ACM Collegiate Programming Contest 2017 (Hong Kong)

Judging Committee

6/24/2017

Venue:	Cyberport, Pokfulam
Time:	2017-6-24 [Sat] 1400-1800
Number of Questions:	7
Input:	Standard Input
Output:	Standard Output
Memory Limit:	64 Megabytes

If you have any problem, please contact the chief judge, Dr. Zhou Xu, through lgtzx@polyu.edu.hk.

Problem A

Berth Allocation

Time Limit: 5 seconds

In a container terminal, such as the Hong Kong Container Terminal, the bottleneck of the traffic is often at the quay. Therefore, the terminal operator has to allocate a limited number of berths of the quay to vessels in an efficient way.

As illustrate in Figure 1 below, consider a container terminal of n berths and m vessels arrived, where each vessel i (for i=1,2,…,m) requires a berth to load and unload containers, and the handling time is t_{ij} if berth j (for j=1, 2, …, n) is allocated to vessel i. For each vessel i=1,2,…m, the terminal manager, Owen, needs to decide on the berth, denoted by $b_i \in \{1,2,\dots,n\}$, as well on the starting time of berthing, denoted by $s_i \geq 0$. It must be satisfied that no two vessels are allowed to occupy the same berth simultaneously, i.e., for any two different vessels i and j, if b_i=b_j, then either $s_i + t_{i,b_i} \leq s_j$ or $s_j + t_{j,b_j} \leq s_i$ must be satisfied. Your task is to help Owen to minimize the total completion time of the vessels, i.e., to minimize $\sum_{i=1}^{m}(s_i + t_{i,b_i})$.

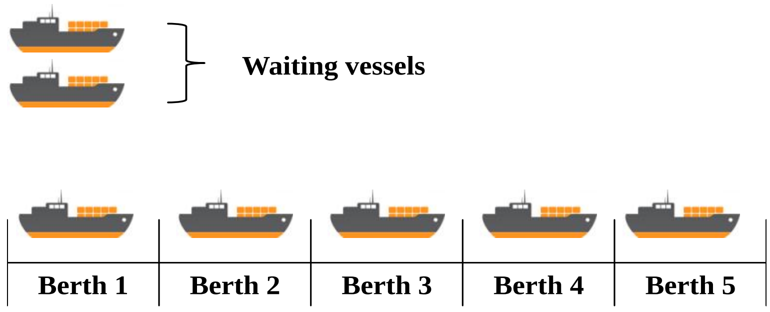


Figure 1. An illustrated example, with a quay of n=5 berths, m=7 vessels, where two vessels are waiting outside the quay.

Input

First line of the input is an integer T indicating the number of test cases. For each case, the first line contains two integers n and m (for $1 \leq n \leq 50$, $1 \leq m \leq 200$). Each of the following m

lines contains n positive integers representing vessel i's handling times t_{i1}, t_{i2}, …, and t_{im} (t_{ij} <=1000) for i=1,2,3…,n.

Output

For each case, output one integer, the minimum total completion time of all the vessels.

Standard Input	**Standard Output**
1 5 7 1 2 3 4 5 2 1 3 4 5 2 3 1 4 5 2 3 4 1 5 2 3 4 5 1 2 2 2 2 2 2 2 2 2 2	11

Problem B
Electric Vehicle Routing

Time Limit: 5 seconds

In recent years, electric vehicles are highly promoted to reduce emissions of greenhouse gas. However, the limited battery capacity of electric vehicles requires visits of recharging stations during the vehicles' tours. In this problem, you are going to plan a route for an electric vehicle to travel from an origin to a destination with the least total cost to charge electricity.

Consider an undirected graph with n nodes and m arcs, denoted by 1, 2, …, n, where each node represents a location, and each edge (i,j) represent a two-way road connecting location i and location j. There are h recharging stations located at nodes s_1, s_2, …, s_h, where $s_p \in \{1,2,\ldots,n\}$ for p=1,2,…,h. A journalist, Kevin, plans to travel from an origin location a to a destination location b. Kevin has an electric vehicle of a battery capacity Q kilowatt-hour (or kWh in short), which can be charged at every recharging station s_j (for j=1, 2, …, h) at a unit charging cost of 1$ per kWh. For each edge (i,j), the consumption of electricity for the car is d_{ij} kWh. Kevin needs to decide a route for his car to travel from origin a to destination b, where the route includes a sequence of locations as well as a plan to charging electricity, so that the car will never be out of electricity. Assuming that the initial charge level of the car is L kWh, your task is to help Kevin to minimize the total charging cost for the car.

