Programming Design, Spring 2015
Lab Exam 2

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

Submission and grading. In this exam, there are three problems. You need to write a C++ program for each problem. 100% of your grades will be based on the correctness of your outputs. The online grading system will input in total 50 sets of testing data and then check your outputs. These 50 sets count for 100 points, i.e., 2 points for each set. Before the due time of the exam, you may upload your programs multiple times. Only the last one you upload will be graded. Unlike what happens with your homework, you will not see the grading results during the exam. For each problem, you will only see the results for the online samples.

To submit your work, please upload the three .cpp files, one for each problem, to the online grading system at https://pdogs.ntu.im/judge/.

Just a reminder: Talking to any other person online, directly or indirectly, is definitely considered cheating.

Problem 1

(30 points) A retail store keeps all the purchasing orders it makes when procuring from manufacturers. In each purchasing order, there are several transactions. In each transaction, there are the following attributes:

- Manufacturer ID: A single capitalized English letter. There are no more than 26 manufacturers.
- Item ID: A positive integer between 1 and 50. There are no more than 50 kinds of items.
- Purchasing quantity: The number of units of the item purchased.
- Unit price: A positive integer between 1 and 100.

For example, $(\text{A}, 15, 1000, 8)$ is a transaction saying that 1000 units of item 15 are purchased from manufacturer A at a unit price \$8. It is clear that \$8000 dollars are paid to manufacturer A. The unit price for a single item may differ from transaction to transaction. Each purchasing order has its issue date. There is at most one purchasing order issued in a day.

Given a set of purchasing orders, the retail store owner now wants to find out the item that counts for the most purchasing costs. It also wants to find out the manufacturer that receives the most payment from it.

Input/output formats

The input consists of 15 files. At the beginning of each file, there is a positive integer n as the number of purchasing orders recorded in the file. You may assume that $1 \leq n \leq 100$. Starting from the second line, the n purchasing orders are provided. For each purchasing order, the first line contains four positive integers t, y, m, and d, which are the number of transactions, issuing year, issuing month, and issuing date. Each of the 2nd to the $(t+1)$th line contains four values a, b, c, and d, where a is the manufacturer ID, b is the item ID, c is the purchasing quantity, and d is the unit price. You may assume that $1 \leq t \leq 10$, a is a capitalized English letter, $1 \leq b \leq 50$, $1 \leq c \leq 10000$, and $1 \leq d \leq 100$. In each line, two consecutive numbers are separated by a white space. At the end of each line, there is a newline character.

As an example, below is an input file:

```
3
3 2015 5 1
A 3 100 4
B 4 80 12
C 16 105 6
2 2015 5 2
A 3 50 4
A 4 100 5
4 2015 5 3
B 1 50 2
C 10 80 2
D 15 10 2
D 16 15 4
```

We can see that there are three purchasing orders, issued on 2015/5/1, 5/2, and 5/3. In total there are 9 transactions. For manufacturers A, B, C, and D, the total payments are \$1100, \$1060, \$790, and \$80, respectively. For items 1, 3, 4, 10, 15, and 16, the total purchasing costs are \$100, \$600, \$1460, \$160, \$20, and \$690, respectively.

Your program should read each input file and then print out four values: the manufacturer ID with the largest total payment, that payment amount, the item ID with the largest total purchasing costs, and that purchasing cost. Two consecutive values should be separated by a white space. For the example provided above, the output should be

```
A 1100 4 1460
```

in a single line.

Problem 2

(40 points) Given a sequence of integers $(x_1, x_2, ..., x_n)$, a leaping sequence of starting position p and step size k is $(x_p, x_{p+k}, x_{p+2k}, ..., x_{p+mk})$, where m is a positive integer so that $p+mk \leq n$ and $p+(m+1)k > n$. For example, suppose the given sequence is $(1, -2, 3, -4, 5, -6, 7, -8, 9, -10)$, then choosing $p = 2$ and $k = 2$ gives us $(-2, -4, -6, -8, -10)$, choosing $p = 2$ and $k = 3$ gives us $(-2, 5, -8)$, and choosing $p = 6$ and $k = 5$ gives us (-6). The sums of the three leaping sequences are -30, -5, and -6, respectively.

