Wesleyan University, Fall 2021, COMP 211
Lab 2: Leap Years and Bit Shifts

1. Overview

The goal in this lab is to familiarize you with data types, conditionals, and loops in C, learn about how to do bit shifts, and give you a chance to do a bit of problem-solving. To this end, you will write two small programs: one to check whether a year is a leap year, and another to use bit shift operations to do multiplication on unsigned integers. For this lab, you should create a new directory `lab2` in your `labs` directory to hold your code for this lab.

I will begin the lab by walking through some sample code (posted as `lab2_example.c`) on data types, conditionals, loops, and bit shifts and then leave you to implementing the leap year and bit shift programs.

Note: the Challenge components below go beyond what I expect you to be able to do in lab, but are there in case you are looking to challenge yourself, or are looking for extra programming practice.

2. A leap-year program: `leap.c`

The algorithm to check whether a year is a leap year is the following:

Leap year: A year is a leap year if it is divisible by 4, and if it is divisible by 100, then it is also divisible by 400.

Your goal here is to write a program that repeatedly prompts the user to enter a year and reports whether or not the year is a leap year. Specifically, your program will sit in a loop, asking the user to input a year, waiting for the input, then checking whether it is a leap year and printing the answer to the screen. After the answer is printed, the program will then query the user if they would like to check another year: depending on the user's answer, the program will either return to the top of the loop or terminate.

You'll need to use the `scanf` and `printf` functions to read in input from the user and print output to the screen. Because we don't know a priori how many times to loop, a `while loop` or a `do while loop` (rather than a `for loop`) plus a `test` to break out of the loop when appropriate is what you should use. Also make sure to think about the data types of the variables you want to use in your program.

Challenge: If you feel comfortable using functions, you might consider putting the leap year check in a function and calling that function in your loop.

3. A bit-shift multiplication program: `bitshifts.c`

You will write a multiplication program that uses only C's bit shift operations, shown in the following table.

Operator	Symbol	Form	Operation
left shift	<<	x << y	all bits in x shifted left y bits
right shift	>>	x >> y	all bits in x shifted right y bits

Write a program that repeatedly prompts the user to enter an unsigned integer. Then use the bit shift left operation to multiply that unsigned integer by two and print it to the screen. Like in the `leap.c` program, after the answer is printed, the program will then query the user if they would like to multiply another number: depending on the user's answer, the program will either return to the top of the loop or terminate

Challenge 1: using only bit shift and addition operations, can you figure out how to extend the multiplication to work so you can multiply by an arbitrary power of 2, not just 2?

Challenge 2: using only bit shift and addition operations, can you figure out how to extend the multiplication to work so you can multiply by an arbitrary unsigned integer?