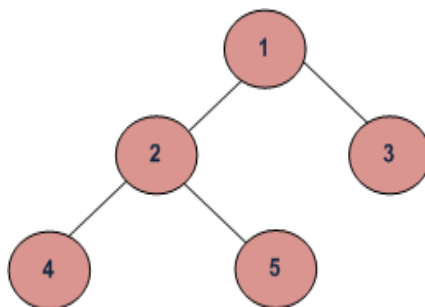


PROGRAM 1: IMPLEMENTATION OF RECURSIVE FUNCTION FOR TREE TRAVERSAL AND FIBONACCI

Aim: To write a program in C++ for implementation of recursive function for tree traversal and fibonacci.

Description *Tree traversal*:

Unlike linear data structures (Array, Linked List, Queues, Stacks, etc) which have only one logical way to traverse them, trees can be traversed in different ways. Following are the generally used ways for traversing trees.



- (a) Inorder (Left, Root, Right) : 4 2 5 1 3
- (b) Preorder (Root, Left, Right) : 1 2 4 5 3
- (c) Postorder (Left, Right, Root) : 4 5 2 3 1

Algorithm *Tree traversal (Recursive Method)*:

Algorithm Inorder(tree)

- Step 1. Traverse the left subtree, i.e., call Inorder(left-subtree)
- Step 2. Visit the root.
- Step 3. Traverse the right subtree, i.e., call Inorder(right-subtree)

Algorithm Preorder(tree)

Step 1. Visit the root.

Step 2. Traverse the left subtree, i.e., call Preorder(left-subtree)

Step 3. Traverse the right subtree, i.e., call Preorder(right-subtree)

Algorithm Postorder(tree)

Step 1. Traverse the left subtree, i.e., call Postorder(left-subtree)

Step 2. Traverse the right subtree, i.e., call Postorder(right-subtree)

Step 3. Visit the root.

Program *Tree traversal (Recursive Method)*:

// C++ program for different tree traversals with recursion

```
#include <iostream>
```

```
using namespace std;
```

```
/* A binary tree node has data, pointer to left child and a pointer to right child */
```

```
struct Node {
```

```
    int data;
```

```
    struct Node *left, *right;
```

```
    Node(int data)
```

```
    {
```

```
        this->data = data;
```

```
        left = right = NULL;
```

```
    }
```

```
};
```

```
/* Given a binary tree, print its nodes in postorder*/
```

```
void printPostorder(struct Node* node)
{
    if (node == NULL)
        return;
    printPostorder(node->left);
    printPostorder(node->right);
    cout << node->data << " ";
}
```

/* Given a binary tree, print its nodes in inorder*/

```
void printInorder(struct Node* node)
{
    if (node == NULL)
        return;
    printInorder(node->left);
    cout << node->data << " ";
    printInorder(node->right);
}
```

/* Given a binary tree, print its nodes in preorder*/

```
void printPreorder(struct Node* node)
{
    if (node == NULL)
        return;
    cout << node->data << " ";
    printPreorder(node->left);
    printPreorder(node->right);
}
```

```
}
```

```
int main()
```

```
{
```

```
    struct Node* root = new Node(1);
```

```
    root->left = new Node(2);
```

```
    root->right = new Node(3);
```

```
    root->left->left = new Node(4);
```

```
    root->left->right = new Node(5);
```

```
    cout << "\nPreorder traversal of binary tree is \n";
```

```
    printPreorder(root);
```

```
    cout << "\nInorder traversal of binary tree is \n";
```

```
    printInorder(root);
```

```
    cout << "\nPostorder traversal of binary tree is \n";
```

```
    printPostorder(root);
```

```
    return 0;
```

```
}
```

Output:

Preorder traversal of binary tree is

1 2 4 5 3

Inorder traversal of binary tree is

4 2 5 1 3

Postorder traversal of binary tree is

4 5 2 3 1

Description *Fibonacci Series (Recursive Method):*

The Fibonacci sequence is a series where the next term is the sum of pervious two terms. The first two terms of the Fibonacci sequence is 0 followed by 1.

The Fibonacci sequence: 0, 1, 1, 2, 3, 5, 8, 13, 21

Algorithm *Fibonacci Series (Recursive Method):*

Step 1: Start

Step 2: Define function fib(i)

Step 2.1: if i=0 return 0 or if i=1 & i=2 return 1 or return fib(i-1) + fib(i-2)

Step 3: Read n

Step 5: Call fib(n)

Step 6: Stop

Program *Fibonacci Series (Recursive Method):*

// C++ program for Fibonacci series using recursion

```
#include <iostream>
```

```
using namespace std;
```

```
int fib(int i)
```

```
{
```

```
    if (i == 0)
```

```
    {
```

```
        return 0;
```

```
    }
```

```
    else if ((i == 1) || (i == 2))
```

```
{  
    return 1;  
}  
else  
{  
    return (fib(i-1) + fib(i-2)); //here's our recursive call  
}  
}
```

```
int main()  
{  
    int n = 40;  
    cout << fib(n);  
    return 0;  
}
```

Output:

Enter the number of elements: 15

Fibonacci Series: 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377

Result:

Thus a program in C++ for implementation of recursive method for tree traversal and Fibonacci has been done successfully.