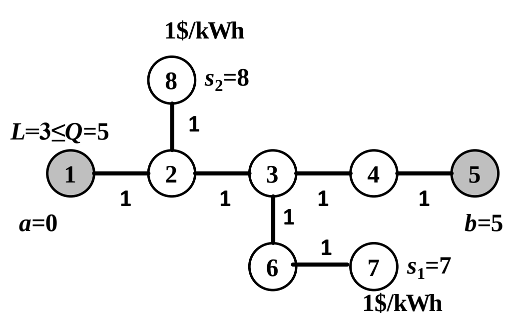


Figure 2. An illustrated example with all d_{ij} equal to 1, where the best routing of the car is to visit node 1, node 2, node 8 (charging electricity of 3 kWh), node 2, node 3, node 4, and node 5, achieving the minimal total charging cost of 3 dollars.

Input

First line of the input is an integer T indicating the number of test cases. For each case, the first line contains six integers n, m, h, a, b, Q, and L, where $1 \leq h, a, b \leq n \leq 1000$, $1 \leq m \leq 10000$, $0 \leq L \leq Q \leq 1000000$. The second line contains h integers representing locations of recharging stations, s_1, s_2, …, s_h. Each of the following m line contains three integers, i, j, and d_{ij}, representing an edge (i,j) and its electricity consumption d_{ij}.

Output

For each case, if the car cannot arrive at the destination b, then output -1, otherwise, output an integer that equals the minimum total charging cost of electricity for the car to travel from origin a to destination b.

Standard Input	Standard Output
2 8 7 2 1 5 5 3 7 8 1 2 1 2 3 1 3 4 1 4 5 1 3 6 1 6 7 1 2 8 1 3 2 1 1 3 3 1 2 1 2 1 2 3 4	3 -1

Problem C

Robot Rescue

Time Limit: 5 seconds

With the rapid development of artificial intelligence, robots today can participate in battles. Consider that in an *xy*-plane, there are n bombs are located at coordinates (x_i,y_i) for i=1,2,…,n, where the x-axis points horizontally to the right, and the y-axis points vertically upwards. A robot is initially located at coordinate (0,0), and facing upwards. A commander, Owen, has developed a program of instructions for the robot to move, so that the robot can be rescued from the bombs and reach a destination at (a,b).

The program contains a main procedure M and a subroutine S. There are three following types of instructions, where $0 \le k \le 1000000$:

- F(k): go forward k step;
- R(k): turn right 90 degree for k times;
- S(k): repeat subroutine S for k times.

The main procedure M can contain all the three types of instructions, while the subroutine S can contain only instructions $F(k)$ and $R(k)$ but not $S(k)$. See Figure 3 for an example.

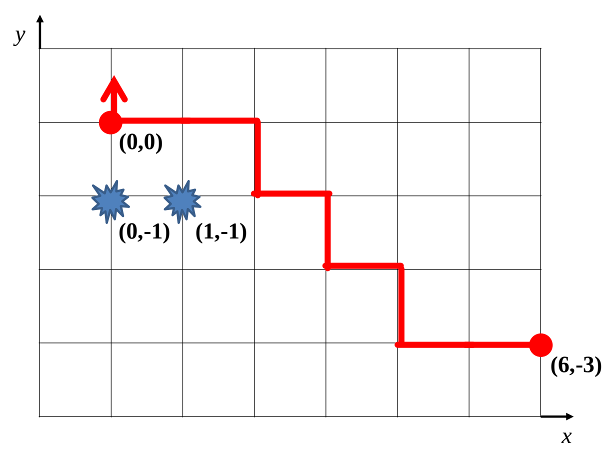


Figure 3. An illustrate example with n=2 bombs located at (0,-1) and (1,-1), where the robot can reach the destination (6,-3) by following a program with a main procedure M that consists of instructions R(1), F(1), S(3), and F(2), and with a subroutine S that consists of instructions, F(1), R(1), F(1), and R(3).

Given a main procedure M and a subroutine S, represented by two strings with spaces to separate instructions, your task is to help Owen determine whether or not by following the program, the robot will reach the destination at (*a*,*b*) without moving to any bomb.

Input

First line of the input is an integer *T* indicating the number of test cases. For each case, the first line contains two integers *a* and *b*, representing the coordinate of the destination. The second line contains an integer *n*, followed by *n* pairs of integers, x_1, y_1, x_2, y_2, …, x_n, y_n, representing the coordinates for the *n* bombs, where $1 \leq n \leq 10000$. The third line contains a string of instructions for the main procedure M, and the third line contains a string of instructions for the subroutine S, where instructions are separated by spaces, and the length of each string is less than 1000.

