovery I made during this project was that guess what, *unsigned* and *signed* variables are different! As a budding computer scientist, I thought myself above measly integers. *Unsigned or signed? Who cares?* Well, unfortunately, whether or not a number is negative can be very important. I spent a few hours scratching my head while trying to figure out why my *uv_census* function wasn't working correctly when it encountered out-of-bounds cells. Essentially, whenever the index of a "cell" I was looking at was negative, I should've been able to tell that it was negative, and take the appropriate steps forward. Instead of using 64-bit signed integers, I was

using 32-bit unsigned integers, thereby losing a sign, and furthermore, my sanity.

4 File reading and writing is weird

Deciphering the ancient C manual pages for various functions like *fscanf(), fopen(), fflush(), fclose(), etc.* was a bit confusing, to say the least. I already knew the general concept of file buffers and the such but distinguishing the different printing functions, as well as understanding *FILE ** types was pretty tricky for me. There's a lot more to think about when writing a program than just its intended functionality. How a program receives data, and outputs it, although somewhat of a dull topic in my opinion, is crucial.

5 Sometimes calloc is better than malloc

I somewhat wrote off calloc after learning that it was slower than malloc and didn't even think about why having 0's would be useful. In this case, initializing the matrix's values to 0 with calloc made my job easier, since the default state of the grid should be *false* anyway.

6 Acknowledgements

Audrey Ostrom's tutoring session was extremely helpful in understanding the general layout of this assignment, and the general order in which we should tackle things. Ben Grant's tutoring session was also very helpful for debugging the issues I mentioned, as well as helping me with swapping pointers.