

Problem A. 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

Problem B. 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 C. Fake Plastic Trees

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

Tree is a recursive structure, which is either:

- **Empty**. Empty tree is denoted as -1 and has a size of 0.
- **Non-empty**. Non-empty tree T is denoted as a pair of two trees (T_1, T_2) , where T_1 is called **left subtree** of T , and T_2 is called **right subtree** of T . If $T = (-1, -1)$, then we call such T a **leaf**. Leaf has a size of 1, and non-leaf has a size of $|T_1| + |T_2|$, where $|T_1|$ is the size of T_1 , and $|T_2|$ is the size of T_2 .

A non-empty tree T is a **Fake Plastic Tree**, if the tree is *balanced*. Formally, Let $T = (T_1, T_2)$. If $|T_1| = |T_2|$ or $|T_1| = |T_2| + 1$ holds, then T is a Fake Plastic Tree.

In computer science, trees are commonly used as a data structure, and they are stored in a memory. At first, there are no trees in the memory, and only an imaginary *null pointer* exists (which corresponds to empty tree, -1). You can allocate a tree in the memory, by setting T_1 and T_2 as either a null pointer or a pointer of an existing tree. Then, the memory is extended by adding $T = (T_1, T_2)$ into its structure. Note that pointer can be described as a small integer, reducing the need for explicitly storing the whole tree.

Formally, memory M is an inductive structure, which at first contains only empty tree -1 . ($M = \{-1\}$). You can expand the memory with following operation $M \leftarrow M \cup \{(T_1, T_2)\}$, where $T_1 \in M, T_2 \in M$. If a tree T is inserted in i -th stage, then it has the **index** $i - 1$. For a tree with index i , their subtrees can be represented as a pair of integer in range $[-1, i - 1]$.

Your task is to construct a memory M , which satisfies the following :

- Every tree in M is either empty or Fake Plastic Tree.
- M has at most 125 non-empty trees.
- There exists a tree $T \in M$, where $|T| = N$ holds. N is an integer, and is given as an input.

Input

The first line contains a single integer T , the number of test cases. ($1 \leq T \leq 2000$)

In the next T lines, a single integer N is given, which indicates the number of leaves your tree should contain. ($1 \leq N \leq 10^{18}$)

Output

For each case, you should print $V+2$ lines, where V is the number of non-empty trees in M . ($1 \leq V \leq 125$).

In the first line, you should print single integer V .

In the next V lines, you should print two space-separated integer L_i, R_i , which indicates the index of left subtree and right subtree for a tree with index i . ($-1 \leq L_i, R_i \leq i - 1$).

In the $V + 2$ -th line, you should print P , the index of the tree which contains N nodes. ($0 \leq P \leq V - 1$).

It is guaranteed that an answer always exists under the given condition.

Example

standard input	standard output
4	1
1	-1 -1
2	0
3	3
4	-1 -1
	-1 -1
	0 1
	2
	3
	-1 -1
	0 0
	1 0
	2
	5
	-1 -1
	0 0
	0 0
	2 1
	1 2
	3

Problem D. 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 $W_1, W_2 \dots W_N$, 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 E. Fractions

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

About 44 days are left before *College Scholastic Ability Test* is held. This exam aims to measure students' achievement of National Curriculum standards and scholastic ability required for college education. (<http://www.kice.re.kr/sub/info.do?m=0205&s=english>)

One of the subjects covered by this test is Mathematics, which consists of 21 multiple choice questions and 9 short-answer questions. The answer of each short-answer question is guaranteed to be a *unique positive integer below 1 000*, as you can see from the answer sheet below.

2018학년도 대학수학능력시험 답안지

② 교시 수학영역

※ 결시자 확인 (수험생은 표기하지 않음)
검은색 컴퓨터용 사인펜을 사용하여 수험번호칸과 열칸을 표기

※ 문제지 표지에 안내된 필적 확인 문구를 아래 '필적 확인란'에 정자로 반드시 기재하여야 합니다.

필적 확인란

성명

수험번호

문형

출수형 ☐

박수형 ☐

※ 문제지의 문항을 확인 후 표기

※ 감독관 확인 (수험생은 표기하지 않음)
본인 여부, 수험번호 및 문형의 표기가 정확한지 확인, 열칸에 서명 또는 날인

서명 또는 날인

문번 답란 문번 답란 문번 답란

22번 23번 24번

25번 26번 27번 28번 29번 30번

※ 단답형 답란 표기방법

- 십진법에 의하되, 반드시 자리에 맞추어 표기
- 정답이 한 자리인 경우 일의 자리에만 표기하거나, 십의 자리에 표기하고 일의 자리에 표기
- 예시
 - [정답 100] → 백의 자리 0, 십의 자리 0, 일의 자리 0
 - [정답 98] → 십의 자리 9, 일의 자리 8
 - [정답 5] → 일의 자리 5 또는 십의 자리 0, 일의 자리 5

한국교육과정평가원

However, the organizers might want to give students short-answer questions with non-integer answers, such as $2\sqrt{3}$ or $\frac{5}{3}$. Usually, the workaround is to write the answer in a canonical form, and then sum up all the integers inside that form and ask students to write that number instead.

