

Problem A. Nutella's Life

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Website `chefforces.at` has just published a schedule of contests for the next year! There will be n contests and no changes in the schedule. Oleg got really excited and decided to maximize the fun.

After a thorough analysis of each contest's problem setters Oleg came up with n integers a_i , one for each contest. The number a_i is the amount of fun Oleg will get while playing the i -th contest. Note that, due to a *notorious coincidence*, some numbers a_i can be negative.

However, Oleg does not want to miss contests and, especially, to miss several contests in a row. Formally, if Oleg decides to skip a contest, and he has already skipped x contests which took place immediately before this one, his total fun decreases by $x + 1$.

Finally, Oleg wants each contest to be at least as fun as the previous one in which he has participated. In other words, if Oleg participates in contests with numbers i and j , $i < j$, then the condition $a_i \leq a_j$ must hold.

Help Oleg to decide in which contests he has to participate in order to maximize the total fun.

Input

The first line contains an integer n , the number of contests in the schedule ($1 \leq n \leq 10^5$).

The second line contains n integers a_i ($-10^9 \leq a_i \leq 10^9$).

Output

The only line of the output must contain an integer: the maximal amount of fun Oleg can get.

Examples

standard input	standard output
7 1 3 2 7 3 2 4	7
7 -3 -4 -2 -2 -6 -8 -1	-11

Problem B. Oleg and Data Science

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Nowadays, everybody has heard about machine learning, neural networks and big data. And so has the student Oleg, who also wants to be in trend. He began to work hard at analyzing various datasets with Python. It's so nice to come to co-working, add a couple of layers to the neural network and, leaning back in the chair, sip the smoothies while the computer processes gigabytes of data! But today, something went wrong, and Oleg asks for your help.

Initially, he had an array a containing very important data: all integers from L to R , inclusive. Then Oleg wrote a function $f(a, m)$ which returns a new array where each integer is replaced by its remainder modulo m . Finally, Oleg mistyped and executed the line $a = f(a, Q)$, thereby replacing the original array a ! To assess the scale of the tragedy, he wants to calculate the number of such positive integers X that, regardless of the content of the original array a , the result of the function $f(a, X)$ will be the same as if Oleg had not executed that hapless line.

If the problem is not clear yet, here is the mathematical statement. It is required to calculate the number of such positive integers X that, for all integers S from the segment $[L, R]$, it is true that

$$((S \bmod Q) \bmod X) = (S \bmod X).$$

Input

The only line contains three integers separated by spaces: L , R and Q ($1 \leq L, R, Q \leq 10^{12}$, $L \leq R$).

Output

If the number of suitable positive integers X is finite, print it. Otherwise, print the word "infinity".

Examples

standard input	standard output
1 1 1	1
3 5 2	2
1 10 10	4
1 1 2	infinity

Problem C. Christmas Garland

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Once upon a time, Nikita was relaxing at home and watching a Christmas garland. The light bulbs were flickering following some strange pattern.

Let us formalize the garland's description. It consists of n colored light bulbs. Every light bulb is either on or off at any moment. Initially, all of them are off.

Sometimes all light bulbs of one color change their states to the opposite. After each such change, Nikita wants to know the number of maximal non-empty continuous segments of lit bulbs. A lit segment is maximal if it is not contained in any other lit segment.

Input

The first line contains integers n , k and q : the number of light bulbs, the number of different colors and the number of changes of the garland ($1 \leq n, q \leq 2 \cdot 10^5$, $1 \leq k \leq n$).

The second line contains n integers c_1, c_2, \dots, c_n : the colors of light bulbs in the garland ($1 \leq c_i \leq k$).

Next q lines describe changes of the garland in the order they happened. Each of these lines contains an integer d_i , the color of light bulbs which have just changed their states ($1 \leq d_i \leq k$).

Output

The output must contain q lines. The i -th line must contain one integer: the number of maximal continuous segments of lit bulbs after i -th change.

Example

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

Problem D. Anna and Lucky Tickets

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Anna is a very diligent student. That is why she attends all classes in the university. Every morning she gets on the bus and, applying E-card to a special device, gets a ticket with an n -digit decimal number on it (possibly with leading zeros).

