#include<queue>

#include <iostream>

using namespace std;

class node{

public:

int data;

node \*left;

node \*right;

node(int d){

data=d;

left=NULL;

right=NULL;

}

};

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;

}

node\* buildTreeFromInPre(int \*ino,int \*pre,int s,int e){

static int i=0;

if(s>e){

return NULL;

}

node \*root=new node(pre[i]);

int index=-1;

for(int j=s;j<=e;j++){

if(ino[j]==pre[i]){

index=j;

break;

}

}

i++;

root->left=buildTreeFromInPre(ino,pre,s,index-1);

root->right=buildTreeFromInPre(ino,pre,index+1,e);

return root;

}

int main() {

int pre[]={1,2,3,4,8,5,6,7};

int ino[]={3,2,8,4,1,6,7,5};

int n=sizeof(ino)/sizeof(ino[0]);

node \*root=buildTreeFromInPre(ino,pre,0,n-1);

bfs(root);

}

Output-

1,

2,5,

3,4,6,

8,7,