

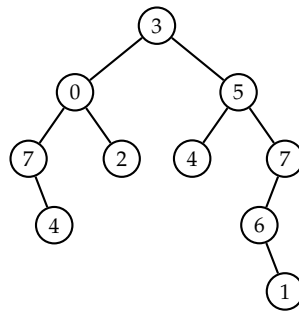
Recursive traversal of a tree

P57669_en

Write a program that reads the description of a natural binary tree and prints its postorder and inorder traversals.

In this exercise as well as in the rest of exercises of this section, if the contrary is not said the description of a tree consists of the number of nodes n followed by the traversal in preorder, which includes the leaves marked with a -1. This traversal has $2n + 1$ elements.

For instance, the tree



would be described as 10 3 0 7 -1 4 -1 -1 2 -1 -1 5 4 -1 -1 7 6 -1 1 -1 -1 -1

To solve this exercise as well as most of the exercises of this section, you will need to store the tree in a vector. Do it using this code (slightly modified if it is necessary):

```

struct Node {
    int value;
    int left , right ;
};

// Reads a tree and stores it in a piece of the vector v starting at the position j.
// Modifies the variable j in order to indicate the next free position of v.
// Returns the position inside of v of the root of the (sub)tree read (o -1).

int tree (int& j, vector<Node>& v) {
    int x;
    cin >> x;
    if (x == -1) return -1;

    int a = j;
    ++j;
    v[a].value = x;
    v[a].left = tree (j, v);
    v[a].right = tree (j, v);
    return a;
}

...

```

```

int main() {
    int n;
    cin >> n;
    vector<Node> v(n);
    int j = 0;
    int a = tree(j, v);
    ...
}

```

With the tree of the instance, the final content of *v* would be

position	0	1	2	3	4	5	6	7	8	9
<i>v.value</i>	3	0	7	4	2	5	4	7	6	1
<i>v.left</i>	1	2	-1	-1	-1	6	-1	8	-1	-1
<i>v.right</i>	5	4	3	-1	-1	7	-1	-1	9	-1

Notice that each position of the vector stores the value of a node, the position of its left child and its right child. Value -1 is used to indicate empty trees. Variable *a* of the main program is the position of the root of the tree, and is -1 if the tree is empty, and 0 if it is not.

Input

Input consists of the description of a natural binary tree.

Output

Your program must print two lines, with the postorder and inorder traversals of the tree. Each element must be preceded by a space.

Sample input

```

10
3 0 7 -1 4 -1 -1 2 -1 -1 5
4 -1 -1 7 6 -1 1 -1 -1 -1

```

Sample output

```

4 7 2 0 4 1 6 7 5 3
7 4 0 2 3 4 5 6 1 7

```

Problem information

Author : Salvador Roura

Translator : Carlos Molina

Generation : 2025-05-14 10:36:52

© Jutge.org, 2006–2025.

<https://jutge.org>