#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\* buildTree(){

int d;

cin>>d;

if(d==-1)

return NULL;

node \*root=new node(d);

root->left=buildTree();

root->right=buildTree();

return root;

}

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 replaceSum(node \*root){

if(root==NULL)

return 0;

if(root->left==NULL && root->right==NULL)

return root->data;

int leftSum=replaceSum(root->left);

int rightSum=replaceSum(root->right);

int temp=root->data;

root->data=leftSum+rightSum;

return temp+root->data;

}

int main() {

node \*root=buildTree();

bfs(root);

cout<<endl;

replaceSum(root);

bfs(root);

}

Input-

8 10 1 -1 -1 6 9 -1 -1 7 -1 -1 3 -1 14 13 -1 -1 -1

Output-

8,

10,3,

1,6,14,

9,7,13,

63,

23,27,

1,16,13,

9,7,13,