To entertain herself during the trip, Anna checks if the ticket is lucky. She knows two methods of verification. According to the first method, the ticket is lucky if the sum of the digits standing on even positions is equal to the sum of the digits standing on odd positions (the positions are numbered from left to right starting from one). According to the second method, the ticket is lucky if the sum of the first $\lfloor n/2 \rfloor$ digits is equal to the sum of the last $\lfloor n/2 \rfloor$ digits (in particular, if the length of the ticket number is odd, then the middle digit is not taken into account).

If the ticket is lucky according to both methods, then Anna is frightened that everything goes too well, and calls such ticket unlucky. Of course, if the ticket is not lucky in any of the ways, it is also called unlucky.

Anna gets very angry when she sees an unlucky ticket whose number is a palindrome (that is, the number stays the same if we read it from right to left instead of from left to right). Help us calculate the number of such tickets! Oh, yes, there can be a lot of them, so output this number modulo $10^9 + 7$.

Input

A single line of input contains an integer n , the length of the ticket number ($2 \leq n \leq 10^6$).

Output

Output an integer: the number of n -digit unlucky tickets whose numbers are palindromes, taken modulo $10^9 + 7$.

Example

standard input	standard output
3	5

Problem E. Octopus

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Vlad is engaged in underwater photography. Most of all he likes photographing squids and octopuses. For example, did you know that an octopus has three hearts? And the fact that they can change the shape and color of the body? And here is one more astonishing... Oh, come on, Vlad, get away from the keyboard, and enough to write about the octopuses!

Vlad is not only a photographer, but also an excellent programmer. He developed a special program for automatic octopus recognition in photographs. The program receives a photograph as the input. And as the output, it gives a representation of octopus in the form of an *undirected graph*. Informally, an undirected graph is a set of points and a set of edges connecting them. The graph obtained by recognizing an octopus image always has a specific form: three or more points form the *body*, and the remaining points are parts of *tentacles* which are attached to body points so that **no more than one** tentacle is attached to each point of the body. The body is a sequence of points in which any two adjacent points, as well as the first and the last points, are connected by an edge, and the number of edges connected to each point is either two or three (if a tentacle is attached to it). A tentacle is a sequence of points in which any two adjacent points are connected by an edge, and the number of edges connected to each point is either two or one (in each tentacle, there is exactly one point with only one edge). Additionally, in this graph, there is no more than one edge between any two points, and every edge connects two different points.

When Vlad collected all his photographs, he launched the recognition process. After several hours of waiting, he got the result. However, because of a tricky bug in the code, one extra edge was added to all the graphs! After rereading the source code of his program, Vlad realized that exactly one edge was indeed added to each graph. Moreover, the added edge connects two different points which were not previously directly connected by an edge. Help Vlad to find the extra edge, and in return, he will tell you more amazing facts about octopuses!

Input

The first line contains integers n and m separated by a space: the number of points and the number of edges in the graph ($1 \leq n, m \leq 10^5$).

The following m lines describe the edges of the graph. Each edge is given by two different integers separated by a space: the numbers of points connected by this edge. The points are numbered starting from one. Each pair of points occurs no more than once.

It is guaranteed that the given graph was obtained by adding exactly one edge to a graph resulting from an octopus image recognition.

Output

In a single line, output two integers separated by a space: the numbers of the points between which you need to remove the edge. You can output the numbers in any order.

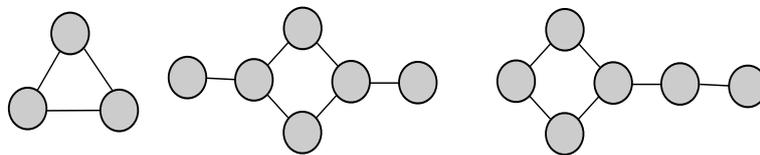
If there are several correct answers, output any one of them.

