Tree Search

You are given a rooted binary tree consisting of N vertices. The vertices are numbered from 1 to N, the root is the vertex number 1. Each of the other vertices has a single parent in the tree. The tree is binary, i.e. each vertex can be a parent of at most two other vertices.

One of the vertices is special. You are trying to guess it. You can ask the questions of the following kind: "Is the special vertex in the subtree of vertex x"? A node y is in the subtree of vertex x if and only if the shortest path between y and 1 goes through vertex x. Note that vertex x is also in its own subtree.

You are allowed to ask this question at most 35 times. After that you should report your guess.

Implementation Details

You should implement the following procedure:

```
int solve(int N, std::vector < int > p)
```

- N: the number of vertices
- p contains exactly $N-1$ elements that describe the tree: the vertex $p[i]$ (where $1 \le p[i] \le i+1$) is the parent of the vertex $i+2$ for each $0 \le i \le N-2$
- No element in p appears more than twice
- This procedure should return the number of the special vertex
- This procedure is called exactly once

The above procedure can make calls to the following procedure:

```
int ask(int x)
```

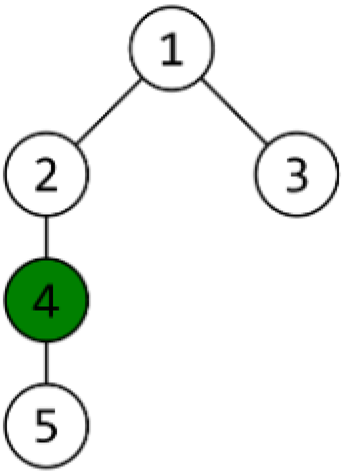
- x: the number of the vertex
- $1 \le x \le N$
- returns 1 if the special vertex is in the subtree of x and 0 otherwise

Example

Consider the following call:

```
solve(5, [1, 1, 2, 4])
```

The tree consists of the edges (1,2), (1,3), (2,4) and (4,5).



Your program made a call

```
ask(4)
```

which returned 1. After that your program made a call

```
ask(5)
```

which returned 0.

Your program concluded that vertex 4 is special and returned 4.

Constraints

- $2 \leq N \leq 100\ 000$

Subtasks

1. (20 points) $N \leq 35$
2. (30 points) $p[i] = i + 1$ for each $0 \leq i \leq N - 2$
3. (15 points) $p[i] = \lfloor i/2 \rfloor + 1$ for each $0 \leq i \leq N - 2$
4. (35 points) No additional constraints.

Sample Grader

The sample grader reads the input in the following format:

- line 1: N
- line 2: $p[0], p[1], ..., p[N-2]$

The sample grader outputs each question in the following format:

- line 1: $?\ x$

The sample grader reads each answer in the following format:

- line 1: y

The sample grader outputs the guess in the following format:

- line 1: $!\ x$