Problem A
Awkward Group
Time Limit: 0.5 Seconds

A community P consists of n persons living in a small town. The persons in P may have friendly relationship with someone, but there are men whom they have never met. A closeness between any distinct persons x and y of P is expressed by a value $f(x,y)$. Here it is assumed that $f(x,y) = f(y,x)$.

A group F of persons in P, that is, a subset of P, is called an *awkward group* if the number of persons in F, denoted by $|F|$, is equal to neither n nor 1 and the maximum closeness between any two distinct persons in F is strictly less than any closeness between a person in F and one not in F, that is, if F satisfies the following conditions:

$$1 < |F| < n \quad \text{and} \quad \max\{\, f(x,y) \mid x \neq y, x \in F,\ y \in F \,\} < \min\{\, f(x',y') \mid x' \in F, y' \in P - F \,\}.$$

Given the values $f(x,y)$ for any distinct persons x and y in P, your program is to find the number of all the possible awkward groups of P.

For example, there are three persons x, y, and z in a community P. The groups F of P such that $1 < |F| < 3$ are $\{x,y\}$, $\{y,z\}$, and $\{z,x\}$. The values for the closeness between two persons are given as $f(x,y) = 8$, $f(y,z) = 3$, and $f(z,x) = 5$. Then the awkward group among the candidates is only $\{y,z\}$. So the number of all the possible awkward groups of P is 1.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing an integer, n ($1 \leq n \leq 1{,}000$), where n is the number of persons in the community P. The persons are represented by integers 1 to n. In the following $n-1$ lines, the values h_{ij} of the closeness are given. In the ith line, there are $n-i$ integers h_{ii+1}, h_{ii+2}, …, h_{in} separated by a single space, where h_{ij}, $j = i+1, \dots, n$, are the values $f(i,j)$ of the closeness between two persons i and j ($1 \leq i < j \leq n$ and $1 \leq h_{ij} \leq 10^6$).

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should contain an integer representing the number of all the possible awkward groups of P.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 3 8 5 3 4 1 3 5	1 2 3

6 4 2 5 1 6 5 10 7 8 9 2 4 3	

Problem B
Card Game
Time Limit: 0.1 Second

Alice is playing a simple but interesting computer game on her computer. The game plays as follows: when Alice starts the game, the computer shows n cards on the monitor in a line. Each card is associated with a number printed on the face and all the cards are faced up so that she can see every number printed on the cards. She and the computer repeatedly pick up a card in turn either from the front or from the rear of the line until there is no card in the line. She always starts the game. In other words, she always picks up a card first. The goal of the game for Alice is to maximize the sum of all the numbers printed on the cards she picks up. She completely understands a strategy to make the sum maximized. Definitely the computer will prevent Alice from making it.

Here is an example. There are four cards in a line and numbers on the cards are (4, 3, 1, 2). Alice first picks up the card of number 4 from the front of the line. Then the computer picks up the card of number 3 from the front. Then Alice picks up the card of number 2 from the rear. Finally, the computer picks up the last card of number 1. This is the best picking order for both Alice and the computer, and Alice makes the sum 6.

You are to make a program that computes the sum of all the numbers on the cards Alice selects among given n cards.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing an integer, n $(1 \leq n \leq 1{,}000)$, where n is the number of cards. In the following line, n numbers are given each of which denotes the number printed on the cards and is separated by a blank. Each number printed on the cards is between 1 and 10,000, inclusive.

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should contain a number which represents the sum of all the numbers printed on the cards Alice selects.

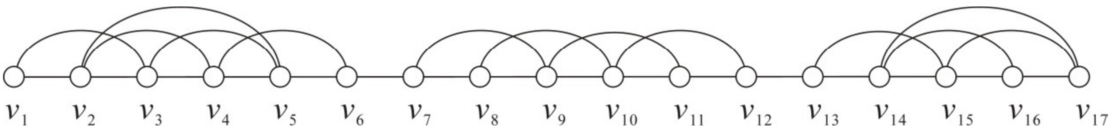
The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 4 1 2 5 2 9 1 1 1 1 2 2 2 2 2	6 8

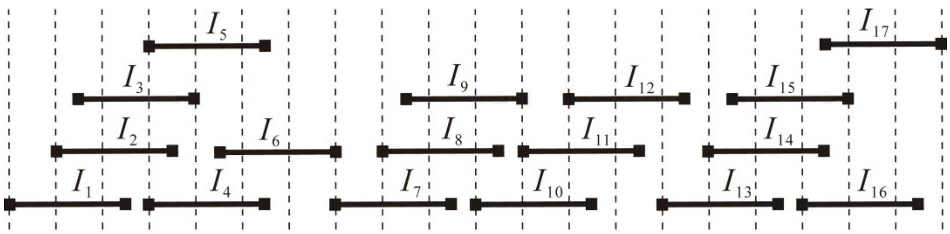
The 40th Annual ACM
International Collegiate Programming Contest
Asia Regional – Daejeon
Nationwide Internet Competition

Problem C
Consecutive Ordering
Time Limit: 0.1 Second

An interval graph is the intersection graph of family $F = \{I_1, I_2, \dots, I_n\}$ of closed intervals on the real line, where two vertices, v_i and v_j, are connected with an edge if and only if their corresponding intervals, I_i and I_j, intersect. The family F is usually called an interval representation for the graph. A unit interval graph is an interval graph with an interval representation in which all the intervals have the same length. Refer to Figure 1 for an example of a unit interval graph and its interval representation.



