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

 class Binary
 {
 private:
  class Node
  {
  public :
   Node *left;
   int value;
   Node *right;

   Node(int v)
   {
    left = right = nullptr;
    value = v;
   }
  };
 public:
  Node *head;

 public:
  Binary():head(nullptr)
  {
  }

  Node* addnode(Node *n,int value)
  {
   if (nullptr==n)
    return new Node(value);

   if (n->value > value)
    n->left=addnode(n->left,value);

   else
    n->right=addnode(n->right,value);

   return n;
  }

  void inorder (Node* n)
  {
   if (n)
   {
    inorder(n->left);
    cout << n->value << endl;
    inorder(n->right);
   }
  }
```

```
void preorder( Node* n)
{
 if (n)
 {   cout << n->value << endl;
 preorder(n->left);
 preorder(n->right);
 }
}

void postorder( Node* n)
{
 if (n)
 {
  postorder(n->left);
  postorder(n->right);
  cout << n->value << endl;
 }
}

void print (Node *n)
{
 static int level=0;
 if (n)
 {
  level++;
  print(n->right);
  cout << setw(level *4) << n->value << endl;
  print(n->left);
  level--;
 }
}



};


int main()
{
 int num;

 Binary b1 ;

 while(cout<<"enter the element(o to stop)",cin>>num,num)
  b1.head = b1.addnode(b1.head,num);

 cout << "Inorder display" << endl;
 b1.inorder(b1.head);
 cout<<endl;

 cout << "PreOrder Display" << endl;
 b1.preorder(b1.head);
```

```
 cout<<endl;

 cout << "PostOrder Display" << endl;
 b1.postorder(b1.head);
 cout<<endl;

 b1.print(b1.head);
}
```