Information

Memory limit

The limit is 512 MiB for each problem.

Source code limit

The size of each solution source code can't exceed 256 KiB.

Submissions limit

You can submit at most 50 solutions for each problem.

You can submit a solution to each task at most twice per 30 seconds. This restriction does not apply in the last 15 minutes of the contest round.

Scoring

Each problem consists of several subtasks. The subtask score is awarded if all tests in the subtask are passed.

The number of points scored for the problem is the total number of points scored on each of its subtasks. The score for the subtask is the maximum number of points earned for this subtask among all the solutions submitted.

Feedback

To get feedback for your solution, go to "Runs" tab in PCMS2 Web Client and use "View Feedback" link. In each problem of the contest you will see the score for each subtask, or the verdict for the first failed test.

Scoreboard

The contestants' scoreboard is available during the contest. Use "Monitor" link in PCMS2 Web Client to access the scoreboard. The standings provided in PCMS2 Web Client are not final. **The scoreboard won't be shown during the last 30 minutes of the contest.**

Problem A. Parking Problem

Time limit: 2 seconds

Paulina wants to visit the conference in file fragmentation that takes place in Innopolis University, so she has driven her car all the way to Innopolis. The University parking is a line of width n meters where the vehicles are parked in the way that they are parallel to each other. This parking is built for cars and motorcycles. The parking is divided into n zones each of width 1 meter. The drivers and riders are very respectful to each other, so they use the zoning: motorcycles occupy exactly one zone of 1 meter width, and cars occupy exactly two adjacent zones of total width of 2 meters.

Paulina wants to enter the parking area. Currently some of the zones of the parking are occupied by the vehicles. There are also a queue of m vehicles that try to enter the parking area apart from Paulina's car. All vehicles will enter the parking in the queue order, and will occupy any vacant place: one zone for motorcycle, and two adjacent zones for car. If there is no place for the vehicle, the driver or rider leaves Innopolis, not occupying any zone.

Paulina's conference presentation time is coming, and as the Innopolis people are very polite, everyone offers Paulina to enter the parking before them, so that she won't be late for her presentation. Paulina doesn't want to abuse her position, but she is worried that if she didn't find a place for her car, she wouldn't make it to the presentation. As an expert in data fragmentation Paulina understands that people can choose places for their vehicles poorly leaving no place for her car. She could easily understand, when she is supposed to enter the parking area to ensure a place for her car.

You are given which of the n parking zones are occupied. You are also given the queue of m motorcycles and cars apart from Paulina's car. You are required to compute for each i from 0 to m, will Paulina be able to park her car, if she enters the parking area after the first i vehicles in the queue, no matter what zones they occupy.

Input

You are required to solve several tests.

The first line contains single integer t — the number of tests in the input ($1 \le t \le 50\,000$).

Then the description of t tests follows. Each of the tests consists of two lines.

The first of these two lines consists of n characters '`.`' and '`X`', the i-th of which denotes the i-th zone: '`.`' for vacant, and '`X`' for occupied ($1 \le n \le 10^5$).

The second of these two lines consists of m uppercase English letters '`C`' and '`M`', denoting cars '`C`', and motorcycles '`M`' ($1 \le m \le n$). The first letter corresponds to the head of the queue, and the last — to the tail.

The total sum of n over all t tests doesn't exceed $5 \cdot 10^5$.

Output

Print t lines: one line for each test.

The line should contain $m + 1$ uppercase English letters '`Y`' and '`N`', for each i from 0 to m print '`Y`', if Paulina will be able to find a place for her car, entering after first i vehicles from the queue, no matter which zones they occupy, and '`N`' otherwise.

Scoring

Subtask	Score	Constraints
1	31	parking has no occupied zones
2	22	at most one zone is occupied on the parking
3	23	the queue consists of motorcycles only
4	24	no additional constraints

Examples

standard input	standard output
1 X..X MM	YNN
5 ..... MMMC ..... CCCM .X..X. MMM XXXXXX CMMCM ..... MM	YYYNN YYNNN YNNN NNNNNN YYY

Note

In the first sample Paulina can park her car in two vacant zones, if she enters the parking area before everyone in the queue. Otherwise, one or no vacant zone left, so she won't be able to park her car.