(a) A unit interval graph G.



(b) An interval representation of the graph G.

Figure 1. A unit interval graph and its interval representation.

The closed neighborhood $N[v_i]$ of a vertex v_i of a graph G refers to the set of vertices adjacent to v_i along with v_i itself, i.e., $N[v_i] = \{v_i\} \cup \{v_j : (v_j, v_i) \text{ is an edge of } G\}$. For the graph G shown in Figure 1(a), we have $N[v_1] = \{v_1, v_2, v_3\}$ and $N[v_5] = \{v_2, v_3, v_4, v_5, v_6\}$. Under the ordering $(v_1, v_2, \dots, v_{17})$ of the vertices of G in which the position of v_i, denoted by $\rho(v_i)$, is i for all v_i, the closed neighborhood $N[v_i]$ of every vertex v_i is *consecutive*, i.e., the positions of vertices in $N[v_i]$ are integers which follow one after another, without gaps, from smallest to largest. An ordering of the vertices of a graph is said to be *consecutive* if the closed neighborhood of every vertex of the graph is consecutive. For the graph G of Figure 1(a), the ordering $(v_1, v_2, \dots, v_{15}, v_{17}, v_{16})$ with v_{16} and v_{17} being switched, where $\rho(v_{17}) = 16$, $\rho(v_{16}) = 17$, and $\rho(v_i) = i$ for all $v_i \neq v_{16}, v_{17}$, is also consecutive. However, $(v_2, v_1, v_3, v_4, \dots, v_{17})$ is not a consecutive ordering because the closed neighborhood of v_5 is not consecutive.

Given n closed intervals of the same length, $I_1, I_2, \dots, I_n$, in a non-decreasing order of their left endpoints and an ordering $(v_{i_1}, v_{i_2}, \dots, v_{i_n})$ of the vertices of the unit interval graph defined on the n intervals, write a program to test whether the ordering is consecutive or not.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case begins with a line containing two positive integers n and l, where $n \leq 100{,}000$ and $l \leq 100{,}000{,}000$, respectively representing the number and the length of intervals. It

follows n lines containing, one by one, the left endpoints of intervals, $I_1, I_2, \ldots, I_n$, in which the left endpoint of I_i is always less than or equal to that of I_j if $i < j$. You may assume that the endpoints, left or right, of intervals are between $-100{,}000{,}000$ and $100{,}000{,}000$, inclusive. It follows n lines again, containing an ordering of the vertices of the unit interval graph defined on the n intervals, one vertex per line.

Output

Your program is to write to standard output. For each test case, print exactly one line that contains an integer indicating whether the given ordering is consecutive. If yes, the integer must be `1`; otherwise `-1`.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 6 5 0 2 3 6 6 9 1 2 3 4 5 6 6 5 0 2 3 6 6 9 2 1 3 4 5 6 4 10 11 12 31 32 3 4 2 1	1 -1 1

Problem D
Diameter
Time Limit: 0.1 Second

The computer network in a country forms a tree, in other words, there is exactly one path between any two nodes. The weight of an edge (j, k), $w(j, k)$, represents the time needed for sending a message from node j to node k. We assume $w(j, k) = w(k, j)$. The length of the path between two nodes s and t is the sum of the weights of edges in the path between s and t. The *diameter* of a tree network is the length of the longest path among the paths between any two nodes in the network. For example, the diameter of the tree network in Figure 1 is 30 by the path between two nodes 1 and 6. The diameter is related to the longest delay in communicating between nodes in a network. So, diameter is one of important parameters in computer networks.

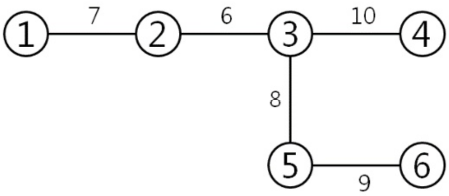


Figure 1. A tree network

Now, assume we can reduce the communication time of an edge by paying a cost to the edge: By paying a cost c to an arbitrary edge (j, k), we can reduce $w(j, k)$ to $max\{w(j, k) - c, 0\}$. Here, the cost c can be any non-negative real number. We want to find the minimum cost such that the tree network has diameter at most D. For example, the minimum cost of the network above for a target diameter D = 26 is 4; by paying a cost 4 to the edge (5, 6), the weight 9 is reduced to 5. For a target diameter D = 0, the minimum cost is 40 because we should reduce all the weights of edges to 0. And, for a target diameter D = 19, the minimum cost is 11.5 by paying a cost 3.5 to the edge (2, 3), paying a cost 0.5 to the edge (3, 4), and paying a cost 7.5 to the edge (3, 5).

You are to write a program, for a given tree network and a target diameter D, that computes the minimum cost.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing two integers n $(1 \leq n \leq 40{,}000)$ and D $(0 \leq D \leq 50{,}000)$, where n is the number of nodes in the tree network and D is a target diameter. In the following $n - 1$ lines, three integers representing an edge are given; the first and second integers represent end-nodes of the edge and the third integer represents the weight of the edge. Each edge weight is given as an integer between 1 and $50{,}000$, inclusive. The node numbers are given as integers between 1 and n, inclusive. Note that although the initial edge weight is given as an integer, the weight can be reduced to a real number by paying a cost of real value.

