

# Ancestor

Input file: `ancestor.in`  
Output file: `ancestor.out`  
Time limit: 1 секунда  
Memory limit: 256 мегабайт

You are given a rooted tree of size  $n$  and  $m$  “IsAncestor” queries to process. Go for it!

## Input

The first line of the input contains a single integer  $n$  ( $1 \leq n \leq 100\,000$ ) — the number of vertices forming the tree. The next line contains  $n$  integers, the  $i$ -th of them is the parent of the node  $i$ . If the corresponding integer is 0, then this node is the root of the tree.

The third line contains an integer  $m$  ( $1 \leq m \leq 100\,000$ ) — the number of queries. Each of the following  $m$  lines contains two distinct integers  $a$  and  $b$  ( $1 \leq a, b \leq n$ ).

Первая строка входного файла содержит натуральное число  $n$  ( $1 \leq n \leq 100\,000$ ) — количество вершин в дереве. Во второй строке находятся  $n$  чисел,  $i$ -е из которых определяет номер непосредственного родителя вершины с номером  $i$ . Если это число равно нулю, то вершина является корнем дерева.

В третьей строке находится число  $m$  ( $1 \leq m \leq 100\,000$ ) — количество запросов. Каждая из следующих  $m$  строк содержит два различных числа  $a$  и  $b$  ( $1 \leq a, b \leq n$ ), meaning the query “is node  $a$  among ancestors of node  $b$ ?”

## Output

For each query print ‘1’ if node  $a$  is an ancestor of node  $b$  and ‘0’ otherwise.

## Examples

<code>ancestor.in</code>	<code>ancestor.out</code>
6	0
0 1 1 2 3 3	1
5	1
4 1	0
1 4	0
3 6	
2 6	
6 5	