Job Interview

Your company is hiring, and N candidates, numbered from 1 to N, have applied for the same position. Each candidate was interviewed by the same three interviewers, and candidate i received the scores a_i, b_i and c_i from the first, the second and the third interviewer, respectively. No interviewer gave the same score to two different candidates.

You are the assistant of the Head of HR, who is a very meticulous person. Your job is to prepare the candidates' resumes for your boss as a single stack, such that whenever a resume lies directly on top of another one, the candidate of the upper resume is *better* than the candidate of the lower one.

Initially, for two different candidates u and v, candidate u is better than candidate v if and only if u received a higher score than v from **at least two** of the three interviewers. It can be seen that for every pair of different candidates (u, v), exactly one of them is considered better than the other.

Note that being better is not necessarily transitive. For example, suppose that there are three candidates with scores $(1, 2, 3)$, $(2, 3, 1)$ and $(3, 1, 2)$. Then candidate 1 is better than candidate 3 (higher scores from the second and the third interviewers), candidate 3 is better than candidate 2 (from the first and the third interviewers), but candidate 2 is better than candidate 1 (from the first and the second interviewers). Even so, the stack with candidate 1 on top, then candidate 3, then candidate 2 at the bottom is valid: only resumes lying directly on top of each other are compared, so it does not matter that candidate 2 is better than candidate 1.

During the day, your colleagues keep bringing you news, which may change who is better than whom. There are Q events, which happen **one after another**. Each event changes only the comparisons described below, and all other comparisons stay the same. There are four types of events:

- `UP` x: it turns out that candidate x is a relative of the CEO. Candidate x is now better than every other candidate.
- `DOWN` x: it turns out that candidate x is a relative of the CEO's sworn enemy. Every other candidate is now better than candidate x.
- `REV` x y: the hiring committee reviews candidates x and y once more and changes its mind about them. If candidate x was better than candidate y, then candidate y is now better than candidate x, and vice versa.
- `STACK`: nothing changes, but your boss asks you for the stack of all N resumes, according to the current comparisons. Formally, the stack is a permutation $p_1, p_2, \ldots, p_N$ of $1, 2, \ldots, N$, listed from top to bottom, such that for every $1 \le i < N$, candidate p_i is better than candidate p_{i+1} at the moment. Determine whether such a stack exists, and if so, find one.

Note:

- At any moment (at the beginning or after every event), the following statement is always true: for every pair of different candidates (u, v), exactly one of them is considered better than the other.
- At the beginning, the better candidate among a pair (u, v) is determined based on two tuples (a_u, b_u, c_u) and (a_v, b_v, c_v), as described above.
- Let $T_{u,v}^{(i)}$ be the better candidate among the pair (u, v) after the i-th event, $T_{u,v}^{(i)} = T_{u,v}^{(i-1)}$ for every pair of candidates (u, v) and every event i, except for the fact that:
  - If the i-th event is `UP` x, $T_{u,x}^{(i)} = x$ for every candidate $u \neq x$.
  - If the i-th event is `DOWN` x, $T_{u,x}^{(i)} = u$ for every candidate $u \neq x$.
  - If the i-th event is `REV` x y, $T_{x,y}^{(i)} \neq T_{x,y}^{(i-1)}$.
- If the i-th event is `STACK`, you need to find a permutation $p_1, p_2, \ldots, p_N$ of $1, 2, \ldots, N$, satisfying that $T_{p_j,p_{j+1}}^{(i)} = p_j$ for every $1 \leq j < N$, or declare that such a permutation does not exist.

Input

The first line contains two integers N and Q ($1 \leq N \leq 10^5$; $1 \leq Q \leq 2 \cdot 10^5$) – the number of candidates and the number of events.

The i-th of the next N lines contains three integers a_i, b_i and c_i ($1 \leq a_i, b_i, c_i \leq 10^6$) – the scores of candidate i from the first, the second and the third interviewer. The values $a_1, a_2, \ldots, a_N$ are pairwise distinct, and so are $b_1, b_2, \ldots, b_N$ and $c_1, c_2, \ldots, c_N$.

Each of the next Q lines describes an event, in the order they happen, in one of the following formats: `UP` x, `DOWN` x, `REV` x y or `STACK` ($1 \leq x, y \leq N$; $x \neq y$).

There are at most 20 `STACK` **events.**

Output

For each `STACK` event, in the order they appear in the input:

- If there is no valid stack, print a single line `NO`.
- Otherwise, print a line `YES`, followed by a line containing N integers $p_1, p_2, \ldots, p_N$ – a valid stack of resumes, from top to bottom.

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

Sample Input 1	Sample Output 1
3 6 3 1 2 1 3 1 2 2 3 STACK UP 2 REV 3 1 STACK DOWN 2 STACK	YES 3 1 2 YES 2 1 3 YES 1 3 2

Sample Explanation

Initially, candidate 1 is better than candidate 2 (higher scores from the first and the third interviewers), candidate 3 is better than candidate 1 (higher scores from the second and the third interviewers), and candidate 3 is better than candidate 2 (higher scores from the first and the third interviewers).

1. `STACK`: the stack $3, 1, 2$ is valid, since candidate 3 is better than candidate 1, and candidate 1 is better than candidate 2.
2. `UP 2`: candidate 2 is now better than candidates 1 and 3, and candidate 3 is still better than candidate 1.
3. `REV 3 1`: candidate 1 is now better than candidate 3.
4. `STACK`: the stack $2, 1, 3$ is valid.
5. `DOWN 2`: candidates 1 and 3 are now better than candidate 2, and candidate 1 is still better than candidate 3.
6. `STACK`: the stack $1, 3, 2$ is valid.

This page is intentionally left blank.