Topic                          Unit No.

## Counting Number of Node

```
int count_Node ( Node *P)
{
    int n = 0;
    if (P == NULL)
        return 0;
    else
    {
        n = 1 + count_Node (p->left)
              + count_Node (p->Right);
        return n;
    }
}
```

Sanjivani Rural Education Society's

# SANJIVANI COLLEGE OF ENGINEERING, KOPARGAON
### (An Autonomous Institute Affiliated to SPPU, Pune)
## DEPARTMENT OF COMPUTER ENGINEERING

Topic                                                                Unit No.

## Height of Binary Tree

```
int Height( BT *p)
{
    if (p== NULL)
        return 0;

int Lht=Height (p→left);
int Rht= Height ( p→right);

    if (Lht > Rht)
    {
        ht= Lht +1;
    }
    else
    {
        ht= Rht+1;
    }
    return ht;

}
```

Height of BT = 3   or 4

Sanjivani Rural Education Society's
# SANJIVANI COLLEGE OF ENGINEERING, KOPARGAON
## (An Autonomous Institute Affiliated to SPPU, Pune)
### DEPARTMENT OF COMPUTER ENGINEERING
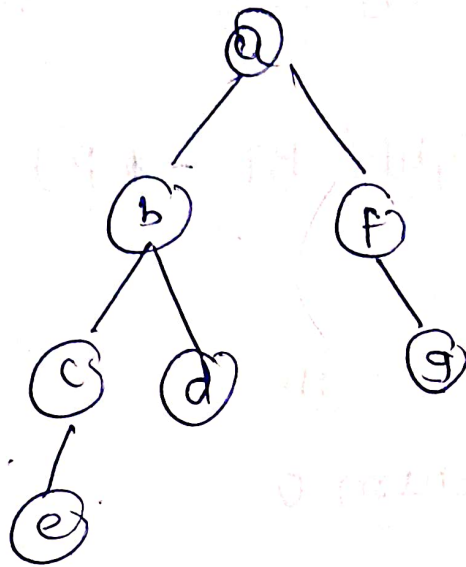
Topic                                    Unit No.

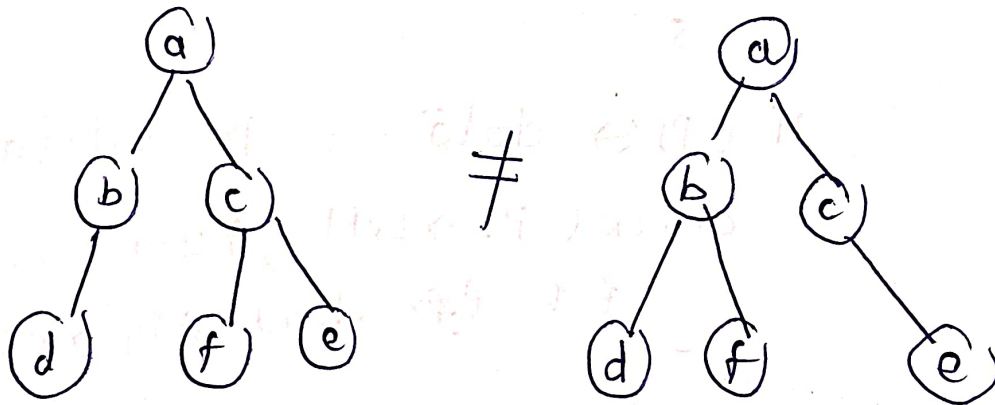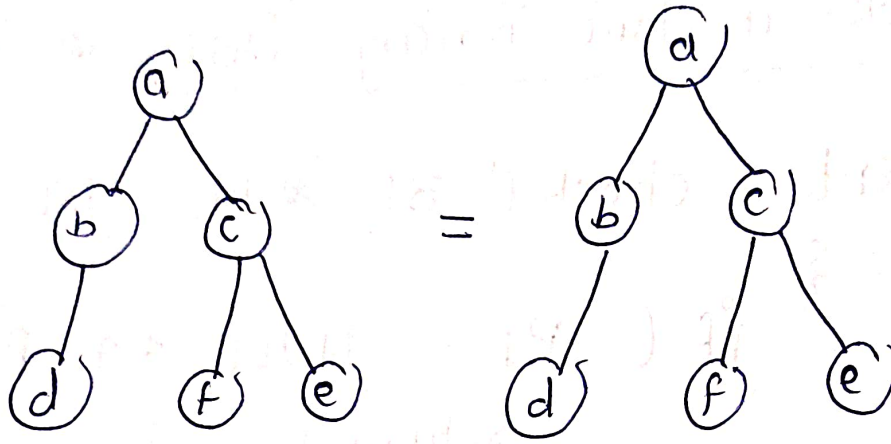## Check if two Binary Trees are Identical ✓

```
int check ( BT * P1, BT * P2)
{
    if ( P1 == NULL && P2 == NULL)
        return 1;
    if ( P1 != NULL && P2 != NULL)
    {
        if ( P1 -> data == P2 -> data &&
            check ( P1 -> Left, P2 -> Left)
            && check ( P1 -> right, P2 -> right))
        {
            return 1;
        }
    }
    else return 0;
}
```

