Experiment no. :-6

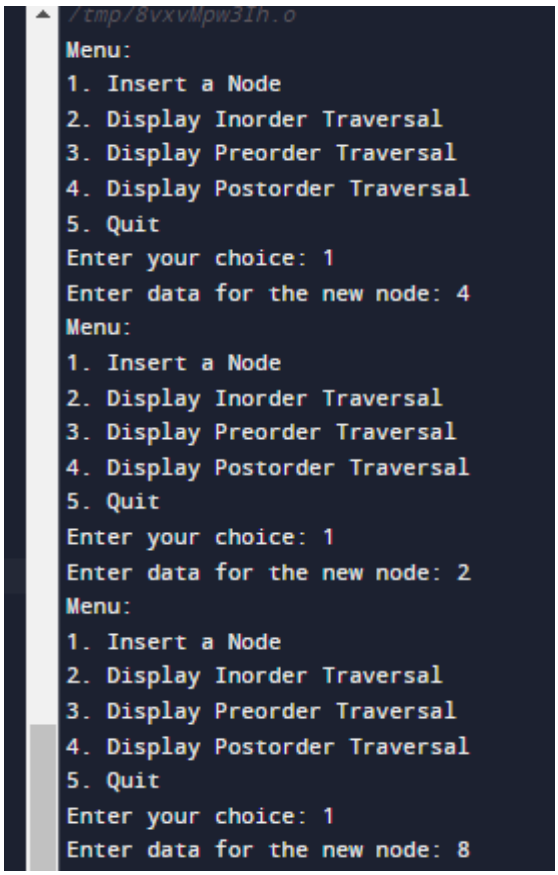
```
#include <stdio.h>
#include <stdlib.h>

// Structure to represent a binary tree node
struct TreeNode {
    int data;
    struct TreeNode* left;
    struct TreeNode* right;
};

// Function to create a new node
struct TreeNode* createNode(int data) {
    struct TreeNode* newNode = (struct TreeNode*)malloc(sizeof(struct TreeNode));
    newNode->data = data;
    newNode->left = newNode->right = NULL;
    return newNode;
}

// Function to insert a node into the binary tree
struct TreeNode* insertNode(struct TreeNode* root, int data) {
    if (root == NULL) {
        return createNode(data);
    }

    if (data < root->data) {
        root->left = insertNode(root->left, data);
    } else if (data > root->data) {
```

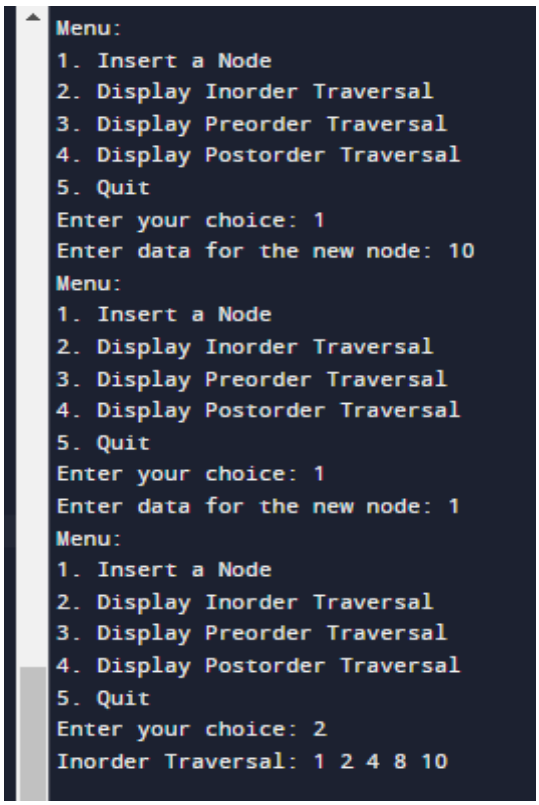
```
        root->right = insertNode(root->right, data);
    }

    return root;
}

// Function to perform an inorder traversal of the binary tree
void inorderTraversal(struct TreeNode* root) {
    if (root == NULL) {
        return;
    }
    inorderTraversal(root->left);
    printf("%d ", root->data);
    inorderTraversal(root->right);
}

// Function to perform a preorder traversal of the binary tree
void preorderTraversal(struct TreeNode* root) {
    if (root == NULL) {
        return;
    }
    printf("%d ", root->data);
    preorderTraversal(root->left);
    preorderTraversal(root->right);
}

// Function to perform a postorder traversal of the binary tree
void postorderTraversal(struct TreeNode* root) {
    if (root == NULL) {
```

