8) Write a program

a) To construct a binary Search tree.

b) To traverse the tree using all the methods i.e., in-order, preorder and post order

c) To display the elements in the tree.

Shashank Patel C J

1BM22CS255

```
#include<stdio.h>
#include<stdlib.h>
struct node{
int data;
struct node* left_side;
struct node* right_side
};
struct node* newnode(int x)
{
   struct node* temp=malloc(sizeof(struct node));
   temp->data=x;
   temp->left_side=NULL;
   temp->right_side=NULL;
   return temp;
}
struct node* insert(struct node* root,int x)
{
```

```
    if(root==NULL)
    {
        return newnode(x);
    }
    else if(x>root->data)
    {
        root->right_side=insert(root->right_side,x);
    }
    else
    {
         root->left_side=insert(root->left_side,x);
    }
    return root;

}
void inorder(struct node* root)
{
    if(root!=NULL)
    {
        inorder(root->left_side);
        printf("%d\n",root->data);
        inorder(root->right_side);
    }
```

```
}
void postorder(struct node* root)
{
   if(root!=NULL)
   {
      postorder(root->left_side);
      postorder(root->right_side);
      printf("%d\n",root->data);
   }
}
void preorder(struct node* root)
{
   if(root!=NULL)
   {
      printf("%d\n",root->data);
      preorder(root->left_side);
      preorder(root->right_side);
   }
}
void main()
{
   struct node* root=NULL;
   root=insert(root,15);
```

```
    root=insert(root,7);
    root=insert(root,50);
    printf("inorder traversal:\n");
    inorder(root);
    printf("preorder traversal:\n");
    preorder(root);
    printf("postorder traversal:\n");
    postorder(root);
}
```

Output:

```
inorder traversal:
7
15
50
preorder traversal:
15
7
50
postorder traversal:
7
50
15

Process returned 3 (0x3)   execution time : 0.031 s
Press any key to continue.
```