Example

standard input	standard output
7 8 1 2 2 3 3 4 4 1 3 5 5 6 5 7 3 6	3 2

Note

Here are examples of graphs which **are** correct representations of octopuses:



Here are examples of graphs which **are not** correct representations of octopuses:

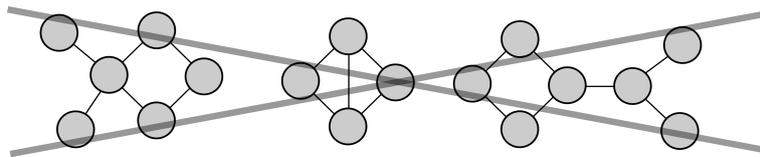
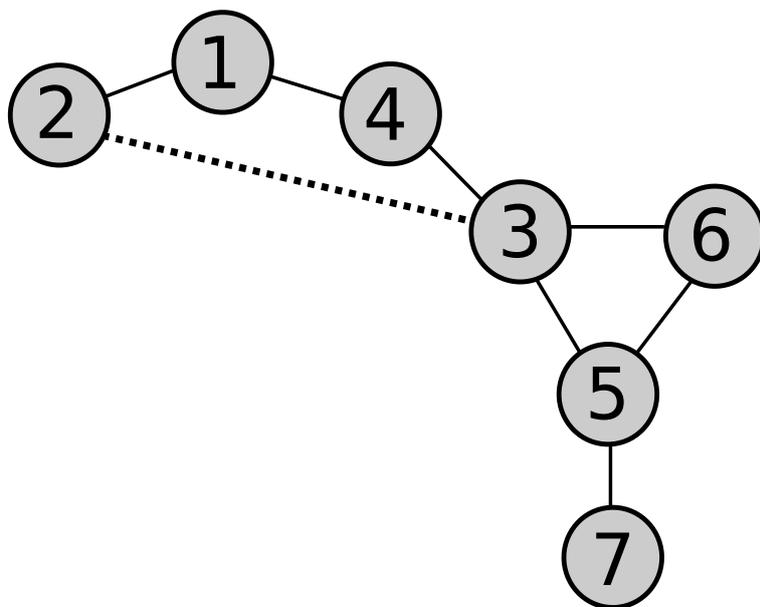


Illustration of the first test:



Problem F. Secret Permutation

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

This is an interactive problem.

There is a secret permutation p of integers from 0 to $n - 1$. The permutation is indexed starting from 0. You have to guess it by asking questions of the form “? a_i b_i c_i ” (a_i , b_i and c_i are integers from 0 to $n - 1$). For each such question, you will get one number in response which equals $p^{-1}(p(a_i) \cdot p(b_i) + p(c_i))$ (all operations are performed modulo n , and $p^{-1}(x)$ is such y that $p(y) = x$). In the end, you have to print the guessed permutation in the form “! $p(0)$ $p(1)$... $p(n - 1)$ ”.

Input

The only line of the input contains an integer n ($1 \leq n \leq 5 \cdot 10^3$).

In each test, the permutation is fixed in before the contest and does not change during the guessing process.

For each test, the length n was picked by the jury, but the permutation was then generated using a pseudorandom number generator. However, the problem has a deterministic solution which works for every possible permutation.

Interaction Protocol

You can ask zero or more questions and make exactly one guess in the end. Print each query on a separate line and do not forget to flush the output buffer. The maximum allowed number of queries, **including** the final guess, is 12512.

Example

standard input	standard output
4	? 1 2 3
3	? 0 3 3
2	? 0 0 1
1	! 2 0 3 1

Problem G. Polygon Rotation

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 256 mebibytes

You are given a convex polygon which contains the origin strictly inside. The polygon is slowly rotated by angle α around the origin in counter-clockwise order. Calculate the swept area, that is, the area of the set of all points which were covered by the polygon at least once during its rotation.

Input

Each test consists of one or more test cases.

The first line contains a positive integer T , the number of test cases. Then T test cases are given.

The first line of each test case contains an integer n , the number of vertices of the polygon ($3 \leq n \leq 10^5$), followed by a real number α , the rotation angle in radians, given with exactly six digits after the decimal point ($0 < \alpha < 2\pi$). The following n lines contain the description of the vertices in counter-clockwise order. Each vertex is described by two integers x_i and y_i which are its coordinates ($-10^9 \leq x_i, y_i \leq 10^9$). It is guaranteed that the polygon has non-zero area, contains the origin strictly inside, and that no three vertices lie on the same line.

The sum of n over all test cases does not exceed 10^5 .

Output

The output must contain T lines, one for each test case.

Each line must contain the answer for the corresponding test case with absolute or relative error no more than 10^{-6} .