In particular, when the answer is a positive rational number $\frac{a}{b}$, the organizers usually ask students to reduce it and sum up the numerator and the denominator of the reduced fraction. For example, when the answer is $\frac{18}{10}$, the student should reduce it to $\frac{9}{5}$ and write the final answer as $9 + 5 = 14$.

28. 한 학생 분수를 가장 간단 해, 약분이 안되는 분수가 약분이 안되는 분수보다 큰 비율은 $\frac{q}{p}$ 이다. $p+q$ 의 값을 구하시오.
(단, p 와 q 는 서로소인 자연수이다.) (40점)

However, when the answer is $\frac{521}{500}$, the reduced fraction is also $\frac{521}{500}$, so the student should write the final answer as $521+500 = 1021$. But this shouldn't happen, since all the answers for the short-answer questions are below 1 000. To avoid this situation, the organizers should make sure that after reducing the fraction, the sum of the numerator and the denominator shouldn't exceed 999. Let's call such fractions as *Suneung Fractions*. For example, $\frac{1996}{2}$ and $\frac{18}{10}$ are *Suneung fractions*, while $\frac{1998}{2}$ and $\frac{521}{500}$ are not.

Suppose that, this year, one of the organizers wrote a problem, and the answer to that problem is $\frac{x}{y}$. Since the problem is not finalized yet, the only thing we know is $A \leq x \leq B$ and $C \leq y \leq D$ holds, for given A, B, C, D . The organizers want to know, among all the pairs (x, y) , how many of $\frac{x}{y}$ is a *Suneung fraction*. Write a program that counts this number.

Input

The first and only line contains four space-separated integers A, B, C and D ($1 \leq A \leq B \leq 10^{12}$, $1 \leq C \leq D \leq 10^{12}$)

Output

Print the number of integral pairs (x, y) ($A \leq x \leq B$, $C \leq y \leq D$), where $\frac{x}{y}$ is a *Suneung fraction*.

Examples

standard input	standard output
5 8 3 6	16
2018 2019 2018 2019	2

Problem F. Game on 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	First
3	Second
5	

Problem G. Histogram Sequence

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

A histogram is a polygon made by aligning N adjacent rectangles that share a common base line. Each rectangle is called a *bar*. The i -th bar from the left has width 1 and height H_i .

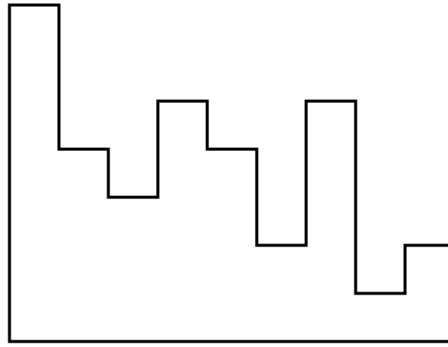


Figure: This picture depicts a case when $N = 9$ and $H = [7, 4, 3, 5, 4, 2, 5, 1, 2]$.

One day, you wanted to find the area of the largest rectangle contained in the given histogram. What you did was to make a list of integers A by the following procedure:

- For each $1 \leq i \leq j \leq N$, calculate the largest area of the rectangle contained in the histogram, where the rectangle's base line coincides with the base line of the $i, i+1, \dots, j-1, j$ -th bar. Add the area to the list A .

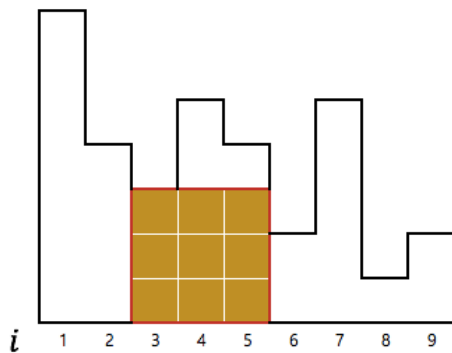


Figure: This picture depicts a case when $i = 3$ and $j = 5$. The area is 9.

