Problem A. LCA

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There are n vertices. There are queries of three types:

1. «`link x y`» — make vertex x the son of y. It is guaranteed that x is a root of some tree and y is in a different tree.

2. «`cut x`» — disconnect x with its father.

3. «`lca x y`» — find the LCA (least common ancestor) of x and y. It is guaranteed that they are in the same tree.

Input

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

Each of the next m lines contains one query.

Output

For each query of the third type, print the answer.

Examples

standard input	standard output

`5 9`
`lca 1 1`
`link 1 2`
`link 3 2`
`link 4 3`
`lca 1 4`
`lca 3 4`
`cut 4`
`link 5 3`
`lca 1 5`
	`1`
	`2`
	`3`
	`2`

`5 16`
`link 3 1`
`lca 3 3`
`lca 1 3`
`lca 3 1`
`lca 1 1`
`cut 3`
`lca 3 3`
`lca 1 1`
`link 1 3`
`link 2 3`
`lca 1 2`
`lca 1 3`
`lca 3 2`
`cut 1`
`link 3 1`
`lca 1 2`
	`3`
	`1`
	`1`
	`1`
	`3`
	`1`
	`3`
	`3`
	`3`
	`1`

Problem B. Distance

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There are n vertices. There are queries of three types:

1. «`link x y`» — make vertex x the son of y. It is guaranteed that x is a root of some tree and y is in a different tree.

2. «`cut x`» — disconnect x with its father.

3. «`lca x y`» — find the LCA (least common ancestor) of x and y and the distance between the vertices. It is guaranteed that they are in the same tree.

Input

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

Each of the next m lines contains one query.

Output

For each query of the third type, print the answer.

Examples

standard input	standard output

5 9
lca 1 1
link 1 2
link 3 2
link 4 3
lca 1 4
lca 3 4
cut 4
link 5 3
lca 1 5
	1 0
	2 3
	3 1
	2 3

5 16
link 3 1
lca 3 3
lca 1 3
lca 3 1
lca 1 1
cut 3
lca 3 3
lca 1 1
link 1 3
link 2 3
lca 1 2
lca 1 3
lca 3 2
cut 1
link 3 1
lca 1 2
	3 0
	1 1
	1 1
	1 0
	3 0
	1 0
	3 2
	3 1
	3 1
	1 2

Problem C. LCT

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There are n vertices. There are queries of three types:

1. «`add x y`» — add an edge between x and y. It is guaranteed that there is no path between them.
2. «`rem x y`» — delete an edge between x and y. It is guaranteed that there is an edge between them.
3. «`conn x y`» — print «`YES`» if there is a path between x and y, or «`NO`» otherwise.

Input

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

Each of the next m lines contains one query.

Output

For each query of the third type print either «`YES`», or «`NO`».

Examples

standard input	standard output
5 11 conn 1 5 add 1 2 add 1 3 add 3 4 add 5 4 conn 1 5 rem 4 5 conn 1 5 rem 3 4 add 3 5 conn 1 5	NO YES NO YES
5 30 conn 3 1 conn 4 3 add 3 5 add 4 5 conn 2 1 conn 2 2 conn 4 3 conn 5 3 rem 5 3 conn 3 5 add 5 2 conn 4 1 rem 4 5 add 4 5 add 2 1 rem 5 4 conn 2 2 conn 4 5 add 3 4 add 3 5 rem 1 2 add 2 1 conn 3 3 rem 3 5 add 4 5 conn 3 4 conn 3 4 rem 1 2 add 5 1 conn 3 1	NO NO NO YES YES YES NO NO YES NO YES YES YES YES

Problem D. Count Vertices

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

There are n vertices. There are queries of two types:

1. «`add x y`» — add an edge between x and y. It is guaranteed that there is no path between them. After that, print the number of vertices in the new component.

2. «`rem x y`» — delete an edge between x and y. It is guaranteed that there is an edge between them. After that, print the number of vertices in the x component, and after that, print the number of vertices in the y component.

Input

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

Each of the next m lines contains one query.

Output

Print the answers for each query.

