/\*\*

\* Definition for singly-linked list.

\* struct ListNode {

\* int val;

\* struct ListNode \*next;

\* };

\*/

struct ListNode\* mergeTwoLists(struct ListNode\* list1, struct ListNode\* list2){

struct ListNode\* tmp;

struct ListNode\* top1=list1;

struct ListNode\* top2=list2;

struct ListNode \*index =NULL;

struct ListNode \*top ;

printf(" pinterTop=%p pinterIndex=%p\n",top,index);

if( top1 == NULL ){

return list2;

}else if( top2 == NULL){

return list1;

}else if( top2 == NULL && top1 == NULL){

return NULL;

}

//index=ans;//如果沒有將ans塞入地址，index都只是null

printf(" pinterTop=%p pinterIndex=%p\n",top,index);

if( top1->val < top2->val){//先將地址塞入ans裡面

index=top1;

top1=top1->next;

}else{

index=top2;

top2=top2->next;

}

printf(" pinterTop=%p pinterIndex=%p\n",top,index);

top =index;//讓top 取得ans的地址

printf("pinterTop=%p pinterIndex=%p\n",top,index);

while( top1 != NULL || top2 != NULL){

printf(" pinterTop=%p pinterIndex=%p\n",top,index);

// printf("%d %d\n",list1->val,list2->val);

// printf("%d %d\n",top1->val,top2->val);

if( top2==NULL ){

index->next=top1;

index=index->next;

top1=top1->next;

}else if( top1==NULL ){

index->next=top2;

index=index->next;

top2=top2->next;

}else if(top1->val < top2->val){

index->next=top1;

index=index->next;

top1=top1->next;

}else if( top1->val == top2->val){

index->next=top1;

index=index->next;

top1=top1->next;

index->next=top2;

index=index->next;

top2=top2->next;

}else if(top1->val > top2->val){

index->next=top2;

index=index->next;

top2=top2->next;

}

}

return top;

}