Problem A. Submissions

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

The legend, example, and note of this problem are used fictitiously. Any resemblance to the actual contests, rules, submissions, or teams is coincidental.

In the International Challenging Puzzle Contest (ICPC), there are m submissions. You are given the list of m submissions ordered by time. A submission can be represented as a tuple (c, p, t, s), which means team c makes a submission on problem p at time t with status s. The status of a submission is either "accepted" or "rejected".

The score of a team is the pair of the number of problems solved by the team and the total time consumed† by the team. The larger the number of problems solved is, the higher the score is. If a tie occurs, the smaller the total time consumed is, the higher the score is.

If team c makes at least one submission with status "accepted" on problem p, we say that team c solves problem p. A team can get a gold medal if the number of teams with higher score is less than $\min(\lceil 0.1 \cdot n \rceil, 35)$, where n is the number of teams that solved **at least** one problem and $\lceil x \rceil$ denotes the smallest integer that is not smaller than x.

You need to find all the teams that can get a gold medal if **at most** one of the m submissions changes its status.

† The total time consumed is the sum of times consumed for all solved problems (0 if no problems are solved). The time consumed for a solved problem is the time of the first submission with status "accepted", plus 20 times the number of submissions on this problem before the first submission with status "accepted". Note that we say submission i is before submission j if and only if submission i appears earlier than submission j in the given list of m submissions.

Input

Each test contains multiple test cases. The first line contains a single integer t ($1 \le t \le 10^5$) denoting the number of test cases. For each test case:

The first line contains a single integer m ($1 \le m \le 10^5$) denoting the number of submissions.

The i-th of the following m lines contains c_i, p_i, t_i, and s_i which mean that team c_i makes a submission on problem p_i at time t_i with status s_i. Specifically:

- c_i is a string of length between 1 and 20 consisting of uppercase letters, lowercase letters, digits and underscores ('`_`'). Note that no two teams have the same name.
- p_i is an uppercase letter.
- t_i is a non-negative integer less than 300.
- s_i is a string, being either "`accepted`" or "`rejected`".

It is guaranteed that $t_i \le t_j$ for all $i < j$. Recall that if $t_i = t_j$ and $i < j$, we still say that the i-th submission came before the j-th submission.

It is guaranteed that the sum of m over all test cases does not exceed 10^5.

Output

For each test case:

Output one integer k on the first line, denoting the number of teams that can get a gold medal if at most one of the m submissions changes its status.

On the second line, output k distinct strings **in any order**, denoting the names of these k teams.

Examples

standard input	standard output
2 5 TSxingxing10 G 0 rejected TSxingxing10 B 83 accepted aoliaoligeiliao J 98 accepted TS1 J 118 accepted TS1 B 263 accepted 12 AllWayTheNorth A 0 rejected YaoYaoLingXian Y 10 accepted XuejunXinyoudui1 X 200 rejected XuejunXinyoudui1 X 200 accepted LetItRot L 215 accepted AllWayTheNorth W 250 accepted ImYourFan I 257 accepted ImYourFan Y 257 accepted AllWayTheNorth T 264 accepted XuejunXinyoudui1 J 294 accepted LetItRot I 299 accepted LetItRot I 299 rejected	2 TSxingxing10 TS1 4 AllWayTheNorth XuejunXinyoudui1 LetItRot ImYourFan

standard input	standard output
2 2 jiangly_fan A 1 accepted jiangly B 23 accepted 3 conqueror_of_tourist A 1 accepted conqueror_of_tourist A 2 accepted tourist B 23 accepted	2 jiangly_fan jiangly 1 conqueror_of_tourist

standard input	standard output
2 13 A A 1 accepted A X 1 accepted K K 1 rejected B B 2 accepted C C 2 accepted D D 2 accepted E E 2 accepted F F 2 accepted G G 2 accepted H H 2 accepted I I 2 accepted J J 2 accepted K K 2 rejected 12 A A 1 accepted A X 1 accepted B B 2 accepted C C 2 accepted D D 2 accepted E E 2 accepted F F 2 accepted G G 2 accepted H H 2 accepted I I 2 accepted J J 2 rejected K K 2 rejected	11 A K B C D E F G H I J 1 A

standard input	*standard output*
2 11 A A 1 accepted B B 1 accepted C C 2 accepted D D 2 accepted E E 2 accepted F F 2 accepted G G 2 accepted H H 2 accepted I I 2 accepted J J 2 accepted K K 2 accepted 12 A A 1 accepted A X 1 accepted K K 1 rejected B B 2 accepted C C 2 accepted D D 2 accepted E E 2 accepted F F 2 accepted G G 2 accepted H H 2 accepted I I 2 accepted J J 2 accepted	2 A B 2 A K

