Problem Set #1 15-295 Spring 2019

Problem A: Voronoi Diagram Returns

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	1024 megabytes

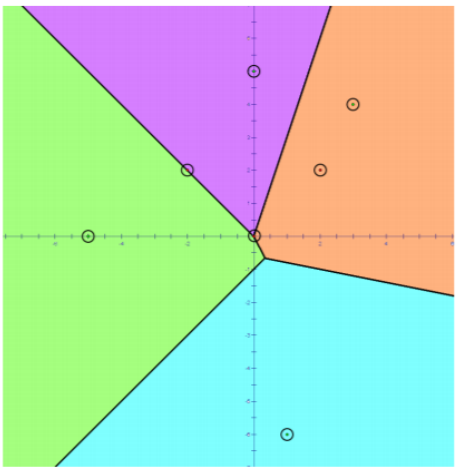

***Figure**: Voronoi Diagram of size 4.*

In the 2-dimensional Cartesian coordinate system, we define the **Voronoi Diagram** of a non-empty set of points S, as a diagram that divides the plane by the criteria "which point in the set S is closest in this location?". More precisely, the Voronoi diagram of a given non-empty point set $\{P_1, P_2, \cdots, P_n\}$ is a collection of **regions**: A point K is included in region i if and only if $d(P_i, K) \leq d(P_j, K)$ holds for all $1 \leq j \leq n$, where $d(X, Y)$ denotes the Euclidean distance between point X and Y.

For example, in the picture above, every location over the plane is colored by the closest point with such location. The points which belongs to a single region is colored by a light color indicating a region, and the points which belongs to more than one region forms lines and points colored black.

There is an algorithm which computes the Voronoi Diagram in $O(n \log(n))$, but it is infamous to be very complicated and hard. In fact, we are lenient problem setters, so we set $n \leq 2000$, which means you can solve this task with slower Voronoi Diagram algorithms!

In this task, you should solve the **point query problem** in Voronoi diagram: in the Voronoi diagram constructed with the set of points $\{P_1, P_2, \cdots, P_n\}$, you should determine which region(s) the point belongs in. More precisely, you will be given q queries of points. For each query point, you should determine the following:

- If it's not included in any region, output `NONE`.
- If it's included in exactly one region, output `REGION X`, where `X` is the index of such region.
- If it's included in exactly two regions, output `LINE X Y`, where `X` and `Y` (`X` < `Y`) are the indices of such two regions.
- If it's included in three or more regions, output `POINT`.

Input

In the first line, the number of points consisting Voronoi diagram n, and the number of queries q are given. $(3 \leq n \leq 2\ 000, 1 \leq q \leq 250\ 000)$

In the ith line of next n lines, two integers indicating x and y co-ordinate of P_i are given. These are the points consisting the Voronoi diagram. All n points are distinct. ($|x|, |y| \leq 10^4$)

In the jth line of next q lines, two integers indicating x and y co-ordinate of Q_j are given. For each point Q_j, you should determine in which region(s) the given point belongs to. ($|x|, |y| \leq 10^4$)

Output

Output consists of q lines. In the jth line, you should print one of following:

- If Q_j is not included in any region, output `NONE`.
- If Q_j is included in exactly one region, output `REGION X`, where `X` is the index of such region.
- If Q_j is included in exactly two regions, output `LINE X Y`, where `X` and `Y` (`X` $<$ `Y`) are the indices of such two regions.
- If Q_j is included in three or more regions, output `POINT`.

Example

standard input	standard output
4 3 -5 0 0 5 3 4 1 -6 -2 2 0 0 2 2	LINE 1 2 POINT REGION 3

Note

Example is illustrated as diagram above.

Problem B: Rising Sun

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Joon has a midterm exam tomorrow, but he hasn't studied anything. So he decided to stay up all night to study. He promised himself that he will not stop studying before the sun rises.

Joon's house is in some mountains. For convenience, let's say that Joon is living in a 2-dimensional coordinate system. The mountains are in the region $y \geq 0$, starting at $x = a_0$, and the boundary of them consists of $2n$ line segments parallel to either $y = x$ or $y = -x$.

More precisely, the boundary of the mountains can be described by $(2n - 1)$ additional integers, where the ith number a_i of them is the x-coordinate of the ith cusp of the mountains. The boundary line starts at $(a_0, 0)$, proceeds parallel to $y = x$ until its x-coordinate reaches a_1, then proceeds parallel to $y = -x$ until its x-coordinate reaches a_2, and so on. After the last step, the line proceeds parallel to $y = -x$ until it meets the x-axis.