Output

Your program is to write to standard output. For each test case, print the minimum cost so that the resulting tree network has diameter at most D. Your output should contain the first digit after the decimal point, rounded off from the second digit.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 6 26 1 2 7 6 5 9 5 3 8 3 2 6 4 3 10 6 0 1 2 7 6 5 9 5 3 8 3 2 6 4 3 10 6 19 1 2 7 6 5 9 5 3 8 3 2 6 4 3 10	4.0 40.0 11.5

Problem E
Grid Aliens
Time Limit: 1 Second

In a planet all aliens $\{a_i\}$ live on points of $N \times N$ grid space. Some aliens like to live with the water and others do not. This tendency is called hydrophilic property. So each alien a_i has its own degree of hydrophilic property as a real number, h_i where $0 \le h_i \le 1$. An alien not living in boundary edges or corners has 4 neighbors (up, down, right, left) in grid. An alien living in edge (corner) points has 3(2) neighbors. Let the friendship of two adjacent neighbor aliens a_i and a_j be determined by the hydrophilic property h_i and h_j as the following:

$$friend(a_i, a_j) = 1 - |h_i - h_j|$$

If two adjacent neighbor aliens have the same degree of hydrophilic property, then the friendship will be perfect such as $friend(a_i, a_j) = 1$. Critical situation, $friend(a_i, a_j) = 0$, may happen if $h_i = 0$, $h_j = 1$. Note that $friend(x, y)$ is not defined among non-adjacent alien pair such as (*a*,*c*) and (*f*,*p*) in Figure 1.

Those aliens decided to set up separating fences to reduce the conflicts between different hydrophilic aliens group. Thus they first need to allocate grid space to two disjoint regions; W region for hydrophilic aliens and Q region for hydrophobic aliens (who dislike the watery environment and have little values of hydrophilic property). So they will set up a closed fence for Q regions against W regions to separate conflicting aliens. In Figure 1, dotted cycles denote the separating fences for Q. Aliens living on grid points $\{b, c, f, x, y\}$ are classified as hydrophilic aliens. And $\{a, d, e, p, q\}$ are aliens living in W region.

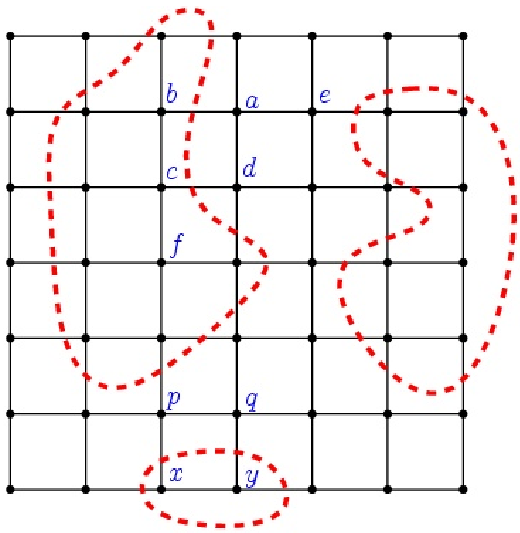


Figure 1. Three fences (dotted red) separate Q regions against R regions

There are so many ways to set up closed fences. When we build the separating fences, it is desirable to allocate the hydrophilic aliens to W region, and hydrophobic aliens to Q region. Also we should minimize the sum $friend(a_i, a_j)$ for $a_i \in Q$ and $a_j \in W$. Here $x \in Q$ means that an alien x is assigned to Q region. All aliens are classified either Q or W exclusively.

So we give a formal objective function K for this (W, Q) separation problem as following:

$$K = \text{maximize}\{\ \text{CostW} + \text{CostQ} - \text{CostWQ}\ \}, \text{ where}$$

$$\text{CostW} = \textstyle\sum_{a_i \in W} h_i, \quad \text{CostQ} = \sum_{a_j \in Q} (1 - h_j),$$
$$\text{CostWQ} = \textstyle\sum_{a_i \in W,\, a_j \in Q} friend(a_i, a_j), \text{ where } (a_i, a_j) \text{ is a grid edge.}$$

You should write a program to compute K given the hydrophilic properties of aliens on grid points.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing an integer N ($3 \leq N \leq 50$) denoting the grid size. In the following N lines, the real numbers denoting the hydrophilic property $h_{i,j}$ of the alien living on a grid position (i, j) are given as $N \times N$ matrix where $0 \leq h_{i,j} \leq 1$. The floating point $h_{i,j}$ has exactly two digits after decimal point.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should contain a real number K with two digits after decimal point.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 4 0.94 0.89 0.99 0.11 0.77 0.87 0.87 0.93 0.78 0.05 0.89 0.15 0.95 0.15 0.14 0.05 5 0.21 0.21 0.83 0.21 0.21 0.21 0.83 0.83 0.83 0.21 0.21 0.83 0.21 0.83 0.21 0.21 0.83 0.83 0.83 0.21 0.21 0.83 0.83 0.21 0.21	12.39 14.67

Problem F
Merging Files
Time Limit: 1 Second

A novelist Mr. Kim usually writes a book by chapters, where each chapter is saved in a different file. After finishing all chapters, he merges the files into one file in the order of the chapters for a new novel. He can only merge two files containing continuing chapters into a new intermediate file and keep merging the original files or intermediate files until he gets one final file. Assuming the cost (e.g., time) it takes for him to merge two files is exactly the same as the size sum of the two files, calculate the minimum cost to complete the final one file.

