#include <iostream>

#include<queue>

using namespace std;

class node{

public:

int data;

node \*left;

node \*right;

node(int d){

data=d;

left=NULL;

right=NULL;

}

};

node\* insertInBST(node \*root,int data){

if(root==NULL){

return new node(data);

}

if(data<=root->data){

root->left=insertInBST(root->left,data);

}

else{

root->right=insertInBST(root->right,data);

}

return root;

}

node\* build(){

int d;

cin>>d;

node \*root=NULL;

while(d!=-1){

root=insertInBST(root,d);

cin>>d;

}

return root;

}

//inorder of bst is always sorted

void inorder(node \*root){

if(root==NULL)

return;

inorder(root->left);

cout<<root->data<<", ";

inorder(root->right);

}

bool search(node \*root,int data){

if(root==NULL)

return false;

if(root->data==data)

return true;

if(data<=root->data){

return search(root->left,data);

}

else{

return search(root->right,data);

}

}

int main()

{

node \*root=build();

inorder(root);

cout<<endl;

int s;

cin>>s;

if(search(root,s)){

cout<<"Present"<<endl;

}

else{

cout<<"Not present"<<endl;

}

}

Input-

5 3 7 1 6 8 -1

8

Output-

1, 3, 5, 6, 7, 8,

Present