ACM ICPC

UM Practice Contest 2

September 22

Problems

A	Commando War
B	Polygon Area
C	Long Journey
D	Number of Paths

Problem A
Commando War

You have a commando squad of N soldiers at your disposal, and are planning an ambush on an important enemy camp located nearby.

You don't want any of your soldiers to know your plan for the other soldiers, so that everyone can cleanly focus only on their task, and every soldier will have a unique responsibility. To enforce this, you brief every individual soldier about their tasks separately, just before sending them to the battlefield. You know that every single soldier needs a certain amount of time to execute their job. You also know very clearly how much time you need to brief every single soldier.

Being anxious to finish the total operation as soon as possible, you need to find an order of briefing your soldiers that will minimize the time necessary for all the soldiers to complete their tasks. You may assume that no soldier has a plan that depends on tasks of their mates. In other words, once a soldier begins a task, they can finish it without making any pauses in between.

Input

Every test case starts with an integer N, denoting the number of soldiers. Each of the following N lines describe a soldier with two integers $B\ (1 \leq B \leq 10000)$ & $J\ (1 \leq J \leq 10000)$. B seconds are needed to brief the soldier, while J seconds are needed to complete their individual tasks. The end of input will be denoted by a case with $N = 0$. This case should not be processed. There will be no more than 100 input sets in one case.

For A-easy, $1 \leq N \leq 10$.
For A-hard, $1 \leq N \leq 1000$.

Output

For each test case, print a line in the format, 'Case X: Y', where X is the case number & Y is the total number of seconds counted from the start of your first briefing till the completion of all jobs.

Sample Input

```
3
2 5
3 2
2 1
3
3 3
4 4
5 5
0
```

Sample Output

```
Case 1: 8
Case 2: 15
```

Example

For the first test case, you can brief the first soldier for 2 seconds, who then immediately goes off to implement their task. You then brief the second soldier for 3 seconds, and the last soldier for 2 seconds. Last soldier takes one second to complete their task. Since other soldiers will have completed their tasks by the time the last solider completes theirs, the total time for the entire operation is 2 + 3 + 2 + 1 = 8 seconds. That is also the minimal possible time for the entire operation.

Problem B
Polygon Area

Given a simple polygon (a polygon that does not intersect with itself), output the area of the polygon.

Input

The first line contains an integer n $(1 \leq n \leq 100)$, indicating the number of polygon points to follow. The following n lines each contains 2 integers, represents a point on the polygon in 2D Cartesian coordinate system. All points are given in either clockwise or counterclockwise order. All coordinates have absolute value of no greater than 2000.

For B-easy, the polygon is convex.
For B-hard, the polygon can be non-convex.

Output

Output the area of the polygon. Your output should have exactly 6 digits of precision after the decimal point.

Sample Input 1

```
4
0 0
10 0
10 10
0 10
```

Sample Output 1

```
100.000000
```

Sample Input 2

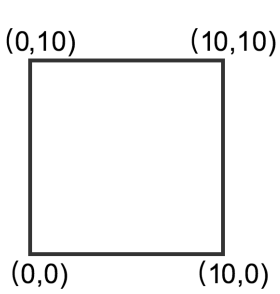
```
4
0 0
4 6
0 3
-4 6
```

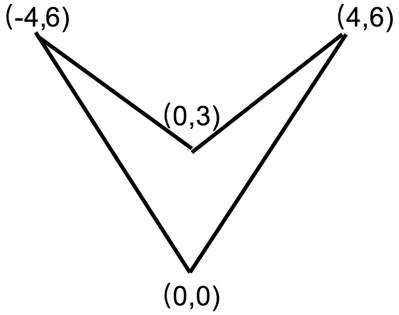
Sample Output 2

```
12.000000
```

Example

Here are the two samples.





Problem C
Long Journey

Jamie is traveling through country "C". Aside from being the only known country with a single letter in its name, C is a very strange country. It has n cities and n directed roads, and all roads forms a perfect cycle of size n.

