1.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 *left;
   struct node *right;
};
typedef struct node *node;
node newNode(int val)
{
   node temp = (node)malloc(sizeof(struct node));
   temp->data = val;
   temp->left = NULL;
   temp->right = NULL;
   return temp;
}
node insertbst(node head, int val)
{
   if (head == NULL)
   {
```

```
        return newNode(val);
    }
    if (val < head->data)
    {
        head->left = insertbst(head->left, val);
    }
    else
    {
        head->right = insertbst(head->right, val);
    }
}

void preorder(node head)
{
    if (head != NULL)
    {
        printf("%d \t", head->data);
        preorder(head->left);
        preorder(head->right);
    }
}
void postorder(node head){
    if (head != NULL){
        postorder(head->left);
        postorder(head->right);
        printf("%d \t",head->data);
```

```
   }
}
void inorder(node head ){
   if(head != NULL){
      inorder(head->left);
      printf("%d \t",head->data);
      inorder(head->right);
   }
}

void main()
{
   node head = NULL;
   while (1)
   {
      printf("1: insert new element \t 2: display preorder 3: display inorder \t 4: display
postorder \t 6: exit\n");
      int ch;
      scanf("%d",&ch);
      switch(ch){
         case 1:
            printf("Enter the value to be inserted: \n");
            int a;
            scanf("%d",&a);
            head = insertbst(head,a);
            break;
```

```
        case 2:
           printf("The elements are : \n");
           preorder(head);
           printf("\n");
           break;
        case 3:
           printf("The elements are : \n");
           inorder(head);
           printf("\n");
           break;
        case 4:
           printf("The elements are : \n");
           postorder(head);
           printf("\n");
           break;
        case 5:
           exit(0);
     }
  }
}
```

output:

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:1

Enter the value to be inserted:

21

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6:

exit

enter the choice:1

Enter the value to be inserted:

22

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:1

Enter the value to be inserted:

23

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:1

Enter the value to be inserted:

24

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:1

Enter the value to be inserted:

25

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:1

Enter the value to be inserted:

26

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:1

Enter the value to be inserted:

27

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:1

Enter the value to be inserted:

28

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:1

Enter the value to be inserted:

29

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:2

The elements are :

21 22 23 24 25 26 27 28 29

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:3

The elements are :

21 22 23 24 25 26 27 28 29

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:4

The elements are :

29 28 27 26 25 24 23 22 21

1: insert new element 2: display preorder 3: display inorder 4: display postorder 6: exit

enter the choice:5

Process returned 0 (0x0) execution time : 26.289 s

Press any key to continue.