Examples

standard input	standard output
6 7	2
add 2 4	2
add 1 3	4
add 3 4	5
add 5 3	6
add 6 4	1 5
rem 1 3	3 2
rem 4 3	
10 10	2
add 2 8	1 1
rem 8 2	2
add 3 4	2
add 1 6	3
add 3 9	1 1
rem 1 6	2 1
rem 3 9	1 1
rem 3 4	2
add 2 10	2
add 5 9	
10 20	2
add 9 5	2
add 6 3	3
add 8 3	3
add 9 1	6
add 3 9	2
add 7 4	7
add 1 2	4 3
rem 9 3	7
add 2 3	8
add 10 2	10
add 7 3	8 2
rem 3 7	5 3
rem 2 3	7
add 2 4	10
add 10 8	3 7
rem 1 2	4 3
rem 10 2	2 1
rem 9 5	2 2
rem 8 3	4
add 5 4	

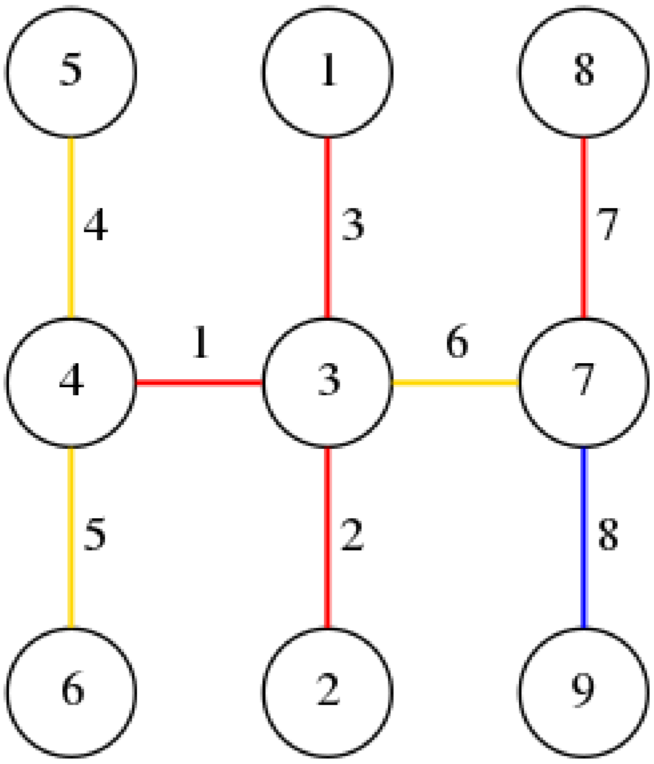
Problem E. Simple Tree Counting

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	512 megabytes

Tlzmybm has a tree (a connected acyclic graph). This tree has n nodes numbered 1 to n. The edges are also numbered 1 to n. The i-th edge connects nodes a_i and b_i.

Furthermore, the edges have colors. The colors are represented as integers. Initially, the i-th edge is colored c_i.

Let's define a component as a maximal subtree whose edges all have the same colors.



In this example, the maximal components (represented as sets of nodes) are $\{1,2,3,4\},\{4,5,6\},\{3,7\},\{7,8\},\{7,9\}$.

Note that a node may appear in more than one component. However, an edge only appears in one component.

Let's define X_i as the component containing the i-th edge. Also, let S_c be the set of components whose common edge color is c. In the example above,

$$X_1 = X_2 = X_3 = \{1,2,3,4\}$$
$$S_{red} = \{\{1,2,3,4\},\{7,8\}\}$$
$$S_{yellow} = \{\{4,5,6\},\{3,7\}\}$$
$$S_{blue} = \{\{7,9\}\}$$

If X is a component, define its productivity $P(X) = \frac{k\times(k-1)}{2}$, where k is the number of nodes in X. If S is a set of components, define its total productivity $T(S)$ as the sum of the productivities of all components in S, i.e., $T(S) = \sum_{X\in S} P(X)$.

Given the tree, your task is to process q operations. There are three kinds of operations:

1. «`1 i c`» — change the color of the i-th edge to c;
2. «`2 l r`» — compute $\sum_{c=l}^{r} T(S_c)$;
3. «`3 i`» — compute $P(X_i)$.

Input

The first line contains one integer n ($1 \le n \le 1.2 \cdot 10^5$) — the number of vertices.

Each of the next $n-1$ lines contains three integers a_i, b_i and c_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$, $1 \le c_i \le 10^6$).

The next line contains one integer q ($1 \le k \le 1.2 \cdot 10^5$) — the number of queries.

Each of the next q lines describes a query.

1. «`1 i c`» ($1 \le i < n$, $1 \le c \le 10^6$);
2. «`2 l r`» ($1 \le l \le r \le n$);
3. «`3 i`» ($1 \le i < n$).

Output

For each query of the second and third types, print out the answer.

Examples

standard input	standard output
5	4
1 2 6	0
1 3 2	
2 4 7	
4 5 7	
3	
2 6 7	
2 3 4	
1 2 7	
5	1
1 2 8	4
1 3 7	
3 4 8	
2 5 3	
3	
3 4	
1 4 2	
2 1 10	
6	7
1 2 2	3
3 2 1	1
4 3 2	1
5 3 1	6
6 4 2	
7	
2 1 2	
3 4	
1 2 2	
2 1 1	
3 4	
1 1 1	
2 2 2	

Problem F. Subtrees and Paths

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There is a tree with n notes, the root of this tree is node 1.