**Topic**

Sanjivani Rural Education Society's
# SANJIVANI COLLEGE OF ENGINEERING, KOPARGAON
(An Autonomous Institute Affiliated to SPPU, Pune)
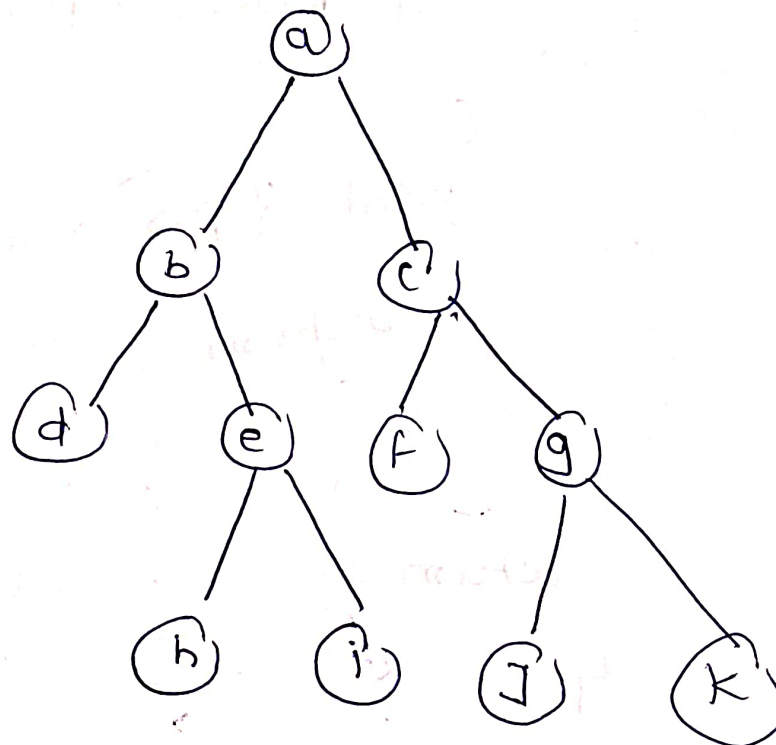## DEPARTMENT OF COMPUTER ENGINEERING

Topic

Unit No.

