Lab prg 8

Write a program

a) To construct binary Search tree.

b) To traverse the tree using all the methods i.e, inorder, preorder and postorder

c) To display the elements in the tree.

node:

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

struct Node {
    int data;
    struct Node *left, *right;
};

struct Node * createNode (int value)
{
    struct Node *newNode = (struct Node*) malloc(sizeof
    newNode->data = value;                    (struct Nod));
    newNode->left = newNode->right = NULL;
    return newNode;
}
struct Node *insert (struct Node *root, int value)
        if (root == NULL)
            return createNode (value);

        if (value < root->data)
            root->left = insert (root->left, value);
        else if (value > root->data)
            root->right = insert (root->right, value);

        return root;
```

```
oid inorder (struct Node * root)
{
    if (root == NULL) return;
    inorder (root -> left );
    printf ("%d", root -> data);
    inorder (root -> right );
}
void preorder (struct Node *root)
{
    if (root == NULL) return;
    printf ("%d", root -> data);
    preorder (root -> left );
    preorder (root -> right);

Void postorder (struct Node * root)
{
    if (root == NULL ) return ;
    post order (root -> left);
    post order (root -> right );
    printf ("%d", root -> data);
}
void display (struct Node * root)
{
    printf ("BST Elements (Inorder): ");
    morder (root );
    printf ("\n");
}
int main ()
{
    struct Node *root = NULL;
    int choice, value ;
    while (1) {
```

```
printf("\n--- Binary Search Tree Menu ---");
printf("1. Insert into BST \n");
printf("2. Inorder Traversal \n");
printf("3. Preorder Traversal \n");
printf("4. Postorder Traversal \n");
printf("5. Display BST \n");
printf("6. Exit \n");
printf("Enter choice: ");
scanf("%d", &choice);
Switch (choice)
{
case 1:
printf("Enter value to insert: ");
scanf("%d", &value);
root = insert (root, value);
  break;

  case 2:
      printf("Inorder Traversal :");
      preorder (root);
      printf("\n");
      break;

  case 3:
    printf("Preorder Traversal :");
    preorder (root);
    printf("\n");
    break;

case 4:
    printf("Postorder Traversal:");
    postorder (root);
    printf("\n");
    break;
```

```
case 5:
    display(root);
    break;
case 6:
    exit(0);
default:
    printf("Invalid choice! Try again.\n");
    }
  }
  return 0;
}
```

Output:

```
---Binary Search Tree Menu---
1. Insert into BST.
2. Inorder Traversal.
3. Preorder Traversal.
4. Postorder Traversal.
5. Display BST
6. Exit
Enter choice: 1
Enter value to insert: 50

---Binary Search Tree Menu---
1. Insert into BST.
2. Inorder Traversal.
3. Preorder Traversal.
4. Postorder Traversal.
5. Display BST
6. Exit.
Enter choice: 1
Enter value to insert: 30
```

--- Binary Search Tree Menu ---
1. Insert into BST
2. Inorder Traversal
3. Preorder Traversal
4 Postorder Traversal
5. Display BST
6. Exit
Enter choice : 1
Enter value to insert : 70.

--- Binary Search Tree Menu ---
1. Insert into BST
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Display BST
6. Exit
Enter choice : 2
: 30 50 70.
Inorder Traversal

--- Binary Search Tree Menu ---
1. Insert into BST
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Display BST
6. Exit
Enter choice : 3
Preorder Traversal : 50 30 70

--- Binary Search Tree Menu ---
1. Insert into BST
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Display BST
6. Exit
Enter choice : 4
Post Traversal : 30 70 50.

---Binary Search Tree Menu---

1. Insert into BST
2. Inorder Traversal
3. Preorder Traversal
4. Postorder Traversal
5. Display BST
6. Exit

Enter choice: 6

MG
12/11/25.