Programming Design, Spring 2013
Homework 09

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 PD008 section).

NO hard copy and NO late submission. The due time of this homework is ***1:00pm, April 29, 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 implement an address book which stores at most 100 contacts' information. For simplicity, let's assume that for each contact we only need to store three pieces of information: sex (a character which may be either "F" or "M"), height (an integer in cm), and weight (an integer in kg). In each line of the testing data, there will be an instruction, which may be one of the following three:

1. Insertion: If the first character of a line is "I", you will then be given one contact's information, first a character for sex, then a three-digit integer for height, then a two- or three-digit integer for weight. These four pieces of information are separated by three white spaces. You need to store this contact into your address book in a sorted manner, i.e., at any time your contacts must be sorted first by the sex (females and then males), then by the height (small to large), and finally by the weight (small to large). In other words, once you obtain the information of a new contact, you need to find the correct position in the address book to insert the contact to keep contacts sorted.
2. Looking up: If the first character of a line is "L", it will be followed by an integer n, and you will be asked to display the information of the contact that is currently placed at the nth position. Between "L" and n there is one white space. The contact's information should be displayed as `(sex, height, weight)`, where `sex`, `height`, and `weight` should be replaced by the contact's corresponding information. Please note that there is a white space following each comma.
3. Summarization: If the first character of a line is "S", it will be followed by another character, which may be "S", "H", and "W".
   (a) If the second character is "S", it will be followed by the third character, which may be "F" or "M". If the third character is "F", display the number of females in the current list; otherwise, display the number of males.
   (b) If the second character is "H" or "W", it will be followed by an integer m in cm or kg, respectively. For the case with "H", display the number of contacts whose height is at least m. For the case with "W", display the number of contacts whose weight is at least m.

   Between the first two inputs, there is a white space. Similarly, between the last two inputs, there is a white space.

In the case of insertion, you should display nothing. If the other two cases, you should display the desired information in one line. Therefore, after you output something according to the above rule, you need to append a new line character or object at the end.

Below is one sample input:

```
I F 160 40
I M 175 80
I F 170 50
I F 165 42
L 3
L 2
I F 160 43
L 3
S S F
S H 170
S W 100
```

Note that after the fourth instruction, the list is

(F, 160, 40), (F, 165, 42), (F, 170, 50), (M, 175, 80)

and after the seventh instruction, the list is

(F, 160, 40), (F, 160, 43), (F, 165, 42), (F, 170, 50), (M, 175, 80)

The expected output is

```
(F, 170, 50)
(F, 165, 42)
(F, 165, 42)
4
3
0
```

Grading criteria

First of all, the TAs will open your CPP file and check whether you implement the program with `struct`. If not, you will get zero point with no exception (i.e., no matter how many points you get through the online grading system). If yes, your grades will be determined in the following way:

- 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 some lines of insertion and 35 lines of looking up and summarization. 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.