

Tree Traversals:

Traversing means visiting each node only once. Tree traversal is a method for visiting all nodes in the tree only once. There are three types of traversals -

- ① Inorder (LDR)
 - ② Preorder (DLR)
 - ③ Post order (LRD)
- L → Left
D → Root
R → Right.

Inorder Traversal

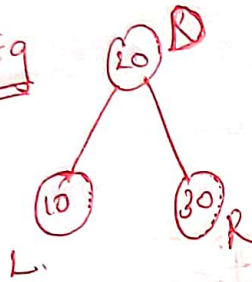
* Inorder traversal of binary tree is performed as traverse the left subtree in order.

* Visit the root node

* Traverse the right subtree in order

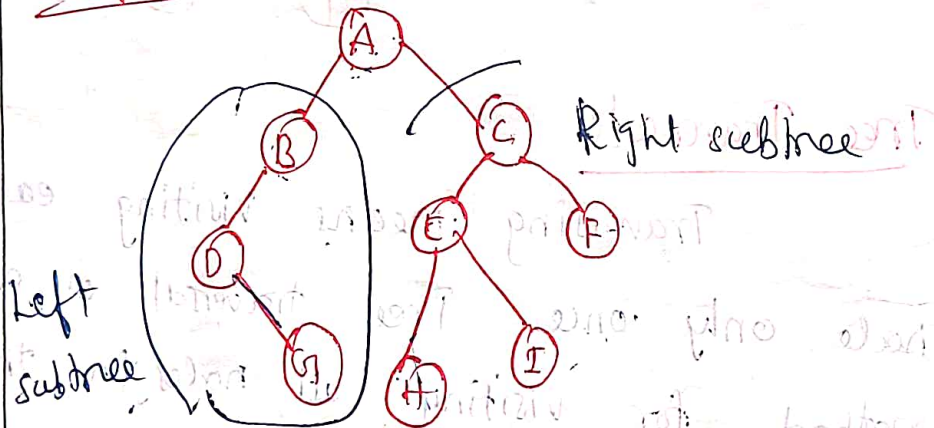
LDR

Eg



10 20 30

Example 2



① Traverse the left subtree in order

D G B A H E I C F

Routine

```
Void Inorder (Tree T)
{
  if (T == NULL)
  {
    Inorder (T->left);
    printf (T->element);
    Inorder (T->right);
  }
}
```

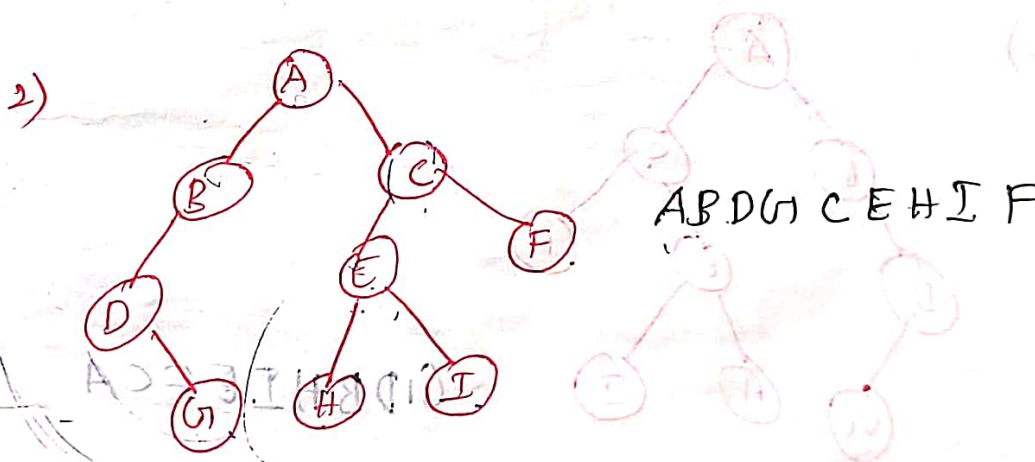
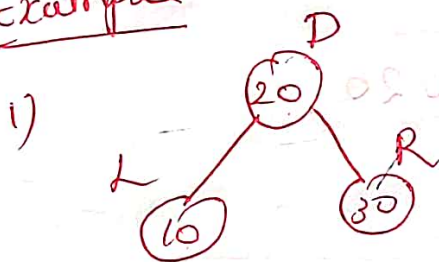
Preorder:- (DLR)

Preorder traversal of binary tree

is performed as

- * Visit the root node
- * Traverse the left sub tree in preorder
- * Traverse the right sub tree in preorder.

