

LAB-10

Write a program to a) to construct a binary search tree b) to traverse the tree using all the methods i.e. in order, preorder, post order c) to display the elements in the tree

```
# void insert()
{
```

```
    int data;
    printf("Enter data to be inserted");
    scanf("%d", &data);
    struct node * temp = (struct node *) malloc (1 * sizeof (struct node));
    temp->value = data;
    temp->left = temp->right = NULL;
    if (root == NULL)
        root = temp;
    else
        search (root);
}
```

```
void search (struct node * t)
{
```

```
    if (temp->value > t->value) && (t->right != NULL)
        search (t->right);
    else if (temp->value > t->value) && (t->right == NULL)
        t->right = temp;
    else if (temp->value < t->value) && (t->left != NULL)
        search (t->left);
    else if (temp->value < t->value) && (t->left == NULL)
        t->left = temp;
}
```



```
void inorder (struct tnode *t)
{
    if (root == NULL)
    {
        printf ("No elements in the tree \n");
        return;
    }
    if (t->left != NULL)
        inorder (t->left);
    printf ("%d \t", t->value);
    if (t->right != NULL)
        inorder (t->right);
}
```

```
void preorder (struct tnode *t)
{
    if (root == NULL)
    {
        printf ("No elements in the tree \n");
        return;
    }
    printf ("%d \t", t->value);
    if (t->left != NULL)
        preorder (t->left);
    if (t->right != NULL)
        preorder (t->right);
}
```

```
void postorder (struct tnode *t)
{
    if (root == NULL)
    {
        printf ("No elements in the tree \n");
        return;
    }
}
```

3

```

if (t->left != NULL)
    postorder (t->left);
if (t->right != NULL)
    postorder (t->right);
printf ("%d -> ", t->value);
    
```

3