## Print All Ancestors of a node in BT



```
int   Traverse ( BT * P,  BT * Target)
  {
      if ( P != NULL)
      {
      if ( P→data == target→data)
          {  return 1;
       }
  }
```

Sanjivani Rural Education Society's
## SANJIVANI COLLEGE OF ENGINEERING, KOPARGAON
(An Autonomous Institute Affiliated to SPPU, Pune)
### DEPARTMENT OF COMPUTER ENGINEERING

Topic                                                        Unit No.
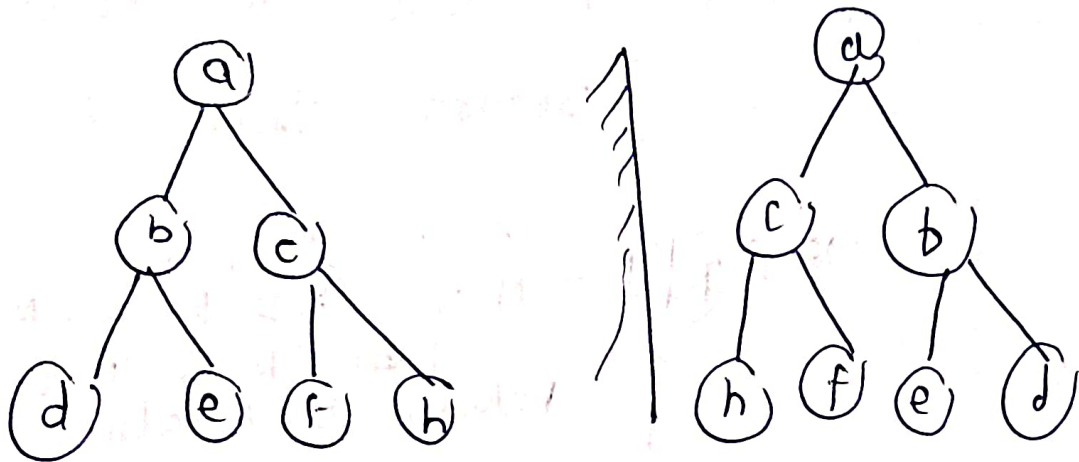
```
if ( Traverse( P → left, target)

        || Traverse (P → right, Target))

    {
      print (P → data)

        return 1;

    }

    return 0;
}
```

```
if (check- left || cheek right)
    print the node & return 1.
```

Topic                                                      Unit No.
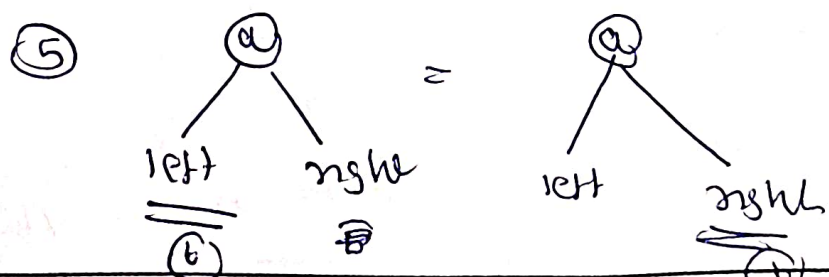
## Check The Tree is mirror of Binary Tree



1.    NULL  =  NULL

2.    (a)   x   NULL

3.    NULL  x  (a)

4.    (a)   x  (b)

5.    

**Topic**                                              **Unit No.**

```
int    isMirror ( Node *R1, Node *R2
{
   if ( R1 == NULL  && R2 == NULL)
   {
      return true; // 1
   }

   if ( (R1 != = NULL || R2 == NULL)
      ||  (R2 != = NULL || (R1 ==NULL
   {  return  0 // False
   }

   if ( R1 → data == R2 → data)
   {

      if (isMirror ( r1 → left , r2 → right)
         && isMirror ( r1 → right , r2 → left))
      {
         return True; // 1
      }
   }
   return  0 // false.
}
```

**Topic**                                                      **Unit No.**

## Convert Binary Tree to Its Mirror Image

```
void mirror ( node *root)
    {
      if ( root != NULL)
        {
      mirror ( root→ Left);
      mirror ( root→ Right);
      temp = root→ Left;
      root→ Left = root→ right;
      root→ right = temp;
        }
      return;
    }
```

**Prepared by :**                                              **Page No.**

**Topic**                                                           **Unit No.**