/\*\*

 \* Definition for a binary tree node.

 \* struct TreeNode {

 \*     int val;

 \*     struct TreeNode \*left;

 \*     struct TreeNode \*right;

 \* };

 \*/

int check;

struct TreeNode\* ancestor;

struct TreeNode\* firstNode;

struct TreeNode\* secondtNode;

struct TreeNode\* pre\_firstNode;

void setAncester(struct TreeNode \*node ,struct TreeNode\* pre\_node, struct TreeNode\* p, struct TreeNode\* q){

    struct TreeNode\* rememberfirst;

    secondtNode=NULL;

    if( check == 0){

        ancestor=pre\_node;

        pre\_firstNode=pre\_node;

        firstNode=node;

    }else if( check == 1){

        pre\_firstNode=pre\_node;

        firstNode=node;

    }else{

        rememberfirst=firstNode;

        printf("run sec ");

        if(firstNode == p){

            secRun(rememberfirst,q);

        }else{

            secRun(rememberfirst,p);

        }

        if( secondtNode == NULL && pre\_firstNode-> != firstNode){

            rememberfirst=pre\_firstNode;

            if(firstNode == p){

            secRun(rememberfirst,q);

            }else{

            secRun(rememberfirst,p);

            }

        }

    }

    printf("check ancestor=%d firstNode=%d\n",ancestor->val,firstNode->val);

    check++;

}

void run( struct TreeNode \*node , struct TreeNode \*pre\_node, struct TreeNode\* p, struct TreeNode\* q){

    if( node == NULL || check > 2){

        //printf("NULL\n");

        return;

    }else{

        printf("%d ancestor=%d\n",node->val,pre\_node->val);

    }

    if( node->val == p->val || node->val == q->val ){

        setAncester(node,pre\_node,p,q);

    }

    if( check > 2){

        return;

    }

    run(node->left,node,p,q);

    run(node->right,node,p,q);

}

void secRun( struct TreeNode \*node , struct TreeNode\* pq){

    if( node == NULL ){

        return;

    }

    if( node == pq){

        printf("ancestor=%d firstNode=%d\n",ancestor->val,firstNode->val);

        secondtNode=pq;

    }

    secRun(node->left,pq);

    secRun(node->right,pq);

}

struct TreeNode\* lowestCommonAncestor(struct TreeNode\* root, struct TreeNode\* p, struct TreeNode\* q) {

    ancestor=root;

    firstNode=root;

    check=0;

    setAncester(root,root,p,q);

    run(root,root,p,q);

    return ancestor;

}