Collaboration Policy

Collaboration between students on programming assignments is **NOT** allowed under any circumstances - you may not even discuss your work with others; you may **NOT** get coding assistance, or copy code, from **ANY** outside source (*books, web sites, current or past students, your own previous submissions if you are repeating the course, etc.*). We test every submission for copying and when we find it we treat it as flagrant academic dishonesty on the part of **all** parties involved, regardless of their roles as provider or consumer. The penalty is generally instant failure in the course, in addition to the usual sanctions applied by Student Judicial Affairs.

As you can see from the submission header below, we ask you to essentially take an oath that the code you submit is *entirely your own work*.

At the same time, we certainly understand very well that you will frequently need help completing your programming assignment, so we provide channels where you may get that help in a way which will strengthen your programming skills: instructor and TA office hours, and the discussion forums where you can ask all the questions you want about the assignment, and even help others with the understanding that you have gained (but not, of course, with any actual code).

Assignment Submission Instructions

Your assignment must be submitted as a *simple text file* named *main.cpp* Files in ANY other format (MS Word document, etc.) or with ANY other name (main, main.doc, main.txt, etc.) will *not* be graded.

Submit your work via the appropriate iLearn assignment link, making sure you use the correct link and click on the attach button

We strongly recommend that:

- 1. you submit at least <u>6 hours</u> before the deadline, even if you haven't completed the program you can resubmit as often as you like, and we will only grade the final submission
- 2. once you have submitted your final attempt, go back and download it back to your system: then run it just to make absolutely sure that it really is the file you intended to submit!

You are budding professionals, and are expected to take full responsibility for abiding by the requirements of a contract.

The **only reason we will ever accept** for a missed deadline is if the system administrators inform me that either the iLearn system or the CS department servers were off-line for a significant period around the time of the deadline (*In which case we will probably have notified you of alternative procedures*).

Remember to include the following header information at the top of your program (DO NOT REMOVE THE // at the beginning of each line, these tell the compiler to ignore these lines)

```
// Course: CS 10 <quarter & year>
11
// First Name:
// Last Name:
// Course username: <enter the username you use to login in the lab>
// Email address: <enter your cs or UCR student email address here>
11
// Lecture Section: <e.g. 001>
// Lab Section: <e.g. 021>
// TA:
11
// Assignment: <assn1, hw2, lab3, etc.>
11
// I hereby certify that the code in this file
// is ENTIRELY my own original work.
11
// _____
```

NOTE: This header MUST appear at the top of EVERY file submitted as part of your assignment (don't forget to fill in *your* details and remove the <> brackets!!).

Copy & paste this header into your file then update personal details.

Assignment Specifications

A ten digit ISBN number uses a checksum as its last digit to verify the first nine digits are valid. Before 2007, all ISBN numbers were composed like this, such as: 0-20-508005-7 or 1-234-56789-X. The first nine digits are assigned by a book's publisher and the last digit is calculated by weighted sum. The X replaces the checksum value of ten as ten cannot be represented by a single digit.

Your Assignment

You must write a program that outputs this checksum value given the first nine digits of the ISBN number. You must utilize the checksum algorithm to properly calculate this value.

ISBN Checksum Algorithm

To compute the weighted sum we start from the left most digit. The sum is computed by adding the first digit plus two times the second digit plus three times the third, etc. all the way to nine times the ninth digit.

Next we take the weighted sum and calculate the remainder after it is divided by eleven. (i.e. weighted sum modulo eleven)

Finally we take this checksum digit and convert it if needed (10 turns into X). We will omit this step and just output the value from 0 through 10 without any conversion.

Input/Output Requirements

Your main function must read in a <u>single</u> integer representing the first 9 digits of an ISBN. Then output ONLY the checksum value. Again, output a 10 not an X if the remainder is 10.

Testing

ISBN (first 9-digits)	Checksum Output Value
123456789	10
00000000	0
987654321	0
047147063	5

Example Run

The input file contains a single nine digit integer, 123456789.

[user@well]\$ g++ main.cpp [user@well]\$./a.out < input.txt 10 [user@well]\$

Marking Guidelines

- Program reads in a single nine digit integer
- Program outputs properly calculated value from zero to ten.

Basic Style Reminders

- indentation all lines inside the main function should be indented 2-4 spaces
- spacing blank lines should be used to separate logic blocks
- no line wraps no line of code should have more than 80 characters
- "meaningful" variable names
- o comments preceding each logic block of code should have a brief explanatory comment