Example

standard input	standard output
1 4 0.785398 -1 -1 1 -1 1 1 -1 1	5.484738133371644

Problem H. Mikhail's Problem

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 512 mebibytes

You are given a string s and several queries. For the i -th query, calculate the number of different palindromic substrings of $s[l_i..r_i]$. A substring is called palindromic if it reads the same from right to left as from left to right. Two substrings are considered different if they differ as strings.

Input

The first line contains a non-empty string s consisting of lowercase English letters. The length of the string does not exceed 10^5 characters.

The second line contains an integer q , the number of queries ($1 \leq q \leq 10^5$). Next q lines contain queries. Each of these lines contains integers l_i and r_i separated by a space ($1 \leq l_i \leq r_i \leq |s|$).

Output

Output q lines. The i -th line must contain one integer: the answer to the i -th query.

Example

standard input	standard output
bbabaabbcabcabc	7
10	7
1 7	8
1 8	8
1 9	6
1 10	7
2 7	8
2 8	8
2 9	4
2 10	3
7 15	
8 15	

Problem I. Rat-O-Matic

Input file: *standard input*
Output file: *standard output*
Time limit: 3 seconds
Memory limit: 256 mebibytes

The company Rapid City Dynamics is famous for its dog-like robots, cheetah-like robots and even human-like robots. But big projects need big money, so they decided to build something simpler yet more popular (and sellable!). So now the company is engineering a new robot called Rat-O-Matic, and you, as Rapid City Dynamics employee, are going to take a part!

The robot itself is a mechanical rat which can move around and generate melodies by touching special frames placed on a plane. You can assume the plane is infinite in all directions and there is a Cartesian coordinate system on it. There are exactly n frames. Each frame is described by three integers x , y and r . It consists of the area enclosed between two axis-aligned squares centered at point (x, y) : the first square's radius is r and the second square's radius is $2r$. The radius of a square is the distance from its center to its side. Moreover, every frame has an associated sound. Currently there are only three possible sounds, let us denote them by “r”, “a” and “t”. It is guaranteed that no two frames intersect or touch.

It is possible to interact with the robot by entering the frame's number on a special keyboard (frames are numbered starting from 1). Initially, the rat stands outside of all frames. After getting a command, the rat starts to move towards the specified frame. Thanks to the patented navigation system, the robot picks a path which intersects the minimal possible number of other frames. When a frame is touched, it produces an associated sound, so the rat generates a melody while moving. The robot stops after touching the specified frame.

It is well known that people like familiar melodies. There are m melodies in your database. Each melody is described by a string of characters which encode sounds. A melody generated by the rat is said to be contained in a melody from the database if and only if the former is a substring (a continuous subsequence) of the latter. And the popularity of a generated melody is the number of melodies from the database which contain the generated melody. Note that, even if some melodies from the database appear to be the same, they still must be counted separately.

In attempts of creating the most entertaining layout of frames, you made q experiments. Each experiment has one of the following types:

1. Deactivate the frame number x . It is guaranteed that the frame is currently active. Deactivated frames do not produce any sound. In the beginning, all frames are active.
2. Activate the frame number x . It is guaranteed that the frame is currently deactivated.
3. Calculate the popularity of a melody generated by the robot while moving towards the frame number x starting outside of all frames. It is guaranteed that this frame is active at the moment.

Help yourself to simulate all experiments using just your computer!

Input

The first line contains an integer n , the number of frames ($1 \leq n \leq 2 \cdot 10^5$). The next n lines describe the frames, i -th of them contains integers x_i , y_i , r_i and a character c_i separated by spaces ($-10^8 \leq x_i, y_i \leq 10^8$, $1 \leq r_i \leq 10^8$, and c_i is either “r”, “a” or “t”). These are coordinates of the center of a frame, the radius of the inner square and the associated sound, respectively.

The next line contains an integer m , the number of melodies in the database ($1 \leq m \leq 2 \cdot 10^5$). The next m lines contain the strings representing the melodies. Each string is non-empty and consists of characters “r”, “a” and “t”. The total length of all strings does not exceed $2 \cdot 10^5$.

