CSci 1113: Introduction to C/C++
Programming for Scientists and Engineers
Homework 1
Spring 2020

Due Date: Friday, February 14, 2020 before 11:55pm.

Instructions: This is an individual homework assignment. There ~~is one~~ **are two** problem worth 20 points **each**. Solve the problem below by yourself (unlike the labs, where you work collaboratively), and submit the solution as a C++ source code file. Here are a few more important details:

1. Unlike the computer lab exercises, this **is not** a collaborative assignment. You must design, implement, and test the solution to each problem on your own without the assistance of anyone other than the course instructor or TAs. In addition, you may not include solutions or portions of solutions obtained from any source other than those provided in class: examples from the textbook, lectures, or code you and your partner write to solve lab problems. Otherwise obtaining or providing solutions to any homework problems for this class is considered academic misconduct. See the "collaboration rules" file on the class website page for more details, and ask the instructor if you have questions.

2. Because all homework assignments are submitted and tested electronically, the following are important:
   - You follow the naming conventions mentioned at the end of the problems.
   - You submit the correct file(s) on gradescope (https://www.gradescope.com/) by the due deadline.
   - You follow the example input and output formats exactly given in each problem description.
   - **Regardless of how or where you develop your solutions, your programs compile and execute on gradescope computers (which run Linux/Ubuntu operating system like the cselabs machines).**

3. The problem descriptions will usually show at least one test case and the resulting correct output. However, you should test your program on other test cases (that you make up) as well. Making up good test cases is a valuable programming skill, and is part of ensuring your code solution is correct.

Problem A: Complex grades (20 points)
Output grades that incorporate the +/- system after the grade letter (no rounding up, so 93.000 is an A, while 92.9999 is an A-). Ask the user first if they want "rough" grades or "finer" grades. The rough grades should just the letter grade without +/- (so 90 is an A and 89 is a B). The fine grades should add "+" or "-" after the grade letter when appropriate. You may assume we always enter either 'f' or 'r' correctly to choose the appropriate type of grades.

Update: we are using the grading scale on the syllabus (taking the higher grade on ties):

```
93.0% -- 100.0%   A
90.0% --  93.0%   A-
87.0% --  90.0%   B+
83.0% --  87.0%   B
80.0% --  83.0%   B-
77.0% --  80.0%   C+
```

```
73.0% --  77.0%   C
70.0% --  73.0%   C-
67.0% --  70.0%   D+
60.0% --  67.0%   D
 0%   --  60.0%   F
```

Hint: There is a smart way that can save you a lot of typing...

(Note: the format has been updated to make it easier for the TAs to grade.)

Example 1 (user input is underlined):

```
Fine or Rough grades? (f/r):
r
Enter a score:
90
A
```

Example 2 (user input is underlined):

```
Fine or Rough grades? (f/r):
f
Enter a score:
90
A-
```

Example 3 (user input is underlined):

```
Fine or Rough grades? (f/r):
f
Enter a score:
1000
A
```

Example 4 (user input is underlined):

```
Fine or Rough grades? (f/r):
f
Enter a score:
87.5
B+
```

Example 5 (user input is underlined):

```
Fine or Rough grades? (f/r):
f
Enter a score:
62
D
```

When you are done, name the source code file hw1A.cpp. Then log into gradescope and upload your file for the "Homework 1A" submission. **If you name your file incorrectly it will be unable to compile and run your code, so you will fail all test cases**. You may submit cpp files as many times as you want until the deadline to try and fix the code if you fail a test case. Following rigorous naming conventions and using test cases are something computer programmers often must do in "real life" programming, and so submitting your program with the correct name and functionality is part of doing this assignment correctly.

Problem B: Zodiac signs (20 points)
Have the user input their birthday month/day in the format (see examples): mm/dd. Then tell them their zodiac sign. The zodiac signs we will use for grading are listed here: http://www.psychicguild.com/horoscopes_explained.php

You **DO** need to check that the month is valid. You may assume the day will be valid.

Aries: March 21 - April 19
Taurus: April 20 - May 20
Gemini: May 21 - June 20
Cancer: June 21 - July 22
Leo: July 23 - August 22
Virgo: August 23 - September 22
Libra: September 23 - October 22
Scorpio: October 23 - November 21
Sagittarius: November 22 - December 21
Capricorn: December 22 - January 19
Aquarius: January 20 - February 18
Pisces: February 19 - March 20

(Hint: you read a very similar input format in a lab problem...)

Example 1 (user input is underlined):
```
Enter the month/day when you were born:
1/2
Capricorn
```

Example 2 (user input is underlined):
```
Enter the month/day when you were born:
11/22
Sagittarius
```

Example 3 (user input is underlined):
```
Enter the month/day when you were born:

111/222
Invalid month
```

When you are done, name the source code file hw1B.cpp. Then log into gradescope and upload your file for the "Homework 1B" submission. **If you name your file incorrectly it will be unable to compile and run your code, so you will fail all test cases**. You may submit cpp files as many times as you want until the deadline to try and fix the code if you fail a test case. Following rigorous naming conventions and using test cases are something computer programmers often must do in "real life" programming, and so submitting your program with the correct name and functionality is part of doing this assignment correctly.