For example, let *C1*, *C2*, *C3* and *C4* be continuing chapter files of size 40, 30, 30 and 50 respectively. We could first merge *C2* and *C3* to get an intermediate file *X1*, where it requires 60 units of cost. Merging *C1* and *X1* to get *X2* requires 100 units. *X2* and *C4* could be merged, requiring 150 units, to obtain the desired file containing all chapters in sequence. Therefore the total units of cost is 60+100+150=310. Alternatively, we could first merge *C1* and *C2* getting *Y1*, then merge *C3* and *C4* getting *Y2* and finally merge *Y1* and *Y2* getting the desired file. This sequence of mergings requires 70+80+150=300 units of cost.

Given sizes of chapter files, write a program to calculate the minimum cost for merging the files into one.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of two lines. The first line contains a positive integer $\boldsymbol{K}$ $(\mathbf{3} \leq \boldsymbol{K} \leq \mathbf{500})$, denoting the number of chapters. The second line contains $\boldsymbol{K}$ positive integers which are sizes of chapters from 1 to $\boldsymbol{K}$. You may assume that sizes of chapters do not exceed 10,000.

Output
Your program is to write to standard output. Print exactly one line for each test case. For each test case, print the minimum cost for merging all chapters given.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 4 40 30 30 50 15 1 21 3 4 5 35 5 4 3 5 98 21 14 17 32	300 864

Problem F
파일 합치기
Time Limit: 1 Second

소설가인 김대전은 소설을 여러 장(chapter)으로 나누어 쓰는데, 각 장은 각각 다른 파일에 저장하곤 한다. 소설의 모든 장를 쓰고 나서는 각 장이 쓰여진 파일을 합쳐서 최종적으로 소설의 완성본이 들어있는 한 개의 파일을 만든다. 이 과정에서 두 개의 파일을 합쳐서 하나의 임시파일을 만들고, 이 임시파일이나 원래의 파일을 계속 두 개씩 합쳐서 소설의 여러 장들이 연속이 되도록 파일을 합쳐나가고, 최종적으로는 하나의 파일로 합친다.두 개의 파일을 합칠 때 필요한 비용(시간 등)이 두 파일 크기의 합이라고 가정할 때, 최종적인 한 개의 파일을 완성하는데 필요한 비용의 총 합을 계산하시오.

예를 들어, C1, C2, C3, C4 가 연속적인 네 개의 장을 수록하고 있는 파일이고, 파일 크기가 각각 40, 30, 30, 50 이라고 하자. 이 파일들을 합치는 과정에서, 먼저 C2 와 C3 를 합쳐서 임시파일 X1 을 만든다. 이 때 비용 60 이 필요하다. 그 다음으로 C1 과 X1 을 합쳐 임시파일 X2 를 만들면 비용 100 이 필요한다. 최종적으로 X2 와 C4 를 합쳐 최종파일을 만들면 비용 150 이 필요하다. 따라서, 최종의 한 파일을 만드는데 필요한 비용의 합은 60+100+150=310 이다. 다른 방법으로 파일을 합치면 비용을 줄일 수 있다. 먼저 C1 과 C2 를 합쳐 임시파일 Y1 을 만들고, C3 와 C4 를 합쳐 임시파일 Y2 를 만들고, 최종적으로 Y1 과 Y2 를 합쳐 최종파일을 만들 수 있다. 이 때 필요한 총 비용은 70+80+150=300 이다.

소설의 각 장들이 수록되어 있는 파일의 크기가 주어졌을 때, 이 파일들을 하나의 파일로 합칠 때 필요한 최소비용을 계산하는 프로그램을 작성하시오.

Input
프로그램은 표준 입력에서 입력 데이터를 받는다. 프로그램의 입력은 T 개의 테스트 데이터로 이루어져 있는데, T 는 입력의 맨 첫 줄에 주어진다. 각 테스트 데이터는 두 개의 행으로 주어지는데, 첫 행에는 소설을 구성하는 장의 수를 나타내는 양의 정수 K ($3 \leq K \leq 500$) 가 주어진다. 두 번째 행에는 1 장부터 K 장까지 수록한 파일의 크기를 나타내는 양의 정수가 K 개 주어진다. 파일의 크기는 10,000 을 초과하지 않는다.

Output
프로그램은 표준 출력에 출력한다. 각 테스트 데이터마다 정확히 한 행에 출력하는데, 모든 장을 합치는데 필요한 최소비용을 출력한다.

두 개의 테스트 데이터에 대한 입출력의 예는 다음과 같다.