Note

In the first case of the first example, `TS1` solves two problems, so they can get a gold medal. `TSxingxing10` can get a gold medal if their first submission changes its status to “accepted”.

In the second case of the first example, `AllWayTheNorth`, `XuejunXinyoudui1`, `LetItRot` and `ImYourFan` have the same score, two problems solved with 514 total time consumed. They can get gold medals simultaneously if no submission changes its status.

Problem B. Festival Decorating

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	1024 mebibytes

To celebrate the coming winter festival in Byteland, the main street, which can be regarded as the x-axis, is decorated with n colorful lamps, labeled by $1, 2, \ldots, n$. The x-coordinate of the i-th lamp is x_i, and the color of the i-th lamp is c_i. No two lamps share the same x-coordinate.

You will be given q queries. In the i-th query, you will be given an integer d_i ($1 \le d_i \le 250\,000$), and you need to find the lamp u ($1 \le u \le n$) with the minimum index such that there is another lamp located at $x_u + d_i$ and the color of that lamp is different from c_u, or determine it is impossible to find such u. **Your answer is considered correct if its absolute or relative error does not exceed 0.5.**

Input

The first line of the input contains two integers n and q ($1 \le n, q \le 250\,000$) denoting the number of lamps and the number of queries.

Each of the next n lines contains two integers x_i and c_i ($1 \le x_i \le 250\,000$, $1 \le c_i \le n$) denoting the x-coordinate and the color of the i-th lamp. It is guaranteed that no two lamps share the same x-coordinate.

Each of the next q lines contains a single integer d_i ($1 \le d_i \le 250\,000$) denoting the i-th query.

Output

For each query, print a line containing a single number: the minimum index u you found. If it is impossible to find such u, print 0 instead.

Your answer is considered correct if its absolute or relative error does not exceed 0.5. Note that this means you can output a non-integer as well.

Formally, let your answer be u, and the jury's answer be u'. Your answer is accepted if and only if:

$$\frac{|u - u'|}{\max(1, |u'|)} \le 0.5.$$

Example

standard input	*standard output*
4 5 3 1 1 2 5 1 6 2 1 2 3 4 5	3 2 1 2 0

Problem C. Yet Another Shortest Path Query

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

In graph theory, a planar graph is a graph that can be drawn on the plane in such a way that its edges intersect only at their endpoints. In other words, it can be drawn so that no edges cross each other. Such a drawing is called a plane graph or a planar embedding of the graph. A plane graph can be defined as a planar graph with a mapping from every node to a point on the plane, and from every edge to a plane curve on that plane, such that the extreme points of each curve are the points mapped from its end nodes, and all curves are disjoint except for their extreme points.

You will be given an undirected planar graph with n vertices and m edges. The vertices are labeled by $1, 2, \dots, n$. The i-th edge connects vertices u_i and v_i, and its length is w_i.

You will then be given q queries. In the i-th query, you will be given two integers s_i and t_i. Please write a program to figure out the length of the shortest path from vertex s_i to vertex t_i such that the path contains **no more than three edges**, or determine that there is no such path.

Input

The first line of the input contains two integers n and m ($2 \leq n \leq 10^6$, $1 \leq m \leq 10^6$) denoting the number of vertices and the number of edges.

In the next m lines, the i-th line contains three integers u_i, v_i and w_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$, $1 \leq w_i \leq 10^8$) describing the i-th edge. It is guaranteed that there is at most one edge between each pair of vertices. It is also guaranteed that the graph is planar.

The next line contains a single integer q ($1 \leq q \leq 10^6$) denoting the number of queries.

