**Tree**

1. **Implementation**

**Structure**

Node{

public:

int data;

Node\* right;

Node\* left;

Node(int d)

{

data = d;

left = NULL;

right = NULL;

}

}

**BuildTree**

Node\* buildTree()

{

Int data;

cout<<”Enter data for root node = “ << endl;

cin >> data;

if(data == -1)

{

return NULL;

}

Node\* newNode = new Node(data);

cout<< “Enter data for the left of << data << endl;

newNode -> left = buildTree();

cout<< “Enter data for the right of << data << endl;

newNode -> left = buildTree();

return newNode;

}

1. **Traversal**

* **Preorder**

**ROOT || LEFT || RIGHT**

**Recursive**

Void preOrder(Node\* root)

{

If(root == NULL)

{

Return;

}

cout<< root -> data << “ “;

preOrder(root -> left);

preOrder(root -> right);

}

**Iterative**