Output

For each case, if the robot reaches the destination at (*a*,*b*) without moving any bomb, then output 0, otherwise, if the robot moves to any bomb, then output an integer representing the first bomb that the robot has moved to, otherwise, output -1.

<table>
<tr><th>Standard Input</th><th>Standard Output</th></tr>
<tr><td>2
6 -3
2 0 -1 1 -1
R(1) F(1) S(3) F(2)
F(1) R(1) F(1) R(3)
100 -20
2 0 -1 -1 -1
R(2) S(20) F(80)
F(1) R(3) F(1) R(1)</td><td>0
1</td></tr>
</table>

Problem D
Cake Cutting Game

Time Limit: 5 seconds

Owen and Kevin are playing a cake cutting game at a birthday party. In the beginning of the game, they are given n slices of cakes where each slice i (for i=1, 2, …, n) is a circular sector of a central angle equal to b_i degree. Each player can only choose one slice, and cut through a radius of the slice, such that each new slice, which is still a sector, must have a central angle greater than or equal to w degree. The two players take turns to cut, until one of them cannot cut any slice and lose the game, while the other wins the game. See Figure 4 for an example.

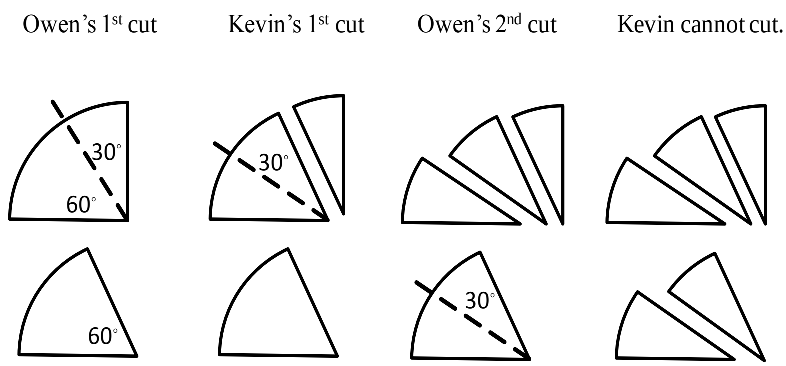


Figure 4. An illustrative example, where n=2, w=30, b_1=90, b_2=60, and Owen wins the game.

Suppose that Owen is always the first one to cut, and that both the two players are super smart, and want to win the game. Your task is to decide who will win the game.

Input

The first line of the input contains the number of test cases T.

For each test case, its first line of the input has two integers, n and w, where $1 \leq n \leq 100000$, $1 \leq w \leq 90$. The second line of the input has n integers, b_1, b_2, …, b_n, where $w \leq b_i \leq 180$ for i=1, 2, …, n.

Output

For each test case, output the name of the player ("Owen" or "Kevin") who will win the game.

Example

Standard Input	Standard Output
1 2 30 90 60	Owen

Problem E

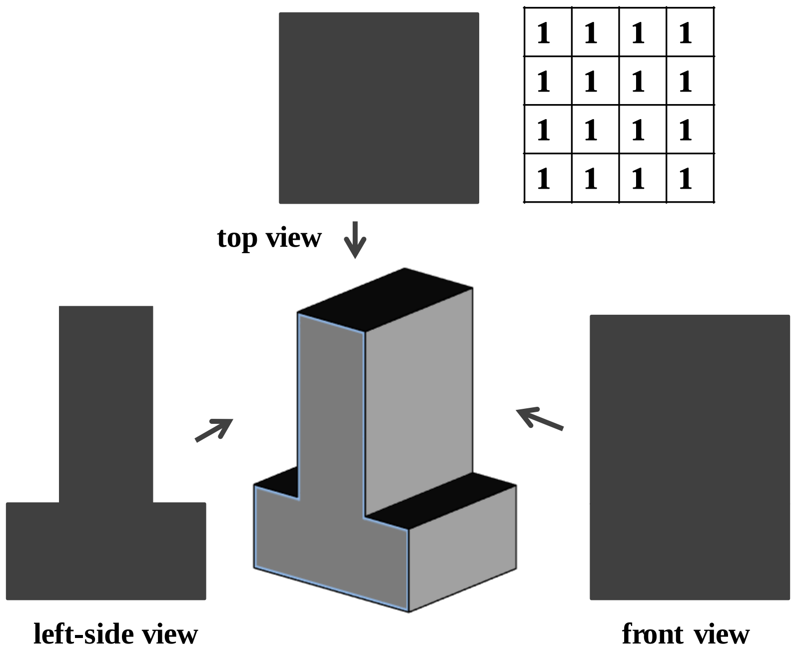
Sculpture Recovery

Time Limit: 5 seconds

Kevin is an artist who creates 3-D sculptures. Whenever Kevin completes a sculpture, he will take digital pictures of the sculpture from the top view, front view, and left-side view, respectively. The pictures of the three views are black and white, with the background being white, and with the sculpture being black. Each of the three digital pictures contains several rows with each row having several pixels, where each pixel is a 1cm by 1cm square, represented by a binary number, equal to 0 if the pixel is white, and equal to 1 if the pixel is black.

