Teddy Jackson

Code Report prog02

Part01 on starts by checking if the inputs are valid using a set of if statements. It then uses malloc to create an array of the requested size. It manually populates the first 2 indices, then iterates through and calculates out the rest of the array. From there it prints out the array in the specified format. In Part01, I picked malloc because I knew I would be using every position in the array, and most importantly it didn't throw an error.

Part 2 has a few major differences. Same idea as part 1 with the starting conditionals to check if the input is valid. Second thing that happens in the main is I check through the inputs to add up the total possible size for the final output of unique numbers. I then use calloc to create an array of this size. I use calloc here, instead of malloc because I don't know how many empty spaces I will leave in the array, so it's important to know where I've been. Another reason for calloc is the fact that all numbers either calculated using a fibonacci sequence, or given as input are going to be greater than 0. Since calloc populates the array it creates with “0” values, there is no chance of ambiguity; if there is a “0” in an index, I haven't touched it.

From here in Part02, we get into the buildSmallSet function. This does the same thing as Part01, calculating a fibonacci sequence using 3 parameters, but after it creates and prints out the smaller array, it then goes index by index and adds any newly encountered numbers to the larger array that was created earlier in the main. It then frees the memory from the smaller array and returns. Once all the smaller arrays have been calculated, printed, and assimilated into the larger array, the program prints out the non-zero contents of the larger array.

I am currently not aware of any valid input combinations that make my program crash. I was getting some trouble earlier in the process when I was still using malloc for part02, but once I substituted calloc, I didn't have a problem. There was a brief period of time where I thought I was crazy, and it was because one of the example outputs on the assignment pdf is incorrect, but I was in the right, so that was pretty vindicating. I definitely have a lot more to say when it comes to testing and debugging my script, but the report guidelines explicitly say to talk about my c program, so I guess I’ll leave it at that.