**Assignment-2**

**Name of Student: Ayush Sanjay Dhangar**

**Batch: 02 Class: SY\_IT-A Roll No: 42**

**PRN: 12210406**

**Subject: Advanced Data Structures**

**Problem Statement:** Write C or C++ program to implement recursive tree traversal (inorder, preorder and post order)

**Code:**

#include <stdio.h>

#include <stdlib.h>

struct binarytree

{

    struct binarytree \*left;

    int data;

    struct binarytree \*right;

};

struct binarytree \*root = NULL;

struct binarytree \*createTree()

{

    struct binarytree \*newnode;

    newnode = (struct binarytree \*)malloc(sizeof(struct binarytree));

    int data;

    char choice;

    printf("Enter data to add in the binary tree: ");

    scanf("%d", &data);

    newnode->data = data;

    getchar();

    printf("Do you want to insert left child to %d? Y/N: ", newnode->data);

    scanf("%c", &choice);

    if (choice == 'Y')

    {

        newnode->left = createTree();

    }

    else

    {

        newnode->left = NULL;

    }

    getchar();

    printf("Do you want to insert right child to %d? Y/N: ", newnode->data);

    scanf(" %c", &choice);

    if (choice == 'Y')

    {

        newnode->right = createTree();

    }

    else

    {

        newnode->right = NULL;

    }

    return newnode;

}

void inOrderTraversal(struct binarytree \*root)

{

    if (root != NULL)

    {

        inOrderTraversal(root->left);

        printf("%d ", root->data);

        inOrderTraversal(root->right);

    }

}

void preOrderTraversal(struct binarytree \*root)

{

    if (root != NULL)

    {

        printf("%d ", root->data);

        preOrderTraversal(root->left);

        preOrderTraversal(root->right);

    }

}

void postOrderTraversal(struct binarytree \*root)

{

    if (root != NULL)

    {

        postOrderTraversal(root->left);

        postOrderTraversal(root->right);

        printf("%d ", root->data);

    }

}

int main()

{

    int choice;

    while (1)

    {

        printf("1. Create Binary Tree\n");

        printf("2. Print Inorder Traversal of Binary Tree\n");

        printf("3. Print Preorder Traversal of Binary Tree\n");

        printf("4. Print Postorder Traversal of Binary Tree\n");

        printf("5. Exit\n");

        printf("Enter your choice: ");

        scanf("%d", &choice);

        switch (choice)

        {

        case 1:

            root = createTree();

            break;

        case 2:

            if (root == NULL)

            {

                printf("First create a tree\n");

            }

            else

            {

                printf("The inorder Traversal of the tree is: ");

                inOrderTraversal(root);

                printf("\n");

            }

            break;

        case 3:

            if (root == NULL)

            {

                printf("First create a tree\n");

            }

            else

            {

                printf("The Preorder traversal of the tree is: ");

                preOrderTraversal(root);

                printf("\n");

            }

            break;

        case 4:

            if (root == NULL)

            {

                printf("First create tree\n");

            }

            else

            {

                printf("The Postorder traversal of the tree is: ");

                postOrderTraversal(root);

                printf("\n");

            }

            break;

        case 5:

            printf("Exiting the program...\n");

            exit(0);

            break;

        default:

            printf("Invalid option, enter a valid choice\n");

            break;

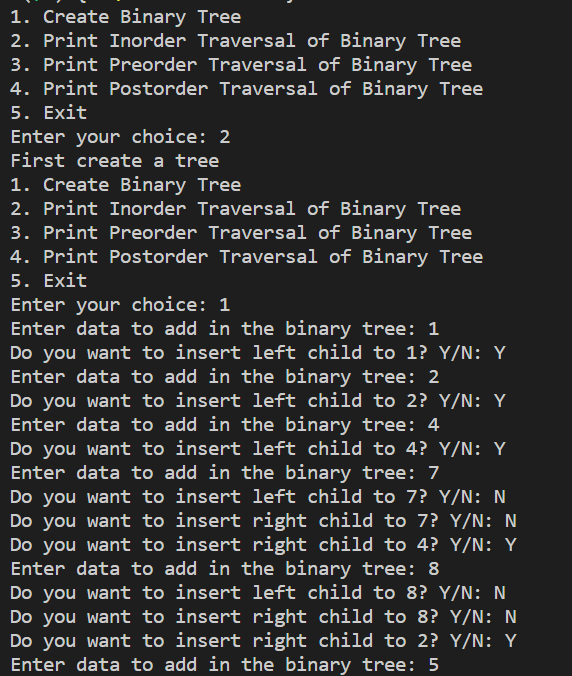
        }

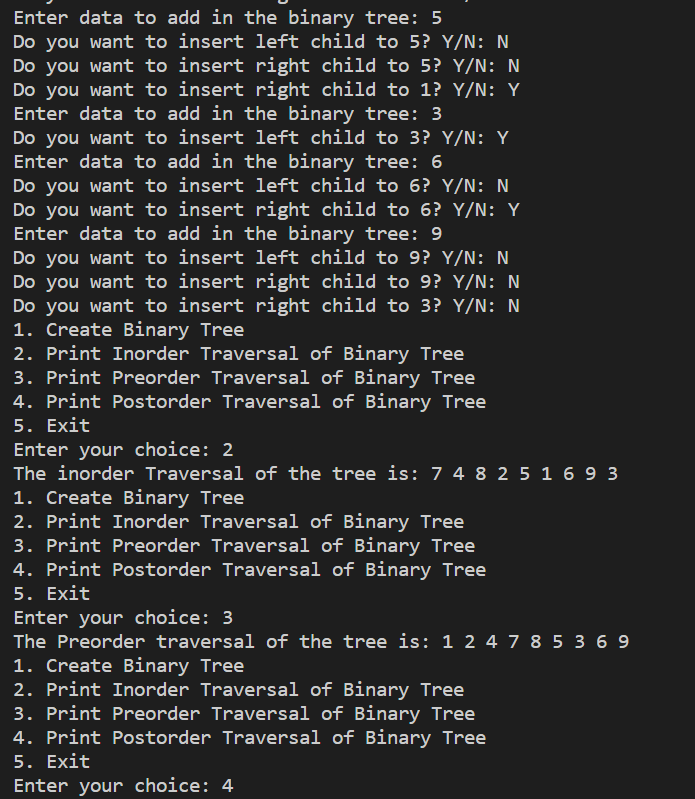
    }

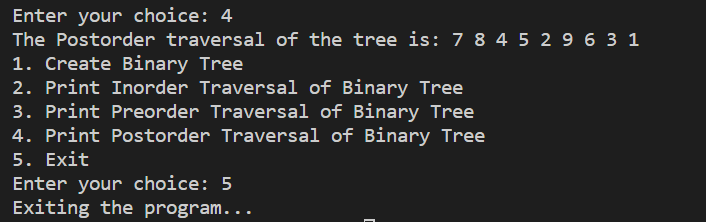
}

**Result:**

**Actual Output:**

****

****

****