In the next q lines, the i-th line contains two integers s_i and t_i ($1 \leq s_i, t_i \leq n$, $s_i \neq t_i$) describing the i-th query.

Output

For each query, print a single line containing an integer denoting the length of the shortest path containing at most three edges. If there is no such path, please print "`-1`" instead.

Examples

standard input	*standard output*
6 9 1 2 4 2 3 6 3 6 5 6 5 3 5 4 2 4 1 3 3 4 9 1 3 100 5 3 1 5 1 3 1 6 3 4 3 5 2 5	6 8 3 1 7
6 4 1 2 1 2 3 1 3 4 1 4 5 1 3 1 4 1 5 1 6	3 -1 -1

Problem D. Operator Precedence

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

Randias is facing his primary school homework:

Find a nonzero integer sequence a of length $2n$ satisfying

$$\begin{aligned}&(a_1 \times a_2)+(a_3 \times a_4)+ \quad \dots \quad +(a_{2n-1} \times a_{2n})\\ &= a_1 \times(a_2 + a_3)\times(a_4 + a_5)\times \dots \times(a_{2n-2} + a_{2n-1})\times a_{2n} \neq 0.\end{aligned}$$

In shorter form, $\sum_{i=1}^{n} a_{2i-1}a_{2i} = a_1 a_{2n} \prod_{i=2}^{n} (a_{2i-2} + a_{2i-1}) \neq 0$.

Of course, Randias knows how to solve it. But he wants to give you a test. Can you solve the question above?

Input

Each test contains multiple test cases. The first line contains a single integer t ($1 \le t \le 10^5$) denoting the number of test cases.

For each test case, the only line contains a single integer n ($2 \le n \le 10^5$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output one line with $2n$ integers: $a_1, a_2, \dots, a_{2n}$ ($1 \le |a_i| \le 10^{10}$).

It can be shown that the answer always exists.

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

Example

standard input	*standard output*
3 2 3 4	1 -3 -3 1 1 -10 6 6 -10 1 1 -15 10 -1 -1 10 -15 1

Problem E. Period of a String

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

Randias has n strings $s_1, s_2, \ldots, s_n$.

For two strings $a = \overline{a_0 a_1 \ldots a_{p-1}}$ and $b = \overline{b_0 b_1 \ldots b_{q-1}}$, if for all i $(0 \le i < q)$, $b_i = a_{i \bmod p}$, we say that a is a period of b.

Now, Randias can perform the following operation:

- Choose one string s_i and choose two indices j and k $(0 \le j, k < |s_i|)$, then swap $s_{i,j}$ and $s_{i,k}$.

He can perform this operation **any** number of times. After all the operations, he wants the following to be true: for each $1 < i \le n$, string s_{i-1} is a period of s_i.

Help him to find the possible final strings, or determine it is impossible.

Input

Each test contains multiple test cases. The first line contains a single integer t $(1 \le t \le 10^4)$ denoting the number of test cases. For each test case:

The first line contains a single integer n $(2 \le n \le 10^5)$.

Then follow n lines. The i-th of these lines contains the string s_i $(1 \le |s_i| \le 5 \cdot 10^6)$. It is guaranteed that the strings only contain lowercase English letters.

It is guaranteed that the sum of n does not exceed 10^5, and the sum of $|s_i|$ does not exceed $5 \cdot 10^6$.

Output

For each test case, if it is possible to make s_{i-1} a period of s_i for all i after some operations, output "YES" (without quotes) on the first line. Then output n strings in n lines. The i-th string s'_i represents the i-th string after all operations. If there are multiple answers, output any one of them.

If it is impossible to do that, output "NO" (without quotes) on the first line.

Example

standard input	*standard output*
4 2 abc abcd 4 bbcaa cabb acabbbb a 3 ab aab bbaaaaab 3 ab aab bbaaaaa	NO YES abbca abbc abbcabb a YES ab aba abaabaab NO

Problem F. Top Cluster

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

Top Cluster is a useful data structure for maintaining data on a tree. Using Top Cluster, we can do range queries efficiently.

