/\*\*

\* Definition for a binary tree node.

\* struct TreeNode {

\* int val;

\* struct TreeNode \*left;

\* struct TreeNode \*right;

\* };

\*/

typedef struct TreeNode TreeNode;

TreeNode\* find( int\* preorder, int preorderSize, int\* inorder, int inorderSize ){

if( preorderSize == 0){

//printf("return\n");

return NULL;

}

TreeNode\* root = (TreeNode\*)malloc(sizeof(TreeNode));

root->val = preorder[0];

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

// for( int i = 0 ; i < preorderSize ; i++){

// printf("%d ",preorder[i]);

// }

// printf("\n");

// for( int i = 0 ; i < inorderSize ; i++){

// printf("%d ",inorder[i]);

// }

//printf("\n");

int over=0;

int \*left\_inorder =malloc(sizeof(int)\*3000);

int \*left\_preorder=malloc(sizeof(int)\*3000);

int countleft\_inorder=0;

int countleft\_preorder=0;

int \*right\_inorder =malloc(sizeof(int)\*3000);

int \*right\_preorder=malloc(sizeof(int)\*3000);

int countright\_inorder=0;

int countright\_preorder=0;

for(int j=0 ; j < inorderSize ; j++){

if( preorder[0] == inorder[j] ){

over =1 ;

}else if( over ==1 ) {

right\_inorder[countright\_inorder]=inorder[j];

//printf("rpp =%d \n",right\_inorder[countright\_inorder]);

countright\_inorder++;

}else if( over == 0 ) {

left\_inorder[countleft\_inorder]=inorder[j];

//printf("lpp =%d \n",left\_inorder[countleft\_inorder]);

countleft\_inorder++;

}

}

for(int i=0;i < preorderSize ; i++){

for(int j=0; j < countleft\_inorder ; j++){

if( preorder[i] == left\_inorder[j] ){

left\_preorder[countleft\_preorder]=preorder[i];

countleft\_preorder++;

}

}

}

for(int i=0 ; i < preorderSize ; i++){

for(int j=0; j < countright\_inorder ; j++){

if( preorder[i] == right\_inorder[j] ){

right\_preorder[countright\_preorder]=preorder[i];

countright\_preorder++;

}

}

}

root->left = find(left\_preorder,countleft\_preorder,left\_inorder,countleft\_inorder);

root->right = find(right\_preorder,countright\_preorder,right\_inorder,countright\_inorder);

return root;

}

void go( struct TreeNode \*ans ){

if(ans == NULL){

return;

}else{

printf("%d ",ans->val);

}

go(ans->left);

go(ans->right);

}

struct TreeNode\* buildTree(int\* preorder, int preorderSize, int\* inorder, int inorderSize){

struct TreeNode \*sol=NULL;

sol=find(preorder,preorderSize,inorder,inorderSize);

return sol;

}