

第六章 数组

蒋玉茹

问题1

- 输入两个人的年龄，按照从小到大的顺序输出

问题2

- 输入三个人的年龄，按照从小到大的顺序输出

问题3

- 输入10个人的年龄，按照从小到大的顺序输出

- 这10个人的年龄怎样存储在内存中？
- 如何排序？

数组

- 定义：一组具有同一属性的数据
- “数”组成的“组”
- “猫”和“狗”可以组成一组吗？
- 整数和小数可以组成一组吗？

定义一维数组

格式

类型符 数组名[常量表达式];

数组元素的类型

数组长度：所含
元素个数

举例

```
int a[10];
```

数组名：a
数组长度：10
数组元素的类型：int

说明

- 1、数组名的命名规则应遵循标识符的命名规则。**
- 2、数组名后是方括号，而非圆括号。**
- 3、常量表达式表示元素的个数，即数组长度。**

数组元素的使用（引用）

- 数组元素？
- 一般形式：数组名[下标]
- 如：a[0]，a[1]，……
- 使用：a[0]=5； t=a[0]；

问题3

- 输入10个人的年龄，按照从小到大的顺序输出
- 这10个人的年龄怎样存储在内存中？

□ `int age[10];`

问题3

- 输入10个人的年龄，按照从小到大的顺序输出
- 这10个人的年龄怎样存储在内存中？
 - `int age[10];`
- 如何输入这N个人的年龄并存储在内存中？

如何输入10个人的年龄并存储在内存中？

□ `int age[10];`

□ `age[i]`

- 这里*i*大于等于0，小于10
- `age[i]`是一个变量
- `age[i]`的类型是int

如何输入10个人的年龄并存储在内存中？

- ☐ scanf (“%d” , &age[0]) ;
- ☐ scanf (“%d” , &age[1]) ;
- ☐ scanf (“%d” , &age[2]) ;
- ☐ scanf (“%d” , &age[3]) ;
- ☐
- ☐ scanf (“%d” , &age[9]) ;

一维数组元素的输入

```
for (i=0; i<10; i++)  
{  
    scanf ("%d", &age[i]);  
}
```

一维数组元素的输出

```
for (i=0; i<10; i++)  
{  
    printf("%d",age[i]);  
}
```

练习

- 输入N个人的年龄，找出最大的年龄
- 输入N个人的年龄，找出最小的年龄
- 输入N个人的年龄，将最大的年龄和最小的年龄交换位置之后，输出

调试及策略

- 观察变量的值，对比实际的值（在观察窗口中看到的值）和理想的值是否一致，如果出现不一致，则错误就隐藏在附近
- 在合适的位置设置断点，以快速发现出错的范围
 - 在输入之后，开始正式计算之前
 - 在每个小的计算步骤完成之后，下一个计算步骤开始之前
- 发现出错的范围之后，重新启动调试，利用单步跟踪（F10）细心发现错误所在。

问题3

- 输入N个人的年龄，按照从小到大的顺序输出
- 这N个人的年龄怎样存储在内存中？
- 如何输入这N个人的年龄并存储在内存中？
- 如何输出N个人的年龄？
- 如何排序？

如何从数组中找到值最大的数组元素

交换

- 有两个杯子中分别装着可乐和芬达，请将两个杯子中的饮料交换



交换

- 有两个杯子中分别装着咖啡和牛奶，还有一个空杯子，请将两个杯子中的饮料交换



如何让两个数的值互换

如何给数组中的元素从大到小排序

- 挑一个最大的放到排头
- 挑剩下的最大的放到剩下的排头

总结：选择排序法

- for ($i=1; i < N; i++$)
 - $\text{max} = a[i];$
 - $\text{maxIndex} = i;$
 - 找到 $i \sim N$ 元素中最大者 $a[k]$, 令 $\text{maxIndex} = k$
 - 将 $a[\text{maxIndex}]$ 与 $a[i]$ 交换位置

