学号	2021144301	姓名	王大殿	实验日期	
实验名称					

实验目的：

实验任务、步骤与结果：

矩阵对角线之和
题目描述
求一个 3×3 矩阵左右对角线元素之和（去掉对角一交叉重复数）。
输入
3 行：
一个 3 行 3 列的矩阵
输出
1 行：
1 个整数，左右对角线之和
代码：

```
#include<stdio.h>
int main(){
    int arr[3][3];
    int sum=0;
    char ch;
    //二维数组输入，元素用空格隔开，输完一行按 enter
    for(int i=0; i<3; i++){
        int len=0;
        do{
            scanf("%d%c",&arr[i][len],&ch);
            len++;
        }while(ch!='\n');
    }
    for(int j=0; j<3; j++){
        //对角相加
        sum+=arr[j][j]+arr[2-j][j];
    }
    sum=sum-arr[1][1];
    printf("%d\n",sum);
    return 0;
}
```

运行结果

```
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/Experiment/实验6$ ./对角线之和
1 2 3
4 5 6
7 8 9
25
```

插入单词

题目描述

现有一组已经排好序的单词，根据字母顺序，将一个新单词插入队列中。

输入

若干行：

第 1 行，整数 n，表示接下来输入 n 个单词

第 2~n+1 行，n 个从小到大排列的单词

第 n+1 行，要插入的单词

输出

若干行：

插入后的单词，每行 1 个

```
#include<stdio.h>
#include<string.h>
int main(){
    int n;
    printf("Input the number of words: ");
    scanf("%d",&n);
    char arr[n+1][20];
    for (int i = 0; i <= n; i++)
    {
        scanf("%s",arr[i]);
    }
    for(int j=0; j<n; j++){
        if(strcmp(arr[n], arr[j])<0){
            //如果最后一个单词比第 j 个单词小，则第 j 个后面的单词向后移一位
            char str_t[20];
            strcpy(str_t,arr[n]);
            for(int k=n-1; k>=j; k--){
                strcpy(arr[k+1],arr[k]);
            }
            //把最后一个单词复制给第 j 个
            strcpy(arr[j],str_t);
            break;
        }
    }
    for (int i = 0; i <=n; i++)
    {
        printf("%s\n",arr[i]);
```

```
        }

        return 0;
    }
```

运行结果：

```
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/Experiment/实验6$ ./WordInsert
Input the number of words: 5
An
apple
bee
bike
meat
frog
An
apple
bee
bike
frog
meat
```

最小等待时间

题目描述

银行一共 n 个客户在排队，每个客户因业务不同，需要的时间从 1~m 分钟不等。银行大堂经理根据每个客户的业务类型，安排业务排队，使 n 个客户的总等待时间最少。例如，某个客户办理业务需要 5 分钟，他面有 4 个客户在等待，则该客户产生的等待时间是 5*4=20 分钟，而如果他只需要 1 分钟，则他产生的等待时间是 4 分钟。

输入

2 行：

第 1 行，1 个整数 n，表示有 n 个客户

第 2 行，n 个整数，表示每个客户需要等待时间

输出

1 行：

整数 n，表示最少需要等待的时间

代码；

```
#include<stdio.h>
void BubbleSort(int a[], int n);
int main(){
    int n;
    printf("Enter the number of customers.\n");
    scanf("%d",&n);
    int a[n];
    int len=0, sum=0;
    char ch;
    do
```

```
    {
        scanf("%d%c",&a[len],&ch);
        len++;
    } while (ch!='\n');
    BubbleSort(a,n);
    for(int i=0; i<n; i++){
        sum+=a[i]*i;
    }
    printf("The minimum waiting time is %d.\n",sum);
    return 0;
}
//冒泡排序
void BubbleSort(int a[], int n){
    for(int i=0; i<n-1; i++){
        for(int j=0; j<n-i-1; j++){
            if(a[j]<a[j+1]){
                int temp=a[j];
                a[j]=a[j+1];
                a[j+1]=temp;
            }
        }
    }
}
```

