Lab 3

Thursday, September 28, 2023

You will get 1 point for attending the lab, making an effort to work on the problems, and **discussing your work with a lab instructor during the lab**.

If you finish during the lab, demonstrate your working code to a lab instructor and they will give you the full 2 points. If you do not finish during the lab, discuss what you've done with a lab instructor and they will give you an opportunity to finish outside of the lab and attend office hours to demonstrate your solution.

Practice Problems. You do not need to submit solutions for these problems and during the lab the teaching assistants will solve these problems if requested.

1. Consider a function that can be used to "order" two integers—i.e., it modifies the values of two variables *a* and *b* so that after `order` is called it is now the case that *a* is the smaller of the two and *b* is the larger of the two.

Complete an implementation of the following prototype for the function `order`.

```
// order(a, b) orders the values pointed to by a and b so that *a <= *b;
//   *diff is set to the absolute value of the difference between *a and *b;
//   returns true if the values were switched and false otherwise
// requires: a, b, and diff point to memory that can be modified
bool order(int * const a, int * const b, int * const diff);
```

Using `asserts` in `main` test that your `order` function correctly handles each possible ordering of the input values *a* and *b* (when *a* starts larger than *b*, when *b* starts larger than *a*, and when *a* and *b* are equal).

2. Consider the function `fib` that computes Fibonacci numbers recursively:

```
1 int fib(int n) {
2   if (n <= 1) {
3     return n;
4   }
5   int F_nsub1 = fib(n - 1);
6   int F_nsub2 = fib(n - 2);
7   return F_nsub1 + F_nsub2;
8 }
```

Suppose `fib(3)` is called from `main`. Describe the contents of the call stack every time it changes until `fib(3)` returns. In particular, what are the contents of the call stack just before the first time line 5 runs? The first time line 7 runs? During the first time `fib(0)` is run?

Assessment Problem. A *Caesar cipher* is a simple way of obscuring words by shifting each letter in a word by a set amount n. For example, when the shift is $n = 3$ the letter `'A'` becomes `'D'` and the letter `'B'` becomes `'E'`. Letters near the end of the alphabet "wrap-around", so that `'Z'` becomes `'C'`.

Write a `void` function `char_caesar` that accepts a single `char` by reference (which represents either a lower-case or upper-case ASCII letter) as well as a single `int` shift amount and then modifies the `char` by shifting it by the given amount.

Test your function in `main` using `assert` on at least five examples including edge cases.

Important: Provide a comment before your declaration of `char_caesar` with a purpose statement in the same format used in the course notes (e.g., see slide 14 of Section 1). Don't forget to include any requirements on the input parameters.