Lovely EMmm likes data structure technologies very much. She is learning Top Cluster now, and she is trying to solve a data structure problem. Can you write a program to solve the problem together with EMmm?

In the problem, you will be given a tree with n vertices, labeled by $1, 2, \ldots, n$. The value of the i-th vertex is a non-negative integer w_i. **All the values are pairwise distinct.**

You will then be given q queries. In the i-th query, you will be given two integers x_i and k_i ($1 \le x_i \le n$, $0 \le k_i \le 10^{15}$), and you need to find the value of $\text{mex}\left(\{w_u \mid \text{dist}(u, x_i) \le k_i \wedge 1 \le u \le n\}\right)$.

Here, $\text{dist}(u, v)$ denotes the length of the shortest path from vertex u to vertex v. In mathematics, the **mex** ("**m**inimum **ex**cluded value") of a set is the smallest non-negative integer that does not belong to the set.

EMmm is good at solving mex problems. She found that when all the values are pairwise distinct, the problem above is equivalent to finding the smallest non-negative integer that either occurred outside the given range, which means $\text{dist}(x_i, u) > k_i$, or never occurred in the whole tree. However, she can't go any further. Can you help her solve the problem?

Input

The first line of the input contains two integers n and q ($1 \le n, q \le 5 \cdot 10^5$) denoting the number of vertices and the number of queries.

The second line contains n integers $w_1, w_2, \ldots, w_n$ ($0 \le w_i \le 10^9$) denoting the values of the vertices. **It is guaranteed that all the values are pairwise distinct.**

Each of the next $n - 1$ lines contains three integers u_i, v_i and ℓ_i ($1 \le u_i, v_i \le n$, $u_i \ne v_i$, $1 \le \ell_i \le 10^9$) denoting a two-way edge between vertices u_i and v_i with length ℓ_i. It is guaranteed that the input forms a tree.

Each of the next q lines contains two integers x_i and k_i ($1 \le x_i \le n$, $0 \le k_i \le 10^{15}$) denoting the i-th query.

Output

For each query, print a single line containing an integer: the mex value that you found.

Example

standard input	*standard output*
5 4 3 9 0 1 2 1 2 10 3 1 4 3 4 3 3 5 2 3 0 1 0 4 6 4 7	1 0 3 4

Problem G. Snake Move

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

Putata is playing a famous snake game on his laptop, where a snake moves around on a grid of size $n \times m$. There may be obstacles in some cells of the grid. The snake can be represented as a sequence of coordinate pairs that determine where its body is located: $(x_1, y_1), (x_2, y_2), \ldots, (x_k, y_k)$. Here, k denotes the length of the snake. The head of the snake is at (x_1, y_1), the tail is at (x_k, y_k), and neighboring parts of the body are located in cells that share a side.

In this game, the snake is programmed with a series of commands represented as a string. There are 5 types of commands that you can use:

- '`L`': Command the snake to move one step left. The head will then move to $(x_1, y_1 - 1)$.
- '`R`': Command the snake to move one step right. The head will then move to $(x_1, y_1 + 1)$.
- '`U`': Command the snake to move one step up. The head will then move to $(x_1 - 1, y_1)$.
- '`D`': Command the snake to move one step down. The head will then move to $(x_1 + 1, y_1)$.
- '`S`': Shorten the length of the snake by one. The tail of the body will be erased. The length will become $k - 1$. Note that you can not do this when $k = 1$.

When the head moves, each part of the body also moves accordingly. Specifically, the i-th part of the body ($2 \le i \le k$) moves to the position where the $(i-1)$-st part was before the command. The snake can not move into a cell with an obstacle, and can not move outside the grid. Besides, the snake cannot collide with itself. So you should guarantee that no two parts of the body will share the same location.

Consider the following corner case: The head is at (x_1, y_1), and the tail is at (x_k, y_k). If the head is moving to (x'_1, y'_1), then it is **allowed** to move to $(x'_1, y'_1) = (x_k, y_k)$: if we think about a real-world scenario, the head moves into the cell just as the tail moves out. In a similar fashion, it is **allowed** to swap the head and the tail by using a single command when $k = 2$.