There are queries:

1. «`add t value`» — add *value* to all vertices of subtree t;
2. «`max a b`» — find the maximum on a path between a and b.

Input

The first line contains one integer n ($1 \le n \le 10^5$) — the number of vertices.

Each of the next $n-1$ lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$).

The next line contains q ($1 \le q \le 10^5$) — the number of queries.

Each of the next q describe queries.

It is guaranteed that $|value| \le 10^4$.

Output

For each query of the second type, print the answer.

Example

standard input	standard output
5 1 2 2 3 2 4 5 1 6 add 4 30 add 5 20 max 4 5 add 2 -20 max 4 5 max 3 4	30 20 10

Problem G. Tourists

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

There are n cities and m roads.

Each city sells some souvenirs. In the i-th city, the price of souvenir is w_i.

There are q queries:

1. «`C a w`» — change the price of the souvenirs in the a-th city to w;

2. «`A a b`» — a tourist is traveling from city a to city b. He wants to buy the cheapest souvenir. However, he does not want to visit any city more than once. Find the minimum price.

Input

The first line contains three integers n, m, and q ($1 \le n, m, q \le 10^5$) — the number of cities, roads, and queries.

Each of the next n lines contain one integer w_i ($1 \le w_i \le 10^9$) — the price of souvenir in the i-th city.

Each of the next m lines contain two integers a_i and b_i ($1 \le a_i, b_i \le n$, $a_i \ne b_i$) — the cities which the i-th road connects. It is guaranteed that there is no more than one road between any pair of cities. It is guaranteed that it is possible to travel from any city to any other.

Each of the next q lines describe a query.

«`C a w`» ($1 \le a \le n$, $1 \le w \le 10^9$).

«`A a b`» ($1 \le a, b \le n$).

Output

For each query of the second type, print the answer.

Examples

standard input	standard output
3 3 3	1
1	2
2	
3	
1 2	
2 3	
1 3	
A 2 3	
C 1 5	
A 2 3	
7 9 4	2
1	1
2	5
3	3
4	
5	
6	
7	
1 2	
2 5	
1 5	
2 3	
3 4	
2 4	
5 6	
6 7	
5 7	
A 2 3	
A 6 4	
A 6 7	
A 3 3	

Problem H. Interesting excursion

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

Flatland has n cities, connected by m one-directional roads.

Tourist company plans to develop a scenic cyclic tour along the roads of Flatland. This tour must start and finish at the same city, visiting some intermediate cities and traveling along some of the roads in their direction. The tour can visit some cities multiple times, but it may not use the same road more than once.

Each road is characterized by the type of its landscape, which is the number from 1 to m. To make the tour really magnificent, every two adjacent roads in the tour must have different landscape types. This also should be true for the first and the last road in the tour, so that you start to travel from any city of the tour.

Help the company to find the tour satisfying these conditions, or report that no such tour exists.

Input

Input contains multiple test cases. The first line contains an integer T — the number of test cases ($1 \le T \le 10^5$).

The first line of each test case description contains two integers n and m ($2 \le n, m \le 2 \cdot 10^5$) — the number of cities and the number of roads. Each of the next m lines contains three integers u_i v_i c_i, meaning that the i-th road starts at city u_i, ends at city v_i and has landscape type c_i ($1 \le u_i, v_i \le n$, $1 \le c_i \le m$, $u_i \ne v_i$).

Sum of all n in all test cases does not exceed $2 \cdot 10^5$. Sum of all m in all test cases does not exceed $2 \cdot 10^5$.

Output

Output the answer of each test case.

If the desired tour does not exist, output the only number «-1». Otherwise, print number k, such that $2 \le k \le m$ — the length of the tour. The next line must contain k numbers $e_1, e_2, \ldots, e_k$ — the numbers of roads in the tour. All numbers e_i must be different. If there are multiple possible tours, output any of them.

Example

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

Problem I. Sushi

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

Bonnie and Clyde have decided to meet at a restaurant to have Sushi. The city is given in the form of an undirected graph with n nodes (numbered from 1 to n) and m edges (without any parallel edges or self loops). Bonnie's house is located at node u and Clyde's house is located at node v. The famous restaurant Sushi Grand is located at node w.

Your task is to find out if Bonnie and Clyde can meet at Sushi Grand such that their path has no common node other than the destination w.

You have to answer q queries. In each query you will be given u, v and w. Find out if Bonnie and Clyde can meet at the Sushi Grand restaurant. If yes, print «`YES`» otherwise print «`NO`».

Input

The first line contains three integers n, m, and q ($1 \le n \le 10^5$, $1 \le m \le min(\frac{n \times (n-1)}{2}, 2 \times 10^5)$, $1 \le q \le 10^5$) — the number of nodes, edges, and queries.

