Login : **team30**

Password : **n_#4zK!knb**

mnpc22
Moroccan National Programming Contest

Saturday October 15, 2022

Moroccan National Programming Contest 2022

Organized by

Moroccan Association For Computing Science (MACS)

Hosted/Sponsored by

1337 Coding School Khouribga

Problem A. Get in the Queue

Balloon Color:	Green
Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

Lekssays went to the hypermarket to buy what he needed. When he finished, he went to pay, then he noticed that all cashiers were queued. Because Lekssays is in a hurry, he asks to help him choose the cashier to pay and go out from the hypermarket the fastest way possible.

Given n cashiers, every cashier i processes a client with a processing rate of r_i seconds. Cashiers can change at certain times, so the one who will replace him/her might have a different r_i.

You are given m clients. Every client i will enter a queue q_i at time t_i. Then, you will be given c changes of cashiers. Every change j will replace the cashier at position c_j, at time t_i, with a new cashier with a new rate r_j.

Lekssays will arrive at time T. Help him choose which queue he needs to enter to go out the fastest way possible.

Input

The first line contains an integer $1 \leq n \leq 10^5$, the number of cashiers present at the queue in the beginning. The next line contains $1 \leq r_i \leq 10^9$, the client's processing rate for the cashier $0 \leq c_i \leq 10^5$.

Next, you are given an integer $1 \leq m \leq 2 * 10^5$, the number of clients entering the queues. The next m lines contain two integers q_i and t_i denoting the queue id and the time of entry, respectively.

The next line contains an integer $0 \leq c \leq 10^5$ the number of cashiers changes. The following c lines contain three integers q_j, t_j, and r_j denoting the queue id where the cashier will be replaced, the time of replacement, and the processing rate of the new cashier.

In the last line, you are given an integer $1 \leq T \leq 10^9$ denoting the time when Lekssays will arrive to pay.

Output

Print the queue id q_i and the time t_i when Lekssays will go out. If there are multiple answers, print the one with the lower queue id.

Example

standard input	standard output
3 1 2 3 7 1 1 2 1 3 1 1 2 3 3 2 3 1 4 3 1 2 3 2 3 4 1 2 3 3	2 7

Note

It is guaranteed that every cashier will serve at least one client and no change of cashiers will happen simultaneously.

Problem B. Diaa does SMS

Balloon Color:	Blue
Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

We all know Diaa as a MACS member, but very few people know that his initial professional profile is as a telecom engineer. Diaa had a critical task in the past to generate a complete list of verbose information about a set of problematic subscribers on the SMS solution.

As lazy as he is, he asked 2 of his colleagues to give him the data, unfortunately, none of the provided data was complete, one set of data contains only the MSIDN and the IMSI , and the other contains only the IMSI and the region.

Your task is to generate one list of the 3 pieces of data correctly mapped.

Input

The first line of the input contains an integer $0 \leq N \leq 100$ indicating the total number of lines in the first list.

Followed by a N lines, each line contains 2 Strings separated by a comma: MSISDN,IMSI

MSISDN is a String starting with '+' and followed by 12 digits.

IMSI is a String composed of 14 digits.

After this list you will find a line containing one integer $1 \leq M \leq 100$, indicating the total number of lines of the second list, followed by M lines.

Each line is composed of 2 Strings separated by a comma: Region,MSISDN

Region is a String composed of Alphabet characters, and MSISDN has the same form as in the first list.

Output

For each MSISDN print one line with the 3 information:

MSISDN,IMSI,region

Sorted Alphabetically by MSIDN then IMSI and by region.

Example

standard input	standard output
5 +212661080808,604011234567890 +212661080899,604011234567855 +212661080811,604011234567443 +212661082208,604011234561234 +212661120808,604011234567444 3 Tangiers,+212661080808 Rabat,+212661080811 Taza,+212661080812	+212661080808,604011234567890,Tangiers +212661080811,604011234567443,Rabat

Note

If you cannot find a correct mapping for a MSISDN, you should ignore it from the final output.

If no mappings could be found print "*nomatch*" without quotes.

Problem C. MH Matches

Balloon Color:	Purple
Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

Mehdi and Houda love challenges; this time, they brought you a nice challenge. You are given a string s. You move it to the right by one position at a time. In addition, you remove the last character at each move. Then, you compare it with the unmoved version to get the number of matches the moving string gives.

Your task is to find the move that gives the maximum character matches between s and its moved versions.

