#include<iostream>

using namespace std; //header fles

////////////////////////////

struct junaid\_node //name of structer

{

junaid\_node \* l\_hassan; //left node

junaid\_node \* r\_bilal; //right node

int data; //data part of node

};

class jbstree //name of class

{

private:

junaid\_node \*ghias; // name of root

public:

jbstree() //class constructer

{

ghias=NULL; //make a root null first

}

void inserting\_node(int r\_node); //inserting values function declaration

void in\_order\_t(); //in\_order function declaration

void inorder(junaid\_node \*p); //passing structre declaration

void pre\_order\_t(); //pre\_order function declaration

void preorder(junaid\_node \*p); //passing structre declaration

void post\_order\_t(); //post\_order function declaration

void postorder(junaid\_node \*p); //passing structre declaration

};//end of class

//////////////////////////////

void jbstree::inserting\_node(int receive\_node) //recevieiving values from below main funtion

{

junaid\_node \*p=new junaid\_node; //creating new structer with p temparary variable p

junaid\_node \*parent; // creating variable of parent

p->data=receive\_node; //put receiving value into data pert of node

p->l\_hassan=NULL; //make left null

p->r\_bilal=NULL; //make right null

parent=NULL;

if(ghias==NULL)

{

ghias=p;

}

else

{

junaid\_node \*ptr;

ptr=ghias;

while(ptr!=NULL)

{

parent=ptr;

if(receive\_node>ptr->data)

ptr=ptr->r\_bilal;

else

ptr=ptr->l\_hassan;

}

if(receive\_node<parent->data)

parent->l\_hassan=p;

else

parent->r\_bilal=p;

}

}

/////////////////////////////////////////

void jbstree::in\_order\_t() //in\_oreder declaration function define now

{

cout<<"\n\nThis is IN\_ORDER tree."<<endl;

inorder(ghias);

}

void jbstree::inorder(junaid\_node \*ptr)

{

if(ptr!=NULL)

{

inorder(ptr->l\_hassan);

cout<<" "<<ptr->data<<" ";

inorder(ptr->r\_bilal);

}

}

//////////////////////////

void jbstree::pre\_order\_t() //pre\_oreder declaration function define now

{

cout<<"\n\nThis is PRE\_ORDER tree."<<endl;

preorder(ghias);

}

void jbstree::preorder(junaid\_node \*ptr) //passing structre

{

if(ptr!=NULL)

{

cout<<" "<<ptr->data<<" ";

preorder(ptr->l\_hassan);

preorder(ptr->r\_bilal);

}

}

//////////////////////

void jbstree::post\_order\_t() //post\_oreder declaration function define now

{

cout<<"\n\nThis is POST\_ORDER tree."<<endl;

postorder(ghias);

}

void jbstree::postorder(junaid\_node \*ptr)

{

if(ptr!=NULL)

{

postorder(ptr->l\_hassan);

postorder(ptr->r\_bilal);

cout<<" "<<ptr->data<<" ";

}

}

/////////////////////////////////////////

int main() //main body of program

{

system("color f9"); //color code

jbstree f; //object of class is f

f.inserting\_node(52); //inserting values

f.inserting\_node(25);

f.inserting\_node(50);

f.inserting\_node(15);

f.inserting\_node(40);

f.inserting\_node(45);

f.inserting\_node(20);

f.in\_order\_t();

f.pre\_order\_t();

f.post\_order\_t();

getchar();

getchar();

return 0;

}

//end of program