You will be given the map of the grid and the body sequence of the snake. Let $f(i, j)$ denote the minimum number of commands that Putata needs to use such that the head of the snake will arrive at (i, j), or 0 if it is impossible. You have to calculate:

$$\left(\sum_{i=1}^{n} \sum_{j=1}^{m} f(i,j)^2 \right) \bmod 2^{64}.$$

Input

The first line of the input contains three integers n, m and k ($1 \le n, m \le 3000$, $1 \le k \le \min\{nm, 10^5\}$) denoting the size of the grid and the length of the snake.

In the next k lines, the i-th line contains two integers x_i and y_i ($1 \le x_i \le n$, $1 \le y_i \le m$, $|x_i - x_{i+1}| + |y_i - y_{i+1}| = 1$) denoting the location of the i-th part of the body. It is guaranteed that all the k pairs (x_i, y_i) are pairwise distinct. It is also guaranteed that each part is in a cell without an obstacle.

In the next n lines, the i-th line contains a string of length m. If cell (i, j) is empty, the j-th character in the i-th of these lines is '`.`'. If cell (i, j) is occupied by an obstacle, the character is '`#`'.

Output

Print a single line containing an integer: the answer to the problem.

Examples

standard input	*standard output*
4 5 5 3 5 3 4 3 3 3 2 4 2	293
2 2 4 1 1 1 2 2 2 2 1	14
5 5 3 1 2 1 1 2 1 ..... .###. .#.#. .###.	407

Problem H. Sugar Sweet II

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

Sugar is sweet.

There are n children asking for sugar. Prof. Chen gives out sugar to the children. The i-th child initially has a_i bags of sugar. There are n events happening in **uniformly randomized order**. The i-th event is:

- If the i-th child has **strictly less** bags of sugar than the b_i-th child, then the i-th child will get extra w_i bags of sugar. Otherwise, nothing happens.

Now, since the events happen in random order, Randias, which is the assistant of Prof. Chen, wants to know the expected number of bags of sugar each child will have after all the events happen.

It can be shown that the answer can be expressed as an irreducible fraction $\frac{x}{y}$ where x and y are integers and $y \not\equiv 0 \pmod{10^9+7}$. Output the integer equal to $x \cdot y^{-1} \pmod{10^9+7}$. In other words, output such an integer a that $0 \le a < 10^9+7$ and $a \cdot y \equiv x \pmod{10^9+7}$.

Input

Each test contains multiple test cases. The first line contains a single interger t ($1 \le t \le 5 \cdot 10^5$) denoting the number of test cases. For each test case:

The first line contains a single integer n ($1 \le n \le 5 \cdot 10^5$) denoting the number of children.

The second line contains n integers a_i ($1 \le a_i \le 10^9$): the initial number of bags of sugar each child has.

The third line contains n integers b_i ($1 \le b_i \le n$).

The fourth line contains n integers w_i ($1 \le w_i \le 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed $5 \cdot 10^5$.

Output

For each test case, output n integers in a line: the expected number of bags of sugar each child will get. Output the answers as integers modulo 10^9+7, as described above.

Example

standard input	standard output
4	500000007 5 5 6
4	5 10 9
2 5 5 2	166666673 5 6
4 2 1 3	500000006 4 3 4 5
3 2 1 4	
3	
5 4 3	
1 1 1	
6 6 6	
3	
5 4 3	
2 3 1	
1 2 3	
5	
2 1 3 2 1	
5 1 1 3 4	
1 3 4 2 4	

Problem I. Dreamy Putata

Input file: *standard input*
Output file: *standard output*
Time limit: 6 seconds
Memory limit: 1024 mebibytes

Putata is dreaming that he got lost in a phantom grid world of size $n \times m$. The rows and columns of the grid are numbered from 0 to $n-1$ and 0 to $m-1$, respectively. Putata has no idea how to escape from the phantom world, so he decides to walk randomly. Assuming Putata is now at (x, y), he will:

- Move to $(x, (y-1) \bmod m)$ with probability $\frac{\ell(x,y)}{100}$.
- Move to $(x, (y+1) \bmod m)$ with probability $\frac{r(x,y)}{100}$.
- Move to $((x-1) \bmod n, y)$ with probability $\frac{u(x,y)}{100}$.
- Move to $((x+1) \bmod n, y)$ with probability $\frac{d(x,y)}{100}$.

