Problem A
Awesome MST Problem

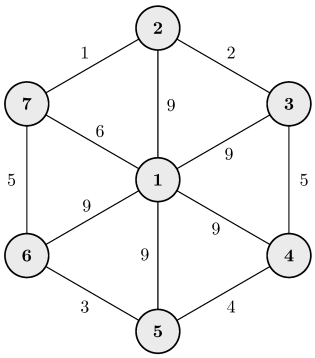
Given a connected, weighted undirected graph with N vertices and M edges. The graph doesn't have any self-loops, and between any pair of vertices, there is at most one edge.

An edge is called **awesome** if there is **at least one** minimum spanning tree containing that edge.

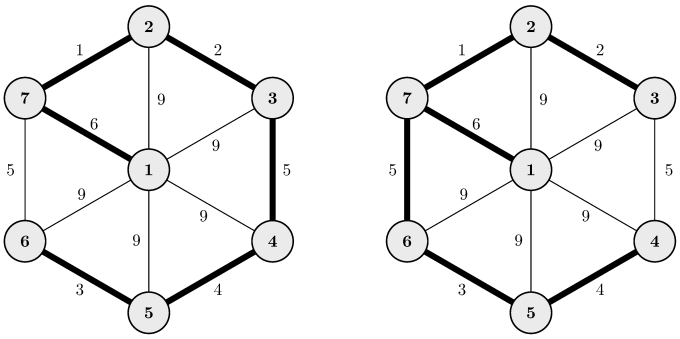
You are given Q queries. In each query, you need to add one new edge to the given graph, and then count the number of **awesome** edges in the new graph. Note that once an edge is added, it stays in the graph forever.

A **minimum spanning tree** of a connected, weighted undirected graph is a subset of its edges that connects all its vertices together, without any cycles and with the minimum possible total edge weight.

For example, consider the following graph:



It has 2 **minimum spanning trees**, presented in two figures below, whose edges are in bold:



The graph has 7 **awesome** edges, which appear in at least one minimum spanning trees: $(2, 7)$, $(2, 3)$, $(7, 1)$, $(7, 6)$, $(3, 4)$, $(6, 5)$ and $(4, 5)$.

Input

The first line of the input contains two integers N and M $(1 \le N, M \le 10^5)$ — the number of vertices and edges of the graph, respectively.

In the next M lines, the i^{th} one contains three integers: u_i, v_i, and w_i ($1 \le u_i, v_i \le N; 0 \le w_i \le 10^9; u_i \ne v_i$), indicating that there is an edge connecting two vertices u_i and v_i of weight w_i in the initial graph.

The next line contains a single integer Q ($1 \le Q \le 10^5$) — the number of queries.

In the next Q lines, each contains three integers x, y and z ($1 \le x, y \le N, 0 \le z \le 10^9, x \ne y$) describing a query in which you need to add an edge of weight z connecting two vertices x and y.

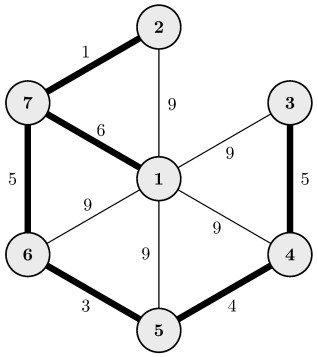
It is guaranteed that at the beginning and after every query, the graph is always connected and no pairs of vertices are connected by more than one edges.

Output

For each query, print a single line containing the number of **awesome** edges in the graph after that query.

Explanation of the sample input

After the first query, the graph is as below:



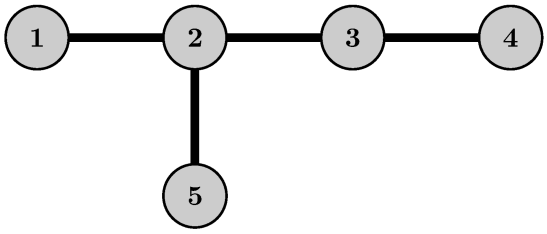
After the second query, the graph is the same as the graph in the problem description.

