```
void insert()
{
    creak ();
    if (root == NULL)
        root = temp;
    else
        search (root);
}


void create()
{
    int data;
    printf ("Enter data of node to be inserted : ");
    scanf (" %d", &data);
    temp= (struct bt node *) malloc ( size of (struct bt node));
    temp -> value = data
    temp ->l = temp->r = NULL


Void search (struct btnode *t)
{
    if ((temp ->value >t->value) && (t->r != NULL))
        search (t ->r);
    else if ((temp -> value > t->value) && (t->r == NULL))
        t ->r = temp;
    else if ((temp -> value < t->value) && (t->l != NULL))
        search (t->l);
    else if ((temp->value < t->value) && (t->l == NULL)
        t->l = temp;
}
```

IBMA(S)11
SWAROOP. S. JADHAV.

```
void inorder (struct bt node *t)


    it (root == NULL)
    {
        print ("No element in a tree to display");
        return ;
    }
    it (t->L != NULL)
        inorder (t->L);
    print (".1.d -> " , t->value);
    it (t->r != NULL)
        inorder (t->r);



void preorde (struct btnode *t)
{
    it (root = =NULL)
    {
        print (" Tree is empty ");
        return ;
    }
    print (".1.d -> ", t-> value);
    it (t->l != NULL)
        preorder (t->L);
    it (t->r != NULL)
        preorder (t-> r);
}
```

IBMI915167

SWAROOP. S. JADHAV.

```
void  postorder (struct btnode *t)
{
    it (root ==NULL)
    {
        printf (" Tree is empty ");
        return;
    }
    id (t-> l!= NULL)
        postorder (t->l);
    it (t->r != NULL)
        postorder (t->r);
    printf (".%d ->", t-> value );
}
```