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.

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

```
    return root;
}
void inorder(struct node *root){
    if(root!=NULL){
        inorder(root->left);
        printf("%d\t",root->data);
        inorder(root->right);
    }
}
void preorder(struct node *root){
    if(root!=NULL){
        printf("%d\t",root->data);
        preorder(root->left);
        preorder(root->right);
    }
}
void postorder(struct node *root){
    if(root!=NULL){
        postorder(root->left);
        postorder(root->right);
        printf("%d\t",root->data);
    }
}
int main()
{
    struct node *root=NULL;
```

```
    root=insert(root,10);
    insert(root,20);
    insert(root,30);
    insert(root,150);
    insert(root,100);
    insert(root,300);

    printf("Inorder traversal:\n");
    inorder(root);
    printf("\n");

    printf("preorder traversal:\n");
    preorder(root);
    printf("\n");

    printf("postorder traversal:\n");
    postorder(root);
}
```

Output:

Output

```
/tmp/l1RWKwiZka.o
Inorder traversal:
10  20  30  100 150 300
preorder traversal:
10  20  30  150 100 300
postorder traversal:
100 300 150 30  20  10
```