You need to perform q operations. Each operation is one of the following:

- "1 x y $c\ell$ cr cu cd" ($0 \le x < n$, $0 \le y < m$, $1 \le c\ell, cr, cu, cd \le 100$, $c\ell + cr + cu + cd = 100$): Change the values of $\ell(x, y)$, $r(x, y)$, $u(x, y)$, and $d(x, y)$ into $c\ell$, cr, cu, and cd, respectively.
- "2 sx sy tx ty" ($0 \le sx, tx < n$, $0 \le sy, ty < m$, $(sx, sy) \ne (tx, ty)$): Assuming Putata is now at (sx, sy), he is wondering what is the expected number of steps that he will take when he reaches the target position (tx, ty) for the first time.

Please write a program to answer his questions.

Input

The first line of the input contains two integers n and m ($3 \le n \le 10^5$, $3 \le m \le 5$) denoting the size of the phantom grid world.

In the next n lines, the i-th line contains m integers $\ell(i-1, 0), \ell(i-1, 1), \ldots, \ell(i-1, m-1)$ ($1 \le i \le n$, $1 \le \ell(\cdot, \cdot) \le 100$).

In the next n lines, the i-th line contains m integers $r(i-1, 0), r(i-1, 1), \ldots, r(i-1, m-1)$ ($1 \le i \le n$, $1 \le r(\cdot, \cdot) \le 100$).

In the next n lines, the i-th line contains m integers $u(i-1, 0), u(i-1, 1), \ldots, u(i-1, m-1)$ ($1 \le i \le n$, $1 \le u(\cdot, \cdot) \le 100$).

In the next n lines, the i-th line contains m integers $d(i-1, 0), d(i-1, 1), \ldots, d(i-1, m-1)$ ($1 \le i \le n$, $1 \le d(\cdot, \cdot) \le 100$).

It is guaranteed that $\ell(i, j) + r(i, j) + u(i, j) + d(i, j) = 100$ holds for all pairs of (i, j) where $0 \le i < n$ and $0 \le j < m$.

The next line contains a single integer q ($1 \le q \le 3 \cdot 10^4$) denoting the number of operations.

Each of the next q lines describes an operation in the format described in the statement above.

Output

For each test query, print a single line containing an integer: the expected number of steps that Putata will take when he reaches the target position (tx, ty) for the first time.

More precisely, assuming the reduced fraction of the answer is $\frac{p}{q}$, you should output the minimum non-negative integer r such that $q \cdot r \equiv p \pmod{10^9 + 7}$. You may safely assume that such r always exists in all test cases.

Example

standard input	standard output
4 3 1 2 3 4 5 6 7 8 9 10 11 12 23 24 25 26 27 28 29 30 31 32 33 34 10 11 12 13 14 15 16 17 18 19 20 21 66 63 60 57 54 51 48 45 42 39 36 33 4 2 0 1 1 1 2 0 0 3 2 1 1 1 25 25 25 25 2 0 0 3 2	76426175 344136684 555192113

Problem J. Mysterious Tree

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

This is an interactive problem.

Randias has an unknown hidden tree with n vertices. The tree is either a **chain** or a **star**. Randias now needs to determine whether the tree is a chain or a star. He can ask a question in the following form, but no more than $\lceil \frac{n}{2} \rceil + 3$ times:

- Is there an edge between vertex u and vertex v ($1 \le u, v \le n$, $u \ne v$)?

Randias needs to determine which of the two kinds the tree is. Help him to ask the questions and determine the answer.

A tree is called a **chain** if and only if there exists a permutation $p_1, p_2, \ldots, p_n$ such that, for every i ($1 \le i < n$), there is an edge (p_i, p_{i+1}) in the tree. Here, a permutation of length n is an array where each integer from 1 to n appears exactly once.

A tree is called a **star** if and only if there exists a vertex u such that, for every other vertex v, there is an edge (u, v) in the tree.

