

已知两个非降序链表序列S1与S2，设计函数构造出S1与S2的交集新链表S3。

输入格式：

输入分两行，分别在每行给出由若干个正整数构成的非降序序列，用-1表示序列的结尾（-1不属于这个序列）。数字用空格间隔。

输出格式：

在一行中输出两个输入序列的交集序列，数字间用空格分开，结尾不能有多余空格；若新链表为空，输出NULL。

输入样例：

1 2 5 -1

2 4 5 8 10 -1

输出样例：

2 5

```
#include<stdio.h>
```

```
typedef struct Node{
    int data;
    struct Node *next;
}Node,*LinkList;
```

```
LinkList init();
```

```
LinkList init(){
    LinkList link;
    link=(LinkList)malloc(sizeof(LinkList));
    link->next=NULL;

    Node* node;
    Node* lnode;
    node=link;
    while(1){
        int temp;
        scanf("%d",&temp);
        if(temp==-1){
            return link;
        }else{
            lnode=(Node*)malloc(sizeof(Node));
            lnode->data=temp;
            lnode->next=NULL;
            node->next=lnode;
            node=node->next;
        }
    }
    return link;
```

```
}
```

```
LinkedList bing(){
    LinkedList list1=init();
    LinkedList list2=init();
    LinkedList list3;
    list3=(LinkedList)malloc(sizeof(LinkedList));
    list3->next=NULL;
    Node* node1;
    Node* node2;
    Node* node3;
    node1=list1->next;
    node2=list2->next;
    node3=list3;
    while(node1&&node2){

        if(node1->data<node2->data){
            node1=node1->next;
        }else if(node1->data>node2->data){
            node2=node2->next;
        }else if(node1->data==node2->data){
            Node* node=(Node*)malloc(sizeof(Node));
            node->data=node1->data;
            node->next=NULL;
            node3->next=node;
            node3=node3->next;
            node1=node1->next;
            node2=node2->next;
        }
    }
    return list3;
}
```

```
int main(){
    LinkedList list=bing();
    Node* node;
    node=list->next;
    while(node){
        printf("%d",node->data);
        node=node->next;
    }
}
```

