CSCI 1300 Introduction to Programming
Instructor: Hoenigman/Lewis
Assignment 6
Due Friday, October 30 by 8am.

For this assignment, include all of your code in one file, called main.cpp, which is the default source file name in a CodeBlocks project. The problems for this assignment are the same problems you solved for your first python assignment, only this time, they need to be solved in C++.

Submitting Your Code to Moodle
You must submit your code to Moodle to get full credit for the assignment, even if the computer science autograder gives you a perfect score.

Please also include comments in your code to describe what your code is doing. Comments should also include your name, recitation TA, and the assignment and problem number. TAs will be checking that you code has comments.

Submitting Your Code to the Autograder:
The computer science autograder, known as COG, can be found here:
https://web-cog-csci1300.cs.colorado.edu

Login to COG using your identikey and password. Select the CSCI1300 - Assignment #06 from the dropdown. Upload your .cpp file and click Submit. **Your file needs to be named main.cpp for the grading script to run.** COG will run its tests and display the results in the window below the Submit button. If your code doesn't run correctly on COG, read the error messages carefully, correct the mistakes in your code, and upload a new file. You can modify your code and resubmit as many times as you need to, up until the assignment due date.

Before you submit your code to COG, make sure it runs on your computer. If it doesn't run on the VM, it won't run on COG.

If you do not get your assignment to run on COG before the assignment deadline, you will have the option of scheduling an interview grade with your TA to get a grade for the assignment. We'll talk more about scheduling the interview in lecture and recitation. Even if you do get the assignment to run on COG, you can schedule the interview if you just want to talk about the assignment and get feedback on your implementation.

What to do if you have questions
There are several ways to get help on assignments in 1300, and depending on your question, some sources are better than others. There is a Peer Discussion Forum on our Moodle page that is a good place to post technical questions, such as how to get user input, or treat that input as an integer. When you answer other students' questions on the forum, please do not post entire assignment solutions. The CAs are

also a good source of technical information. If, after reading the assignment write-up, you need clarification on what you're being asked to do in the assignment, the TAs and the course instructors are better sources of information than the discussion forum or the CAs.

Problems:

For each of the following problems, write the code to solve the problem. All code can go in your *main()* function and should be in the same order as the question order, i.e the code to answer problem 1 comes before the code to answer problem 2, which comes before the code to answer problem 3.

1. A day has 86,400 seconds (24*60*60). Given a number of seconds in the range of 0 to 86,400, output the time as hours, minutes, and seconds for a 24-hour clock. For example, 70,000 seconds is 19 hours, 26 minutes, and 40 seconds. Your program should have user input that is the number of seconds to convert, and then use that number in your calculations. Your output should be displayed as "*The time is X hours, Y minutes, and Z seconds*".

2. Take one sheet of paper, and fold it in half, then fold it in half again, and again, and again. Can you fold it 30 times? Pretending that you can (you probably won't be able to fold it more than eight times), how thick would it be after 30 times? Assume the paper is 1/200 cm thick. Write a program to solve this puzzle. Your program should have one user input that is the number of folds. The program should display the thickness of the paper in centimeters in the following way: "*The thickness after X folds will be Y cm*". (There are several, equally correct, ways to solve this problem. To recognize the pattern, you may want to write out by hand the answer for 0,1, 2, 3, 4, 5, 6, … n folds.)

3. The U.S. Census provides information about the current U.S. population as well as approximate rates of change. Three rates of change are provided:
    a. There is a birth every 7 seconds
    b. There is a death every 13 seconds
    c. There is a new immigrant every 35 seconds

    Using those three rates of change, and a current U.S. population of 307,357,870, write a program to calculate the U.S. population in exactly one year (365 days). Your program should output the result in a nicely formatted print statement: "The population will be X". The correct population is 310,338,194.

    Hints: You will need to calculate the number of seconds in one year. Also, a population cannot have fractional people, e.g there can be 10 people, but not 10.2 people.