In this problem, the interactor is **adaptive**, which means that the secret tree is not fixed beforehand. Instead, the interactor can change the tree arbitrarily during the interaction. Nevertheless, at every moment, the tree will be consistent with all the answers you got.

Input

Each test contains multiple test cases. The first line contains a single integer t ($1 \le t \le 250$) denoting the number of test cases.

For each test case, the first line contains one integer n ($4 \le n \le 1000$) denoting the number of vertices. It is guaranteed that the sum of n over all test cases does not exceed 1000.

Interaction Protocol

You can ask at most $\lceil \frac{n}{2} \rceil + 3$ questions in every test case. To ask a question, output a line of the form "`?` u v" ($1 \le u, v \le n$, $u \ne v$). Then you should read the response from standard input.

In response to the query, the interactor will output a line with a single integer: 1 if there is an edge between u and v in the tree, or 0 if there is no such edge.

To give your answer, print a line of the form "`! 1`" if you determined that the tree is a chain, or "`! 2`" if you determined that it is a star. The output of the answer is **not** counted towards the limit of $\lceil \frac{n}{2} \rceil + 3$ queries.

After printing the answer, your program should process the next test case, or terminate if there are no more test cases.

After printing each line, do not forget to output end of line and flush the output. To do the latter, you can use `fflush(stdout)` or `cout.flush()` in C++, `System.out.flush()` in Java, or `stdout.flush()` in Python.

Example

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

Problem K. Card Game

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

Randias is playing a card game. In this game, each card has a number written on it. For cards with numbers $a_1, a_2, \ldots, a_m$, Randias will play the game in the following way.

Initially, all cards are in his hand. Randias will maintain a card sequence (initially empty). In the i-th operation, Randias will put the i-th card (this card has number a_i written on it) at the end of the card sequence. Then:

- If there are no other cards in the sequence with number a_i written on them, the i-th operation ends.
- Otherwise, let the j-th card in the card sequence have number a_i written on it. Randias will take away all cards between the j-th card and the newly placed card, including the j-th card and the newly placed card.

For example, let $a = [2, 1, 3, 1, 2, 3]$, and the card sequence $s = []$ initially.

After the 1-st operation, $s = [2]$.

After the 2-nd operation, $s = [2, 1]$.

After the 3-rd operation, $s = [2, 1, 3]$.

After the 4-th operation, $s = [2]$ (cards $1, 3, 1$ are taken away).

After the 5-th operation, $s = []$ (cards $2, 2$ are taken away).

After the 6-th operation, $s = [3]$.

Now, Randias is given n cards $a_1, a_2, \ldots, a_n$. He has q queries. The i-th query is a pair of integers ℓ_i, r_i. With this query, Randias wants to know how many cards will be left in the card sequence if the initial list of cards is $a_{\ell_i}, a_{\ell_i+1}, \ldots, a_{r_i}$.

For some reason, Randias hopes you can answer the questions **online**. That is, you need to decode the next question with the answer for the previous question.

Input

The first line contains two integers n and q ($1 \le n, q \le 3 \cdot 10^5$) denoting the number of cards and the number of queries.

The following line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le n$).

Each of the following q lines contains two integers ℓ'_i and r'_i ($0 \le \ell'_i, r'_i \le 2n$). Let the answer for the last query is *lastans*. Then $\ell_i = \ell'_i \oplus lastans$ and $r_i = r'_i \oplus lastans$ are the next query. In these formulas, $\oplus$ is the bitwise exclusive OR operation. It is guaranteed that, after decoding, $1 \le \ell_i \le r_i \le n$. If you haven't answered any queries before, $lastans = 0$.

Output

For each query, output a line with one integer: the answer to that query.

Examples

standard input	*standard output*
5 5 3 3 1 1 1 5 5 3 4 3 3 0 5 3 5	1 2 1 0 1
7 7 2 4 1 2 3 1 2 1 6 0 4 3 3 0 4 0 3 0 6 2 7	2 1 1 1 2 3 0

Note

For the first example, the segments in the queries are $[5, 5]$, $[2, 5]$, $[1, 1]$, $[1, 4]$, and $[3, 5]$.

