Post-project Writeup - Polynomial

James Viner

Project Summary

The task was to create a static library for the creation and manipulation of polynomials by an end-user of another developer. The polynomial came in the form of a struct with a coefficient and an exponent and built-in mathematical operations and certain specific tests were required to be implemented to ensure some level of basic functionality.

Challenges

I'm left in a strange place when it comes to ranking this project. While it wasn't as complex and certainly didn't cause as much time-crunching in the 25th hour as ws or mastermind, this project also did require an understanding of linked lists and recursion in order to make it both function and read well. So, while I guess at face value I'd say this wasn't as difficult as some previous assignments, this one definitely wore me out the most in comparison to the other two-day projects. I hesitate to call it burn-out, because I feel like I haven't been in the field long enough to justify using that term, but it was definitely an up-hill battle to fight the mental fatigue that always comes when I drastically underestimate the complexity of some small part of the project.

On a different note, I always find it funny reading your code (assuming that it's just Liam reading this). There's a level of fundamental understanding and familiarity that I can't expect to have with C in comparison to you, but somehow that doesn't make it any less upsetting to find that what a function that took me roughly half an hour to figure out and write can literally be written in a single line. I'm talking about the sizeof_poly() function in my program, which performs a task that you were able to get the same functionality from using snprintf()'s return value, which I feel like is a level of esoteric familiarity with C that I'd be simultaneously eager to and terrified of achieving.

I tend to ramble in the challenges section, I've noticed. It's hard to find broad-strokes topics or concepts that I struggled with on a project given that it's usually just minor details that hold me up for hours instead. This time around though, I can happily say that I know the broad-strokes topic that I should do more research into: pointers. I'm not exaggerating when I say that I had to get a whiteboard and spend the better half of an hour mapping out exactly what your poly_add() function actually was doing. I thought I understood the use of pointers and recursion from the topics we went over, but actually being thrown into the deep-end on it felt genuinely terrible and I probably would've had to come up with some sort of ugly iterative solution instead had I not just lifted and modified many of your functions to suit the program.

Successes

I would like to apologize for the snprintf line in the poly_to_string() function. I'm entirely sure there are better ways to accomplish that task, but I am motivated by forces beyond my comprehension to implement the first solution to a problem that pops into my head, and the topic of the day happened to be ternary operators. I'm calling it a success, because it actually works and writes to memory the strings in the same way as the poly_print() function prints them to the screen. Deep down, I know it's a failure to find some sort of nicer, easier solution, but I won't lie in that looking at it makes me laugh, and sometimes that's enough for me.

I know we were given most of the implementation for insertion sort, but I still feel like reaching for a doubly-linked list and implementing it in my library felt like a win. In fact, basically any time I used the prev pointer felt pretty satisfying and I'm curious to see how many other people approached the project that way. Maybe it's not even necessary and I'm over-complicating things, but maybe it was at least a little clever. Who knows?

Lessons Learned

Liberal use of code-borrowing absolutely saved this project for me, given that over half of my functions were based on or directly taken from existing code that we were given this week. Also, both recursion and the fancy pointer-wizardry on display that took an actual graph for me to understand were unironically awesome to see in action and learn about. I'm very excited to see about more concepts related to DSA, because I hadn't even considered a lot of the structures and sorting methods that I found while doing research during this project. I still probably have to learn to pace myself so that I'm not sitting at a desk all of Friday until 4AM (or later), but I'd like to think I'm getting at least a little faster with writing code and recovering from my mistakes. Overall, I'm pleased with the way things are going and am hopefully optimistic for the rest of the module. I sincerely hope that writing that doesn't cause me to incur some level of divine punishment for my hubris.