

## Homework 6

Due Date: 3/13/15

Files to submit: **fib.c**, **bin\_str.c**, **ReadMe.txt**

- All programs must compile without warnings when using the -Wall option
- If you are working in a group **ALL** members must submit the assignment on SmartSite
- Submit only the files requested
  - Do **NOT** submit folders or compressed files such as .zip, .rar, .tar, .targz, etc
- All output must match the provided solution in order to receive credit
  - We use a program to test your code so it must match exactly to receive credit
- All input will be valid unless stated otherwise
- The examples provided in the prompts do not represent all possible input you can receive. Please see the Tests folder for each problem for more adequate testing
- You may assume all inputs are valid unless otherwise specified
- All inputs in the examples in the prompt are underlined
- If you have questions please post them to Piazza

### Restrictions

- No global variables are allowed
- Your main function may only declare variables and call other functions.

1. (10 minutes) Write a program called **fib.c** that calculates the nth fibonacci number. The equation for the Fibonacci numbers is defined as follows:

$$Fib(0)=0$$

1.  $Fib(1)=1$

$$Fib(N)=Fib(N-1)+Fib(N-2)$$

2. Name your executable **fib.out**
3. Your program should accept N as a command line argument
4. You MUST solve this program RECURSIVELY
5. Unlike the example in class, you may only have **ONE RECURSIVE CALL** in your function
  1. Hint pointers help make this possible.
6. Here are the [first 100 numbers in the Fibonacci sequence](#)

7. Examples

1. `./fib.out 0`

The 0th fibonacci number is 0.

2. `./fib.out 1`

The 1th fibonacci number is 1.

3. `./fib.out 10`

The 10th fibonacci number is 55.

2. (5 minutes) Write a program called **bin\_str.c** that completes a binary number. A binary number is made up of 0's and 1's but the input strings you will receive can also contain 'x's. An x represents a digit that can be either a 0 or a 1. Your program should display all the possible binary digits that can be formed. For example the string x1x0 could represent either 0100, 0110, 1100, or 1110.

1. Name your executable **bin\_str.out**
2. Your program should accept the binary string as a command line argument
3. You MUST solve this program RECURSIVELY

4. Examples

1. `./bin_str.out 0110`  
0110

2. `./bin_str.out 01x0`  
0100  
0110

3. `./bin_str.out xx`  
00  
01  
10  
11

```
4../bin_str.out 101x100x11x
10101000110
10101000111
10101001110
10101001111
10111000110
10111000111
10111001110
10111001111
```