Tree Game

A complete binary tree of depth h is given. You can asign the value 0 or 1 to each leaf. Its internal nodes compute their values based on the values of their children, which is 0 if both children have value 1, and 1 otherwise. You play a game with a computer. The computer can ask you for a value of any leaf, and you can tell him either 0 or 1. The computer wants to know the root value, and he will keep asking until he is absolutely sure what the root value is. Your goal is to make him ask as many questions as possible. It is known that you can make a computer ask for all leaves. For a given sequence of leaves determine a sequence of 0's and 1's as answers to those leaves such that at no point before asking the last leaf value can the computer be sure of the root value. You must answer each question optimally (i.e. you should not make use of the knowledge of what the computer's (i+1)st query will be when you answer the ith query).

Input

In the first line of input the number h is given ($1 \leq h \leq 15$). In the second line a space-separated list of 2^h numbers are given. They are a permutation of the numbers 1, 2, ..., 2^h, and they represent the order of asked leaves (leaves of a tree are indexed from left to right).

Output

On a single line write a space-separated sequence of 0's and 1's corresponding to the values of leaves in the given order of being asked. If there are multiple solutions, write any of them. Do not output a response for the last query.

Example

Input:

```
3
5 2 7 3 1 6 8 4
```

Output:

```
1 1 1 1 0 0 1
```