The length of the list A is exactly $\frac{N(N+1)}{2}$ since you chose each pair (i, j) exactly once. To make your life easier, you sorted the list A in non-decreasing order. Now, to find the largest area of the rectangle contained in the histogram, you just need to read the last element of A , $A_{N(N+1)/2}$.

However, you are not satisfied with this at all, so I decided to let you compute some part of the list A . You have to write a program that, given two indices L and R ($L \leq R$), calculate the values $A_{L..R}$, i.e. $A_L, A_{L+1}, \dots, A_{R-1}, A_R$.

Input

The first line of the input contains an integer N ($1 \leq N \leq 300\,000$) which is the number of bars in the histogram.

The next line contains N space-separated positive integers $H_1, H_2, \dots, H_N$ ($1 \leq H_i \leq 10^9$), where H_i is the height of the i -th bar.

The last line contains two integers L and R ($1 \leq L \leq R \leq \frac{N(N+1)}{2}$, $R - L + 1 \leq 300\,000$).

Output

Print $R - L + 1$ integers. The j -th ($1 \leq j \leq R - L + 1$) of them should be the $(L + j - 1)$ -th element of the list A , i.e. A_{L+j-1} .

Example

standard input	standard output
9 7 4 3 5 4 2 5 1 2 42 45	12 12 14 15

Problem H. Timsort

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 128 megabytes

Python uses a sorting algorithm called Timsort. In short, the algorithm splits the list into mostly sorted subarrays, sorts each subarray using insertion sort, and merges the sorted subarrays. If the list is already mostly sorted, this algorithm runs very quickly.

In this problem, let's focus on the splitting part:

1. Take the longest non-decreasing ($a_0 \leq a_1 \leq a_2 \leq \dots$) or strictly decreasing ($a_0 > a_1 > a_2 > \dots$) subarray that starts from the current index.
2. If the length of the subarray is less than $MINRUN$, take more elements until the length equals $MINRUN$ or all elements are taken. Let's call these additional elements "bad elements".

MINRUN=3

2	4	4	3	-1	-2	-2	5	6	5	
---	---	---	---	----	----	----	---	---	---	--

MINRUN=4

BAD

BAD BAD

2	4	4	3	-1	-2	-2	5	6	5	
---	---	---	---	----	----	----	---	---	---	--

MINRUN=5

BAD BAD

BAD

2	4	4	3	-1	-2	-2	5	6	5	
---	---	---	---	----	----	----	---	---	---	--

If $MINRUN$ is too small, there are too many subarrays to merge; if $MINRUN$ is too large, it takes too much time to apply insertion sort. For this reason, it is important to set an appropriate value of $MINRUN$. Also, $N/MINRUN$ should be close to a power of 2 to ensure a balanced merging, but we will not consider this in this problem. For each $MINRUN$ value, find the number of subarrays and bad elements.

Input

The first line consists of a single integer N ($5 \leq N \leq 100,000$), the length of the array. The second line consists of N integers, the elements of the array. The absolute value of each element is at most 10^9 . The third line consists of a single integer Q ($1 \leq Q \leq 100,000$), the number of queries. Each of the next Q lines contains a single integer $MINRUN$ ($2 \leq MINRUN \leq N$).

Output

For each query, print the number of subarrays and bad elements in one line.

Example

standard input	standard output
15	5 0
2 4 4 3 -1 -2 -2 5 6 5 4 3 2 3 4	4 3
3	3 5
3	
4	
5	

Problem I. Utilitarianism

Input file: **standard input**
Output file: **standard output**
Time limit: 8 seconds
Memory limit: 1024 megabytes

In RUN-land, there are n cities numbered 1 to n . Some pairs of cities are connected by a bidirectional road. It happens that there are $n - 1$ roads in total and that for any two cities, there is a unique path from one to the other. Also, each road is assigned an integer called the *value*.

Today, to honor the k co-founders of RUN-land, Alex, the king of RUN-land, has decided to choose k different roads and give one road to each of the k co-founders. To prevent unnecessary conflicts, there should be no city that is connected to more than one chosen roads.

In this process, Alex, the king of RUN-land, does not care about who gets what road. Instead, Alex, the king of RUN-land, is only interested in the sum of the values of the k chosen roads. Your task is to choose the roads to maximize this sum.

Input

The first line contains two integers n and k ($2 \leq n \leq 250000, 1 \leq k \leq n - 1$), which are the number of cities in RUN-land, and the number of roads to choose. Each of the next $n - 1$ lines contains three integers u, v, c ($1 \leq u, v \leq n, -10^6 \leq c \leq 10^6$), which means that the city u and the city v are directly connected with a bidirectional road with value c .