Sample Input 1	Sample Output 1
7 10 2 7 1 5 6 3 4 5 4 6 7 5 3 4 5 1 7 6 1 2 9 1 3 9 1 4 9 1 5 9 2 1 6 9 2 3 2	6 7

Problem B
Backbone Network

FPT Telecom is planning to build a new Internet backbone network to better serve their customers. The network should consist of n stations, which are numbered from 1 to n. It should also consist of $n-1$ bidirectional links, each connects two distinct stations. These links should guarantee reachability between every pair of stations. In other words, the network should form a tree.

The *realiability* of a network is the expected number of pairs of stations that are still connected if one link is down, assuming that links are down with equal probability.



For example, from the above network:

- If the link 1 — 2 is down, all 6 pairs of stations among $2, 3, 4, 5$ are connected.
- If the link 2 — 3 is down, there are 4 pairs of connected stations: $(3, 4), (1, 2), (1, 5), (2, 5)$.
- If the link 3 — 4 is down, all 6 pairs of stations among $1, 2, 3, 5$ are connected.
- If the link 2 — 5 is down, all 6 pairs of stations among $1, 2, 3, 4$ are connected.

Thus, the reliability of this network is $\frac{6+4+6+6}{4} = 5.5$.

Obviously, we want to build a network with maximum reachability. Given the number of stations n, your task is to build such a network.

Input

The input contains a single integer n ($2 \le n \le 50$).

Output

- The first line contains a single number — the maximum reliability of a network.
- In the next $n-1$ lines, each contains two integers u and v ($1 \le u, v \le n$) meaning that there is a link connecting two stations u and v.

Your answer will be considered correct if its relative or absolute error doesn't exceed 10^{-6}. Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, iff $\frac{|a-b|}{max(1,b)} \le 10^{-6}$.

If there are multiple solutions, you can output any of them.

Sample Input 1	Sample Output 1
3	1.000000 1 2 2 3

Problem C
Coloring Polygon

You are given a convex polygon with n vertices. The vertices are numbered from 1 to n in close-wise direction. A diagonal is a line segment joining two non-consecutive vertices of the polygon. Formally, a diagonal connects two vertices u and v of the polygon where $|u - v| \notin \{1, n - 1\}$.

This problem contains two steps:

1. Firstly, you need to partition this polygon into $n - 2$ triangles using $n - 3$ diagonals. Every valid partition is uniquely defined by a set of $n-3$ diagonals of the polygon where the chosen diagonals only intersect other at their ends. In other words, if two diagonals of a valid partition share some common point, the common point must be some vertex of the polygon. You are given k diagonals $(u_1, v_1), (u_2, v_2), \ldots, (u_k, v_k)$, you need to find $n - 3 - k$ other diagonals to form a valid partition.

2. Secondly, you need to color all $n - 2$ triangles of the polygon using m colors (which are numbered from 1 to m), so that every triangle is colored by exactly one color, and two triangles sharing a side must be colored differently.

Let z be a valid partition which can be obtained in the first step, we define $f(z)$ as the number of ways we can color triangles in the second step. You have to calculate two values:

- $P = \min f(z)$, i.e the minimum possible number of valid colorings in the second step, considering all valid partitions in the first step.
- Q is the number of valid partition z such that $f(z) = P$; in other words, the number of valid partitions in the first step which results in the minimum number of valid colorings in the second step.

Input

The first line of the input contains 3 integers n, k and m ($3 \le n \le 10^5, 0 \le k \le n-3, 1 \le m \le 10^5$).

In the next k lines, the i^{th} one contains two integers u_i and v_i ($1 \le u_i, v_i \le n$) representing a diagonal joining two vertices u_i and v_i of the polygon.

It is guaranteed that these k diagonals only intersect at their ends.

Output

Print two integers in a single line — the values of P and Q respectively. Since these values can be rather large, you should print them modulo $998\,244\,353$.

Sample Input 1	Sample Output 1
4 0 2	2 2

Sample Input 2	Sample Output 2
6 3 3 2 4 4 6 6 2	24 1

Problem D
Distinctive Number

This year, the organizing comittee of ICPC prepares a game for students. The game is very simple, each student can select one positive integer and send his/her selection to the game master. After all selections are received, the winning number is the *smallest unique number*. For example, if the selections are $1, 1, 2, 2, 2, 3, 4, 5, 5, 7$, the winning number is 3. There could be a case where there is no winning number (i.e. all selections are not unique).

