Yesenia Molano Sanchez seu346

CS 3333 Project Report

In my project I decided to choose Topic 2 from Category A which is Primes, GCD, LCM and Topic 4 from Category B which is Binomial Theorem.

In Topic 2 – from Category A, I created 3 functions called *primes*, *LCM*, and *GCD*.

In my *prime* function I only had one argument which was the first integer from the user input. Inside *prime* I calculated all the primes for any given number. In the process of coding this function I struggled with understanding how to keep looping to get the smallest prime, which is two. At first, I made this function more complicated than it was supposed to be. I was allocating two arrays and using nested for loops. To get through this challenge I decided to just print directly from the function instead of saving it returning it to the *main* and then looping to print it.

In my *LCM* function I had two arguments *a* and *b* which were from the user input. Inside this function I allocated two arrays for *a* and *b* each of the size of the other integer. After this I looped through the array for each variable and saved the data for the multiples up until it multiplied by the other variable. After this I do a nested for loop to loop through the values in the matrix and compare it with the value inside the other matrix until a match is found. One of the challenges I faced with this function is returning *lcm*. At first, I was returning *lcm* until the end, but this was not returning the right least common multiple. To fix this I decided to return *lcm* right after I found the value instead of waiting until the end.

In my *GCD* function I had three arguments, *a, b* and *lcm*. The input *a* and *b* are from the user and *lcm* is what the previous *LCM* function returned. In this function I truly didn't struggle it was straight forward process. I just multiplied *a* and *b* and divided it by *lcm*. After this I returned it to the main function.

In Topic 3 – from Category B, I created 2 functions called *printBinomial*, and *comb*.

In my *printBinomial* function I just printed the format of the function expansion, (coefficient)x^(n-k)y^(k). But from this function I call *comb* which is the function that calculates the coefficient for each variable. In this *printBinomial* function I did not face any challenges.

In my *comb* function I had two arguments, which were passed in from *printBinomial's* loop to get the value of n and r. In this function I calculated the combination of (n, r). In this function I did not run into any issues to get n! I just looped through and saved it to a multiplying total. From this function I returned the combination number to *printBinomial.*

To run my project files: gcc main.c calls.c -lm

For this project I worked alone, with no grade for other group members. I give myself a 10.