In the first test of the second sample, after the first three motorcycles enter the parking are, they could park in the following way: "M.M.M.", then Paulina won't be able to find two adjacent zones to park her car.

Problem B. Diamond Hands

Time limit: 2 seconds

The "Diamond Hands" corporation has a long and eventful history. Starting from its foundation, they had a lot of successful and unsuccessful days. For simplicity, let's consider a day successful if its stock price increased by one (in abstract units). Similarly, we consider a day unsuccessful if its stock price decreased by one. As it often happens, successful days come in long streaks, as do unsuccessful days. There is no middle ground: every day is either successful or unsuccessful.

You would like to figure out which days were successful and unsuccessful for the corporation. To accomplish it, you obtained the historical stock price data: n pairs (d_i, p_i), meaning that after d_i days after issuing stock the difference with the starting stock price was p_i units (p_i can be an arbitrary integer, including any negative number).

Represent the corporation's history with the **minimum** number of successful or unsuccessful day streaks, or report that data contains an error and it's impossible to achieve it. If there are multiple possible answer with the minimum number of streaks, output any one of them.

Input

The first line of the input contains an integer n ($1 \le n \le 200\,000$). Next n lines contain a pair of integers d_i p_i each ($1 \le d_i \le 10^8$; $-10^8 \le p_i \le 10^8$; $d_i < d_{i+1}$ for all i from 1 to $n-1$).

Output

If there are errors in the historical stock price data, output -1. Otherwise, print k, the number of streaks, in the first line. Next, print k lines describing successful or unsuccessful streaks. Each line should contain a pair l_i c_i ($1 \le l_i \le 10^8$; $c_i \in \{$`+`, `-`$\}$), meaning that the next streak lasted for l_i days, and was successful if $c_i =$ `+`, or unsuccessful if $c_i =$ `-`.

The description of streaks must be done in chronological order, starting from the day stock was issued and ending on day d_n. This means that the sum of all l_i must be equal to d_n.

Scoring

Subtask	Points	Constraints
1	6	$n \le 2000$, if an answer exists, then $k = 1$
2	8	$n \le 2000$, if an answer exists, then $k \le 2$
3	10	$n \le 200\,000$, if an answer exists, then $k \le 2$
4	28	$n \le 2000$, $d_i, p_i \le 2000$
5	17	$n \le 2000$, $d_i, p_i \le 10^8$
6	31	$n \le 200\,000$, $d_i, p_i \le 10^8$

Examples

standard input	standard output
4 2 2 3 3 5 1 7 1	3 3 + 3 - 1 +
2 3 -3 7 -3	2 5 - 2 +
1 1 0	-1

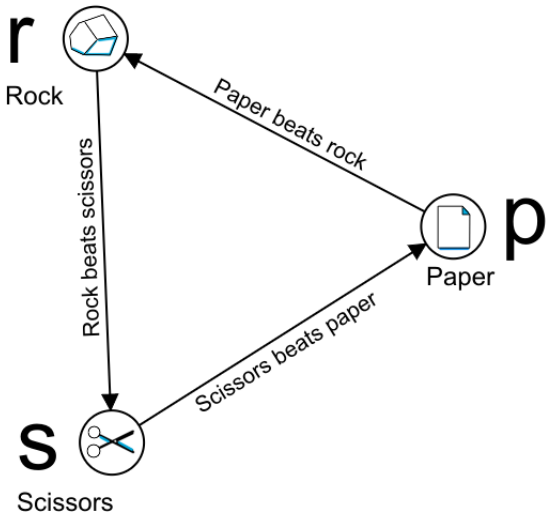
Note

In the first example, first three days are successful, so after 2 days the difference is 2, and after 3 days the difference is 3. Next three days are unsuccessful, so after 5 days the difference becomes 1, and after 6 days the stock price is back to the initial value. The last seventh day is successful, so the final difference after 7 days is 1.