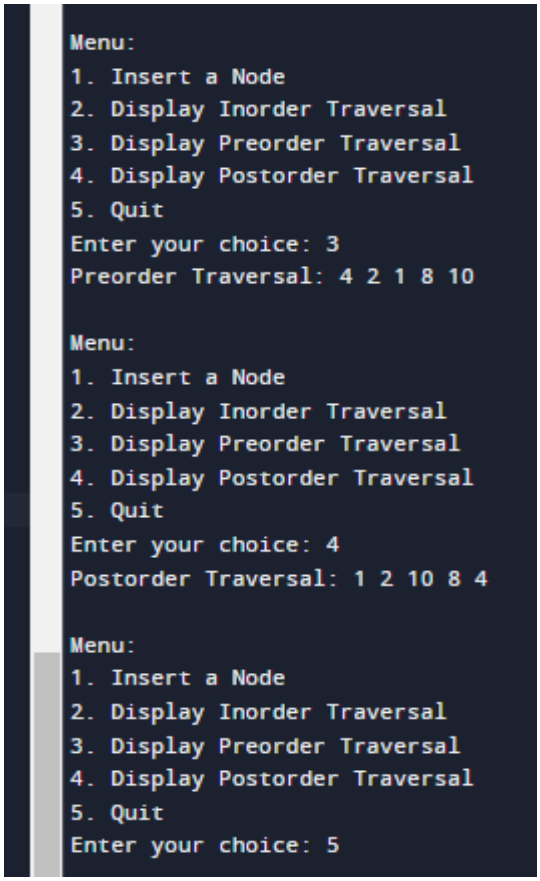
```
        return;
    }
    postorderTraversal(root->left);
    postorderTraversal(root->right);
    printf("%d ", root->data);
}

int main() {
    struct TreeNode* root = NULL;
    int choice, data;

    while (1) {
        printf("\nMenu:\n");
        printf("1. Insert a Node\n");
        printf("2. Display Inorder Traversal\n");
        printf("3. Display Preorder Traversal\n");
        printf("4. Display Postorder Traversal\n");
        printf("5. Quit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter data for the new node: ");
                scanf("%d", &data);
                root = insertNode(root, data);
                break;
            case 2:
                printf("Inorder Traversal: ");
```

```
            inorderTraversal(root);
            printf("\n");
            break;
        case 3:
            printf("Preorder Traversal: ");
            preorderTraversal(root);
            printf("\n");
            break;
        case 4:
            printf("Postorder Traversal: ");
            postorderTraversal(root);
            printf("\n");
            break;
        case 5:
            // Free memory and exit
            free(root);
            exit(0);
        default:
            printf("Invalid choice. Please try again.\n");
      }
   }

   return 0;
}
```

Output :-


/tmp/8vxvMpw31h.o
Menu:
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Quit
Enter your choice: 1
Enter data for the new node: 4
Menu:
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Quit
Enter your choice: 1
Enter data for the new node: 2
Menu:
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Quit
Enter your choice: 1
Enter data for the new node: 8


Menu:
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Quit
Enter your choice: 1
Enter data for the new node: 10
Menu:
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Quit
Enter your choice: 1
Enter data for the new node: 1
Menu:
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Quit
Enter your choice: 2
Inorder Traversal: 1 2 4 8 10


Menu:
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Quit
Enter your choice: 3
Preorder Traversal: 4 2 1 8 10
Menu:
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Quit
Enter your choice: 4
Postorder Traversal: 1 2 10 8 4
Menu:
1. Insert a Node
2. Display Inorder Traversal
3. Display Preorder Traversal
4. Display Postorder Traversal
5. Quit
Enter your choice: 5