6910 Cutting Tree

Tree in graph theory refers to any connected graph (of nodes and edges) which has no simple cycle, while forest corresponds to a collection of one or more trees. In this problem, you are given a forest of N nodes (of rooted trees) and K queries. Each query is in the form of:

- `C` x : remove the edge connecting node and its parent. If node has no parent, then ignore this query.
- `Q` a b : output '`YES`' if there is a path from node to node in the forest; otherwise, '`NO`'.

For example, let the initial forest is shown by Figure 1.

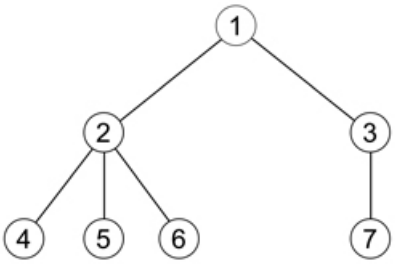

Figure 1.

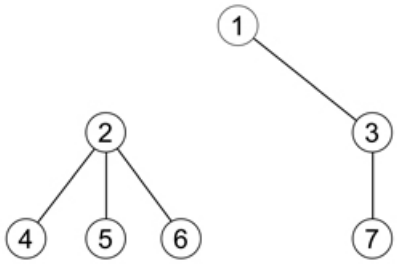

Figure 2.

Let's consider the following queries (in order):

1) `Q` 5 7 : output `YES`.
2) `C` 2 : remove edge (2, 1) — the resulting forest is shown in Figure 2.
3) `Q` 5 7 : output `NO`, as there is no path from node 5 to node 7 in Figure 2.
4) `Q` 4 6 : output `YES`.

Input

The first line of input contains an integer T ($T \leq 50$) denoting the number of cases. Each case begins with two integers: N and K ($1 \leq N \leq 20,000$; $1 \leq K \leq 5,000$) denoting the number of nodes in the forest and the number of queries respectively. The nodes are numbered from 1 to N. The next line contains N integers P_i ($0 \leq P_i \leq N$) denoting the parent of i-th node respectively. $P_i = 0$ means that node i does not have any parent (i.e. it's a root of a tree). You are guaranteed that the given input corresponds to a valid forest. The next K lines represent the queries. Each query is in the form of '`C` x' or '`Q` a b' ($1 \leq x, a, b \leq N$), as described in the problem statement above.

Output

For each case, output '`Case #`X`:`' in a line, where X is the case number starts from 1. For each '`Q` a b' query in the input, output either '`YES`' or '`NO`' (without quotes) in a line whether there is a path from node a to node b in the forest.

Explanation for 2nd sample case:

The initial forest is shown in Figure 3 below.

1) C 3 : remove edge (3, 2) — the resulting forest is shown in Figure 4.

2) Q 1 2 : output `YES`.

3) C 1 : remove edge (1, 2) — the resulting forest is shown in Figure 5.

4) Q 1 2 : output `NO` as there is no path from node 1 to node 2 in Figure 5.

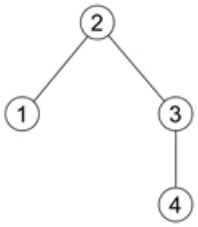

Figure 3.

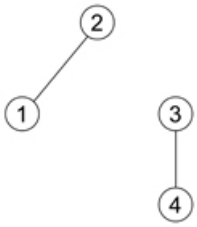

Figure 4.

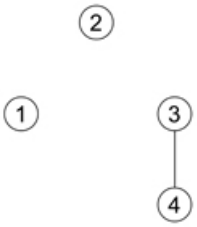

Figure 5.

Sample Input

```
4
7 4
0 1 1 2 2 2 3
Q 5 7
C 2
Q 5 7
Q 4 6
4 4
2 0 2 3
C 3
Q 1 2
C 1
Q 1 2
3 5
0 3 0
C 1
Q 1 2
C 3
C 1
Q 2 3
1 1
0
Q 1 1
```

Sample Output

```
Case #1:
YES
NO
YES
Case #2:
YES
```

```
NO
Case #3:
NO
YES
Case #4:
YES
```