/\*\*

 \* Definition for singly-linked list.

 \* struct ListNode {

 \*     int val;

 \*     struct ListNode \*next;

 \* };

 \*/

struct ListNode\* addTwoNumbers(struct ListNode\* l1, struct ListNode\* l2){

    struct ListNode\* index=malloc( sizeof(struct ListNode));//宣告一個答案的節點

    struct ListNode\* result=index;//把答案的初始節點丟給result

    int val1=0;

    int val2=0;

    int carry=0;//carry是進位

    int res=0;

    while( l1 != NULL ||  l2 != NULL){

        struct ListNode\* next=malloc( sizeof(struct ListNode));//宣告答案的下一個節點

        val1=0;

        val2=0;

        if( l1 != NULL ){//取得l1的數值

            //printf("L1=%d \n",l1->val);

            val1=l1->val;

            l1=l1->next;

        }

        if( l2 != NULL ){//取得l2的數值

            //printf("L2=%d \n",l2->val);

            val2=l2->val;

            l2=l2->next;

        }

        res=val1+val2+carry;//取得l2的數值

        carry=res/10;//取進位數

        res=res%10;//取餘數

        // printf("( val1=%d+ val2=%d )=%d",val1,val2,res);

        // printf("\n");

        index->val=res;//把餘數給現在的答案節點

        if( l1 == NULL && l2 == NULL && carry ==0 ){//如果進位沒有值就將next放入NULL結束

            index->next=NULL;

        }else if( l1 == NULL && l2 == NULL && carry !=0 ){//如果進位有值就將next放入新節點，並且carry的進位數值放入新節點中

            index->next=next;

            index=index->next;

            index->val=carry;

            index->next=NULL;

        }else{//如果l2或l2還有數值就繼續

            index->next=next;

            index=index->next;

        }

    }

    return result;

}