Assignment 8

Due on: November 3rd, Sunday 6pm
No interview grading

*All problems should be done just in main()

Task 1: C++ practice problems

1. Write a program named ***task1_1.cpp*** that asks the user to input an integer that represents a length of time in seconds. The program should then output the number of hours, minutes, and seconds that corresponds to that number of seconds. For example, if the user inputs 50391 total seconds then the program should output 13 hours, 59 minutes, and 51 seconds.

2. The Babylonian algorithm to compute the square root of a number ***n*** is as follows:

   I. Make a ***guess*** at the answer (you can pick n/2 as your initial guess).
   II. Compute ***r = n / guess***
   III. Set ***guess = (guess+ r) / 2***
   IV. Go back to step 2 for as many iterations as necessary. The more that steps II and III are repeated, the closer ***guess*** will become to the square root of ***n***.

   Write a program named ***task1_2.cpp*** that inputs an integer for ***n***, iterates through the Babylonian algorithm five times, and outputs the answer as a double to two decimal places. Your answer will be most accurate for small values of ***n***.

3. One way to measure the amount of energy that is expended during exercise is to use metabolic equivalent (MET). Here are some METS for various activities:

   Running 6 MPH: 10 METS
   Basketball: 8 METS
   Sleeping: 1 MET

   The number of calories burned per minute may be estimated using the formula:

   Calories/Minute = 0.0175 * MET * Weight (kg)

   Write a program named ***task1_3.cpp*** that asks the user to enter a subject's weight in pounds, the number of METS for an activity, and the number of minutes spent in that activity, and then outputs the estimate for total number of calories burned. One kilogram is equal to 2.2 pounds (you should declare this conversion factor as a constant.)

4. Write a program that prints the numbers from 1 to 170. But for multiples of three, print "Habit" instead of the number and for the multiples of seven, print "rabbit". For numbers which are multiples of both three and seven, print "HabitRabbit". Name your source file ***HabitRabbit.cpp***.

Expected output:
1
2
Habit
4
5
Habit
Rabbit
8
Habit
10
11
Habit
...
20
HabitRabbit
22
...

Task 2: Increase the sum of dice values while avoiding three

Write a program named ***task2.cpp*** that displays the sum of dice values when the game is over. Here, the dice value is one of $\{1, 2, 3, 4, 5, 6\}$ and can be determined using the `rand()` function. The program asks the user to input a single boolean variable to indicate whether it's a human mode or a computer mode (1 = human mode, 0 = computer mode). Plays out the entire turn until the player either rolls a 3 or chooses to hold. The program should then display the turn total as an integer value. The turn total is the sum of the dice values before the end of the game. If the game ends because the dice value is 3, the turn total is 3. In the human mode, the program asks the user to input a single boolean variable to indicate whether it's a roll or a hold (1 = roll, 0 = hold) every trial. The computer player should:

- Roll until it reaches at least some minimum turn total

- Have some randomness in its decision making

- Be capable of rolling or holding past the minimum

Instructions to submit your Assignment

Please zip all your source code files, make the file name <first_name>_<last_name>.zip and upload it on Moodle. Remember that your file must be free of errors upon compiling it with standard g++ compiler. Failing to do so will result in deduction of points from your total score for this assignment. Keep in mind the Honor code and ensure that you do not violate any of the rules it entails.