The interior of the mountains is the region below the boundary and above the x-axis. Thus, the interior and the boundary are disjoint.

At some point between $x = a_0$ and $x = a_{2n-1}$, there is Joon's house on the boundary of the mountains. The size of his house is neglectable compared to the mountains.

Currently, the sun is at the origin and it rises vertically ($+y$ direction), 1 per minute. Joon can see the sun if the interior of the mountains does not intersect the straight line segment connecting Joon's house and the sun. Joon is completely exhausted and wants to know when can he stop studying. But as you may expect, he is out of his mind, so he cannot do such difficult math. Help him!

Input

The first line of the input contains an integer n ($1 \leq n \leq 10^3$).

The next line contains $2n$ integers, the ith of which is the integer a_{i-1} ($1 \leq a_0 < \cdots < a_{2n-1} \leq 10^6$).

The last line contains an integer x, the x-coordinate of Joon's house ($a_0 \leq x \leq a_{2n-1}$).

It is guaranteed that the boundary of the mountains is in the region $y \geq 0$.

Output

Print exactly one integer m, the smallest integer such that Joon can see the sun after m minutes.

Examples

standard input	standard output
2 1 4 6 7 7	5
2 3 4 5 7 7	0
3 4 9 12 13 14 16 15	8

Note

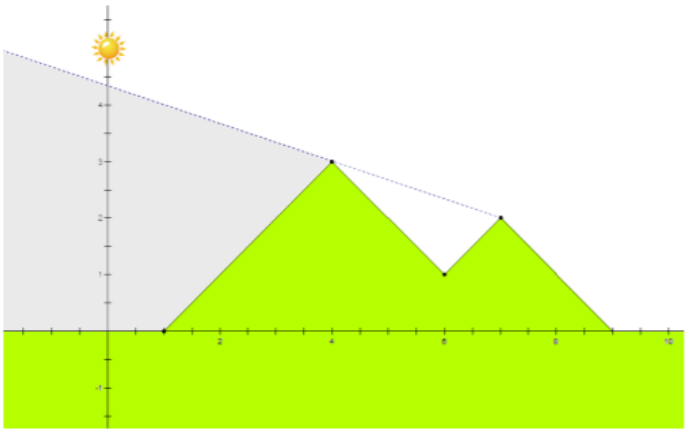

***Figure**: Illustration of the first example.*

Problem C: Game on a plane

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

You are given N points on a plane. These points are precisely the set of vertices of some regular N-gon. Koosaga, an extreme villain, is challenging you with a game using these points. You and Koosaga alternatively take turns, and in each turn, the player

1. chooses two of the given points, then
2. draws the line segment connecting the two chosen points.

Also, the newly drawn line segment must not intersect with any of the previously drawn line segments in the interior. It is possible for two segments to meet at their endpoints. If at any point of the game, there exists a convex polygon consisting of the drawn line segments, the game ends and the last player who made the move wins.

Given the integer N, Koosaga is letting you decide who will move first. Your task is decide whether you need to move first or the second so that you can win regardless of Koosaga's moves.

Input

The input consists of many test cases. The first line contains an integer T ($1 \leq T \leq 5000$), the number of test cases. Each of the following T test cases is consisted of one line containing the integer N ($3 \leq N \leq 5000$).

Output

For each test case, print one line containing the string `First` if you need to move first or `Second` if you need to move second so that you can win regardless of Koosaga's moves.

Example

standard input	standard output
2 3 5	First Second

Problem D: Electronic Circuit

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

Joon is taking General Physics II and he is now studying electronic circuits. An electronic circuit consists of several nodes and undirected wires each connecting two distinct nodes. Moreover, a circuit has two distinctive end nodes; a source node and a sink node, where a voltage is applied (usually it is applied by additional wire with a battery connecting the two nodes, but we will neglect it). Each wire has a resistance, and Joon should know how to calculate the composite resistance of a circuit.

By the way, Joon hates complicated things. So he only cares about circuits that can be made by series and parallel compositions, since they are easy to calculate the composite resistance. He calls them 'nice' circuits; formally, a nice circuit can be defined as follows.

- A circuit with a single wire connecting two end nodes is nice.
- A circuit obtained by merging the sink node of a nice circuit C_1 and the source node of a nice circuit C_2 into a single node is nice. The source node and the sink node of the obtained circuit are the source node of C_1 and the sink node of C_2, respectively.
- A circuit obtained by merging the two source nodes of nice circuits C_1 and C_2 into a single node, and merging the two sink nodes of C_1 and C_2 into a single node, is nice. The two end nodes of the obtained circuit are the respective merged end nodes.

