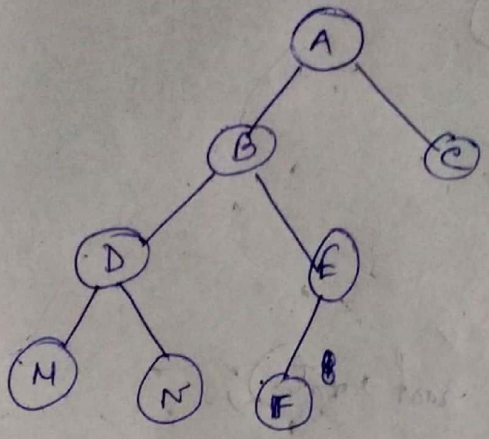


Pre order- A B D M N E F C

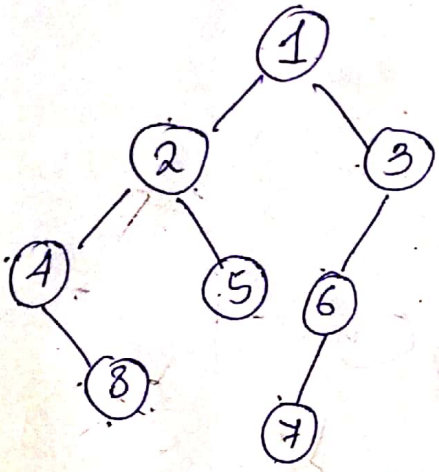
In order- M D N B F E A C

Post order- M N D F E B C A

```
void preorder (Node * root, char arr[])
{
    printf(
    if (root == NULL)
        return;
    else
    {   printf("%c", root->data);
        preorder (root->left, arr);
        preorder (root->right, arr);
    }
}
```

~~void preorder (node * root)~~

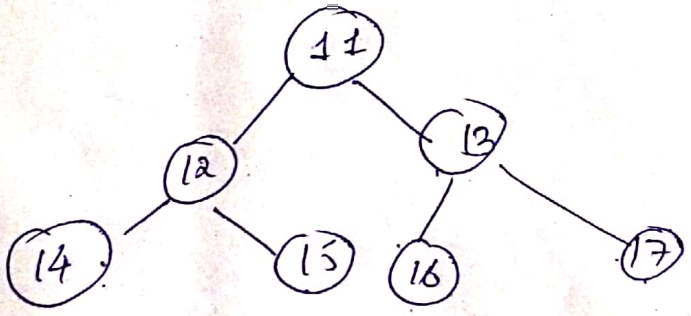
```
{
}
void preorder (node * root)
{   if (root != NULL)
    {   printf ("%c", root->data);
        preorder (root->left);
        preorder (root->right);
    }
}
Void   inorder (node * root)
{       if (root != NULL)
        {   inorder (root->left);
            printf ("%c", root->data);
            inorder (root->right);
        }
}
void   postorder (node * root)
{       if (root != NULL)
        {   postorder (root->left);
            postorder (root->right);
            printf ("%c", root->data);
        }
}
```



Inorder → 4 8 2 5 1 7 6 3

preorder → 1 2 4 8 5 3 6 7

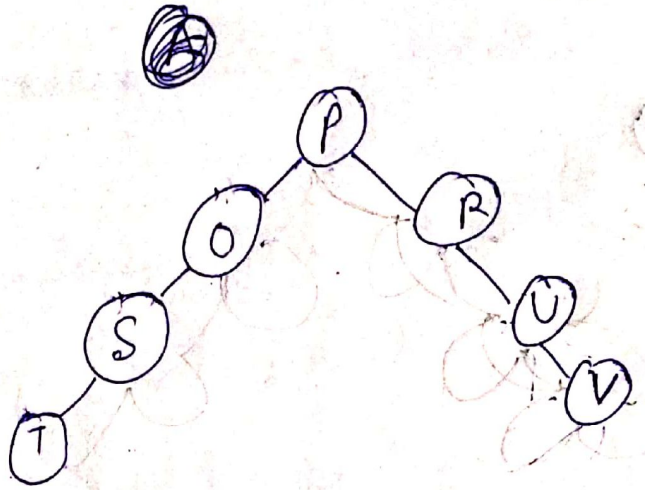
postorder → 8 4 5 2 7 6 3 1 .



Preorder → 11, 12, 14, 15, 13, 16, 17

Inorder → 14, 12, 15, 11, 16, 13, 17

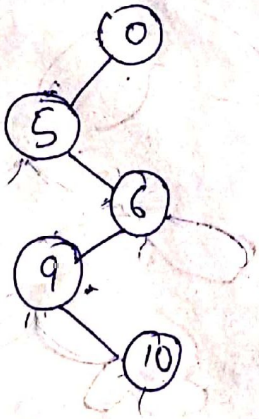
Postorder → 14, 15, 12, 16, 17, 13, 11



Preorder- P O S T R U V

Inorder - T S O P R U V

Postorder- T S O V U R P



Preorder - 0, 5, 6, 9, 10

Inorder - 5, 9, 10, 6, 0

Postorder - 10, 9, 6, 5, 0.