

CS 255 (Fall 2022)
Assignment 1: Basics of C
Due: 9/15 (11:59pm)

The purpose of this assignment is to reinforce some of the basics of C that we have been learning by writing some code. For this assignment you will write four functions to perform the tasks described below. You should write these functions in a single file, called `hw1.c`. You will certainly want to write a `main` method to test your programs as you are writing them, but your final submission should have no `main` method defined.

Problem 1: Fibonacci Numbers

Write a function called `fib` with the following header

```
void fib(int n)
```

The function should print out the first `n` Fibonacci numbers. The Fibonacci numbers are a sequence of numbers where the first two numbers are 1, and every number beyond that is the sum of the previous two numbers. The numbers in the printout should be separated by a comma and a single space. As an example, if we call

```
fib(10)
```

The printout should be exactly:

```
1, 1, 2, 3, 5, 8, 13, 21, 34, 55
```

Problem 2: Reverse String

Write a function called `reverse` with the following header

```
void reverse(char *s)
```

Remember from lecture that while the argument looks a little odd, `s` is just a character array, and can be accessed by the usual index convention (e.g. `s[0]`, `s[1]`, etc.).

This function should print two lines. The first line should just be the given string in `s`. The second line should be the string in `s` in reverse order. For example, if `s` contains the string "Hello World!", then the output should be as follows:

```
Hello World!  
!dlorW olleH
```

The next terminal prompt after the printout should not be on the same line as your printout.

Problem 3: Prime Factors

Write a function called `factors` with the following header

```
void factors(long num)
```

The program should print a list of the prime factors of `num`, with repetition. For example, if the function is called

```
factors(120)
```

The result should be

```
2, 2, 2, 3, 5
```

Problem 4: Arithmetic

Write a function called `arithmetic` with the following header

```
void arithmetic(void)
```

The function will prompt the user for an input with the following:

Operation:

The user will enter a simple integer arithmetic expression, with two operands and a single operation (addition, subtraction, multiplication or division).

For example, a user might enter

Operation: 13 + 45

Then your program should print:

Sum: 58

If the user enters a subtraction:

Operation: 40 - 100

Then the program should print

Difference: -60

If the user enters a multiplication:

Operation: 3 * 7

Then the program should print:

Product: 21

Finally, if the user enters a division:

Operation: 20 / 7

Then the program should print two lines:

Quotient: 2

Remainder: 6

You will need to be a little careful about division with negative numerators. The rules are that the remainder always should be positive and less than the denominator and the equation:

$$\text{numerator} = \text{quotient} * \text{denominator} + \text{remainder}$$

should be true.

For example, if the user enters

Operation: -20 / 7

the program should print

Quotient: -3

Remainder: 1

Since $0 \leq 1 < 7$ and $-20 = -3 * 7 + 1$. You can assume that the denominator will always be positive, for simplicity.

The program should continue to prompt the user in a loop until they give the EOF. (Hint: It may be simpler to use `getchar()` rather than `scanf()`.)

As an additional hint, you will probably want to use a switch statement to handle deciding which arithmetic operation to perform.