8/10/25

8) write a program:

a) To construct a binary search tree

b) To traverse the tree using all the methods i.e., inorder, preorder & post order

c) To display the elements in the tree.

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

struct node{
        int data;
        struct node· *left, *right;

};
struct node *createnode(int value){
      struct node *newnode=(structnode*)malloc
                        (sizeof(structnode));
       newnode->data=value;
       newnode->left=newnode->right=NULL;
       return newnode;
 }
struct node *insert(struct node *node, 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;
}

void 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->left;
}
void postorder(struct node *root)
        if(root==NULL){
               return;
        }
        Postorder(root->left);
        Postorder(root->right;
         Printf("%d",root->data);
}
void display(structnode *root){
        PrintfC"BST Elements (Inorder): ");
        inorder(root);
        Printf("\n");
}
int main(){
      struct node *root=NULL;
      int choice ,value;
      while(1){
          Printf("\n--Binary search Tree menu--\n");
          Printf("1)Insert intoBST \n");
          Printf("2).~~Inser~~order Traversal\n");
          Printf("3.Preorder Traversal\n");
          Printf("4.postorder Traversal\n");
          Printf("5.Display BST \n");
Printf("6.
Printf("6. exit\n");
Printf("Enter choice: ");
scanf("%d", &choice);
switch(choice){
    case1:
        Printf("Enter value to insert: ");
        scanf("%d", &value);
        root = insert(root, value);
        break;
  case2:
      Printf("Inorder Traversal : ");
      inorder(root);
      printf("\n");
      break;
 case3:
     Printf("Preorder Transversal: ");
     Preorder(root);
     Printf("\n");
     break;
case4:
    Printf("Post order Transversal :");
    postorder(root);
    Printf("\n");
    break;
case5:
  display(root);
   break;
case6:
  exit(0);
default:
    Printf("Invalid choice! Try again\n");
} }
}return 0;
}
```

O/P:-

```
-- 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: 20

Enter choice 1
enter value to insert: 10

enter choice 1
Enter value to insert: 30

Enterchoice 2
Inorder Transversal : 10 20 30

Enter choice: 3
Preorder Traversal : 20 10 30

Enter choice: 4
Post order Traversal : 10 30 20

Enter choice: 5
BST Elements (Inorder) : 10 20 30
Enter choice: 6
```

8/1/25