The game master receives N selections, one-by-one. Instead of calculating the winning number after having received all selections, he wants to keep track of the winning number as selections are sent in. Your task is to help the game master deal with this.

Input

- The first line of the input contains a single integer N — the number of selections ($1 \leq N \leq 5 \cdot 10^5$).
- The i-th line of the next N lines contains a single integer s_i denoting the i-th selection ($1 \leq s_i \leq 10^9$).

Output

You should print N lines. The i-th one contains the winning number after the i-th selection is sent. In case there is no winning number, print -1 instead.

Sample Input 1	Sample Output 1
6 5 4 4 3 5 3	5 4 5 3 3 -1

Problem E
Even Paths

Last year, Arthur participated in the Vietnamese Robotic Olympiad with a malfunctioning robot. This year, he decided to participate one more time.

The challenge for contestants this year is much harder. The playing field is a rectangular board of size $r \cdot c$, with r rows numbered from 1 to r from top to bottom and c columns numbered from 1 to c from left to right. The cell which lies on ith row and jth column is denoted as (i, j). Each cell contains either a light or a stone, but not both. These lights can be turned on or off. Initially, some of them are on.

The challenge's objective is to switch off all the lights. In order to do so, contestants' robots can execute several runs. In each run, a robot starts at an arbitrary cell on the board, and then performs a sequence of moves. There are several rules for a valid run:

- In each step, a robot has to move from the current position to a side-adjacent cell. In other words, it has to move to the next cell towards any of the four directions: up, down, left or right.
- Robots must always stay inside the board.
- Robots must not visit any cells containing stones.
- In a run, a robot can not visit a cell more than once. However, a robot can visit a cell multiple times during different runs.
- The number of visited cells in a run **must be even**.

From all the above rules, it can be easily seen that a valid run is uniquely defined by a tuple (x, y, s), where (x, y) represents its starting position, and s is a string containing characters **U**, **D**, **L**, **R** representing its sequence of moves. Here **U**, **D**, **L**, **R** stands for up, down, left and right, respectively. The length of the string s **must be odd**.

According to the rules of the Olympiad, while executing a run, whenever a robot passes through a cell, it can choose to turn off the light there. However, as Arthur's robot is buggy again, his robot decides to **always switch the light** (on to off, off to on) when it visits a cell. In other words, all the lights of all cells on its path will be switched.

With this buggy robot, it is even hard just to turn off all the lights. Your task is to help Arthur to accomplish that. He will be extremely thankful if you manage to turn off all the lights. The number of runs does not need minimizing, but the number of moving steps over all runs should not exceed 10^6.

Input

The input starts with an integer t — the number of test cases ($t \leq 100\,000$). Then t test cases follow, each test case is presented as below:

- The first line contains two integers r and c ($1 \leq r, c \leq 10^6$) representing the size of the board.

- The last r lines represent the board. Each contains a string of length c, whose characters are either `.`, `X` or `#`, where:
  - character `.` represents a cell with a light initally turned off,
  - character `X` represents a cell with a light initally turned on,
  - character `#` represents a cell with a stone.

The sum of $r \cdot c$ over all test cases of an input does not exceed 10^6.

Output

For each test case:

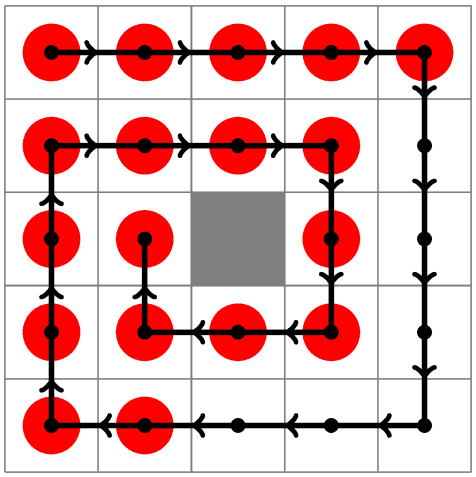
- If it is impossible to turn off all the lights using at most 10^6 moving steps, print -1.
- Otherwise, the first line contains an integer k $(0 \le k \le 10^6)$ representing the number of runs. Then k lines follow, each contains two integers x, y and a string s to describe a run as presented above. The length of all these strings should not exceed 10^6.