Problem C. RPS string

Time limit: 2 seconds

The robots battle in rock-paper-scissors takes place in Innopolis. All advanced robots are engaged in full-fledged work, so the simplest bots participate in the battles. Each of them always play the same shape: a rock, or scissors, or a paper.



The robots stand in a line. Then the judge can choose two neighboring robots, and make them play one round of rock-paper-scissors. If one of them beats the other, the loser is eliminated from the game and is removed from the sequence of robots, the sequence of the remaining ones doesn't change. There are two versions of the rules. In the first of them, if the robots played the same shape, the judge can independently choose which one to remove from the game. In the second version, if the robots played the same shape, both robots remain. The robot wins if all other robots are eliminated from the game.

However, the judge has become very bored with the competition, so he wants to find out for each robot whether the judge can choose pairs in each round so that this robot wins. Help him!

Input

You will need to solve several testcases.

The first line contains an integer t ($t \geq 1$) — the number of tests in the input.

Next, t tests are given, each consisting of an integer and a string.

The integer d denotes the version of the rules that is used in the current robot layout $d \in \{1, 2\}$.

Then the string s follows, it contains n ($1 \leq n \leq 5 \cdot 10^5$) lowercase English letters '`r`', '`p`' and '`s`', describing the list of robots. More precisely, the characters '`r`', '`p`' and '`s`' denote robots that play rock, paper, and scissors, respectively.

The total sum of n for t given tests does not exceed $5 \cdot 10^5$.

Output

Output t lines: one for each test.

The line should contain n digits '`0`' and '`1`', for each i from 1 to n print '`1`' if the judge can choose such pairs for each round that the i-th robot wins, and '`0`' otherwise.

Scoring

Subtask	Score	Constraints
1	7	$d = 1, \sum n \leq 20$
2	9	$d = 1, \sum n \leq 200$
3	10	$d = 1, \sum n \leq 5000$
4	11	$d = 1, \sum n \leq 5 \cdot 10^5$
5	7	$\sum n \leq 20$
6	13	$\sum n \leq 200$
7	10	$\sum n \leq 5000$
8	33	$\sum n \leq 5 \cdot 10^5$

Example

standard input	standard output
4 1 rpspp 2 pps 1 rpsps 2 rpssprsrr	10111 001 10101 110010001

Note

First test example: "rpspp", $d = 1$. Let's identify the robots with integers from 1 to 5 from left to right.

1. In order for the robot 1 to win, the judge can act as follows:
   - The judge chooses robots 4 and 5, they play the same shape, so the judge decides that the 5th robot is eliminated. Robots {1, 2, 3, 4} remain.
   - The judge chooses robots 2 and 3, the 2nd one is eliminated. Robots {1, 3, 4} remain.
   - The judge chooses robots 3 and 4, the 4th one is eliminated. Robots {1, 3} remain.
   - The judge chooses robots 1 and 3, the 3rd one is eliminated. The robot 1 wins.
2. For the robot 2, the judge can't get rid of the 1st robot without removing the 2nd, so the 2nd can't win.
3. In order for the robot 3 to win, the judge can act as follows:
   - The judge chooses robots 4 and 5, they play the same shape, so the judge decides that the 5th robot is eliminated. Robots {1, 2, 3, 4} remain.
   - The judge chooses robots 1 and 2, the 1st one is eliminated. Robots {2, 3, 4} remain.
   - The judge chooses robots 2 and 3, the 2nd one is eliminated. Robots {3, 4} remain.
   - The judge chooses robots 3 and 4, the 4th one is eliminated. The robot 3 wins.
4. In order for the robot 4 to win, the judge can act as follows:
   - The judge chooses robots 4 and 5, they show the same shape, so the judge decides that the 5th robot is eliminated. Robots {1, 2, 3, 4} remain.
   - The judge chooses robots 2 and 3, the 2nd one is eliminated. Robots {1, 3, 4} remain.
   - The judge chooses robots 1 and 3, the 3rd one is eliminated. Robots {1, 4} remain.
   - The judge chooses robots 1 and 4, the 1st one is eliminated. The robot 4 wins.
5. In order for the robot 5 to win, the judge can act as follows:

