## PROGRAM 2: IMPLEMENTATION OF ITERATION FUNCTION FOR TREE TRAVERSAL AND FIBONACCI

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

**Algorithm *Tree Traversal (iteration method) :***

Step 1: Start

Step 2: Create an empty stack S.

Step 3: Initialize current node as root

Step 4: Push the current node to S and set current = current->left until current is NULL

Step 5: If current is NULL and stack is not empty then

a) Pop the top item from stack.

b) Print the popped item, set current = popped item->right

c) Go to step 3.

Step 6: If current is NULL and stack is empty then we are done.

Step 9: Stop

**Program *Tree Traversal (iteration method) :***

```
// C++ program to print inorder traversal with iteration method
// using stack.
#include<bits/stdc++.h>
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;
    struct Node* right;
    Node (int data)
    {
        this->data = data;
        left = right = NULL;
    }
};

/* Iterative function for inorder tree traversal */
void inOrder(struct Node *root)
{
    stack<Node *> s;
    Node *curr = root;

    while (curr != NULL || s.empty() == false)
    {
        /* Reach the left most Node of the curr Node */
        while (curr !=  NULL)
```

```
        {
            /* place pointer to a tree node on the stack before
               Traversing the node's left subtree */
            s.push(curr);
            curr = curr->left;
        }

        /* Current must be NULL at this point */
        curr = s.top();
        s.pop();
         cout << curr->data << " ";
          /* we have visited the node and its left subtree. Now, it's
              Right subtree's turn */
        curr = curr->right;
    } /* end of while */
}
 /* Driver program to test above functions*/
int main()
{
    /* Constructed binary tree is
              1
            /   \
          2      3
        / \
      4    5   */
```

```
    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);
    inOrder(root);
     return 0;
}
```

**Output:**

4 2 5 1 3

**Algorithm F*ibonacci (iteration method):***

Step 1: Start

Step 2: Read number

Step 3: Initialize n1=0 n2=1

Step 4: Repeat Steps for i -> 2 to number

-> n3=n1+n2

-> Write n3

-> n1=n2

-> n2=n3

Step 7: Stop

**Program F*ibonacci (iteration method):***

**// C++ program to print Fibonacci series with iteration method**

```
#include<iostream>
int main()
```

```
{
 int  n1=0,n2=1,n3,i,number;
 cout<<"Enter the number of elements: ";
 cin>>number;
 cout<<n1<<" "<<n2;  //printing 0 and 1
 for(i=2;i<number;++i)//loop starts from 2 because 0 and 1 are already
                                printed
 {
  n3=n1+n2;
  cout<<" "<<n3;
  n1=n2;
  n2=n3;
 }
  return 0;
 }
```

Output:

**Enter the number of elements: 15**

**0 1 1 2 3 5 8 13 21 34 55 89 144 233 377**

**Result:**

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