Each of the next m lines contains two v_i and u_i ($1 \le v_i, u_i \le n$).

Each of the next q lines contains three integers v_i, u_i, and w_i ($1 \le v_i, u_i, w_i \le n$).

Output

For each query print «`YES`» or «`NO`».

Examples

standard input	standard output

4 4 3
1 2
2 3
2 4
3 1
2 4 1
1 4 3
1 4 2
	NO
	YES
	YES

5 6 6
1 2
1 3
2 4
3 5
4 1
5 1
1 4 3
1 5 2
4 5 2
2 3 1
2 1 5
2 5 4
	NO
	NO
	YES
	YES
	NO
	YES

Problem J. Tree-Tac-Toe

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

The tic-tac-toe game is starting on a tree of n vertices. Some vertices are already colored in white while the remaining are uncolored.

There are two players — white and black. The players make moves alternatively. The white player starts the game. In his turn, a player must select one uncolored vertex and paint it in his color.

The player wins if he paints some path of three vertices in his color. In case all vertices are colored and neither player won, the game ends in a draw.

Could you please find who will win the game or whether it ends as a draw, assuming both players play optimally?

Input

The first line contains a single integer T ($1 \le T \le 50\,000$) — the number of test cases. Then descriptions of T test cases follow.

The first line of each test contains a single integer n ($1 \le n \le 5 \cdot 10^5$) — the number of vertices in the tree.

Each of the following $n - 1$ lines contains integers v, u ($1 \le v, u \le n$) denoting an edge of the tree connecting vertices v and u.

The last line of a test case contains a string of letters '`W`' (for white) and '`N`' (for not colored) of length n denoting already colored vertices. Vertexes already colored in white are denoted as '`W`'.

It's guaranteed that the given edges form a tree, that there is at least one uncolored vertex and that there is no path of three white vertices.

It's guaranteed that sum of all n among all test cases is at most $5 \cdot 10^5$.

Output

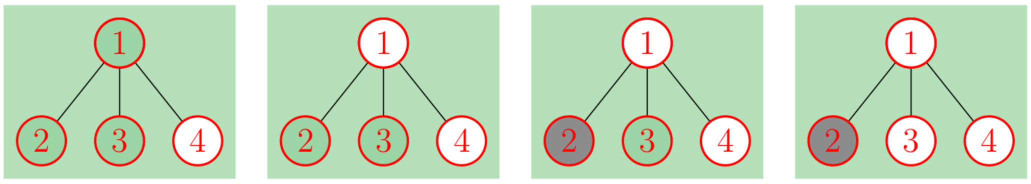
For every test case, print either "`White`", "`Draw`" or "`Black`", depending on the result of the game.

Example

standard input	standard output
2 4 1 2 1 3 1 4 NNNW 5 1 2 2 3 3 4 4 5 NNNNN	White Draw

Note

In the first example, vertex 4 is already colored in white. The white player can win by coloring the vertex 1 in white first and the remaining vertex on his second turn. The process is illustrated with the pictures below.



In the second example, we can show that no player can enforce their victory.

Problem K. Elephants

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

This is an interactive problem.

Dancing Elephants is a spectacular show in Pattaya that features N elephants dancing on a line, known as the stage.

After years of training, elephants in this show are capable of many amazing dances. The show consists of a series of acts. In each act, exactly one elephant performs a cute dance while possibly moving to a different position. The show producers want to produce a photo book that contains pictures of the entire show.

After each act, they want to take pictures of all elephants as seen by the spectators. At any time during the show, multiple elephants may share the same position. In that case, they simply stand behind one another at the same position.

A single camera can take a picture of a group of elephants if and only if all their positions lie on some segment of length L (including both its endpoints). As the elephants can spread out across the stage, multiple cameras may be needed in order to take simultaneous snapshots of all the elephants.

In this task, you have to determine the minimum number of cameras needed to take the pictures after each of the acts. Note that the number of cameras needed may increase, decrease, or stay the same between acts.

Interaction Protocol

The first line contains three integers n, l, and m ($1 \le n, m \le 1.5 \times 10^5$, $0 \le l \le 10^9$) — the number of elephants, the length of segment, and the number of acts.

Each of the next n lines contains x_i ($0 \le w_i \le 10^9$) — the locations of the i-th elephant. Their locations increase.

For each of the next m queries, you need to read two integers i and x ($0 \le i < n$, $0 \le x \le 10^9$) — the index of the elephant, which is moving, and the location where it stopped. After each query, print the answer, the symbol of end of line, and make `flush` operation.

If you read -1, please exit your program.

Example

standard input	standard output

4 10 5
10
15
17
20
2 16
1 25
3 35
0 38
2 0
	1
	2
	2
	2
	3