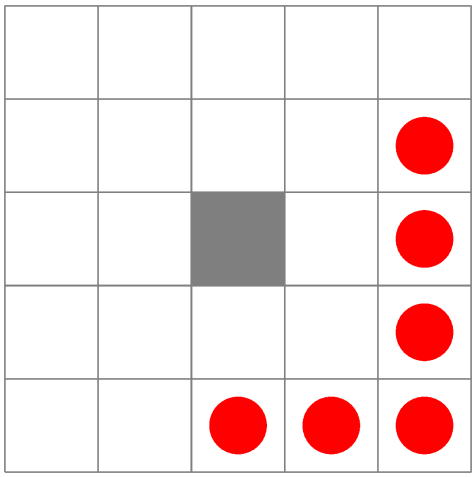
If there are multiple solutions, you can output any of them.

Explanation of the sample

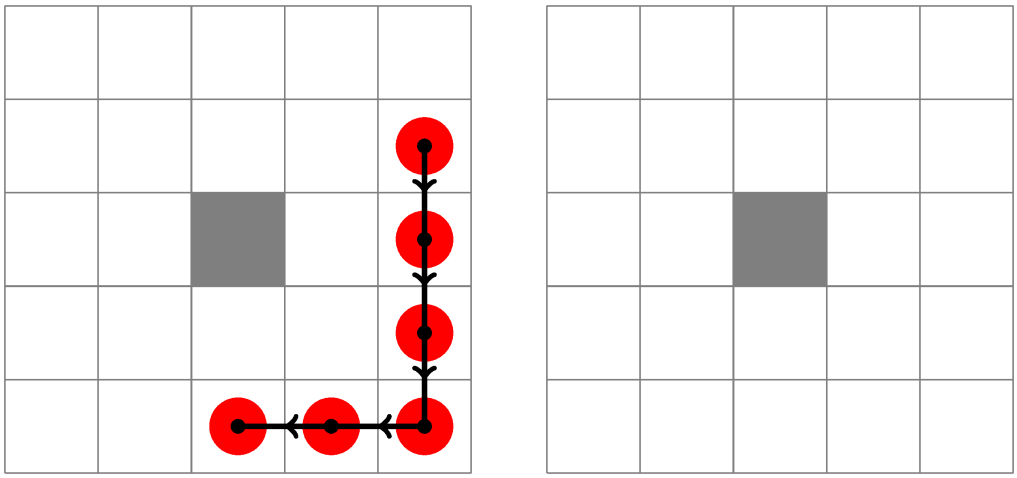
The following figures demomstrate the above sample. In each figure, red circles represent lights which are turned on, blank cells represent cells with lights which are turned off, and grey cells represents cells with stones. Black bold segments and arrows represent runs.

These two figures below demonstrate the first run. The left one represents the state of the board before the run and how robot moves during the run, while the right one represents the state of the board after the run.





These two figures below demonstrate the second run. The left one represents the state of the board before the run and how robot moves during the run, while the right one represents the state of the board after the run.



Sample Input 1	Sample Output 1
1 5 5 XXXXX XXXX. XX#X. XXXX. XX...	2 1 1 RRRRDDDDLLLLUUURRRDDLLU 2 5 DDDLL

Problem F
Final Ranking

After the ICPC Vietnam National contest, the judges proposed a way to give k types of medals to the teams as below:

- The rank of a team equals *the number of teams that have higher score* plus 1
- k types of medals are numbered from 1 to k, medals numbered with smaller indices represent better rankings.
- If the rank of a team is r and $r \leq k$, this team will get the medal of type r.

This is an example of a final ranking from the ICPC national contest where 3 types of the medals (gold, silver, bronze) will be awarded. You can see that there might be types of medals that are not given to any teams and there might be more than k teams which are awarded medals.

In this example, medals distribution per types are 2 gold, 0 silver and 2 bronze or formally $[2, 0, 2]$.

Rank	Team	Score	Medal
1	HCMUS-RationalKittens	12	Gold
1	PENDLE	12	Gold
3	Vaccinated	11	Bronze
3	Meld	11	Bronze
7	HCMUS-FlamingTomatoes	10	
7	HCMUS-Clique	10	
7	NANO	10	
7	HCMIU_ThinkingTourists	10	
9	Secret	9	
9	ETHEREUM	9	

Given n and k, you task is to calculate the number of possible medal distributions.

