

A binary search tree is a binary tree that satisfies the following properties:

- The left subtree of a node contains only nodes with keys less than the node's key.
- The right subtree of a node contains only nodes with keys greater than the node's key.
- Both the left and right subtrees must also be binary search trees.

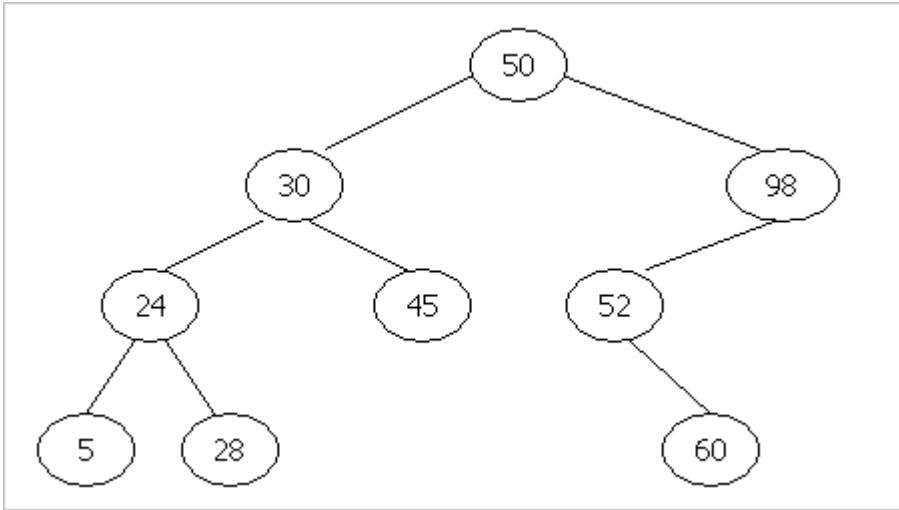


Figure 1. Example binary search tree

*Pre-order traversal* (Root-Left-Right) prints out the nodes key by visiting the root node then traversing the left subtree and then traversing the right subtree. *Post-order traversal* (Left Right-Root) prints out the left subtree first and then right subtree and finally the root node. For example, the results of pre-order traversal and post-order traversal of the binary tree shown in Figure 1 are as follows:

**Pre-order:** 50 30 24 5 28 45 98 52 60

**Post-order:** 5 28 24 45 30 60 52 98 50

Given the pre-order traversal of a binary search tree, your task is to find the post-order traversal of this tree.

## Input

The keys of all nodes of the input binary search tree are given according to pre-order traversal. Each node has a key value which is a positive integer less than  $10^6$ . All values are given in separate lines (one integer per line). You can assume that a binary search tree does not contain more than 10,000 nodes and there are no duplicate nodes.

## Output

The output contains the result of post-order traversal of the input binary tree. Print out one key per line.

## Sample Input

```
50
30
24
5
28
45
98
52
60
```

## Sample Output

```
5
28
24
45
30
60
52
98
50
```