Sample Input	Output for the Sample Input
2 4 40 30 30 50 15 1 21 3 4 5 35 5 4 3 5 98 21 14 17 32	300 864

Problem G
Monotone Walkway
Time Limit: 0.2 Seconds

There is a famous walkway which is a Jeju Olle trail course. The entrance is one of the westernmost positions of this walkway. Since the walkway is a simple path which consists of horizontal sections and vertical sections, a walker should turn 90 degree left or right at every corner of the path. Also, a walker can move from the entrance to the exit without moving from east to west, i.e., without decreasing the x-coordinate of one's current position. So, this path is called a *monotone walkway*. Figure 1 shows an example of a monotone walkway.

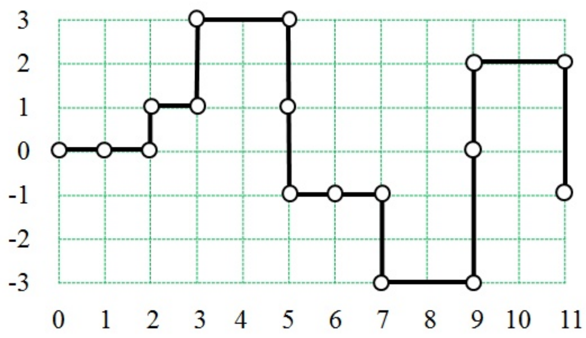


Figure 1. An example of a monotone walkway

There are n cafés along the walkway here and there. Specifically, there exist cafés on the entrance, the exit, and all corners of the path. Mr. Kim, a manager of Jeju Olle trail courses, has the location coordinates of all cafés. The coordinate of the entrance is always the origin $(0,0)$. He is going to number all cafés from 1 to n. The café located on the entrance is assigned the number 1. The remaining cafés are assigned the numbers according to the order they are visited. Traversing the walkway from the entrance to the exit, a café A which is visited before another café B is assigned the smaller number than the café B. In the walkway of Figure 1, the cafés located on the coordinates $(3,1)$ and $(9,0)$ are assigned the number 5 and 14, respectively, and the café on the exit is assigned the number 17. He wants to do his work without having to walk the path. To help him, you are to write a program which carries out the work of numbering the cafés.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing an integer, n ($2 \le n \le 100{,}000$), where n is the number of cafés. In the following n lines, each coordinate of the n cafés is given line by line. Each coordinate is represented by two numbers x and y, which are the x-coordinate and the y-coordinate of the café, respectively ($0 \le x \le 100{,}000$ and $-100{,}000 \le y \le 100{,}000$). The coordinate of the entrance is always $(0,0)$. Note that there are no two cafés which have a same coordinate. The last line of the input has an integer m ($1 \le m \le 10$) and m additional integers between 1 and n, inclusively, which are the café numbers.

Output

Your program is to write to standard output. Print exactly one line for each of m integers given as café numbers in each test case. The line should contain two integers x and y representing the coordinate of the café whose number is k for a given integer k.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 17 3 3 5 3 11 2 9 2 2 1 3 1 5 1 0 0 1 0 2 0 9 0 11 -1 9 -3 6 -1 7 -1 7 -3 5 -1 3 5 14 17 4 0 0 0 1 1 1 1 0 5 1 4 1 3 1	3 1 9 0 11 -1 0 0 1 0 0 0 1 1 0 0

Problem G
모노톤길
Time Limit: 0.2 Seconds

제주 올레길 코스로 유명한 산책로가 있다. 산책로의 입구는 산책로에서 가장 서쪽에 위치하고 있다. 이 산책로는 단순 경로이며 수평 구간과 수직 구간으로만 구성되어 있어 모든 코너에서 90 도 각도로 왼쪽 또는 오른쪽으로 회전만 할 수 있다. 또한 입구에서 출구 방향으로 걸어갈 때 동쪽에서 서쪽으로 이동을 전혀 하지 않아도, 즉, 보행자의 현재 위치를 나타내는 좌표의 x축 값이 작아지는 경우가 없이도 출구까지 도달할 수 있다. 그래서 이 산책로를 모노톤길이라고 부른다. 그림 1은 모노톤길의 예를 보여준다.

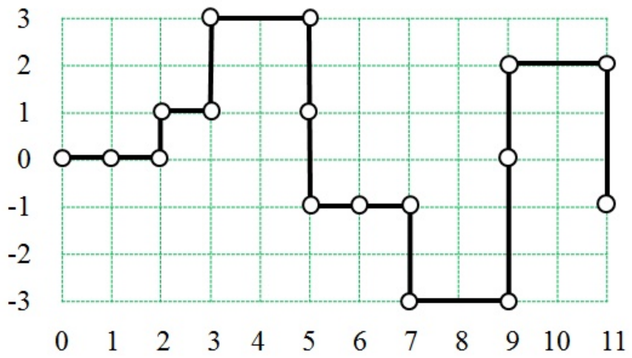


그림 **1.** 모노톤길의 예

이 산책로에는 n개의 카페가 곳곳에 들어서 있다. 특히 입구와 출구, 그리고 모든 코너에는 카페가 들어서 있다. 올레길 코스 관리자인 김씨는 이 산책로에 들어서 있는 모든 카페들의 위치 좌표를 가지고 있다. 입구 좌표는 항상 원점 $(0,0)$이다. 그는 이들 카페에 1부터 n까지의 일련번호를 붙이려고 한다. 입구의 카페는 1번, 그 다음부터는 길을 따라가면서 만나는 순서대로 번호를 배정한다. 입구에서 출구로 갈 때 카페 A를 카페 B보다 먼저 만나게 된다면 A에는 B보다 더 작은 번호를 배정한다. 따라서 그림 1 의 산책로에서 좌표 $(3,1)$에 위치한 카페의 번호는 5 이고, 좌표 $(9,0)$에 위치한 카페는 14번이고, 출구의 카페는 17번이다. 그는 산책로를 직접 걷지 않고 카페의 좌표들만으로 이 작업을 수행하고 싶어 한다. 그를 도와서 카페에 번호를 붙이는 작업을 수행하는 프로그램을 작성하시오.