Input

The first line contains a string s composed only of characters `m`, `h` where $1 \leq |s| \leq 5000$.

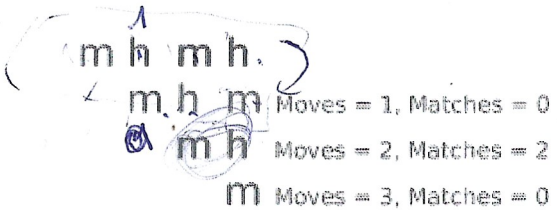
Examples

standard input	standard output
mhmh	2 2
mh	0 0
mhmmhmhh	4 3

Note

This problem is an easier version of *Problem I - XYZ Matches*.

Explanation of Testcase #1:



Problem D. A Multiplication Problem

Balloon Color:	White
Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

You are with your favorite math professor, explaining how hard it is to do multiplication in your head. Luckily, you are smart enough to write a program to automate this for you.

Your task is simple, write a program that verifies if the multiplication of two given numbers $A * B$ is larger or equal to C.

Input

First and only line contains 3 integers $0 \leq A, B, C \leq 10^{18}$.

Output

Print Yes if the multiplication of A and B is larger or equal to C and No otherwise.

Examples

standard input	standard output
10 10 100	Yes
6 7 43	No

Problem E. No Name Problem

Balloon Color:	Yellow
Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

You are given an array A of size n and a number k. You are tasked to find the size of the maximum subarray where the difference between the minimum element in the subarray and the maximum element of the subarray does not exceed k.

That is $Max(A[i..j]) - Min(A[i..j]) <= k$.

Input

The first line of the input consists of two integers, $1 \leq n \leq 10^5$ and $0 \leq k \leq 2 \times 10^9$.

The next line of the input consists of the elements of the array where $-10^9 \leq A_i \leq 10^9$

Output

Output one single number, the size of the maximum subarray where the maximum gap does not exceed k.

Examples

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

Problem F. A New Scheduler

Balloon Color:	Orange
Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

You are given a list of task schedules that should be run on a multi-core CPU. Each task takes one CPU slice but should be executed in the specified time range. Knowing the number of cores in the new CPU model, can you make a scheduler that will optimally allocate CPU cores to tasks so that all tasks are executed or report that it is impossible?

Input

The first line of input contains two numbers $1 \le N \le 1000$ number of tasks and $1 \le C \le 100$ the number of CPU cores.

N lines follow, each line contains two integers $1 \le s_i \le e_i \le 100$ the time range (in quantum) where we can execute task i.

Output

If we can run every task within the given constraints print Yes otherwise, print No.

Examples

standard input	standard output
4 3 1 2 2 3 2 3 1 1	Yes
4 1 1 2 2 3 2 3 1 1	No

Note

A CPU core can only execute **at most one** single task in a given slice.

Problem H. Oriented Lamps

Balloon Color:	Red
Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

Given an $N \times N$ grid, and a bunch of lamps identified by (x_i, y_i) positions, each lamp can be set either horizontally or vertically (but not both) and can propagate light in that direction up to a distance K.

What is the maximum distance $1 \leq K$, such that when all the lamps are turned on, no cell in the grid is highlighted by 2 or more lamps shooting light in the same direction?

You are free to configure the direction of each lamp independently as long as the condition is not broken.

Input

First line contains 2 integers, $2 \leq N \leq 10^9$ and $1 \leq M \leq 2000$, size of the grid side, and number of lamps.

M lines follow, each containing 2 integers $1 \leq x, y \leq N$ positions of the lamps

Output

Print a single integer K denoting the maximum distance, -1 if no configuration will allow you to do it, and **inf** if the distance can be as big as you want.

Examples

standard input	standard output
4 2 2 2 3 4	inf
3 6 1 3 1 2 1 1 3 2 3 1 3 3	-1

Note

Each lamp, when turned on, will light up to $2 * K + 1$ cells (either horizontally or vertically), where K is the maximum distance for the lamps.

Problem I. XYZ Matches

Balloon Color:	Gold
Input file:	standard input
Output file:	standard output
Time limit:	1 second
Memory limit:	256 megabytes

Nour and Manal brought you another nice challenge. You are given a string s. You move it to the right by one position at a time. In addition, you remove the last character at each move. Then, you compare it with the unmoved version to get the number of matches the moving string gives.

Your task is to find the move that gives the maximum character matches between s and its moved versions.

Input

The first line contains a string s composed only of characters `x`, `y`, and `z` where $1 \leq |s| \leq 10^5$.

