```
#include <iostream>
using namespace std;

class BinaryTree
{
        class Node
        {
        public:
                Node *left;
                int value;
                Node *right;

        public:
                Node(int val)
                        :left(nullptr),value(val),right(nullptr)
                {}
        };

public:
        Node *root;

        BinaryTree()
                :root(nullptr)
        {}

        //accept value from user
        void insert (int value)
        {
                root = addNode(root, value);
        }
```

```
//add new nodes to the tree
Node* addNode(Node *node,int value)
{
	//first node in the tree
	if(!node)
		return new Node(value);

	//if value is greater than root node->value
	if(value > node->value)
		node->right = addNode(node->right,value);

	//if value is smaller than root node->value
	else
		node->left = addNode(node->left,value);

	return node;
}

//Printing tree in Inorder
void Inorder(Node *node)
{
	if(node)
	{
		Inorder(node->left);
		cout<<node->value<<"\t";
		Inorder(node->right);
	}
}

//Printing tree in Preorder
void Preorder(Node *node)
```

```
    {
        if(node)
        {
            cout<<node->value<<"\t";
            Preorder(node->left);
         Preorder(node->right);
        }
    }

    //Printing tree in Postorder
    void Postorder(Node *node)
    {
        if(node)
        {
            Postorder(node->left);
         Postorder(node->right);
            cout<<node->value<<"\t";
        }
    }
};

int main()
{
    //creating object of binary tree
    BinaryTree b;
    int num;

    // taking value from user to add a node
    while(cout<<"enter a value to insert in tree(0 to stop)"<<endl,
             cin>>num,
                num)
    {
        //inserting a nodea
        b.insert(num);
    }

    //Displaying in Inorder
    b.Inorder(b.root);
    cout<<endl;

    //Displaying in Preorder
    b.Preorder(b.root);
    cout<<endl;

    //Displaying in Postorder
    b.Postorder(b.root);
    cout<<endl;

    return 0;
}



/*output
enter a value to insert in tree(0 to stop)
5
enter a value to insert in tree(0 to stop)
7
enter a value to insert in tree(0 to stop)
1
enter a value to insert in tree(0 to stop)
2
```

```
enter a value to insert in tree(0 to stop)

9

enter a value to insert in tree(0 to stop)

4

enter a value to insert in tree(0 to stop)

0

1       2       4       5       7       9

5       1       2       4       7       9

4       2       1       9       7       5

Press any key to continue . . .

*/
```