Input

Input contains 2 integers n and k ($1 \leq k \leq n \leq 10^6$).

Output

Print a single integer denoting the number of possible medal distributions. Since this number could be rather large, you should output it modulo $10^9 + 7$.

Explanation of the samples

With $n = 4$ and $k = 3$, there are 8 possible rankings:

Ranking	Medal distribution
$[1, 1, 1, 1]$	$[4, 0, 0]$
$[1, 1, 1, 4]$	$[3, 0, 0]$
$[1, 1, 3, 3]$	$[2, 0, 2]$
$[1, 1, 3, 4]$	$[2, 0, 1]$
$[1, 2, 2, 2]$	$[1, 3, 0]$
$[1, 2, 2, 4]$	$[1, 2, 0]$
$[1, 2, 3, 3]$	$[1, 1, 2]$
$[1, 2, 3, 4]$	$[1, 1, 1]$

With $n = 4$ and $k = 2$, there are also 8 possible rankings but only 6 medal distributions:

Ranking	Medal distribution
$[1, 1, 1, 1]$	$[4, 0]$
$[1, 1, 1, 4]$	$[3, 0]$
$[1, 1, 3, 3]$	$[2, 0]$
$[1, 1, 3, 4]$	$[2, 0]$
$[1, 2, 2, 2]$	$[1, 3]$
$[1, 2, 2, 4]$	$[1, 2]$
$[1, 2, 3, 3]$	$[1, 1]$
$[1, 2, 3, 4]$	$[1, 1]$

Sample Input 1	Sample Output 1
4 3	8

Sample Input 2	Sample Output 2
4 2	6

Problem G
Group Testing

During the first outbreak of the COVID-19 pandemic, extensive population-wide testing proved to be one of the best strategy to control and prevent the outbreak. However, PCR testing costs both time and money. An effective way to speed up the process and to save our resources is to implement group testing (or pool testing) strategy.

In this method, instead of doing test for individuals, we mix the samples of p individuals and run the test. If the result is negative, we can safely conclude that all p individuals are negative. Otherwise, if the result is positive, individual testing is used to determine who has the virus.

Today, we have a line of n people coming back from foreign countries. They are numbered 1 to n in the order they are lining up to do the testing before immigrate to Vietnam. Person 1 is in the front of the line, an person n is in the back of the line.

Historical data shows that these people have very little chance of getting the virus because they had already tested negative before their flights. While arriving to Vietnam, there should be at most 2 people with the virus. And if there are exactly two people with the virus, they must be in *consecutive* positions.

The challenge is that we only have 10 test kits left. Each test takes hours to finish so we want to do 10 tests in parallel. We need to determine whether there are people with the virus or not; and if there are at least one people with the virus, we need to determine potision of exactly one person with the virus.

More formally, you are about to write a program which reads the number of people n, then provides a strategy to do 10 tests in parallel. After that, your program receives the outcome of all 10 tests and concludes the result as below:

- If no people have the virus, your program should conclude that there is no positive case.
- If exactly one person at position z has the virus, your program should find out this position z.
- If exaclty two people at positions z and $z + 1$ have the virus, your program should find out either position z or $z + 1$. Any of these value will be accepted.

Interaction protocol

- Firstly, your program reads the integer n — the number of people ($1 \le n \le 1000$).
- Secondly, your program prints 10 lines, each line represents a test you would like to do in the following format:
  - Each line starts with an integer p ($0 \le p \le n$) — the number of people involved in this test.
  - Then the line follows by p distinct integers $x_1, x_2, \ldots, x_p$ ($1 \le x_i \le n$) — the indices of the people involved in this test.
- Thirdly, your prgram reads 10 lines representing the outcome of all 10 tests. Each line is either `POSITIVE` or `NEGATIVE`. `NEGATIVE` means that everyone involved in the test

does not have the virus; while `POSITIVE` means that **at least one** involved person has the virus, but we do not know how many people or which exact people have the virus.

Please be aware that you can not read any of these lines from the standard input before having printed exactly 10 **lines in the previous step.**

- Finally, your program concludes the result by print out a single integer:
  - If no people have the virus, print -1.
  - Otherwise, print z ($1 \leq z \leq n$) meaning that the person at position z has the virus.
- After that, your program should terminate with exit code 0.