Problem L. Master of Both V

Input file: `standard input`
Output file: `standard output`
Time limit: 6 seconds
Memory limit: 1024 mebibytes

Prof. Chen is the master of data structures and computational geometry. Recently, he taught Putata and Budada the definition of a convex polygon. A convex polygon is a simple polygon (that is, no two vertices coincide and no two edges intersect unless two consecutive edges intersect at a vertex) with all interior angles strictly less than π.

Putata and Budada solved the convex checker problem. But Prof. Chen asked them to go further. Now, they have to maintain a multiset of segments S, initially empty, and support the following two types of queries:

- "`+` px py qx qy": insert a segment with endpoints (px, py) and (qx, qy) into the multiset S.
- "`-` i", erase the segment inserted in the i-th query. It is guaranteed that the i-th query is an insertion query, and the corresponding segment is currently in the multiset.

After each query, Putata and Budada need to answer if there exists a convex polygon $\mathcal{C}$ with the following property. Let the vertices of the convex polygon be $p_0, p_1, p_2, \ldots, p_{m-1}$ in counter-clockwise order. For every segment $u \in S$, there exists an integer $j \in \{0, 1, 2, \ldots, m-1\}$ such that $u \subseteq p_j p_{(j+1) \bmod m}$. For two segments e and f, we say $e \subseteq f$ if and only if, for every point $z \in e$, this point $z \in f$.

Please help Putata and Budada to solve this problem.

Input

Each test contains multiple test cases. The first line contains a single integer t ($1 \le t \le 5 \cdot 10^5$) denoting the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 5 \cdot 10^5$) denoting the number of queries.

Each of the following n lines contains one query. The query begins with a character op ($op \in \{+, -\}$).

If $op =$ `+`, then four integers px, py, qx, and qy ($-10^9 \le px, py, qx, qy \le 10^9$) follow, denoting an inserting query. It is guaranteed that $px \neq qx$ or $py \neq qy$.

Otherwise, an integer i ($1 \le i \le n$) follows, denoting an erasing query. It is guaranteed that the i-th query is an inserting query, and the corresponding segment is currently in the multiset.

It is guaranteed that the sum of n over all test cases does not exceed $5 \cdot 10^5$.

Output

For each test case, print a line consisting of `0`s and `1`s. The i-th character must be `1` if the answer is true after the i-th query, otherwise it must be `0`.

Example

standard input	*standard output*
4 8 + 0 0 1 0 + 5 5 1 3 + 2 0 2 1 + 9 7 6 2 + 1 2 2 2 - 4 + 0 1 0 2 - 2 5 + 0 0 1 1 + 0 1 1 2 + 0 2 1 3 - 2 + 1 1 10 10 4 + 0 0 1 1 + 0 0 1 0 + 0 0 0 1 - 1 4 + 0 0 1 1 + 0 0 1 1 - 1 - 2	11000001 11011 1101 1111

Problem M. V-Diagram

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

A 1-indexed integer sequence a of length n is a V-diagram if $n \geq 3$ and there exists an index i $(1 < i < n)$ satisfying the following:

- $a_j > a_{j+1}$ for $1 \leq j < i$;
- $a_j > a_{j-1}$ for $i < j \leq n$.

Given a V-diagram a, find a V-diagram b with the maximum possible average such that b is a consecutive subsequence of a.

A consecutive subsequence of a sequence can be obtained by removing some (possibly zero) elements from the beginning and end of the sequence.

Input

Each test contains multiple test cases. The first line contains a single integer t $(1 \leq t \leq 10^5)$ denoting the number of test cases. For each test case:

The first line contains one integer n $(3 \leq n \leq 3 \cdot 10^5)$ denoting the length of the integer sequence a.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq 10^9)$ denoting the sequence a itself.

It is guaranteed that a is a V-diagram, and the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, output a real number denoting the maximum possible average.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-9}.

Formally, let your answer be x, and the jury's answer be y. Your answer will be considered correct if and only if $\frac{|x-y|}{\max(1,|y|)} \leq 10^{-9}$.

Example

standard input	*standard output*
2 4 8 2 7 10 6 9 6 5 3 4 8	6.75000000000000000000 5.83333333333333303727