

CS2311 Assignment 2 – Calculation with large numbers

Deadline: 27-Nov-2013 (Wed) 1:00pm

1. Background of Assignment

1.1 Problem Statement:

C++ has the built-in integer type **int**, which is usually good enough for most arithmetic operations. However, being limited in storage size, a type **int** variable can only store numbers in range -2147483648 to +2147483647. If the magnitude of the input number (or the result of calculation) is too large, **overflow** will occur and results in wrong answer.

In this assignment, you will build your own number type, using a **character array** to emulate a normal number. By representing each digit as individual array cell, your program can perform **addition** and **subtraction** of very big numbers, which was not possible with classical **int** type.

1.2 Techniques assessed:

String, CString Library, character array, loops, conversion between ASCII and number, functions and parameter passing.

2. Program Requirement

2.1.1 Basic Requirement - Integer Arithmetic between two numbers (50%)

You'll write a program which reads in an arithmetic expression, performing either addition or subtraction between 2 non-negative integers. Both integers are expressed in decimal, with no more than 50 digits each. You may assume the numbers are always expressed in its simplest form (*e.g. 123 will NEVER be written as 0123 or 00123*). The program supports only 2 types of operator: either Add (+) or Subtract (-). The operator is always separated from the numbers with at least 1 space character. After calculation, the program will print the answer in the form:

<number1> <operator> <number2> = <answer>

Similarly, the output number should be expressed in its simplest form. You may refer to the next session for sample input/output. (*Note: If your program generates extra output or does not follow the output format, the PASS3 system will consider your program wrong (as in your tutorials)*)

2.1.2 Sample Input/Output for Integer Arithmetic

Input	123456 + 789012
Output	123456 + 789012 = 912468

Input	999999 - 555
Output	999999 - 555 = 999444

Input	10000 - 1
Output	10000 - 1 = 9999

Input	1010 + 101
Output	1010 + 101 = 1111

Input	123 - 4567
Output	123 - 4567 = -4444

2.2.1 Advanced Requirement - Arithmetic between two real numbers (30%)

In this session you will enhance your program in 2.1 such that it could handle addition or subtraction between two non-negative real numbers. Besides the integer part (*which is up to 50 places*), the number could also contains the **decimal dot** as well as the fractional part (*also up to 50 places*). The input may or may not be in its simplest form (*e.g. 0.123 could be expressed as .123 or 00.12300*) but the output should be simplest, without unneeded decimal dot and redundant zero prefix or suffix.

2.2.2 Sample Input/Output for Real Number Arithmetic

Input	0.123 + 10.111
Output	0.123 + 10.111 = 10.234

Input	1234.5678 - 204.507
Output	1234.5678 - 204.507 = 1030.0608

Input	1 - 0.1
Output	1 - 0.1 = 0.9

Input	0103.406080 + 20.050709000
Output	0103.406080 + 20.050709000 = 123.456789

Input	100 - 234.56
Output	100 - 234.56 = -134.56

3. Marking

The mark of assignment is based on the level of achievement as following:

Basic Requirement (50%)

Input contains two non-negative integers in simple form.

Advanced Requirement (30%)

Input contains two real numbers possibly with fractional part.

Programming style and efficiency (20%)

Programming styles like indentation, proper naming of variables, efficiency of the program and proper use of functions & parameter passing will be marked in this part.

4. Remarks

- | In the PASS3 system, there will be separate entries for Basic Requirement and Advanced Requirement respectively. Depending on the available time, students may wish to challenge both or only the basic one.
- | Students may “Test” the assignment as many times as he/she wish, but please click on the “Submit” button for the final version.
- | Students may use the same or different C++ sources for the basic and advanced levels. Note that “Submit” in basic level does not imply automatic submission to the advanced level (and vice versa). Students need to submit two times to the entries respectively.
- | Note that no manual judgment or code change will be applied to the marking of program correctness. If you submit source code which is non-compilable in PASS3, or if your code output extra content such that PASS consider your program wrong, you will get zero mark in correctness. (*i.e. maximum score is 20% from programming style*)
- | Plagiarism is strictly prohibited in this course. Cases detected by the PASS3 system will be given zero mark (regardless of the program correctness) and subjects to further punishment.
- | Debugging and testing are key activities of programming. Student helpers and Instructors will not be responsible for these parts. However, you may discuss with them for hints and advices. In fact, able to locate bugs and fix them will help you write a correct code in the future.
- | Students in need may visit the **Programming Clinic**, which is a Q&A helpdesk service by senior students. For information, let's refer to: <http://courses.cs.cityu.edu.hk/clinic>