입력(Input)
입력 데이터는 표준입력을 사용한다. 입력은 T 개의 테스트 데이터로 구성된다. 입력의 첫 번째 줄에는 테스트 데이터의 개수 T가 정수로 주어진다. 각 테스트 데이터의 첫 번째 줄에는 카페의 수를 나타내는 정수 n $(2 \le n \le 100{,}000)$이 주어진다. 그 다음 n개의 줄에는 각 줄마다 각 카페의 좌표 (x,y)를 나타내는 두 개의 정수 x와 y가 주어진다($0 \le x \le 100{,}000$ 이고 $-100{,}000 \le y \le 100{,}000$). 입구 좌표는 항상 $(0,0)$이다. 어떤 두 카페도 동일한 좌표를 가지는 경우는 없다. 마지막 줄에는 정수 m $(1 \le m \le 10)$과 m 개의 정수가 주어진다. m 개의 각 정수는 1 이상 n 이하로서 카페의 번호를 나타낸다.

출력(Output)

출력은 표준출력을 사용한다. 각 테스트 데이터에 대해, 카페 번호로서 주어진 m개의 정수에 대한 답을 순서대로 한 줄에 하나씩 출력한다. 정수 k에 대한 답은 번호가 k인 카페의 좌표 (x, y)를 나타내는 두 개의 정수 x와 y이다.

다음은 두 개의 테스트 데이터에 대한 입력과 출력의 예이다.

입력예제(Sample Input)	출력예제(Output for the Sample Input)
2 17 3 3 5 3 11 2 9 2 2 1 3 1 5 1 0 0 1 0 2 0 9 0 11 -1 9 -3 6 -1 7 -1 7 -3 5 -1 3 5 14 17 4 0 0 0 1 1 1 1 0 5 1 4 1 3 1	3 1 9 0 11 -1 0 0 1 0 0 0 1 1 0 0

Problem H
Palindromic Numbers
Time Limit: 0.001 Second

A number is a palindrome if it reads the same backward or forward. For example, 747 is a palindrome. 255 is also a palindrome as it is FF in hexadecimal. Given a positive integer, write a program whether it can be a palindrome with a certain base B ($2 \leq B \leq 64$). Note that the base is the number of unique digits (including zero) used to represent numbers. For example, for the decimal system the base B is 10.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of a line containing a positive integer between 64 and 1,000,000, inclusively.

Output
Your program is to write to standard output. Print exactly one line for each test case. The line should contain 1 if the number can be a palindrome with a certain base B ($2 \leq B \leq 64$), and 0 otherwise.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 747 255 946734	1 1 0

Problem H
회문인 숫자
Time Limit: 0.001 Second

어떤 숫자를 왼쪽부터 읽어도, 오른쪽부터 읽어도 같을 때 이 숫자를 회문인 숫자라고 한다. 예를 들어, 747 은 회문인 숫자이다. 255 도 회문인 숫자인데, 16 진수로 표현하면 FF 이기 때문이다. 양의 정수를 입력 받았을 때, 이 숫자가 어떤 B 진법($2 \leq B \leq 64$)으로 표현하면 회문이 되는 경우가 있는지 알려주는 프로그램을 작성하시오. B진법이란, 한 자리에서 숫자를 표현할 때 쓸 수 있는 숫자의 가짓수가 B라는 뜻이다. 예를 들어, 십진법에서 B는 10 이다.

입력(Input)

입력 데이터는 표준입력을 사용한다. 입력은 T개의 테스트 데이터로 구성된다. 입력의 첫 번째 줄에는 테스트 데이터의 수를 나타내는 정수 T가 주어진다. 각 테스트 데이터는 64 이상 1,000,000 이하인 하나의 정수로 주어진다.

출력(Output)

출력은 표준출력을 사용한다. 하나의 테스트 데이터에 대한 답을 하나의 줄에 출력한다. 각 테스트 데이터에 대해, 주어진 수가 어떤 B진법 ($2 \leq B \leq 64$)으로 표현하여 회문이 될 수 있다면 1 을, 그렇지 않다면 0 을 출력한다.

다음은 세 개의 테스트 데이터에 대한 입력과 출력의 예이다.

