Contains x

Send Feedback

Given a generic tree and an integer x, check if x is present in the given tree or not. Return true if x is present, return false otherwise.

Input format :

The first line of input contains data of the nodes of the tree in level order form. The order is: data for root node, number of children to root node, data of each of child nodes and so on and so forth for each node. The data of the nodes of the tree is separated by space.
The following line contains an integer, that denotes the value of x.

Output format :

The first and only line of output contains true, if x is present and false, otherwise.

Constraints:

Time Limit: 1 sec

Sample Input 1 :

```
10 3 20 30 40 2 40 50 0 0 0 0
40
```

Sample Output 1 :

```
true
```

Sample Input 2 :

```
10 3 20 30 40 2 40 50 0 0 0 0
4
```

Sample Output 2:

```
false
```

Count nodes

Send Feedback

Given a tree and an integer x, find and return the number of nodes which contains data greater than x.

Input format:

The first line of input contains data of the nodes of the tree in level order form. The order is: data for root node, number of children to root node, data of each of child nodes and so on and so forth for each node. The data of the nodes of the tree is separated by space.
The following line contains an integer, that denotes the value of x.

Output Format :

The first and only line of output prints the count of nodes greater than x.

Constraints:

Time Limit: 1 sec

Sample Input 1 :

10 3 20 30 40 2 40 50 0 0 0 0
35

Sample Output 1 :

3

Sample Input 2 :

10 3 20 30 40 2 40 50 0 0 0 0
10

Sample Output 2:

5

Node with maximum child sum

Send Feedback

Given a generic tree, find and return the node for which sum of its data and data of all its child nodes is maximum. In the sum, data of the node and data of its immediate child nodes has to be taken.

Input format :

The first line of input contains data of the nodes of the tree in level order form. The order is: data for root node, number of children to root node, data of each of child nodes and so on and so forth for each node. The data of the nodes of the tree is separated by space.

Output format :

The first and only line of output contains the data of the node with maximum sum, as described in the task.

Constraints:

Time Limit: 1 sec

Sample Input 1 :

5 3 1 2 3 1 15 2 4 5 1 6 0 0 0 0

Sample Output 1 :

1

Structurally identical

Send Feedback

Given two generic trees, return true if they are structurally identical. Otherwise return false.

Structural Identical

If the two given trees are made of nodes with the same values and the nodes are arranged in the same way, then the trees are called identical.

Input format :

The first line of input contains data of the nodes of the first tree in level order form. The order is: data for root node, number of children to root node, data of each of child nodes and so on and so forth for each node. The data of the nodes of the tree is separated by space.
The following line of input contains data of the nodes of the second tree in level order form. The order is: data for root node, number of children to root node, data of each of child nodes and so on and so forth for each node. The data of the nodes of the tree is separated by space.

Output format :

The first and only line of output contains true, if the given trees are structurally identical and false, otherwise.

Constraints:

Time Limit: 1 sec

Sample Input 1 :

```
10 3 20 30 40 2 40 50 0 0 0 0
10 3 20 30 40 2 40 50 0 0 0 0
```

Sample Output 1 :

```
true
```

Sample Input 2 :

```
10 3 20 30 40 2 40 50 0 0 0 0
10 3 2 30 40 2 40 50 0 0 0 0
```

Sample Output 2:

false

Next larger

Send Feedback

Given a generic tree and an integer n. Find and return the node with next larger element in the tree i.e. find a node with value just greater than n.

Note: Return NULL if no node is present with the value greater than n.

Input format :

The first line of input contains data of the nodes of the tree in level order form. The order is: data for root node, number of children to root node, data of each of child nodes and so on and so forth for each node. The data of the nodes of the tree is separated by space.
The following line contains an integer, that denotes the value of n.

Output format :

The first and only line of output contains data of the node, whose data is just greater than n.

Constraints:

Time Limit: 1 sec

Sample Input 1 :

10 3 20 30 40 2 40 50 0 0 0 0
18

Sample Output 1 :

20

Sample Input 2 :

```
10 3 20 30 40 2 40 50 0 0 0 0
21
```

Sample Output 2:

```
30
```

Second Largest Element In Tree

Send Feedback

Given a generic tree, find and return the node with second largest value in given tree.

Note: Return NULL if no node with required value is present.

Input format :

The first line of input contains data of the nodes of the tree in level order form. The order is: data for root node, number of children to root node, data of each of child nodes and so on and so forth for each node. The data of the nodes of the tree is separated by space.

Output Format :

The first and only line of output contains data of the node with second largest data.

Constraints:

Time Limit: 1 sec

Sample Input 1 :

```
10 3 20 30 40 2 40 50 0 0 0 0
```

Sample Output 1 :

```
40
```

Replace with depth

Send Feedback

You are given a generic tree. You have to replace each node with its depth value. You just have to update the data of each node, there is no need to return or print anything.

Input format :

The first line of input contains data of the nodes of the tree in level order form. The order is: data for root node, number of children to root node, data of each of child nodes and so on and so forth for each node. The data of the nodes of the tree is separated by space.

Output format:

The updated tree is printed level wise. Each level is printed in a new line. Please refer to sample output 1 for more details.

Constraints:

Time Limit: 1 sec

Sample Input 1:

```
10 3 20 30 40 2 40 50 0 0 0 0
```

Sample Output 1:

```
0
1 1 1

2 2
```