Example



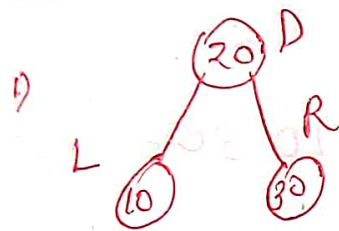
Routine

```
void Preorder (T)
{
    if (T != NULL)
    {
        printf ("Element ");
        Preorder (T->left);
        Preorder (T->right);
    }
}
```

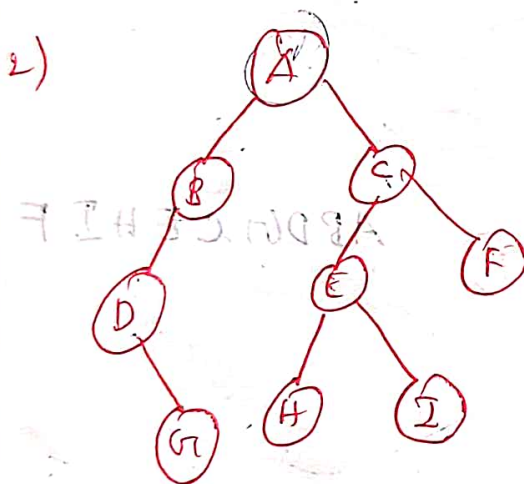
Post order (L.R.D)

- * Traverse the left subtree in post order
- * Traverse the right subtree in post order
- * Visit the root node.

Example



10 30 20



GIDBHIEFCA

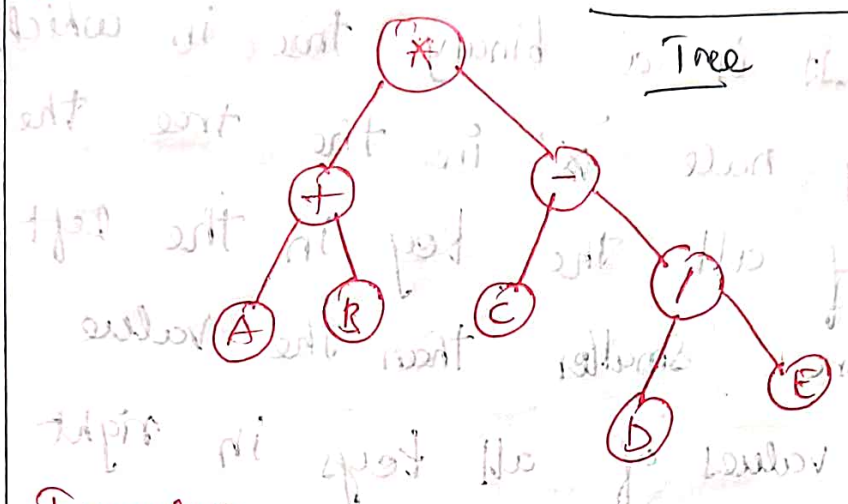
Routine:

void Postorder (Tree T)

```
if (T != NULL)
{
    Postorder (T->left);
    Postorder (T->right);
    Print ("Element");
}
```

Example I

Find Inorder, Preorder & post order traversals of given



Inorder

A + B * C - D / E

Pre order

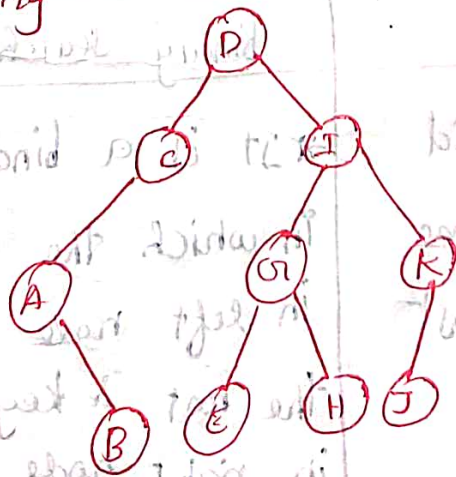
* + AB - C / DE

Post order

AB + CDE / - *

② Find Pre order Inorder and post order

of following



Inorder

ABCDEGH IJK

Pre order

D CAB I G E H K J

Post order

BAC E H G J K I D