运行结果：

```
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/Experiment/实验6$ ./最小等待时间
Enter the number of customers.
5
7 3 2 1 4
The minimum waiting time is 20.
```

字符统计

题目描述

有一段文章，共有 3 行文字，每行有 80 个字符。要求分别统计出其中英文大写字母、小写字母、空格、数字以及其他字符的个数。

代码：

```
#include<stdio.h>
#include<string.h>
int main(){
    char s[3][80];
    //大写字母个数，小写字母个数，空格个数，数字个数，其他字符个数
    int Upper=0, Lower=0, space=0, number=0, other=0;
    for(int i=0; i<3; i++){
        int j=0;
```

```
            do{
                scanf("%c",&s[i][j]);
                if(s[i][j]==' '){
                    space++;
                }else if (s[i][j]>=65 && s[i][j]<=90)
                {
                    Upper++;
                }else if (s[i][j]>=97 && s[i][j]<=122)
                {
                    Lower++;
                }else if (s[i][j]>=48 && s[i][j]<=57)
                {
                    number++;
                }else{
                    other++;
                }
            }while(s[i][j]!='\n');
        }
        printf("Upper: %d\n",Upper);
        printf("Lower: %d\n",Lower);
        printf("space: %d\n",space);
        printf("number: %d\n",number);
        printf("other: %d\n",other-3);    //减去三个'\n'
        return 0;
}
```

运行结果：

```
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/Experiment/实验6$ ./字符统计
Hello, I love you!
But,how about you?
yeah...
Upper: 3
Lower: 28
space: 5
number: 0
other: 7
```

杨辉三角

题目描述

杨辉三角，是二项式系数在三角形中的一种几何排列。在欧洲，这个表叫做帕斯卡三角形。帕斯卡（1623----1662）是在 1654 年发现这一规律的，比杨辉要迟 393 年，比贾宪迟 600 年。

代码：

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

```
int main(){
    int n;
    printf("Enter a number.\n");
    scanf("%d",&n);
    int a[n][n];
    //三角的两边赋值为 1
    for(int i=0; i<n; i++){
        a[i][i]=1;
        a[i][0]=1;
    }
    //从第三行开始，中间的每一个元素，等于该元素上面一个元素与左上元素之和
    for(int i=2; i<n; i++){
        for(int j=1; j<i; j++){
            a[i][j]=a[i-1][j]+a[i-1][j-1];
        }
    }
    for(int i=0; i<n; i++){
        for(int j=0; j<=i; j++){
            printf("%6d",a[i][j]);
        }
        printf("\n");
    }
    return 0;
}
```

运行结果：

```
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/Experiment/实验6$ ./PascalTriangle
Enter a number.
10
     1
     1     1
     1     2     1
     1     3     3     1
     1     4     6     4     1
     1     5    10    10     5     1
     1     6    15    20    15     6     1
     1     7    21    35    35    21     7     1
     1     8    28    56    70    56    28     8     1
     1     9    36    84   126   126    84    36     9     1
```

成绩排序

题目描述

给出班里某门课程的成绩单，请你按成绩从高到低对成绩单排序输出，如果有相同分数则名字字典序小的在前。

输入

第一行为 n (0 < n < 20)，表示班里的学生数目；

接下来的 n 行，每行为每个学生的名字和他的成绩, 中间用单个空格隔开。名字只包含字

母且长度不超过 20，成绩为一个不大于 100 的非负整数。
输出
n 行
把成绩单按分数从高到低的顺序进行排序并输出，每行包含名字和分数两项，之间有一个空格。
代码：

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
//定义学生结构体
struct student
{
	char name[20];
	float score;
	struct student *next_stu;
};
//定义头节点
struct student *stu_head;

void createStuList(int n);
void displayList();
void listSort(int n);

int main(){
	int n;
	printf("Input the number of students: ");
	scanf("%d",&n);
	createStuList(n);
	listSort(n);
	displayList();
	return 0;
}
//创建链表
void createStuList(int n){
	//current 指向当前节点，temp 用于指向上一个节点
	struct student *current, *temp;
	int i;
	//开辟一块内存空间给头节点
	stu_head=(struct student *)malloc(sizeof(struct student));
	if(stu_head==NULL){
		printf("Memory can not be allocated.");
	}else{
		printf("Input data for student 1: ");
		scanf("%s %f",stu_head->name,&stu_head->score);
```

