Trees are dynamic nonlinear data structure. A special type of trees is binary tree that each node contains at most two children. A famous variation of binary tree is the binary search tree (BST). Every node in a BST follows the BST rule that the keys of all nodes in the left subtree are less than such node, and the keys of all nodes in the right subtree are greater than such node. The construction of a BST bases on the insertion of nodes from an empty binary tree. The insertion procedure finds an appropriate leaf node according to the BST rule to insert. A good property for BST is that the inorder traversal results in a sorted sequence.
Your goal is to write a program to construct a BST by insertion, print the contents of the BST by inorder traversal, and delete the BST by postorder traversal.
Requirement: Use self-referential structure to build a BST.
Prohibited: Static as well as dynamic array, any sorting methods for array.

Input

Each line contains an integer to be inputted, and input ends with an EOF marker.

Output

A sorted sequence of inputted values without duplicate values. Each line contains an integer.

Sample Input

```
6
1
2
4
2
3
1
5
```

Sample Output

```
1
2
3
4
5
6
```