Program - 14

Title : Write a Program to Perform the Tree Traversal Technique

Code:

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

// Define the tree node structure
struct node {
   int data;
   struct node* left;
   struct node* right;
};

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

// Function to insert a value into the tree
struct node* insert(struct node* root, int val) {
   if (root == NULL) {
      return createNode(val);
   } else if (val < root->data) {
      root->left = insert(root->left, val);
   } else {
      root->right = insert(root->right, val);
   }
   return root;
}

// Function to perform in-order traversal
void inOrder(struct node* root) {
   if (root != NULL) {
      inOrder(root->left);
      printf("%d ", root->data);
      inOrder(root->right);
   }
}

// Function to perform pre-order traversal
void preOrder(struct node* root) {
   if (root != NULL) {
      printf("%d ", root->data);
      preOrder(root->left);
      preOrder(root->right);
   }
}

// Function to perform post-order traversal
void postOrder(struct node* root) {
   if (root != NULL) {
      postOrder(root->left);
```

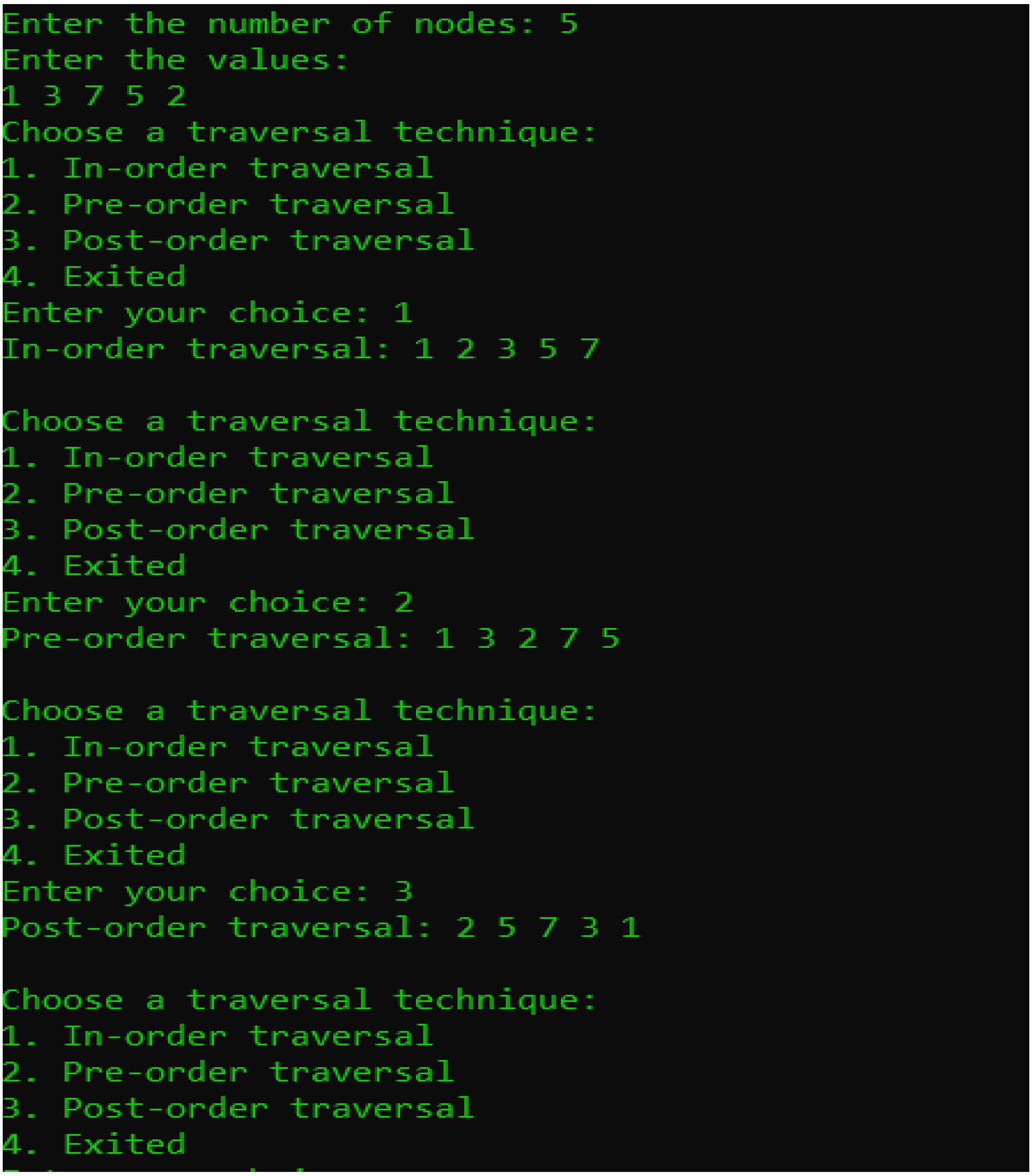
```
        postOrder(root->right);
        printf("%d ", root->data);
    }
}

// Main function to test the tree traversal program
int main() {
    struct node* root = NULL;
    int n, val;
    printf("Enter the number of nodes: ");
    scanf("%d", &n);
    printf("Enter the values:\n");
    for (int i = 0; i < n; i++) {
        scanf("%d", &val);
        root = insert(root, val);
    }
    int choice=0;
    while(choice<4){
    printf("Choose a traversal technique:\n");
    printf("1. In-order traversal\n");
    printf("2. Pre-order traversal\n");
    printf("3. Post-order traversal\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);
    switch (choice) {
        case 1:
            printf("In-order traversal: ");
            inOrder(root);
            break;
        case 2:
            printf("Pre-order traversal: ");
            preOrder(root);
            break;
        case 3:
            printf("Post-order traversal: ");
            postOrder(root);
            break;
        case 4:
            printf("Program Exited ");

            break;
        default:
            printf("Invalid choice.\n");
            break;
    }
    printf("\n\n");
    }

    return 0;
}
```

Output:



```
Enter the number of nodes: 5
Enter the values:
1 3 7 5 2
Choose a traversal technique:
1. In-order traversal
2. Pre-order traversal
3. Post-order traversal
4. Exited
Enter your choice: 1
In-order traversal: 1 2 3 5 7

Choose a traversal technique:
1. In-order traversal
2. Pre-order traversal
3. Post-order traversal
4. Exited
Enter your choice: 2
Pre-order traversal: 1 3 2 7 5

Choose a traversal technique:
1. In-order traversal
2. Pre-order traversal
3. Post-order traversal
4. Exited
Enter your choice: 3
Post-order traversal: 2 5 7 3 1

Choose a traversal technique:
1. In-order traversal
2. Pre-order traversal
3. Post-order traversal
4. Exited
```

Date : ___/____/______

Teacher Sign ……………………………………………………………