He made a circuit with his wires to calculate the composite resistance, but his friend Pringles screwed up his circuit, so now Joon does not know what the end nodes are. To make things worse, he is not even sure whether the circuit is nice or not.

Joon will give you the circuit. He kindly asks you whether the circuit can be nice by appropriately choosing two end nodes. Be careful that there may be multiple wires connecting two nodes.

Input

The first line contains two integers, n and m ($2 \leq n \leq 10^5$, $1 \leq m \leq 3 \times 10^5$), where n is the number of nodes and m is the number of wires. All nodes are numbered from 1 to n.

In the following m lines, each line contains two integers u and v ($1 \leq u, v \leq n$, $u \neq v$), which represents a wire connecting u and v. It is guaranteed that every node is attached to at least one wire; otherwise the node does not exist!

Output

Print `Yes` if the given circuit can be nice, or `No` otherwise.

Examples

standard input	standard output
4 6 1 2 2 3 2 3 3 4 1 4 1 4	Yes
4 6 1 2 1 3 1 4 2 3 2 4 3 4	No
9 12 1 9 1 4 5 4 6 5 1 5 8 1 3 6 6 8 3 8 2 9 9 7 7 2	Yes

Problem E: Fascination Street

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	1024 megabytes

A street named *Fascination Street* is divided into N equal length of blocks. For each block i ($1 \leq i \leq N$), it has block $i-1$ in its left side if $i > 1$, and block $i+1$ in its right side if $i < N$.

Unlike its name, the street is infamous to be a dark and eerie place in the night. To solve this, Robert decided to install the streetlight for some of the blocks. The cost of installing a streetlight for i-th block is W_i, and the total cost is the sum of each installation cost. After installing, every block should either have a streetlight, or have a streetlight in it's left or right block.

Robert also has some tricks to reduce the cost. Before installing the streetlight, Robert selects two distinct blocks i and j, and exchange their position. After the operation, the cost of installation is exchanged. In other words, the operation simply swaps the value of W_i and W_j. This operation have no cost, but Robert can only perform it at most K times.

Now, given the array W and the maximum possible number of operations K, you should find the minimum cost of lighting the whole street.

Input

The first line contains two space-separated integers N, K. N is the number of blocks, and K is the maximum possible number of operations. ($1 \leq N \leq 250000, 0 \leq K \leq 9$)

The second line contains N space-separated integers $W_1, W_2 \cdots W_N$, where W_i is the cost of installing a streetlight for i-th block. ($0 \leq W_i \leq 10^9$)

Output

Print a single integer which contains the minimum cost of lighting the whole street.

Examples

standard input	standard output
5 0 1 3 10 3 1	4
5 1 1 3 10 3 1	2
12 0 317 448 258 208 284 248 315 367 562 500 426 390	1525
12 2 317 448 258 208 284 248 315 367 562 500 426 390	1107

Problem F: Dumae

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	1024 megabytes

Do you know *Dumae*? It is a nickname of the most famous restaurant nearby KAIST, *Dumae Charcoal-grilled Barbecue*. Because *Dumae* is a very famous restaurant, lots of KAIST students stand in line even though it has not opened yet. Students wonder how long they have to wait, so they started to guess their order.

There are N students in waiting line and each of them has a distinct student ID from 1 to N. Student i (student with student ID i) guessed that he/she is either L_i-th, $(L_i + 1)$-th, ..., $(R_i - 1)$-th, or R_i-th person in the line. (i.e. the number of people standing relatively in front of him/her is in the interval $[L_i - 1, R_i - 1]$) Also, M claims are made, of which the i-th says that student v_i can see student u_i in the waiting line. It means student u_i is relatively in front of student v_i.

You wonder if all of students' guesses and claims were right. Find an order of waiting line that satisfies all the guesses and claims, or report that such an order does not exist.

Input

The first line contains two space-separated integers N, M. $(1 \leq N \leq 300\,000, 0 \leq M \leq 1\,000\,000)$

In the next N lines, two space-separated integers L_i, R_i are given. $(1 \leq L_i \leq R_i \leq N)$

In the next M lines, two space-separated integers u_i, v_i are given. $(1 \leq u_i \leq N,\ 1 \leq v_i \leq N,\ u_i \neq v_i)$

Output

If there is no answer that satisfies the condition, print -1.

Otherwise, print N lines. In the i-th line, print the student ID of the i-th student from the front.

Examples

standard input	standard output
3 3 1 3 1 3 1 3 1 2 2 3 3 1	-1
3 3 1 3 1 3 1 3 1 2 2 3 1 3	1 2 3