#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);

}

void bfs(node \*root){

queue<node\*> q;

q.push(root);

q.push(NULL);

while(!q.empty()){

node\* f = q.front();

if(f==NULL){

cout<<endl;

q.pop();

if(!q.empty()){

q.push(NULL);

}

}

else{

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

q.pop();

if(f->left){

q.push(f->left);

}

if(f->right){

q.push(f->right);

}

}

}

return;

}

int main()

{

node \*root=build();

inorder(root);

cout<<endl;

bfs(root);

}

Input-

5 3 7 1 6 8 -1

Output-

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

5,

3,7,

1,6,8,