

### 练习 13

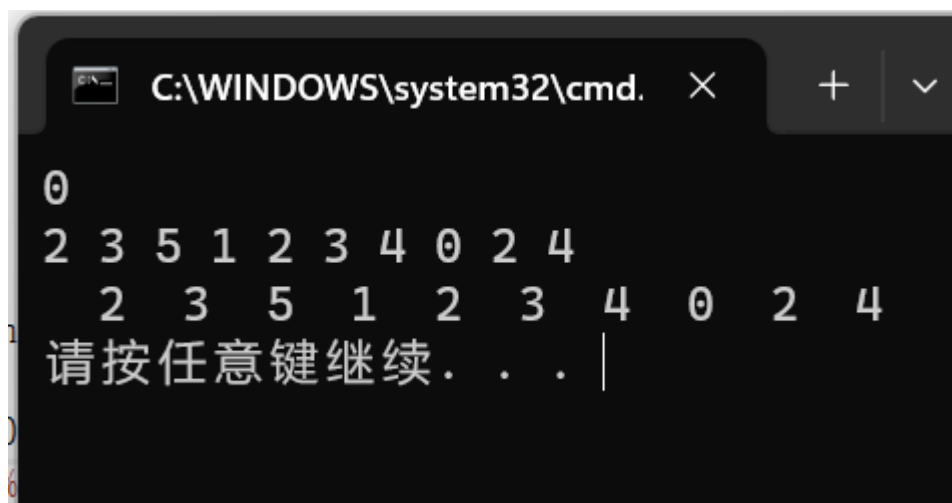
#### 第 4 题

代码:

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

```
union grades{
    int a[10];
    float b[10];
};

void main(){
    char sign;
    union grades g;
    int i;
    scanf("%c",&sign);
    if(sign=='0'){
        for(i=0;i<10;i++)
            scanf("%d",&g.a[i]);
        for(i=0;i<10;i++)
            printf("%3d",g.a[i]);
        printf("\n");
    }
    if(sign=='1'){
        for(i=0;i<10;i++)
            scanf("%f",&g.b[i]);
        for(i=0;i<10;i++)
            printf("%6.2f",g.b[i]);
        printf("\n");
    }
}
```



```
C:\WINDOWS\system32\cmd. x + v
1
92.0 93.5 89.5 82.0 95.0 100 98.5 94.5 99 99
92.00 93.50 89.50 82.00 95.00 100.00 98.50 94.50 99.00 99.00
请按任意键继续. . . |
```

第五题:

代码:

```
int nodelength( struct node *head) {
    struct node *p=head;
    int count=0;
    while(p!=NULL) {
        count++;
        p=p->next;
    }
    return count;
}
```

当给定链表如下（标黄）时，结果如下图:

```
#include<stdio.h>
struct node{
    int data;
    struct node *next;
};
int nodelength( struct node *head) {
    struct node *p=head;
    int count=0;
    while(p!=NULL) {
        count++;
        p=p->next;
    }
    return count;
}
void main() {
    struct node a[5]={0, &a[1]}, {1, &a[2]}, {2, &a[3]}, {3, &a[4]}, {4, NULL}};
    int b=nodelength(a);
    printf("%d\n", b);
}
```

结果



第 6 题:

以升序列表为例, 编写 nodecat 函数合并两列表, 返回头指针 ch 值。

代码: (标黄为合并链表函数)

```
#include<stdio.h>

struct node{
    int data;
    struct node *next;
};

typedef struct node NODE;

NODE *nodecat(NODE *ah, NODE *bh) {
    NODE *p=ah, *q=bh, *ch, *c;
    if(ah==NULL) return bh;
    if(bh==NULL) return ah;

    if(ah->data<bh->data){
        ch=ah;
        p=ah->next;
    }
    else{
        ch=bh;
        q=bh->next;
    }
    c=ch;
    while(p!=NULL&&q!=NULL){
        if(p->data<q->data){
            c->next=p;
            p=p->next;
        }
        else{
            c->next=q;
            q=q->next;
        }
        c=c->next;
    }
    if(p==NULL)
```