Sample communication

standard input	standard output
10	
	2 1 2 2 2 3 2 3 4 2 4 5 2 5 6 2 6 7 2 7 8 2 8 9 2 9 10 2 10 1
NEGATIVE NEGATIVE NEGATIVE NEGATIVE NEGATIVE POSITIVE POSITIVE POSITIVE NEGATIVE NEGATIVE	
	7

Note

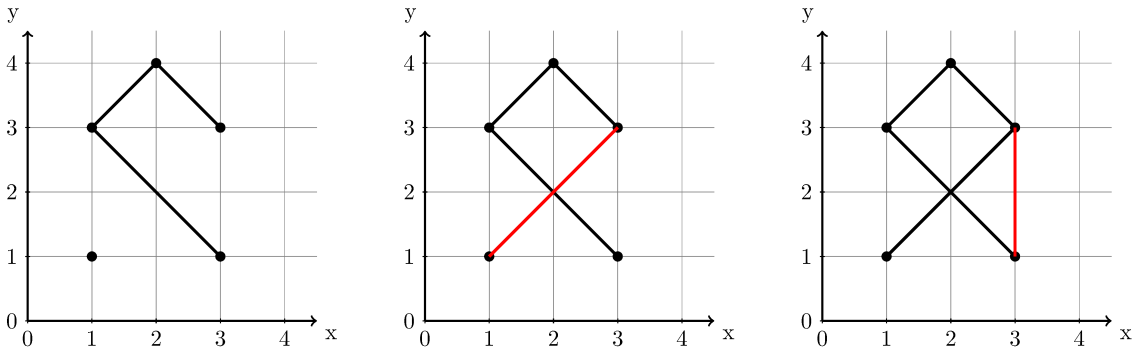
When you write the solution for the interactive problem it is important to keep in mind that if you output some data it is possible that this data is first placed to some internal buffer and may be not directly transferred to the interactor. In order to avoid such situation **you have to use special 'flush' operation each time you output some data**. There are these 'flush' operations in standard libraries of almost all languages. For example, in C++ you may use 'fflush(stdout)' or 'cout « flush' (it depends on what do you use for output data — 'scanf/printf' or 'cout'). In Java you can use method 'flush' for output stream, for example, 'System.out.flush()'. In Python you can use 'stdout.flush()'.

Problem H
Hoang & Vuong

Hoang and Vuong are excellent students. They have won many prizes in coding competitions. Today, they are playing a game of *points and polygons*.

At the begining, there are n given points on the Cartesian plane. These points are numbered from 1 to n. The i^{th} point has coordinates (x_i, y_i). Among them, no 3 points are collinear.

Initially, there are no line segments on the plane. Hoang and Vuong take alternative turns. In each turn, a player selects 2 given points that are not directly connected by a line segment and draw a line segment between them. If a player does not have any valid moves, the other one is the winner of the game. After a player's turn, if he creates a *convex polygon* whose vertices are among the original n points, he is the winner of the game.



In the example above, the first figure shows a state of a game with 5 points $(1, 1)$, $(1, 3)$, $(2, 4)$, $(3, 1)$, $(3, 3)$ after 3 moves.

In the 4^{th} turn, Vuong connects $(1, 1)$ and $(3, 3)$. Note that, the created square (whose vertices are $(1, 3)$, $(2, 2)$, $(3, 3)$, $(2, 4)$) is not counted because $(2, 2)$ is not a given point. Hence, the game still continues after this turn.

In the 5^{th} turn, Hoang connects $(3, 3)$ and $(3, 1)$. This time, a new convex polygon $(1, 3)$, $(3, 1)$, $(3, 3)$, $(2, 4)$ appears. All vertices of this polygon are among the n given points, thus the game ends and Hoang is the winner.

Assuming that both Hoang and Vuong play optimally and Hoang plays first, your task is to identify the winner of the game.

Input

The first line of the input contains t — the number of testcases ($1 \le t \le 100$). Then t testcases follow, each is presented in the following format.

- The first line contains an integer n — the number of points ($2 \le n \le 64$).
- The next n lines, each contains 2 integers x_i and y_i ($0 \le |x_i|, |y_i| \le 10^6$).

