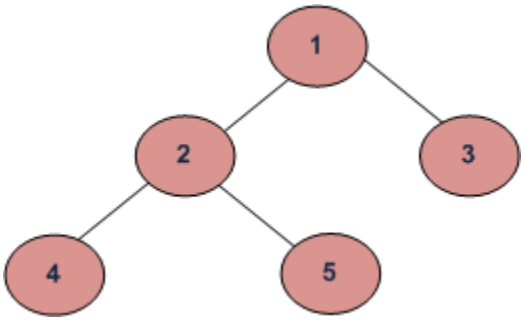
Write a program to Calculate Size of a tree | Recursion

Size of a tree is the number of elements present in the tree. Size of the below tree is 5.



Size() function recursively calculates the size of a tree. It works as follows:
Size of a tree = Size of left subtree + 1 + Size of right subtree.

Size of Binary Tree

size(tree)
1. If tree is empty then return 0
2. Else
    (a) Get the size of left subtree recursively i.e., call
        size(tree->left-subtree)
    (a) Get the size of right subtree recursively i.e., call
        size(tree->right-subtree)
    (c) Calculate size of the tree as following:
          tree_size = size(left-subtree) + size(right-
                        subtree) + 1
    (d) Return tree_size

```
// A recursive C++ program to
// calculate the size of the tree
#include <bits/stdc++.h>
using namespace std;

/* A binary tree node has data, pointer to left child
and a pointer to right child */
class node
{
    public:
    int data;
    node* left;
    node* right;
};

/* Helper function that allocates a new node with the
given data and NULL left and right pointers. */
node* newNode(int data)
{
    node* Node = new node();
    Node->data = data;
    Node->left = NULL;
    Node->right = NULL;

    return(Node);
}

/* Computes the number of nodes in a tree. */
int size(node* node)
{
    if (node == NULL)
        return 0;
    else
        return(size(node->left) + 1 + size(node->right));
}

/* Driver code*/
int main()
{
    node *root = newNode(1);
```

```
    root->left = newNode(2);
    root->right = newNode(3);
    root->left->left = newNode(4);
    root->left->right = newNode(5);

    cout << "Size of the tree is " << size(root);
    return 0;
}
```