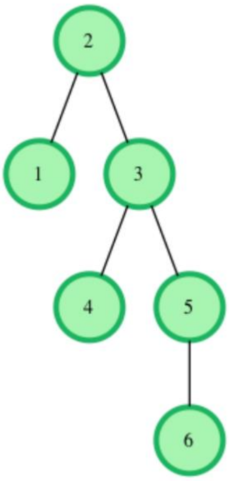
Binary Search Tree: Lowest Common Ancestor

You are given the elements of a binary search tree (BST) and two values ***v1*** and ***v2***. First, you need to create the BST, the you should print the lowest common ancestor (LCA) of ***v1*** and ***v2*** in the BST.



For example, in the diagram above, the lowest common ancestor of the nodes ***4*** and ***6*** is the node ***3***, which is the lowest node that has nodes ***4*** and ***6*** as descendants.

Input Format

The first line contains an integer, ***n***, the number of nodes in the tree.

The second line contains space-separated integers representing ***node.data*** values.

The third line contains two space-separated integers, ***v1*** and ***v2***.

Constraints

- **1 ≤ n, node.data ≤ 25**
- **1 ≤ v1, v2 ≤ 25**
- **v1, v2, where v1 ≠ v2**
- It is guaranteed that the tree will contain nodes with *data* equal to ***v1*** and ***v2***.

With the support of the Erasmus+ Programme of the European Union

Output Format

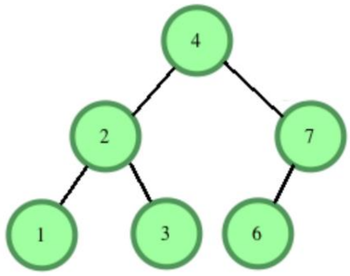
Return the value to the node that is the lowest common ancestor of ***v1*** and ***v2***.

Sample

Sample input	Sample output

6
4 2 3 1 7 6
1 7
	4

Explanation



The figure represents the BST created with the sample input.

v1 = 1 and v2 = 7.

LCA of ***1*** and ***7*** is ***4***, the root in this case.

Return the value to the node.