```
            stu_head->next_stu=NULL;
            //将头节点地址赋给 temp
            temp=stu_head;
            for(i=2; i<=n; i++){
                //创建一个新的节点
                current=(struct student *)malloc(sizeof(struct student));
                if(current==NULL){
                    printf("Memory can not be allocated.");
                }else{
                    //给新节点赋值
                    printf("Input data for student %d: ",i);
                    scanf("%s %f",current->name,&current->score);
                    current->next_stu=NULL;
                    //将上一节点的 next 指针指向新节点
                    temp->next_stu=current;
                    //temp 指向新节点
                    temp=current;
                }
            }
        }
}
//链表排序
void listSort(int n){
    struct student *temp=stu_head;
    struct student *current, *next;
    if(temp->next_stu==NULL || temp==NULL){
        return;
    }else{
        //冒泡排序
        for(int i=0; i<n-1; i++){
            temp=stu_head;
            for(int j=0; j<n-i-1; j++){
                current=temp;
                next=temp->next_stu;
                //交换数据域
                if(current->score<next->score){
                    float score_t=current->score;
                    char name_t[20];
                    current->score=next->score;
                    next->score=score_t;
                    strcpy(name_t,current->name);
                    strcpy(current->name,next->name);
                    strcpy(next->name,name_t);
                }
```

```
                    temp=temp->next_stu;
                }
            }
        }
}
//打印链表
void displayList(){
    struct student *temp;
    if(stu_head==NULL){
        printf("List is empty.");
    }else{
        temp=stu_head;
        while (temp != NULL)
        {
            printf("name: %s score: %.1f\n",temp->name,temp->score);
            temp=temp->next_stu;
        }
    }
}
```

运行结果：

```
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/Experiment/实验6$ ./成绩排序
Input the number of students: 4
Input data for student 1: kitty 80
Input data for student 2: Hanmeimei 90
Input data for student 3: Joey 92
Input data for student 4: Tim 28
name: Joey score: 92.0
name: Hanmeimei score: 90.0
name: kitty score: 80.0
name: Tim score: 28.0
```

谁考了第 k 名

题目描述

在一次考试中，每个学生的成绩都不相同，现知道了每个学生的学号和成绩，求考第 k 名学生的学号和成绩。

输入

第一行有两个整数，分别是学生的人数 n（1≤n≤100），和求第 k 名学生的 k（1≤k≤n）。

其后有 n 行数据，每行包括一个学号（整数）和一个成绩（浮点数），中间用一个空格分隔。

输出

1 行

输出第 k 名学生的学号和成绩，中间用空格分隔。（注：请用%g 输出成绩）

代码：

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

```
#include<string.h>
#define MAXSIZE 100
typedef struct {
	char no[20];
	float score;
}stu;

typedef struct{
	stu *elem;
	int length;
}SqList;

int InitList(SqList *L);
void InputList(SqList *L, int n);
void DisplayList(SqList *L);
void SortList(SqList *L);

int main(){
	SqList *L;
	InitList(L);
	int n, k;
	printf("Input the total number of student and the rank you want: ");
	scanf("%d%d",&n,&k);
	InputList(L,n);
	SortList(L);
	printf("\n%s %g\n",L->elem[k-1].no,L->elem[k-1].score);
	//DisplayList(L);
	return 0;
}
//顺序表初始化
int InitList(SqList *L){
	L->elem=(stu *)malloc(sizeof(stu)*MAXSIZE);
	if (!L->elem)
	{
		exit(0);
	}
	L->length=0;
	return 1;
}
//输入顺序表的内容
void InputList(SqList *L, int n){
	for (int i = 0; i < n; i++){
		scanf("%s %f",L->elem[i].no,&L->elem[i].score);
		L->length++;
```

