

CS 2022 – Fall 2009
Assignment #1
9/07/2009
Due: Friday 9/11/2009 11:59 PM

In this assignment you are asked to write three small C programs to demonstrate command of the basic principles we have discussed in class so far.

Fibonacci Number Calculator

The Fibonacci numbers are the following sequence of numbers: 1, 1, 2, 3, 5, 8, 13, 21, 34 ... where the first number, F(1), is 1, and the second number, F(2), is 1, then every number after that is defined by this simple recursive formula:

$$F(n) = F(n-1) + F(n-2)$$

The Fibonacci sequence is named after the Italian mathematician Leonardo of Pisa who was also known as Fibonacci. The story goes that Fibonacci discovered this sequence when studying the reproductive habits and population growth of rabbits!

Fibonacci numbers are used as a prime example (along with factorials) in teaching recursive functions. However, in fact, there is a closed form expression that calculates the nth Fibonacci number without the need to compute the ones that came before it. That expression is:

$$F(n) = \frac{\varphi^n - (1-\varphi)^n}{\sqrt{5}} \text{ where } \varphi \text{ is the golden ratio}$$

Your job here is to write two C programs `fib1.c` and `fib2.c` that calculate a Fibonacci number using the **recursive definition** and **not the closed form expression**. The desired number will be passed to your program as a command line argument. `fib1.c` should use recursive function calls to compute the answer, and `fib2.c` should not make any recursive calls but should use loops to compute the series.

After your programs are compiled the user should be able to run them as follows:

- `:-> ./fib1 1`
1
- `:-> ./fib1 7`
13
- `:-> ./fib2 8`
21
- `:-> ./fib2 9`
34

Anagrams Lister

Your third small program, `anagram.c`, will take in a string as a command-line argument and will output all anagrams (rearrangements of letters) of that string, each on a separate line. For example:

- ```
:~> ./anagram "abc"
abc
acb
bac
bca
cab
cba
```

Check the last slide of lecture 2 for a hint!

## Tips

Here are some tips you might find helpful in completing this assignment:

- Use the `atoi` function to convert a string argument into an integer
- Use the `printf` function to print to the screen
- If you run into an infinite-loop and your program does not terminate, use `<ctrl>-c` to terminate your program forcefully. If that does not work, start another terminal and type `"killall name_of_your_app"`.

## Submission and Testing

Submit your work on CMS (<http://cms.csuglab.cornell.edu/>) by the deadline. Make sure you have been added to CMS early on and otherwise contact me via email. You should submit the **source code** (the `.c` files) of your application and **not** the compiled binaries.

Your programs will be compiled with `gcc` and tested on a linux environment. You are free to choose the environment of your liking to develop your solutions, but keep in mind that testing will be on a fixed environment, and your application is expected to run on that.

## Academic Integrity Reminder

Remember that you may have general discussions about how to approach this problem with your peers, but you should work on the final solution by yourself alone. If you are stuck or are having trouble, you may email me or talk to me after class on Monday or during my office hour on Wednesday.

Good Luck!