Programming Design, Spring 2016
Homework 6

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

Please upload one PDF file for Problem 1 and two CPP files for Problems 2 and 3 to PDOGS at http://pdogs.ntu.im/judge/. Each student must submit her/his individual work. No hard copy. No late submission. The due time of this homework is ***2:00 am, April 4, 2016***. Please answer in either English or Chinese.

Before you start, please read Sections 5.20–5.22 and Chapter 19 of the textbook.[1]

The TA who generates the testing data and grades this homework is ***Chien Huang***.

Problem 1

(20 points) Design an algorithm and express your algorithm with pseduocodes for the following tasks. Please also write relevant comments.

(a) (5 points) Given a sequence of numbers $(x_1, x_2, ..., x_n)$, find an index $i^* \in \{1, 2, ..., n-2\}$ such that

$$x_{i^*} + x_{i^*+1} + x_{i^*+2} \geq x_i + x_{i+1} + x_{i+2} \quad \forall i \in \{1, 2, ..., n-2\}.$$

(b) (5 points) Given ten numbers 1, 2, ..., 10, list all the possible ways of selecting three numbers out of the ten. For each way, list the three numbers from small to large.

(c) (10 points) Given a set of n jobs, let b_j be the benefit of completing job j and w_j be the number of minutes needed to complete job j. Moreover, there are m persons, each of them has K minutes to do these jobs. Allocate these jobs to these persons to minimize the difference of benefits obtained by the two persons earning the highest and lowest benefits.

Note. This problem is said to be *NP-hard*, which means most researchers believe that there is no polynomial-time algorithm that finds an optimal solution. Therefore, your algorithm does not need to find an optimal solution. However, it MUST be a polynomial-time algorithm, and its steps MUST make sense in finding a near-optimal solution.

Problem 2

(40 points) A baseball batter records his batting records in the past n *plate appearances* as x_1, x_2, ..., and x_n.[2] $x_i \in \{-1, 0, 1, 2, 3, 4\}$, where -1 means a *walk*, 0 means an *out*, 1 means a *single*, 2 means a *double*, 3 means a *triple*, and 4 means a *home run*. Let's assume these are the only outcomes of his batting. Singles, doubles, triples, and home runs are all called *hits*. The *total bases* is the total number of bases earned by hits, where a single, a double, a triple, and a home run gives 1, 2, 3, and 4 bases. The number of *at bats* is defined as the number of plate appearances minus the number of walks.

He wants to calculate three performance measurements of batting:

1. *AVG* (batting average): The number of hits divided by the number of at bats.
2. *OBP* (on-base percentage): The number of hits and walks divided by the number of plate appearances.

[1] The textbook is *C++ How to Program: Late Objects Version* by Deitel and Deitel, seventh edition.
[2] Please see `http://ppt.cc/UBTO7` for more information.

3. *SLG* (slugging percentage): The number of total bases divided by the number of at bats.

Please help him calculate the three performance indicators to the third digit after the decimal point (e.g., 0.345, 0.292, 1.124, etc.). The fractional parts starting from the fourth digit should be truncated (e.g., 0.3452, 0.3455, and 0.3459 are all considered as 0.345). Note that while AVG and OBP are within 0 and 1, SLG is within 0 and 4.

Hint. The functions `setprecision` and `fixed` in the C++ standard library `<iomanip>` may be helpful. Please see `http://www.cplusplus.com/reference/iomanip/setprecision/` and `http://www.cplusplus.com/reference/ios/fixed/` for more details.

Input/output formats

There are 15 input files. In each file, there are n integers x_1, x_2, ..., and x_n. Two consecutive integers are separated by a white space. It is known that $x_i \in \{-1, 0, 1, 2, 3, 4\}$ and $1 \leq n \leq 500$. For example, the following input

```
-1 -1 0 1 0 2 0 -1 4
```

means that the players historical records are walk, walk, out, single, out, double, out, walk, and home run.

Given this input, your program should calculate the AVG, OBP, and SLG, and print each of them to the third digits after the decimal point. For the example above, your output should be

```
0.500 0.666 1.166
```

because AVG $= \frac{3}{6} = 0.5$, OBP $= \frac{6}{9} \approx 0.666$, and SLG $= \frac{7}{6} \approx 1.166$. Note that the fractional parts starting from the fourth digit should be truncated.

What should be in your source file

Your .cpp source file should contain C++ codes that will both read testing data and complete the above task. For this problem, you are allowed to use only techniques covered so far. NO other techniques are allowed. Finally, you should write relevant comments for your codes.

Grading criteria

- 30 points will be based on the correctness of your output. PDOGS will compile your program, feed testing data into your program, and check the correctness of your outputs. Each fully correct set of outputs gives you 2 points.
- 10 points 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.

Problem 3

(40 points) There are two boxes, M white balls, and N black balls in front of you. You need to put m white balls and n black balls into box 1 and $M - m$ white balls and $N - n$ black balls into box 2. There are two restrictions on your choices of m and n: $m_L \leq m \leq m_H$ and $n_L \leq n \leq n_H$, respectively. After you put the balls into boxes, your opponent will randomly select a box and then randomly draw a ball from the selected box. If the ball drawn by your opponent is white, you win; otherwise, you lose. If the box has no ball, you also lose.

Given M, N, m_L, m_H, n_L, and n_H, please find (m, n) to maximize your chance of winning. If multiple pairs of (m, n) result in the same maximum winning rate, choose the one with the smallest m; if there is still a tie, choose the one with the smallest n.

Input/output formats

There are 15 input files. In each file, there are six integers M, N, m_L, m_H, n_L, and n_H. Two consecutive integers are separated by a white space. It is known that $1 \leq M \leq 100$, $1 \leq N \leq 100$, $0 \leq m_L \leq m_H \leq M$, and $0 \leq n_L \leq n_H \leq N$. For example, the following input

```
50 50 0 50 1 25
```

means that you have 50 white balls, 50 black balls, and you are required to choose m and n subject to $0 \leq m \leq 50$ and $1 \leq n \leq 25$.

Given this input, your program should print out m and then n so that (m, n) maximizes the winning chance. For the example above, your output should be

```
12 1
```

where the winning probability 0.679929 (that should not be printed out) is the maximum.

What should be in your source file

Your .cpp source file should contain C++ codes that will both read testing data and complete the above task. For this problem, you are allowed to use only techniques covered so far. NO other techniques are allowed. Finally, you should write relevant comments for your codes.

Grading criteria

- 30 points will be based on the correctness of your output. PDOGS will compile your program, feed testing data into your program, and check the correctness of your outputs. Each fully correct set of outputs gives you 2 points.
- 10 points 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.