Kudaramuni G.M
1BM18CS085.

LAB-4

```
Insert (Node * node, int data)
  if( node== = NULL
  return newNode (data)
  else if data > node -> data
node -> lchild = insert (node -> lchid, data)
  else if data > node -> data
   node -> rchild = insert (node -> rchild, data)
node -> height = 1 + max(height(node -> lchid), heigh(
                    node -> rchild));
  balance = getbalance(node)
  if (balance <-1 && data > node -> rchild -> data)
  node = (node) # left rotate
  if (balance > 1 && data < node -> lchild -> data)
    node = right rotate (node)
    if (balance > 1 && data > node -> lchild -> data)
      node -> lchild = leftrotate (node -> lchild)
      node = left Rotate (node)
      return node
Delete (node* root, int data
      if (root = = Null)
        return root
    if(data < root -> data)
     root -> lchid = Delete (root -> lchild, data)
     else if (data > root -> data)
   root -> rchild = Delete (root -> rchild, data)
  else
{   if root -> lchild is Null or root rchild is null
    {
```

01

1BM18CS085.

```
{
    temp = root->left child ? root->lchild : root->rchild.
     if temp is Null
        temp = root
        root = Null
       else
        root = temp
        free(temp)}
    {else
      temp = minvaluenode(root->rchild)
     root->data = temp->data
      root->rchild = Delete (root->rchild, temp->data)
    }}
     if root is null
        return root
root->height = 1 + max(height(root->rchild), height(root->lchild
    balance = Get Balance(root);
if(balance > 1 && getbalance(root->lchild) < 0)
   root->lchild = leftrotate (root->lchild)
   root = rightRotate (root
if(balance > 1 && getbalance(root->lchild >= 0)
   root = rightrotate (root)
if (balance < -1 && getbalance(root->rchild) <= 0)
     root = leftrotate (root);
if (balance < -1 && get balance (root->rchild) > 0)
   root->rchild = right Rotate(root->rchild)
    root = left Rotate (root)
return root.
```

02

End.