

Practical no: 11

Objective: To implement a C program that creates a binary tree, and performs in-order, pre-order, and post-order traversals, displaying each traversal result to understand tree traversal techniques and their applications.

Program Codes: Following is the code of this problems in C:-

1. Practical11.c

```
#include <stdio.h>
#include <stdlib.h>

struct tnode {
    struct tnode* left;
    int data;
    struct tnode* right;
};

struct tnode* createBiTree() {
    int x;
    struct tnode* newNode = (struct tnode* ) malloc(sizeof(struct tnode));
    printf("\nEnter data: ");
    scanf("%d", & x);
    if (x == -1) {
        return NULL;
    }
    newNode → data = x;
    printf("Enter the left child of %d", x);
    newNode → left = createBiTree();
    printf("Enter the right child of %d", x);

    newNode → right = createBiTree();
    return newNode;
}

void inOrderTraversal(struct tnode* root) {
    if (root == NULL) {
        return;
    }
}
```

```

    }
    inOrderTraversal(root → left);
    printf("%d ", root → data);
    inOrderTraversal(root → right);
}

void preOrderTraversal(struct tnode* root) {
    if (root == NULL) {
        return;
    }
    printf("%d ", root → data);
    preOrderTraversal(root → left);
    preOrderTraversal(root → right);
}

void postOrderTraversal(struct tnode* root) {
    if (root == NULL) {
        return;
    }
    postOrderTraversal(root → left);
    postOrderTraversal(root → right);
    printf("%d ", root → data);
}

int main() {
    struct tnode * root = createBiTree();
    printf("\nThis is In-order traversal:\n");
    inOrderTraversal(root);
    printf("\nThis is Pre-order traversal:\n");
    preOrderTraversal(root);

    printf("\nThis is Post-order traversal:\n");
    postOrderTraversal(root);
    return 0;
}

```

Output: Following is the output of the program:-

```
C:\Users\DV yadav\Documents\Grad-Workdocs\Practicles\DSA>gcc Practical11.c && a.exe

Enter data: 1
Enter the left child of 1
Enter data: 2
Enter the left child of 2
Enter data: 3
Enter the left child of 3
Enter data: 4
Enter the left child of 4
Enter data: -1
Enter the right child of 4
Enter data: -1
Enter the right child of 3
Enter data: -1
Enter the right child of 2
Enter data: -1
Enter the right child of 1
Enter data: -1

This is In-order traversal:
4 3 2 1
This is Pre-order traversal:
1 2 3 4
This is Post-order traversal:
4 3 2 1
C:\Users\DV yadav\Documents\Grad-Workdocs\Practicles\DSA>
```