Output

If there is no way to choose k roads to satisfy the conditions, print **Impossible**. Otherwise, print one integer, the maximum sum of the values of the k chosen roads.

Examples

standard input	standard output
5 1 1 2 2 2 3 3 2 4 10 4 5 6	10
5 2 1 2 2 2 3 3 2 4 10 4 5 6	9
5 3 1 2 2 2 3 3 2 4 10 4 5 6	Impossible

Problem J. 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 i th number a_i of them is the x -coordinate of the i th 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 i th of which is the integer a_{i-1} ($1 \leq a_0 < \dots < a_{2n-1} \leq 10^6$).

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

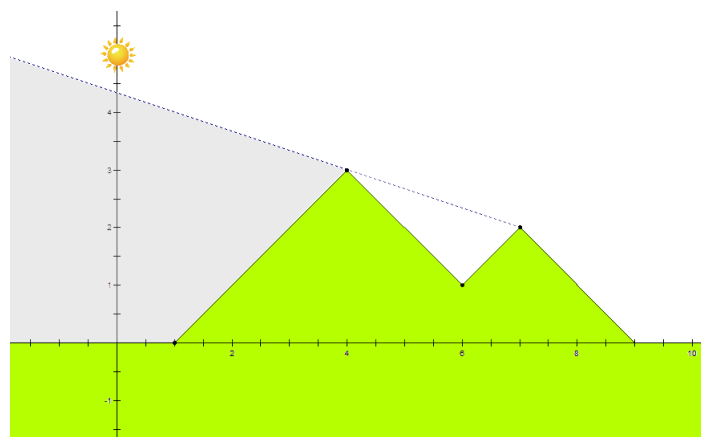


Figure: Illustration of the first example.

Problem K. Voronoi Diagram Returns

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

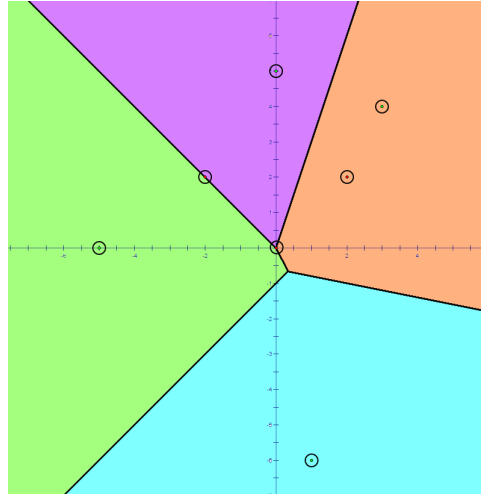


Figure: Voronoi Diagram of size 4.

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 $\{P_1, P_2, \dots, P_n\}$ 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 $\{P_1, P_2, \dots, P_n\}$, 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 i th 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 j th 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 j th 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 L. Repetitive Palindrome

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

You are given a string s consisting of lowercase alphabets, and an integer k .

Make a new string t by concatenating k copies of s . Determine whether t is a palindrome, e.g. is the same backward as forward.

Input

The first line contains a string s consisting of lowercase alphabets. ($1 \leq |s| \leq 250000$)

The second line contains an integer k . ($1 \leq k \leq 10^{18}$)

Output

If t is a palindrome, print YES. If not, print NO.

Examples

standard input	standard output
abc 3	NO
abba 1	YES

Problem M. Coke Challenge

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

Mr. Jeong really loves coke. He loves so much that he drinks coke everyday without exception. One day, he decided to open a coke contest in Daejeon. To the winner, a box of cokes will be given!

N people participate in the contest. Each participant is given K mL of coke, and the one who finishes his/her coke earliest wins the contest. But it is painful to drink the whole coke in one time, so each person repeats drinking and taking a rest. More specifically, as soon as the contest starts, the i th participant starts to drink for t_i seconds, then takes a rest for s_i seconds, and repeats this process until no more coke remains. Moreover, everyone drinks A mL per second. The contest is over if one of the participants finished his/her coke.

Given the information of N participants, determine after how many seconds the contest is finished, since the contest begins.

Input

The input starts with the first line containing three integers N ($2 \leq N \leq 1000$), K ($1 \leq K \leq 10000$), and A ($1 \leq A \leq 100$). The i th of the next N lines contains two integers t_i ($1 \leq t_i \leq 100$) and s_i ($1 \leq s_i \leq 100$), the information of i th participant. K is a multiple of A .

Output

Write a single integer, the answer to the question.

Examples

standard input	standard output
2 100 1 10 5 5 10	145
4 100 2 30 30 49 2 50 50 20 10	50