

# Tree Query

You are given a full binary tree, and there is an integer representing the data at each node of the tree. The other members of your band are asking you questions in the following form. They give you an integer  $X$ , and you have to tell them which leaf of the tree will get  $X$  for if you follow the path from it to the root of the tree and sum the numbers in each node. If there are two such leaves, choose the one with a smaller ID. The root of the tree has an ID of 0. Its left child has an ID of 1, and its right one an ID of 2. In more mathematical form, if you have a node with an ID of  $x$ , then the left child of  $x$  is  $2*x+1$  and the right one is  $2*x+2$ . The members of your band will be impressed if you answer their questions fast and correctly. Cool!

## Input

The input consists of multiple test cases. Each test case starts with the number of nodes in the tree  $N$  ( $1 \leq N \leq 1000$ ). The last test case will have  $N = 0$ , so at this point you should quit. As you know a full binary tree with  $N$  nodes will have  $N = (2^x) - 1$  nodes. The nodes are given IDs using the scheme described above, and the root has an ID of 0. On the next  $N$  lines of the input, you will find the the data that is stored in the corresponding node. On the next line after this is the integer  $M$  ( $1 \leq M \leq 1000$ ) - the number of question the band members are going to ask you. On each of the next  $M$  lines you will find one integer representing the question.

## Output

For each of the questions you have to output the minimum ID of a leaf node that meets the requirements. If no leaf meets the requirements output "NOT FOUND". Refer to the example tests.

## Example

### Input:

```
3
1
2
3
4
0
3
4
5
0
```

### Output:

```
NOT FOUND
1
2
NOT FOUND
```