- The judge chooses robots 4 and 5, they show the same shape, so the judge decides that the 4th robot is eliminated. Robots {1, 2, 3, 5} remain.
- The judge chooses the robots 2 and 3, the 2nd one is eliminated. Robots {1, 3, 5} remain.
- The judge chooses robots 1 and 3, the 3rd one is eliminated. Robots {1, 5} remain.
- The judge selects robots 1 and 5, the 1st one is eliminated. The robot 5 wins.

Second test example: "pps", $d = 2$. Let's identify the robots with integers from 1 to 3 from left to right.

1. For the robot 1 the judge can't get rid of the 2nd robot without removing the 1st, so the 1st can't win.
2. For the robot 2, the judge can't get rid of the 1st robot without removing the 2nd, so the 2nd can't win.
3. In order for the robot 3 to win, the judge can act as follows:
   - The judge selects robots 2 and 3, the 2nd one is eliminated. Robots {1, 3} remain.
   - The judge selects robots 1 and 3, the 1st one is eliminated. The robot 3 wins.

Problem D. Long puzzle

Time limit: 2 seconds

You have a one-dimensional puzzle. Every piece of the puzzle can be described by three values: length, type of the left border, and type of the right border. Borders can be one of three types: straight, convex, and concave. Pieces couldn't be reversed, i.e. you can't swap left and right borders of a piece. Any convex border can be connected with any concave border and vice versa. You can't connect pieces by two straight borders.

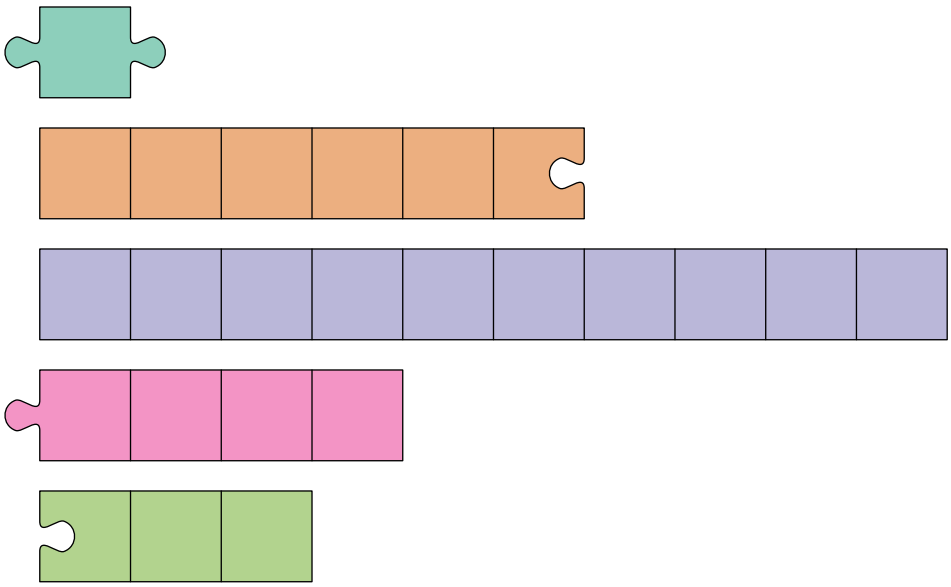


Figure 1: Example of pieces

You want to connect several (possibly one) pieces one after another in order to get a part of length l. The left and the right borders of the part should be straight. Find a number of sets of pieces, such that you can build desired part using all pieces from the set. The number could be large, so calculate it modulo 1 000 000 007. Note that you should find the number of sets of pieces, not the number of different ways of connecting them.

Input

The first line contains two integer numbers n and l — the number of pieces and desired length of a part ($1 \le n \le 300$, $1 \le l \le 300$).