Output

For each test case, output in one line the winner of the game (either `Hoang` or `Vuong`).

Sample Input 1	Sample Output 1
1 3 0 2 2 0 2 2	Hoang

Problem I
ICPC Hardest Problem

Given a positive integer N with at most 10^5 digits. Find a positive integer M such that:

- N is a substring of M^2,
- M has at most $10^5 + 10$ digits.

An integer x is a substring of y if x appears in a contiguous subsequence of y.
For example:

- 33 is a substring of 33,
- 34 is a substring of 1345,
- 14 is **not** a substring of 1234.

Input

The input contains a single positive integer N ($1 \le N < 10^{10^5}$).

Output

Output a single positive integer M ($1 \le M < 10^{10^5+10}$) satisfying the given conditions.

Sample Input 1	Sample Output 1
1	1

Problem J
Joining Strings

Given a string f_0 with at most 1000 characters, you need to proceed q queries, each query is one of the following:

- `1 u c`: Create a new string f_i by appending character c to string f_u. In other words, let $f_i = f_u + c$.
- `2 u c`: Create a new string f_i by prepending character c to string f_u. In other words, let $f_i = c + f_u$.
- `3 u v`: Create a new string f_i by concatenating two strings f_u and f_v. In other words, let $f_i = f_u + f_v$.
- `4 u`: Create a new string f_i as a reversion of string f_u. In other words, let $f_i = reverse(f_u)$.
- `5 u v`: Let $f_i = f_{i-1}$. Consider all strings of the form $prefix(f_u) + suffix(f_v)$, count the number of unique strings.

Please note that during the ith query, we always create a new string f_i, while all previously created strings remain unchanged.

Input

The first line contains the non-empty string f_0 with at most 1000 lowercase English characters.

The second line contains a single integer q ($1 \le q \le 10^5$) — the number of queries.

q lines follow, the i-th one describes a query in one of the 5 types presented above. All parameters of the ith query satisfy $0 \le u, v < i$ and c is a lowercase English characters.

Output

For each query of type 5, output a single line containing the answer. Since the number of unique strings can be rather large, you need to output it modulo $10^9 + 7$.

Explanation of the sample input

Below are the strings created after each query:

- $f_0 =$ `ab`
- $f_1 =$ `abz`
- $f_2 =$ `yab`
- $f_3 =$ `abzab`
- $f_4 =$ `bay`

- $f_5 =$ `bay`

In the 5-th query, all prefixes of f_2 are:

- `y`
- `ya`
- `yab`

All suffixes of f_4 are:

- `y`
- `ay`
- `bay`

All the strings of the form $prefix(f_2) + suffix(f_4)$ are:

- `yy`
- `yay`
- `ybay`
- `yay`
- `yaay`
- `yabay`
- `yaby`
- `yabay`
- `yabbay`

Amongst these 9 strings, there are 7 different strings.

Sample Input 1	Sample Output 1
ab 5 1 0 z 2 0 y 3 1 0 4 2 5 2 4	7

Problem K
K Query

Given a sequence of integers $(a_1, a_2, \ldots, a_n)$, for a pair of indices (i, j) such that $1 \leq i \leq j \leq n$, we define $f(i, j)$ as below: considering all pairs of elements a_u and a_v such that $i \leq u \leq v \leq j$, $f(i, j)$ is the sum of absolute difference over all these pairs.

For example, considering the sequence $(1, 1, 2, 3)$, we have:

- $f(1, 1) = |1 - 1| = 0$.
- $f(1, 2) = |1 - 1| + |1 - 1| + |1 - 1| = 0$.
- $f(2, 4) = |1 - 1| + |1 - 2| + |1 - 3| + |2 - 2| + |2 - 3| + |3 - 3| = 4$.

You are given q queries. In each query, you are given 3 integers x, y and k. You need to count the number of pairs of indices (i, j) such that:

- $x \leq i \leq j \leq y$,
- $f(i, j) \leq k$.

Input

The first line of the input contains a single integer n $(1 \leq n \leq 2\,000)$ — the length of the sequence.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(-10^9 \leq a_i \leq 10^9)$.

The third line contains a single integer q $(1 \leq q \leq 2 \cdot 10^5)$ — the number of queries.

