Programming Design, Spring 2013
Homework 10

Instructor: Ling-Chieh Kung
Department of Information Management
National Taiwan University

To submit your work, please upload the following one file to the online grading system PDOGS at http://stella.im.ntu.edu.tw/online-judgement/.

1. A CPP file for Problem 1 (to the PD009 section).

NO hard copy and NO late submission. The due time of this homework is ***1:00pm, May 20, 2013***.

Problem 1

(100 points) Please write a C++ program according to the following instructions.

What should your program do

In this homework, you need to process strings, find numbers embedded in strings, and calculate their sum. Each line of the input data contains a string, which consists of several "words". Between two words, there is a white space. A word may or may not be a number, which is defined according to the following rule:

- A number must start with either a digit (0, 1, 2, ..., or 9), a plus sign (+), and a minus sign (−).
- A character that is not the first one can only be either a digit or a decimal point (.).
- Among all the characters that are not the first one, there can be at most one decimal point.

Please note that a number cannot start with a decimal point. All words that do not satisfy the above rule are not numbers. Your task is to find all the numbers and sum them up. It is allowed that a number contains several leading zeros. In this case, these zeros should be removed (when the following character is not a decimal point) or combined into just one (when the following character is a decimal point) when evaluating the value of the number. Words that are not numbers should be ignored when you calculate the summation. Once you calculate the sum, output it and then output a newline character or object. If there is no number in a line, the output should be zero.

When you output a number, simply use `cout` directly without any output manipulator.

Below are some examples:

- Input: `123 456`. Output: `579`.
- Input: `1.23 4.5.6 10`. Output: `11.23`. Please note that `4.5.6` is not a number.
- Input: `12 -150 +6`. Output: `-132`. All these three words are numbers.
- Input: `123 613 this is 5`. Output: `741`. Your program should simply ignore `this` and `is`.
- Input: `01 002 3 4 05 + 6 - 7 8 i g * .9`. Output: `36`. Please note that the value of `01` is 1 and that of `002` is 2. Please also note that `.9` is not a number.
- Input: `00000.1`. Output: `0.1`.
- Input: `there is no number`. Output: `0`.

Some suggestions

Below are some suggestions that may be helpful. You certainly do not need to follow all of them if you have other ways that work.

1. When you read the input data, do not use `cin >>`, because you have no idea how many words are in a line contain and whether the next word is a number or not. You are suggested to first read the whole line into a C++ string and then split the string into several substrings. To read a line without being interrupted by white spaces, you may refer to

   http://www.cplusplus.com/reference/string/string/getline/.

2. How to find all those substrings? You are suggested to use the function `find()` provided in `<string>`. Read the slides, any book you have, or the following web page

   http://www.cplusplus.com/reference/string/string/find/.

3. Once you are able to split a line into multiple words, you need to determine whether a word is a number. To do so, you may write a loop to check each character. The function `isdigit()` provided in `<cctype>` may help. The following web page

   http://www.cplusplus.com/reference/cctype/isdigit/

   contains a basic introduction.

4. Once you have a number, you need to get its value. This can be done by combining `c_str()`, which is a member function of the class `string`, and `atof()`, which is provided in `<cstdlib>`. The former (as well as all the members of the class `string`) is introduced on

   http://www.cplusplus.com/reference/string/string/

   and the latter on

   http://www.cplusplus.com/reference/cstdlib/atof/.

Please note that all the above functions are introduced somewhere in the slides used in lectures.

Grading criteria

Your program will be graded based on the following criteria:

- 70% of your grades for this program will be based on the correctness of your output. The online grading system will input a set of testing data, which includes 35 lines of strings. You may only see the grades of running your program on these data but cannot see the inputs and outputs. The 35 output lines count for 70 points, i.e., 2 points for each line.
- 30% of your grades for this program will be based on how you write your program, including the logic and format. Please try to write a robust, efficient, and easy-to-read program.