입력 예제(Sample Input)	출력 예제(Output for the Sample Input)
3 747 255 946734	1 1 0

Problem I

Particle

Time Limit: 0.1 Second

Sheldon is a physicist who studies high-energy particles' movement and reactions. Sheldon developed a brilliant machine, named ICPC (Integrated Cool Particle Capturer) that can capture a single particle in a 2-dimensional rectangle R of size $W \times H$ in its hull. After a particle gets captured by ICPC, its movement is limited in the rectangle R while retaining its original speed. More specifically, any particle P captured by ICPC obeys the following rules:

(i) P always stays in the rectangle R without any speed reduction for all times.
(ii) P moves straight until it hits any side of R.
(iii) If P hits a side of R, then P perfectly reflects, as a light ray hits a flat mirror.
(iv) If P reaches a corner of R, then P moves back in the direction opposite to the original direction

Sheldon is doing his experiments with his machine ICPC. He captured a particle P in ICPC. When he has first observed the particle P, P was located at (x_0, y_0) in R, moving in direction vector (a, b). Note that the coordinate of a location in the rectangle R is represented by (x, y) with $0 \leq x \leq W$ and $0 \leq y \leq H$. In the rectangle R, there are two special locations A and B whose coordinates are (x_1, y_1) and (x_2, y_2), respectively. Sheldon would like to know which of A and B is first reached by P, or if P does reach neither A nor B. You can help Sheldon by your programming skills.

Given all necessary values variables W, H, x_0, y_0, x_1, y_1, x_2, y_2, a, b, your program is to decide one of the three possibilities: P first reaches A before B, P first reaches B before A, or P can reach none of them. To make things simpler, assume that the particle P is very small as just a point. In addition, all the variables W, H, x_0, y_0, x_1, y_1, x_2, y_2, a, b appeared above will be given as integers.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case consists of a single line with 10 integers separated by a single space that represent W, H, x_0, y_0, x_1, y_1, x_2, y_2, a, b in order. Each of the 10 integers falls into the following range: $2 \leq W, H \leq 1{,}000{,}000$, $0 < x_0 < W$, $0 < y_0 < H$, $0 \leq x_1, x_2 \leq W$, $0 \leq y_1, y_2 \leq H$, and $-10{,}000 \leq a, b \leq 10{,}000$. Also, note that $(x_0, y_0) \neq (x_1, y_1)$, $(x_0, y_0) \neq (x_2, y_2)$, $(x_1, y_1) \neq (x_2, y_2)$, and $(a, b) \neq (0,0)$.

Output

Your program is to write to standard output. Print exactly one line for each test case. The line should consist only of "A" if P reaches A before P reaches B, "B" if P reaches B before P reaches A, or "O" if P can reach none of A and B forever.

The following shows sample input and output for three test cases.

Sample Input	Output for the Sample Input
3 5 5 1 3 4 2 2 1 1 1 5 3 1 2 0 2 3 0 3 2 5 3 2 2 2 1 3 2 1 3	A B O

Problem J
Pythagorean Expectation
Time Limit: 0.1 Second

A formula similar to the famous Pythagorean Theorem has been used in baseball. This is called *Pythagorean Expectation*, invented by Bill James. This expectation is used to estimate how many games a baseball team should have won during a season. The Pythagorean Expectation W for a baseball team is defined as follows.

$$W = \frac{S^2}{S^2 + A^2},$$

where S is the number of runs the team scored, and A is the number of runs the team allowed. The formula name comes from the resemblance to the Pythagorean Theorem.

This expectation empirically tells how the baseball team actually performs by comparing the real winning ratio. For example, Hanwha Eagles has scored 619 runs, but allowed 889 runs during 2014 season, that is, $S = 619$, $A = 889$, so $W = 619^2/(619^2 + 889^2) = 0.326$. Since each team plays 128 games according to the KBO rule, Eagles should have won $0.326 \times 128 = 41.728$ games. But Eagles actually won 49 games, showing they were not bad at all. For Lotte Giants with $S = 715$ and $A = 719$, $W = 0.497$, so they should have won $0.497 \times 128 = 63.616$ games. Since Giants actually won 58 games, one may say that they performed a bit poor for the season.

Given records of the games between n baseball teams, you are to write the largest and the smallest Pythagorean expectations of the teams.

Input
Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing two integers, n $(2 \leq n \leq 1{,}000)$ and m $(2 \leq m \leq 1{,}000)$, where n is the number of baseball teams, and m is the total number of games between n teams. In the following m lines, a record for each game is given line by line. Each record is represented by four numbers a, b, p, q, separated by a single space, which means team a has played against team b, and team a has scored p runs and team b has scored q runs. Note that $1 \leq a \neq b \leq n$, and p and q are non-negative integers no more than 20. Note that teams in a test case may have different number of games. If a team has no scored and allowed runs, then its expectation value becomes zero.

Output
Your program is to write to standard output. Print two integers for each test case. The first line should contain the integer part of the value, the largest Pythagorean Expectation of the teams times 1,000. The second line should contain the integer part of the value, the smallest Pythagorean Expectation of the teams times 1,000.

The following shows sample input and output for two test cases.

