#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\* buildTreeFromArray(int \*a,int s,int e){

if(s>e){

return NULL;

}

int mid=(s+e)/2;

node \*root=new node(a[mid]);

root->left=buildTreeFromArray(a,s,mid-1);

root->right=buildTreeFromArray(a,mid+1,e);

return root;

}

void print(node \*root){

if(root==NULL)

return;

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

print(root->left);

print(root->right);

}

int main() {

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

int n=7;

node \*root=buildTreeFromArray(a,0,n-1);

bfs(root);

}

Output-

4,

2,6,

1,3,5,7,