For a given sequence of integers, we are interested in finding the maximum-sum leaping sequence within a set of step sizes $\{1, 2, ..., K)$. For each step size k, possible starting positions that are worth of trying are 1, 2, ..., and k. Therefore, given K, the number of leaping sequence that may need to be investigated is $1 + 2 + \cdots + K = \frac{K(K+1)}{2}$. As an example, for the sequence $(1, -2, 3, -4, 5, -6, 7, -8, 9, -10)$, if we are given any $K \geq 2$, the maximum-sum leaping sequence is always with step size $k = 2$ and starting position $p = 1$. As another example, for the sequence $(4, 7, -3, -8, 5, 4, 2, -2)$ with $K = 3$, the six leaping sequences under considerations are:

k	p	leaping sequence	sum
1	1	$(4, 7, -3, -8, 5, 4, 2, -2)$	9
2	1	$(4, -3, 5, 2)$	8
2	2	$(7, -8, 4, -2)$	1
3	1	$(4, -8, 2)$	-2
3	2	$(7, 5, -2)$	10
3	3	$(-3, 4)$	1

It is clear that the maximum-sum leaping sequence is obtained with $k = 3$ and $p = 2$.

In this problem, you will be given a sequence of length n and a positive integer $K \leq n$. You need to find a maximum-sum leaping sequence for a step size $k \leq K$ and starting position $p \leq k$. You then

output k, p, and the maximum-sum. If there are multiple maximum-sum leaping sequences, you only need to output the one whose k is the smallest; if there are still multiple, then among that maximum sum leaping sequences with the smallest k, you only need to output the one whose p is the smallest.

Input/output formats

The input consists of 20 files. In each file, there are two lines of integers:

- In the first line, there are two integers n and K, which are the number of values in the sequence and the maximum step size to consider. It is given that $1 \leq n \leq 10000$ and $1 \leq K \leq n$.
- In the second line, there are n integers x_1, x_2, ..., and x_n. They form the sequence to be considered. It is given that $-100000 \leq x_i \leq 100000$.

Your program should read each input file and then print out three integers for the maximum-sum leaping sequence: the step size k, starting position p and the maximum sum. For example, suppose we have

```
8 3
4 7 -3 -8 5 4 2 -2
```

as a file of input, the output should be

```
3 2 10
```

in a single file.

Problem 3

(30 points) Consider a network $G = (V, E)$ where V is the set of nodes and E is the set of undirected edges. In this problem, we assume that the graph is complete, i.e., there is exactly one edge between two nodes. Therefore, if the number of nodes is n, the number of edges must be $\frac{n(n-1)}{2}$. The edge connecting nodes i and j is denoted as $[i, j]$. For edge $[i, j]$, there is an associated edge weight $w_{ij} > 0$.

In this problem, let's consider nodes as cities, edges as roads, and edge weights as the time it takes to go through those roads. For cities and roads in the real world, sometimes traveling times do not satisfy the triangular inequality.[1] As an example, to move from Taichung to Hualien, passing through the Central Mountain Range may take longer time than going though the Taichung-Taipei-Hualien route.

Now, given a network, we would like to find the number of triangles that violate the triangular inequality. More precisely, we ask for all sets of three nodes i, j, and k, how many of them satisfies $d_{ij} + d_{jk} \leq d_{ik}$. Note that an equality counts!

Input/output formats

The input consists of 15 files. In each file, there are $n + 1$ lines of positive integers:

- In the first line, there is one single integer n, which is the number of nodes in the network. It is given that $3 \leq n \leq 100$. At the end there is a newline character.
- In the ith line, $i = 2, 3, ..., n + 1$, there are n integers $w_{(i-1),1}$, $w_{(i-1),2}$, ..., and $w_{(i-1),n}$. These are the traveling times from node $i-1$ to the other nodes. Two consecutive numbers are separated by a white space. At the end there is a newline character. It is given that $w_{ij} = w_{ji}$ and $0 < w_{ij} \leq 10000$ for all $i = 1, ..., n$, $j = 1, ..., n$, $i \neq j$. It is also given that $w_{ii} = 0$ for all $i = 1, ..., n$.

[1] The triangular inequality is a property satisfied by any triangle in geometry: Given any triangle, the sum of the lengths of any two edges is larger than that of the last edge.

Your program should read each input file and output the number of triangles that violate the triangular inequality. As an example, suppose you are given

```
4
0 2 4 6
2 0 8 3
4 8 0 5
6 3 5 0
```

as a file of input, the output should be

```
3
```

in a single file. The three triangles that violate the triangular inequalities are $\{1, 2, 3\}$, $\{1, 2, 4\}$, and $\{2, 3, 4\}$, respectively. Note that for the last triangle, the three edge weights are 3, 5, and 8. As $3 + 5$ is not greater than 8, this triangle also violates the triangular inequality.