Sample Input	Output for the Sample Input
2 3 5 1 2 3 5 1 3 10 1 1 2 0 7 2 3 9 3 3 2 4 5 4 6 1 2 0 11 1 3 17 13 1 4 17 1 2 3 7 12 2 4 19 17 3 4 17 0	871 100 753 103

Problem J
피타고라스 기대값
Time Limit: 0.1 Second

유명한 피타고라스의 정리와 유사한 형태의 피타고라스 기대값 (Pythagorean Expectation)이 야구와 같은 스포츠 경기에 자주 사용된다. 피타고라스 기대값은 빌 제임스(Bill James)에 의해 정의되었으며, 이 값은 특정 야구팀이 한 시즌 동안 얼마나 잘 했는지를 평가하는 지표 중 하나로 사용된다. 한 야구팀의 피타고라스 기대값 W는 아래 식과 같이 정의된다.

$$W = \frac{S^2}{S^2 + A^2},$$

여기서 S는 해당 팀의 총 득점 수를, A는 해당 팀의 총 실점 수를 나타낸다.

이 기대값을 실제 승률과 비교하여, 해당 팀이 한 시즌을 얼마나 잘 보냈는지 평가할 수 있다. 예를 들어, 한화 이글스는 2014 시즌에 619 득점과 889 실점을 했다. 즉, $S = 619$, $A = 889$이 되어, 한화 팀의 피타고라스 기대값은 $W = 619^2/(619^2 + 889^2) = 0.326$이 된다. KBO 규정에 따르면, 한 시즌당 한 팀은 128번의 경기를 해야 하므로, 한화는 $0.326 \times 128 = 41.728$ 경기에서는 승리했어야 한다. 실제 한화는 49 경기에서 승리했기 때문에, 2014 시즌을 그리 나쁘지 않게 보냈음을 알 수 있다. 반면에 롯데 자이언츠는 $S = 715$ 이고 $A = 719$ 이 되어, $W = 0.497$ 이 된다. 따라서 $0.497 \times 128 = 63.616$ 경기 이상을 이길 것으로 기대되었지만, 실제론 58 경기만 승리했다. 결국, 롯데에게 2014년은 기대에 많이 못 미친 시즌이었음을 알 수 있다.

n개의 팀에 대한 기록이 주어지면, 이 기록으로부터 팀 별 피타고라스 기대값을 계산한 후, 그 중 최대 기대값과 최소 기대값을 출력하는 프로그램을 작성하시오.

입력(Input)
입력 데이터는 표준입력을 사용한다. 입력은 T개의 테스트 데이터로 구성된다. 입력의 첫 번째 줄에는 입력 데이터의 수를 나타내는 정수 T가 주어진다. 각 테스트 데이터의 첫 줄에는 두 양의 정수 n ($2 \le n \le 1{,}000$)과 m ($2 \le m \le 1{,}000$)이 주어진다. 여기서, n은 팀 개수이며, m은 전체 경기 수이다. 다음의 m개의 줄에는 각 경기에 대한 정보가 주어진다. 하나의 경기는 네 개의 정수 a, b, p, q로 주어지는 데, 팀 a와 팀 b가 경기를 했고 팀 a는 p 득점을 팀 b는 q 득점을 했다는 의미이다. 여기서, $1 \le a \ne b \le n$이며, p와 q는 모두 음이 아닌 20 이하의 정수이다. 하나의 테스트 데이터에서 팀 당 경기수가 반드시 같을 필요는 없다. 만약, 어떤 팀의 총 득점과 총 실점이 모두 0 이라면, 그 팀의 기대값은 0 으로 정의한다.

출력(Output)
출력은 표준출력을 사용한다. 각 테스트 데이터에 대해, 두 정수 값을 한 줄에 하나씩 출력해야 한다. 첫 번째 줄에는 최대 피타고라스 기대값에 1,000 을 곱한 값의 정수부분을 출력하고, 두 번째 줄에는 최소 피타고라스 기대값에 1,000 을 곱한 값의 정수부분을 출력한다.

다음은 두 개의 테스트 데이터에 대한 입력과 출력의 예이다.

입력예제(Sample Input)	**출력예제(Output for the Sample Input)**
2 3 5 1 2 3 5 1 3 10 1 1 2 0 7 2 3 9 3 3 2 4 5 4 6 1 2 0 11 1 3 17 13 1 4 17 1 2 3 7 12 2 4 19 17 3 4 17 0	871 100 753 103

Problem K
Registration
Time Limit: 1 Second

Print out your ICPC team number and team name.

Input
No input is given for this problem.

Output
Your program is to write to standard output. Print exactly two lines. The first line should contain your team number, and the second line should contain your full team name, even if your team name contains one or more blanks (a character of ASCII 32).

The following shows sample input and output, where the team number is `123` and the team name is `Your_ICPC_Team_Name`. Notice that no input is given.

Sample Input	Output for the Sample Input
	123 Your_ICPC_Team_Name

Problem K
등록
Time Limit: 1 Second

자신의 ICPC 팀 번호와 팀 이름(team name)을 그대로 출력하는 프로그램을 작성하시오.

Input

이 문제는 입력이 없다.

Output

표준출력(standard output)으로 출력해야 한다. 첫 줄에 자신의 팀 번호, 둘째 줄에 팀 이름을 출력한다. 출력할 팀 이름은 공백문자(ASCII 코드 32 번인 문자)를 포함하더라도, 공백문자를 포함하여 완전한 이름을 출력해야 한다.

다음은 팀 번호가 `123` 번, 팀 이름(team name)이 `Your_ICPC_Team_Name` 인 경우의 입출력 예제이다. 참고로 입력이 없는 것에 주의한다.

Sample Input	Output for the Sample Input
	123 Your_ICPC_Team_Name