```
    }
}

void DisplayList(SqList *L){
    for(int i=0; i<L->length; i++){
        printf("%s %g\n",L->elem[i].no,L->elem[i].score);
    }
}
//选择排序
void SortList(SqList *L){
    for(int i=0; i<L->length-1; i++){
        int max=i;
        for(int j=i+1; j<L->length; j++){
            if(L->elem[j].score>L->elem[max].score){
                max=j;
            }
        }
        float temp=L->elem[i].score;
        L->elem[i].score=L->elem[max].score;
        L->elem[max].score=temp;
        char no_t[20];
        strcpy(no_t,L->elem[i].no);
        strcpy(L->elem[i].no,L->elem[max].no);
        strcpy(L->elem[max].no,no_t);
    }
}
```

运行结果：

```
Input the total number of student and the rank you want: 5 3
2001 67.8
2002 90.3
2003 61
2004 68.4
2005 73.9

2004 68.4
```

整数奇偶排序

题目描述

给定 10 个整数的序列，要求对其重新排序。排序要求:

1.奇数在前，偶数在后；

2.奇数按从大到小排序；

3.偶数按从小到大排序。

输入

1 行：

输入一行，包含 10 个整数，彼此以一个空格分开，每个整数的范围是大于等于 0，小于等

于 100。
输出
1 行：
按照要求排序后输出一行，包含排序后的 10 个整数，数与数之间以一个空格分开。

代码：

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

void DisplaySeq(int *);
void SortMax(int *, int);
void SortMin(int *, int);

int main(){
	int seq_origin[10], seq_sorted[10];
	int odd[10], even[10];
	for (int i = 0; i < 10; i++)
	{
		scanf("%d",&seq_origin[i]);
	}
	int count_o=0, count_e=0;
	//判断奇偶，分别放到奇数数组和偶数数组
	for(int i=0; i<10; i++){
		if(seq_origin[i]%2==0){
			even[count_e]=seq_origin[i];
			count_e++;
		}else if (seq_origin[i]%2!=0)
		{
			odd[count_o]=seq_origin[i];
			count_o++;
		}
	}
	//分别给奇数组和偶数组排序
	SortMax(odd, count_o);
	SortMin(even, count_e);
	//将奇数数组与偶数数组输入到新数组中
	for (int i = 0; i < count_o; i++)
	{
		seq_sorted[i]=odd[i];
	}
	for (int i =0; i < 10; i++)
	{
		seq_sorted[i+count_o]=even[i];
	}
```

```
    DisplaySeq(seq_sorted);
    return 0;
}

void DisplaySeq(int *seq){
    for (int i = 0; i < 10; i++)
    {
        printf("%d ",seq[i]);
    }
    printf("\n");
}
//冒泡排序
void SortMax(int *seq, int n){
    for(int i=0; i<n-1; i++){
        for(int j=0; j<n-i-1; j++){
            if(seq[j]<seq[j+1]){
                int temp=seq[j];
                seq[j]=seq[j+1];
                seq[j+1]=temp;
            }
        }
    }
}
//选择排序
void SortMin(int *seq, int n){
    for(int i=0; i<n-1; i++){
        int min=i;
        for(int j=i+1; j<n; j++){
            if(seq[j]<seq[min]){
                int temp=seq[j];
                seq[j]=seq[min];
                seq[min]=temp;
            }
        }
    }
}
```

运行结果：

```
(base) daniel@DESKTOP-PG8AU51:~/WPS/计算机/程序设计基础/Experiment/实验6$ ./整数奇偶排序
4 7 3 13 11 12 0 47 34 98
47 13 11 7 3 0 4 12 34 98
```

发现问题与分析：
备注：

说明：

1、 各项记录应完备详尽。

2、 各栏目如果长度不够可以自行调整大小，但应注意排版的美观。

3、 可将实现过程的代码、运行效果等内容以帖图方式放在相应栏目。