At one of the travel stops, Jamie finds out that he lost his passport! In order to keep traveling, he decides to draw a fake one. Obviously, this probably will not going to work forever, but luckily, Jamie draws very, very well, so his passport really looks just like a real one, enough to cheat lots of officials in country C.

To travel in country C, you need to show your passport before you enter any road. Each road has a caution-value of a_i. Assume that Jamie's drawing skill is B. If $B \geq a_i$, then the official at the beginning of the road i will let Jamie go through the road, without further checking the passport. Otherwise, Jamie will be stopped. Please help Jamie to determine how far he can go.

Input

The first line contains two integers: n, m, which represents the number of cities in country C, and the number of queries.
The following line contains n integers a_i $(0 \leq i < n, 1 \leq a_i \leq 10^9)$. Note that for simplicity, assume that all cities in country C forms a perfect cycle, and the cities on the cycle are numbered $0, 1, \ldots, n-1$ in order, and a_i represents the caution-value of the directed road from city i to city $i+1$ ($n-1$ to 0 as well).
The following m lines, each line contains a query of two integers $c_i (0 \leq c_i < n)$ and $b_i (1 \leq b_i \leq 10^9)$. Help Jamie determine how far he can go, if his drawing skill is b_i, and starts from city c_i.

For C-easy, $2 \leq n, m \leq 1000$.
For C-hard, $2 \leq n, m \leq 200000$.

Output

For each query, output a line that contains a single integer, which represents the farthest city ID of the city that Jamie can go. If Jamie can go through the whole country and back to the city he starts, output "I will be back!" instead.

Sample Input

```
2 4
10 11
1 10
1 11
0 10
0 11
```

Sample Output

```
1
I will be back!
1
I will be back!
```

Example

For query 1, Jamie is stuck at city 1, so that's the furthest he can go. For query B, his drawing skills can get him from 1 to 0, then from 0 back to 1, so he will be back! For query 3, he can travel from 0 to 1 but no further, so 1 is the furthest he can go. For the final query, his drawing skill can help him travel the entire circuit and so he will be back again!

Problem D
Number of Paths

You are given a directed unweighted graph. Find the number of different paths of certain length that you can take from the starting vertex to all the other vertices. Length of "one step" is a move from any one vertex to any other vertex in the direction of the connecting edge. Note that two paths are different if there is at least one edge that is different on the path.

Input

The first line contains three integers $n\,(1 \le n \le 100), m(1 \le m \le n(n-1)), t$ – representing the number of vertices, the number of edges, and the number of steps (i.e. the path length).
The next m lines, each contains two integers $a_i, b_i\ (1 \le a_i \ne b_i \le n)$, representing a directed edge from a_i to b_i. Note that all edges are unique.

For D-easy, $1 \le t \le 10000$.
For D-hard, $1 \le t \le 10^9$.

Output

Output contains n lines, and the i-th line is a number of different paths starting from vertex 1 and ending at vertex i that have exactly the length t. Since the number of paths can get very large, output the number of paths modulo 1000000007.

For example, 1000000008 = 1 (mod 1000000007)

Sample Input 1

```
3 3 100
1 2
2 3
3 1
```

Sample Output 1

```
0
1
0
```

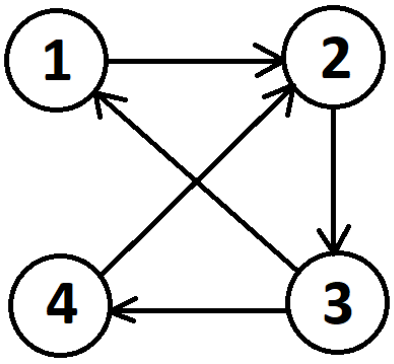
Sample Input 2

```
4 5 4
1 2
2 3
3 1
3 4
4 2
```

Sample Output 2

```
0
2
0
0
```

Diagram for Sample 2



You can get from vertex 1 to vertex 2 using two different paths of length 4. You cannot get to any other vertex using path of length 4.