The next line contains an integer q , the number of experiments ($1 \leq q \leq 2 \cdot 10^5$). The next q lines describe experiments, j -th of them contains a character t_j (“-”, “+” or “?”) and an integer x_j separated

by a space. The character encodes the type of the experiment (“-” is 1, “+” is 2 and “?” is 3), and the integer is the number of the frame.

It is guaranteed that there is at least one experiment of the third type.

Output

For every experiment of the third type, print a single line containing an integer which is the popularity of the generated melody.

Example

standard input	standard output
3	1
3 3 4 r	2
2 4 1 a	3
14 4 1 t	2
3	
rat	
rara	
aaa	
6	
? 3	
? 2	
- 1	
? 2	
+ 1	
? 2	

Problem J. Readability

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

The Martian alphabet consists of 10^9 integers from 1 to 10^9 . Odd integers are vowels, and even integers are consonants. A word is a sequence of such integers.

Why do Martians have such a big alphabet? It gives the language an interesting property: changing the order of the integers in a word does not change its meaning. But, it does change its pronunciation. If each pair of adjacent integers in a word contains one vowel and one consonant, such word is called *readable*.

Vasya Pupkin decided to open a restaurant of Earthern cuisine on Mars. He bought n wooden integers for its name and placed them in a row on n pedestals, numbered from 1 to n . The integer placed on i -th pedestal is denoted as a_i .

Now Vasya wants to reorder the integers, because he wants the name of the restaurant to be readable. But moving an integer from pedestal i to pedestal j costs $|i - j|$ Martian rubles. Integers are flat, so two or more of them can be placed on one pedestal while reordering, but in the end, each pedestal must have exactly one integer on it.

Help Vasya to find a readable word formed by reordering the integers that requires him to pay as little Martian rubles as possible. If there are several such words, choose one that is the smallest lexicographically, because that will put the name of the restaurant earlier in the telephone directory.

A sequence of integers b_i is lexicographically smaller than a sequence of the same length c_i if there exists such index j that $b_i = c_i$ for all $i < j$, but $b_j < c_j$.

Input

The first line contains an integer n , the number of Martian letters in the word ($1 \leq n \leq 10^5$).

The second line contains n space-separated integers a_i : the letters themselves ($1 \leq a_i \leq 10^9$).

It is guaranteed that the solution exists.

Output

Output one line. It must contain n space-separated integers: the required readable word.

Example

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

Problem K. Robotobor

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

After the great success of Rat-O-Matic robot (see problem “I” if you like long stories), you have been promoted to CEO of Ural branch of Rapid City Dynamics company. Inspired by the local traditions, you have decided to create a very special robot called Robotobor!

Robotobor can move on a rectangular grid which consists of $n \times m$ cells. Some cells are blocked, so the robot cannot visit them. Every second, the robot moves to a cell which shares an edge with the current cell.

The movement of the robot is controlled by a program. The program consists of zero or more lines. Each line is a non-empty string of characters. Each character is one of the following: “U” (up), “D” (down), “L” (left) or “R” (right). Each character means that the robot must move to the neighboring cell in the respective direction. The program is executed line by line, every line is executed from left to right.

The program is said to be valid if and only if the following conditions hold:

- During the program execution, the robot does not visit blocked cells and does not leave the grid.
- The length of each line is at most 100 characters.
- Every line is a palindrome, that is, it reads the same right to left as left to right (Ural traditions, as wild as they are).

You want to find a valid program which moves the robot from the cell S to the cell F and consists of the minimum possible number of lines. Note that it is **not necessary** to minimize the total length of the lines. Can you repeat your previous success?

Input

The first line contains integers n and m , the number of rows and the number of columns of the grid ($1 \leq n, m \leq 50$).

The next n lines describe the grid. Each of these lines contains m characters. Each symbol is either “.” (empty cell), “#” (blocked cell), “S” (starting cell) or “F” (target cell). You can assume that starting and target cells are not blocked. It is guaranteed that the grid contains exactly one “S” and exactly one “F”.

Output

If there is no program which satisfies all conditions, then the only line of the output must contain the number -1 .

Otherwise, output the program you found. The first line must contain the number of lines k which must be the minimum possible. The next k lines must contain the program itself. If there are several possible answers, output any one of them.

Examples

standard input	standard output
3 5 S.... .#... ..#F.	2 RR DRD
2 2 F# #S	-1