Given the pictures of the three views of Kevin's new sculpture, your task is to recover the sculpture. Since there can be multiple possible sculptures with the same pictures of the three views, you need to find the one with the largest volume.

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1



0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0
1	1	1	1
1	1	1	1

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

Figure 5. An illustrated example with three views and their digital pictures for a sculpture of a volume equal to 64.

Input

The first line of the input contains the number of test cases T, and cases are separated by empty lines.

For each test case, its first line contains three integers, a,b and h, where $1<=a,b,h<= 200$. It is followed by one $b*a$ binary matrix, one $h*a$ binary matrix, one $h*b$ binary matrix, representing the digital pictures of the top view, the front view, and the left-side view of Kevin's sculpture respectively, where the three matrices are separated by empty lines.

Output

For each test case, output one integer, the largest possible volume of Kevin's new sculpture.

Example

Standard Input	**Standard Output**
2 4 4 6 1111 1111 1111 1111 1111 1111 1111 1111 1111 1111 0110 0110 0110 0110 1111 1111 2 4 7 11 11 11 11	64 34

10 01 11 11 11 11 11 0101 1111 1111 0111 0011 1111 0001	

Problem F

Perspective

Time Limit: 5 seconds

Owen likes drawing. In drawing, *perspective* is a way of portraying three dimensional objects on a two-dimensional flat surface by suggesting depth or distance. The two most common perspectives are one-point perspective and two-point perspective. In one-point perspective, all the lines that are directly parallel with the viewer's line of sight converge at a unique vanishing point on the horizon line. In two-point perspective, it contains two vanishing points on the horizon line, to which two sets of parallel lines converge.

You task is to develop an intelligent program to help Owen judge whether a given drawing has one-point perspective, two-point perspective, or neither of them. To achieve this, the program needs to first extend all the segments in the drawing to lines. If all the lines (except horizontal or vertical lines) pass through the same point, the drawing must have one-point perspective. If all the lines pass through two points, the drawing must have two-point perspective.

Input

The first line of the input contains the number of test cases T.

For each test case, its first line contains an integer n with $1 \leq n \leq 1000$, which is the number of segments in the given drawing. Each of the following n lines contains four integers $x1$, $y1$, $x2$, $y2$ with $-10^6 \leq x1, x2, y1, y2 \leq 10^6$, representing the coordinates of the two end points of each segment. It is possible that some segments are co-linear, i.e., they are lying on the same line.

Output

For each test case, output one line. If the drawing has one-point perspective, print "One-point perspective". Otherwise, if the drawing has two-point perspective, print "Two-point perspective". Otherwise, print "Other". Notice that you should print “Other” for the following cases.

1. All the segments are parallel to the *x*-axis or *y*-axis.

2. After excluding horizontal and vertical segments, there exists only one remaining segment.
3. After excluding horizontal and vertical segments, all the remaining segments are co-linear.
4. After excluding horizontal and vertical segments, all the remaining segments are parallel.

Example

Standard Input	Standard Output
3 7 -8 -8 -2 -8 -8 -8 -8 -6 -2 -8 -2 -6 -8 -6 -2 -6 -2 -8 1 -4 -2 -6 1 -3 -8 -6 -4 -4 4 3 3 -2 -2 3 3 5 -7 -4 -2 -1 -1 -4 -2 1 6 3 0 0 1 1 2 3 3 4 1 9 2 10	One-point perspective Two-point perspective Other

Problem G

Pointer

Time Limit: 5 seconds

Kevin likes to play a toy pointer. At the beginning, it points to the *North*. Now Kevin plays with the pointer by following a given sequence of operations. Each operation turns the pointer 90 degree in clockwise or anticlockwise direction. Your task is to output the final pointing direction of the pointer after all the operations.

Input

The first line of the input contains the number of test cases T.

For each test case, there are n lines of strings consisting of only letters 'A' and 'C' (without quotes). For each string, the letter 'A' means to turn the pointer in 90 degree anticlockwise direction, 'C' means to turn the pointer 90 degree in clockwise direction. The length of any strings will not exceed 100.

Output

For each test case, output a letter ('N' for North, 'S' for South, 'W' for West, 'E' for East) corresponding to the final pointing direction.

Example

Standard Input	Standard Output
2 AC AAAC	N S