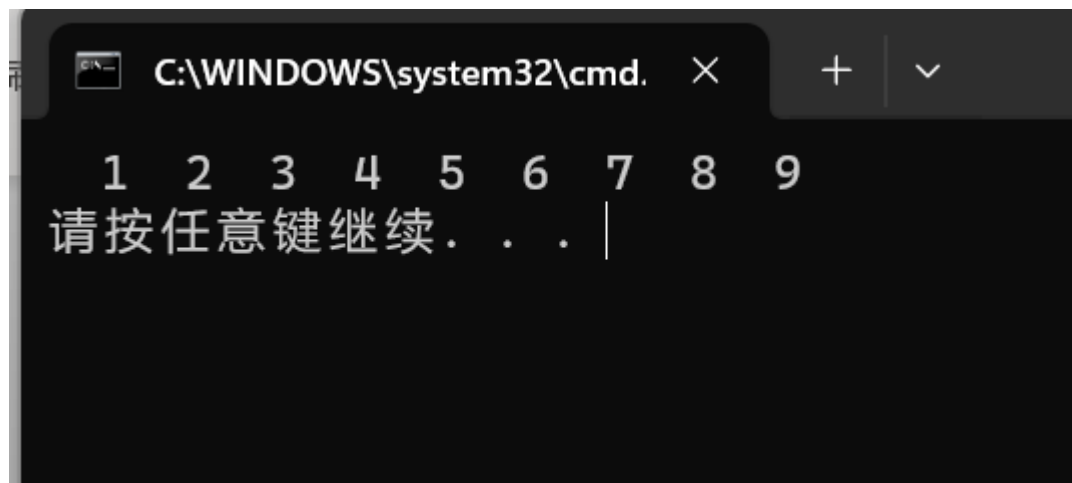
```

        c->next=q;
    else c->next=p;
    return ch;
}

void main() {
    NODE a[5]={ {1,&a[1]}, {3,&a[2]}, {6,&a[3]}, {8,&a[4]}, {9, NULL}};
    NODE b[4]={ {2,&b[1]}, {4,&b[2]}, {5,&b[3]}, {7, NULL}};
    NODE *c=nodecat(a,b), *p=c;
    while(c!=NULL) {
        printf("%3d",c->data);
        c=c->next;
    }
    printf("\n");
}

```

结果:



约瑟夫问题:

```

#include<stdio.h>
#include<stdlib.h>
typedef struct ysf{
    int n;
    struct ysf *next;
}Y;
Y *f(int m){
    Y *head,*temp,*p;
    int i;
    head=NULL;
    p=NULL;
    for(i=1;i<=m;i++){
        temp=(Y*) malloc(sizeof(Y));
        temp->n=i;
        temp->next=NULL;
    }
}

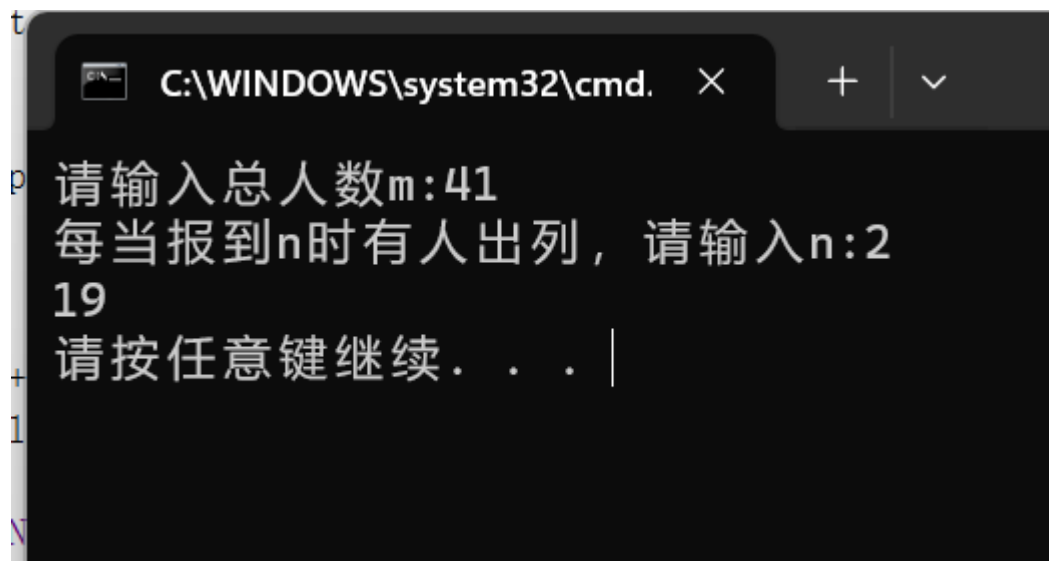
```

```

        if(head==NULL) {
            head=temp;
            p=temp;}
        else{
            p->next=temp;
            p=temp;
        }
        p->next=head;
    }    return head;
}

void main() {
    int n,m,i;
    Y *head,*p,*temp;
    printf("请输入总人数m:");
    scanf("%d",&m);
    head=f(m);
    temp=head;
    p=NULL;
    printf("每当报到n时有人出列，请输入n:");
    scanf("%d",&n);
    while(temp->next!=temp) {
        for(i=1;i<n;i++) {
            p=temp;
            temp=temp->next;
        }
        p->next=temp->next;
        temp=temp->next;
    }
    printf("%d\n",(*temp).n);
    free(temp);
}

```



```

C:\WINDOWS\system32\cmd.
请输入总人数m:41
每当报到n时有人出列，请输入n:2
19
请按任意键继续. . . |

```