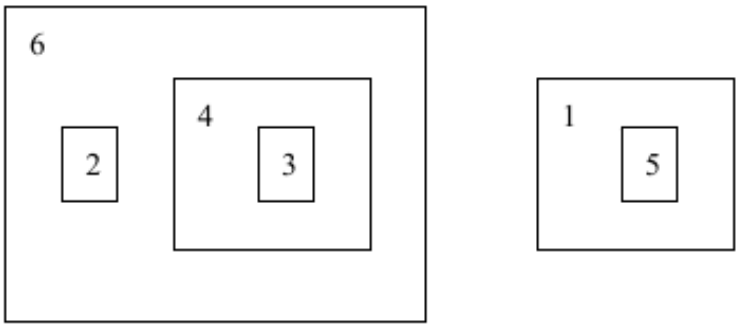
4393 Box

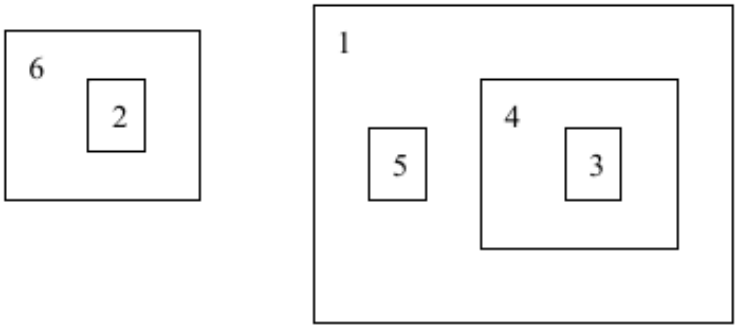
There are N boxes on the ground, which are labeled by numbers from 1 to N. The boxes are magical, the size of each one can be enlarged or reduced arbitrarily.

Jack can perform the "`MOVE` x y" operation to the boxes: take out box x; if $y = 0$, put it on the ground; Otherwise, put it inside box y. All the boxes inside box x remain the same. It is possible that an operation is illegal, that is, if box y is contained (directly or indirectly) by box x, or if y is equal to x.

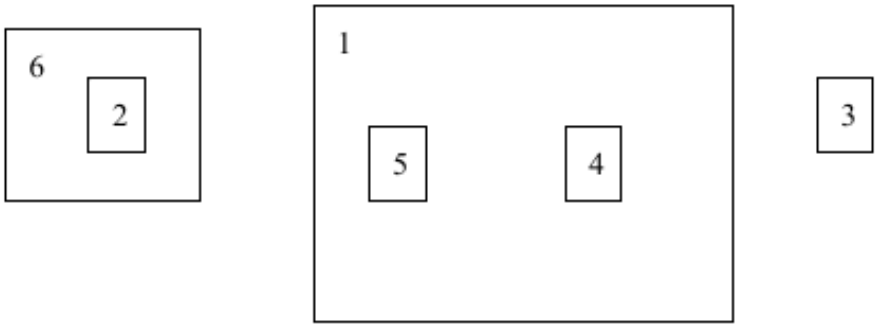
In the following picture, box 2 and 4 are directly inside box 6, box 3 is directly inside box 4, box 5 is directly inside box 1, box 1 and 6 are on the ground.



The picture below shows the state after Jack performs "`MOVE 4 1`":



Then he performs "`MOVE 3 0`", the state becomes:



During a sequence of `MOVE` operations, Jack wants to know the root box of a specified box. The root box of box x is defined as the most outside box which contains box x. In the last picture, the root box of box 5 is box 1, and box 3's root box is itself.

Input

Input contains several test cases.

For each test case, the first line has an integer N ($1 \leq N \leq 50000$), representing the number of boxes.

Next line has N integers: $a_1, a_2, a_3, \ldots, a_N$ ($0 \leq a_i \leq N$), describing the initial state of the boxes. If a_i is 0, box i is on the ground, it is not contained by any box. Otherwise, box i is directly inside box a_i. It is guaranteed that the input state is always correct (No loop exists).

Next line has an integer M ($1 \leq M \leq 100000$), representing the number of '`MOVE`' operations and queries.

On the next M lines, each line contains a '`MOVE`' operation or a query:

1. `MOVE` x y, $1 \leq x \leq N$, $0 \leq y \leq N$, which is described above. If an operation is illegal, just ignore it.
2. `QUERY` x, $1 \leq x \leq N$, output the root box of box x.

Output

For each query, output the result on a single line. Use a blank line to separate each test case.

Sample Input

```
2
0 1
5
QUERY 1
QUERY 2
MOVE 2 0
MOVE 1 2
QUERY 1 6
0 6 4 6 1 0
4
MOVE 4 1
QUERY 3
MOVE 1 4
QUERY 1
```

Sample Output

```
1
1
2

1
1
```