The following n lines contain a description of the pieces. Every line contains a_i, b_i and c_i — the length of the piece, type of its left border, and type of its right border, accordingly ($1 \le a_i \le l$; $b_i, c_i \in \{$"`in`","`out`","`none`"$\}$). String "`in`" denotes concave border, "`out`" — convex, "`none`" — straight.

Output

Output single integer — the number of sets of pieces, such that you can build desired part using these pieces, modulo 1 000 000 007.

Scoring

Subtask	Score	Constraints
1	20	$n \leq 20$
2	20	$b_i \in \{$“in”,“none”$\}$, $c_i \in \{$“out”,“none”$\}$
3	20	$n, l \leq 50$
4	20	$n, l \leq 100$
5	20	No additional constraints

Examples

standard input	standard output
5 10 1 out out 6 none in 10 none none 4 out none 3 in none	3
4 5 1 none out 1 in out 2 in out 1 in none	1

Note

Pieces of the puzzle from the first example correspond to the previous picture.

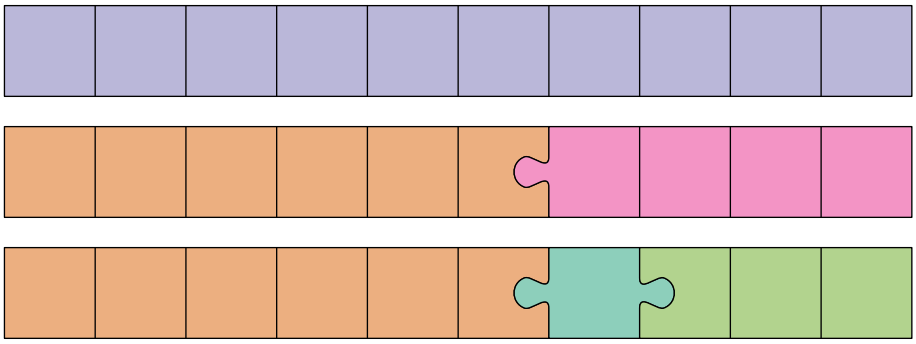


Figure 2: Sets of pieces, such that you can build desired part using them

Problem E. Rooted MST

Time limit: 3 seconds

You are given a simple undirected weighted graph with $n + 1$ vertices numbered $0, 1, \ldots, n$ and $n + m$ edges.

The weight of an edge between vertices 0 and i is a_i for $1 \le i \le n$.

The weight of an edge between vertices u_i and v_i is w_i for $1 \le i \le m$.

You need to answer q queries, in each query, you are given two integers i, w and you need to change the weight of an edge from 0 to i to w and find the weight of the minimum spanning tree in the graph.

Note that changes to the weights are permanent, i.e. they stay after each query.

Input

The first line of input contains two numbers n, m ($2 \le n \le 300\,000, 0 \le m \le 300\,000$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Each of the next m lines contains three integers u_i, v_i, w_i ($1 \le u_i, v_i \le n, 0 \le w_i \le 10^9$).

It is guaranteed that the given graph is simple, in other words, it contains no loops and multiple edges.

The next line contains one integer q ($1 \le q \le 300\,000$).

Each of the next q lines contains two integers i, w ($1 \le i \le n, 1 \le w \le 10^9$).

Output

For each query print one integer: the weight of the minimum spanning tree in the graph after the first i queries.

Scoring

Subtask	Score	Constraints
1	10	$n, m, q \le 2000$
2	10	All weights are either 1 or 2
3	10	$w = 1$ in all queries
4	10	$i = 1$ in all queries
5	10	$i \le 5$ in all queries
6	10	$m = n - 1, u_i = v_i - 1$
7	20	$n, m, q \le 150\,000$
8	20	No additional constraints

Example

standard input	standard output
5 7 3 2 1 2 1 1 5 1 1 3 2 2 5 2 4 5 2 3 4 1 2 4 2 1 2 1 10 3 2 2 3 4 1 3 2 5 1 5 3 3 1 2 3 4 3 5 1	6 6 5 5 5 6 6 6 6 5