Output

The output contains two integers c and m denoting the maximum character matches and the move where they have been gotten. Print the lowest move if multiple moves give the maximum character matches.

Examples

standard input	standard output
xxxyz	2 1
zxzzxzxxx	4 2
xyz	0 0

Note

Refer to the explanation of testcases in *Problem C - MH Matches.*

Problem J. ZigZag Jumps

Balloon Color:	Black
Input file:	standard input
Output file:	standard output
Time limit:	2 seconds
Memory limit:	256 megabytes

ZigZag just got a new array A as a present and has been playing with it all day long. ZigZag uses the array to jump around the elements. He came up with the following rules for his jumps:

ZigZag can only jump from index i to index j such that $j > i$ in the following way:

- during odd-numbered jumps (i.e., jumps 1, 3, 5, and so on), you jump to the smallest index j such that $A_i < A_j$ and $A_j - A_i$ is minimal among all possible j's.
- during even-numbered jumps (i.e., jumps 2, 4, 6, and so on), you jump to the smallest index j such that $A_i > A_j$ and $A_i - A_j$ is minimal among all possible j's.

It may be that there is no legal jump for some index i. In this case, the jumping stops.

ZigZag considers the jumps to be satisfying if he can reach the end of the array (i.e., index $n - 1$ where n is the size of the array) respecting the rules explained above.

ZigZag is now wondering that given an Array A, he can choose any index where he can start his ZigZag jumps following the rules above; how many valid starting positions are there?

Input

The first line of the input consists of an integer n denoting the number of elements in the array A. The second line of the input contains n elements of the array such that $0 \leq A_i \leq 10^9$.

Output

Print one integer, the number of valid starting positions.