In the next q lines, each line contains three integers x, y and k $(1 \leq x \leq y \leq n, 0 \leq k \leq 10^{18})$ describing a query.

Output

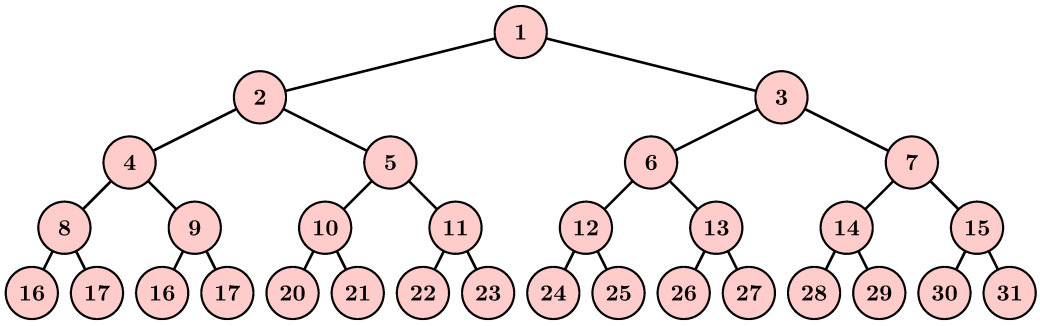
For each query, print its answer in a single line.

Sample Input 1	Sample Output 1
4 1 1 2 3 3 1 1 0 1 2 0 2 4 2	1 3 5

Problem L
Left Or Right

Given a complete binary tree of $2^{30} - 1$ vertices, where vertices are numbered from 1 to $2^{30} - 1$. The root is vertex 1. Every vertex i where $1 \le i \le 2^{29} - 1$ has exactly 2 children, the left one is vertex $2 \cdot i$ and the right one is vertex $2 \cdot i + 1$.

The below figure demonstrates 31 first vertices of this binary tree:



For every non-root vertex n in this tree, the path from root to this vertex always consists of several "moving down" steps, in each step we move from the current vertex to either its left or right child. Therefore, the path can be represented by a string consisting of `L` and `R`, which mean moving to the left child and to the right child, respectively.

Given some vertex n, your task is to find the string representing the path from root to this vertex.

Input

The input starts with a positive integer t ($t \le 1000$) — the number of test cases. Then each test case is printed in a single line with a single integer n ($2 \le n \le 2^{30} - 1$).

Output

For each test case, print a string representing the path from root to vertex n.

Sample Input 1	Sample Output 1
2 3 4	R LL

Problem M
Millionplex

We are quite familiar with some of the names of large numbers like: million (10^6), billion (10^9) or trillion (10^{12}). But there are a lot more names which are unfamiliar, such as: quadrillion (10^{15}), quintillion (10^{18}), sextillion (10^{21}), septillion (10^{24}), octillion (10^{27}), nonillion (10^{30}), decillion (10^{33}), undecillion (10^{36}), duodecillion (10^{39}), tredecillion (10^{42}), quattuordecillion (10^{45}), quindecillion (10^{48}), sexdecillion (10^{51}), septendecillion (10^{54}), octodecillion (10^{57}), novemdecillion (10^{60}), vigintillion (10^{63}), centillion (10^{303}), etc.

Learning more about larger numbers, John Horton Conway and Richard K. Guy have suggested that N-*plex* can be used as a name for 10^N. Thus, millionplex is a number which starts with a digit 1 followed by a million 0s.

Hieu is fascinated by large numbers and researching about them. His work involves understanding positive integers up to a millionplex. Today, Hieu is calculating a function $f(n)$ which equals to the sum of squares of all its "subnumbers". A "subnumber" of a positive integer n is a number formed by a *contiguous* sequence of digits of n. In this problem, we only consider decimal representation of numbers.

For example: $f(2207) = 2207^2+220^2+207^2+22^2+20^2+07^2+2^2+2^2+0^2+7^2 = 4963088$.

Given a positive integer up to a millionplex, your task is to calculate $f(n)$. Since this number could be rather large, you should calculate it modulo $10^9 + 7$.

Input

The input contains a single integer n ($1 \le n \le 10^{10^6}$).

Output

Print a single integer — the value $f(n)$ modulo $10^9 + 7$.

Sample Input 1	Sample Output 1
2207	4963088