/\*\*

 \* Definition for a binary tree node.

 \* struct TreeNode {

 \*     int val;

 \*     struct TreeNode \*left;

 \*     struct TreeNode \*right;

 \* };

 \*/

void same( struct TreeNode\* root, struct TreeNode\* subRoot ,bool \*retAns ){

    struct TreeNode\* subIndex=subRoot;

    struct TreeNode\* rootIndex=root;

    if( subIndex != NULL && rootIndex != NULL){//subRoot跟root都有值就比較看看

        //printf("rootIndex=%d subIndex=%d\n",rootIndex->val,subIndex->val);

        if( subIndex->val != rootIndex->val){

            \*retAns=false;

            return;

        }

    }else if( subIndex == NULL && rootIndex == NULL ){

    }else{//一個有值一個為NULL就將retAns改為false

            \*retAns=false;

            return;

    }

    if( subIndex->left != NULL && rootIndex->left != NULL){//subRoot跟root的左側不為null就往左側找找

        same( subIndex->left , rootIndex->left ,retAns);

    }else if( subIndex->left == NULL && rootIndex->left == NULL ){

    }else{//一個有值一個為NULL就將retAns改為false

            \*retAns=false;

            return;

    }

    if( subIndex->right != NULL && rootIndex->right != NULL){//subRoot跟root的右側不為null就往右側找找

        same( subIndex->right , rootIndex->right ,retAns);

    }else if( subIndex->right == NULL && rootIndex->right == NULL ){

    }else{//一個有值一個為NULL就將retAns改為false

            \*retAns=false;

            return;

    }

}

void runRoof( struct TreeNode\* root, struct TreeNode\* subRoot ,bool \*ans){//走root迴圈

    struct TreeNode\* index=subRoot;

    bool retAns=true;

    if( root != NULL && subRoot != NULL  ){//如果root值與subRoot的值相同就用same去比對看看是否相同

        //printf("root=%d index=%d\n",root->val,index->val);

        same(root,subRoot,&retAns);

    }else if( root == NULL && subRoot == NULL){//如果root值與subRoot都是NULL就結束這樣答案就應該是true

        return;

    }else{

        \*ans=false;

        return;

    }

    if( retAns == true){//如果same有找到答案就將ans給true的值

        \*ans=true;

        return;

    }

    if( root->left != NULL ){

       runRoof(root->left,subRoot,ans);

    }

    if( root->right != NULL ){

       runRoof(root->right,subRoot,ans);

    }

}

bool isSubtree(struct TreeNode\* root, struct TreeNode\* subRoot){

    bool ans = false;

    runRoof(root,subRoot,&ans);

    return ans;

}