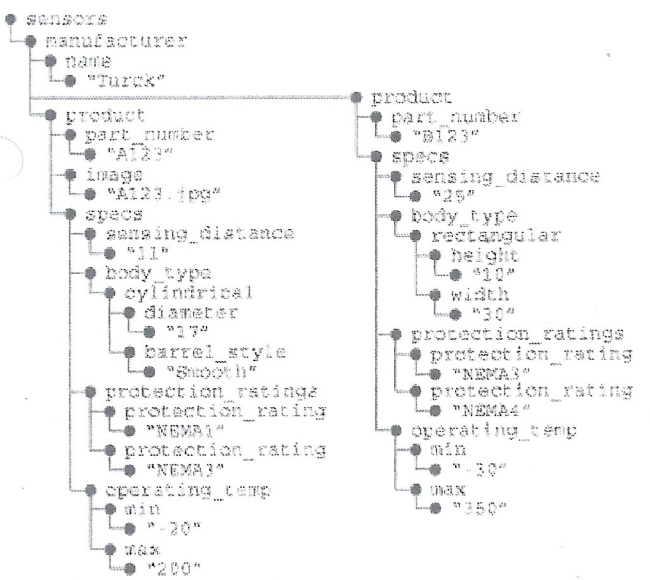
Examples

standard input	standard output
5 10 13 12 14 15	2
5 10 11 14 11 10	2

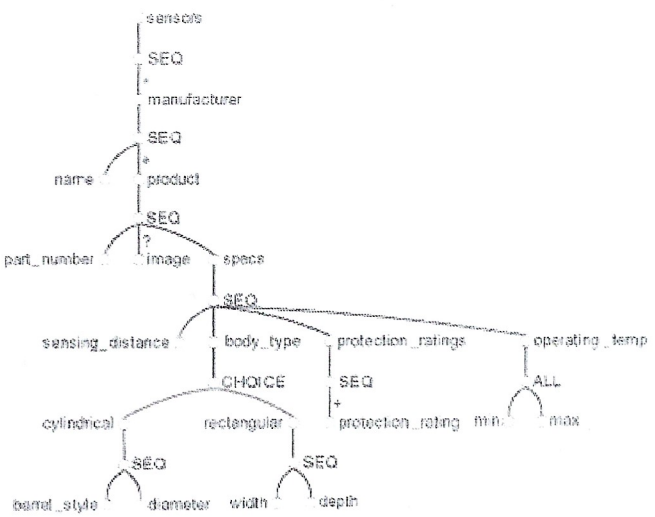
Problem K. XML COV

Balloon Color:	`Pink`
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

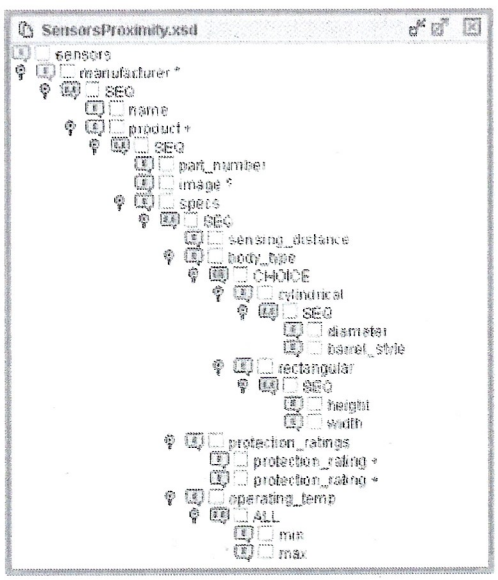
XML (Extensible Markup Language) is a markup language and file format for storing, transmitting, and reconstructing arbitrary data. It defines a set of rules for encoding documents in a human-readable and machine-readable format. –*Wikipedia*



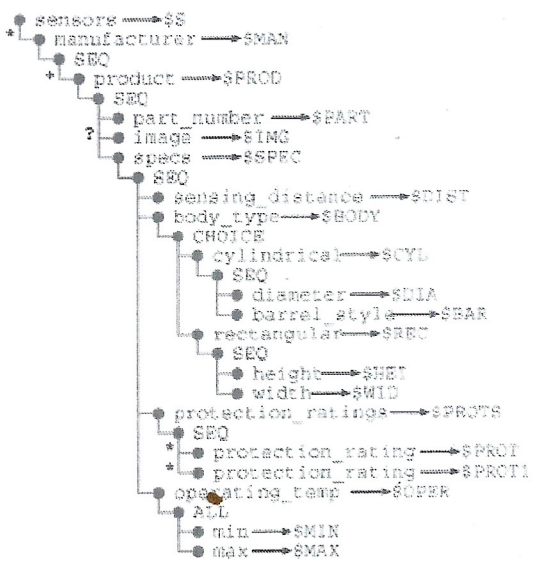
(a) Data Set (*loto*)



(b) XML Schema



(c) Expanded Schema Tree (*EST*)



(d) Expanded Schema Tree (*EST*)

Example of an XML schema

Anouar is a computer science student, he loves learning new technologies and algorithms. Unfortunately, this year one of his teachers asked him to make an XML schema coverage tool using the programming language of his choice. The task is not hard, but Anouar hates XML, so he needs your help.

We will use the term **object** to denote nodes with children nodes, and **attribute** to denote nodes with no children nodes also known as *leaf nodes*.

You will be given a stream of queries consisting of two types:

- **Type 1:** You are given a path expression representing access to a node in the XML schema.

  The path syntax is similar to a filesystem path where the first '/' represents the root node of the schema, and the rest of the nodes are separated with a '/'.

  Example from the XML schema (b) above: '/sensors/manufacturer/SEQ/name'.

  Note: Accessing an **object** is equivalent to accessing all its children nodes.

- **Type 2:** You are given a path expression representing a node in the XML tree. Print the coverage of that node.

  The coverage of a node X is defined as: $coverage(X) = \frac{A(X)}{T(X)}$ such as:

  $A(X)$ = number of accessed attributes under the node X

  $T(X)$ = number of attributes under the node X

A node is considered **accessed** if and only if it has been accessed at least once via a **Type 1** query.

Input

The first line contains 2 integers N ($1 \leq N \leq 10^4$) and Q ($1 \leq Q \leq 10^4$), the number of attributes and the number of queries accordingly.

The following N lines contain each a path representing an attribute in the XML schema.

The last Q lines contain an integer $T \in \{1, 2\}$ the query type followed by a string P the path to the node.

Output

Print a line for each **Type 2** query, the coverage for the given node.

The answer will be considered correct if the absolute or relative error doesn't exceed 10^{-6}.

Example

standard input	standard output
7 4 /user/firstname /user/lastname /user/email /user/birthdate/day /user/birthdate/month /user/birthdate/year /user/address 1 /user/email 1 /user/birthdate/day 1 /user/address 2 /user	0.428571

Note

The coverage of the node */user* is the number of visited attributes under */user* divided by the total number of attributes in the schema.

The visited attributes are */user/email*, */user/birthdate/day* and */user/address* therefore the numerator is 3.

The total number of attributes under */user* is equal to 7.

$coverage(/user) = 3/7 = 0.4285714285$