选择排序法程序

```
#include<stdio.h>
int main()
{
    int a[10]={2, 5, 3, 8, 6, 4, 1, 0, 9, 7};
    int i=0, j=0, max=0, maxIndex=0, t=0;
    for(i=0; i<10; i++)
    {
        max=a[i];
        maxIndex=i;
        for(j=i; j<10; j++)
        {
            if(max<a[j])
            {
                max=a[j];
                maxIndex=j;
            }

            t=a[i];
            a[i]=a[maxIndex];
            a[maxIndex]=t;
        }

        for(i=0; i<10; i++)
        {
            printf("%d ", a[i]);
        }

        return 0;
    }
}
```


冒泡排序法

N个数的排序

- 自底向上，逐步实现
 - 发现原子功能，逐步扩展

从小到大排序

49 38 65 97 76 13 27

排序结果



从小到大排序

49 38 65 97 76 13 27

排序结果



从小到大排序

49 38

排序结果



原子功能

□ 两个数 x, y 从小到大排序

□ $a[i]$ 和 $a[i+1]$ 的从小到大排序

计算机的原始智商和能力

□ 智力不高

- 每次只能比较两个数
- 每次只能交换两个数

□ 穷举

从小到大排序

49 38 65 97 76 13 27

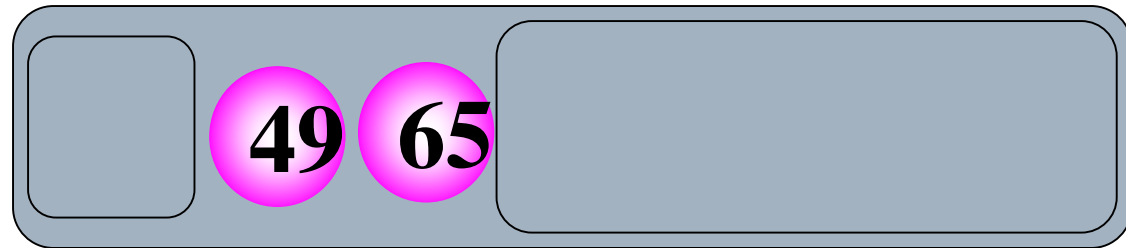
从小到大排序

49 38

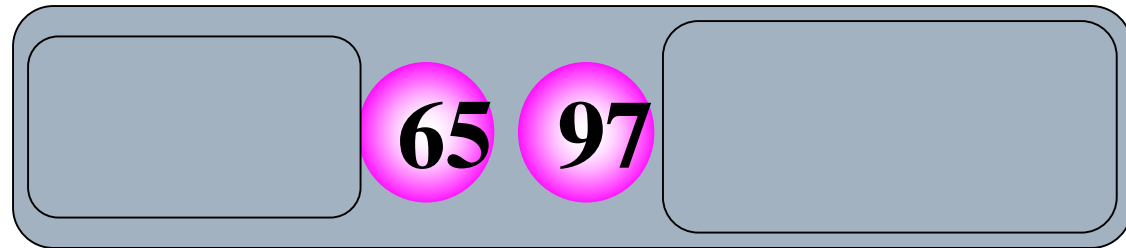
从小到大排序

38 49

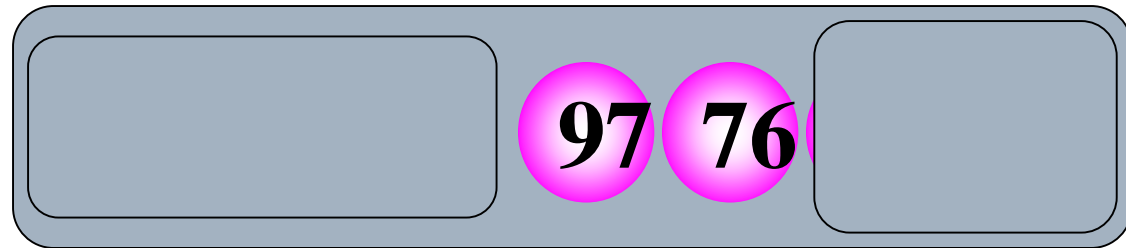
从小到大排序



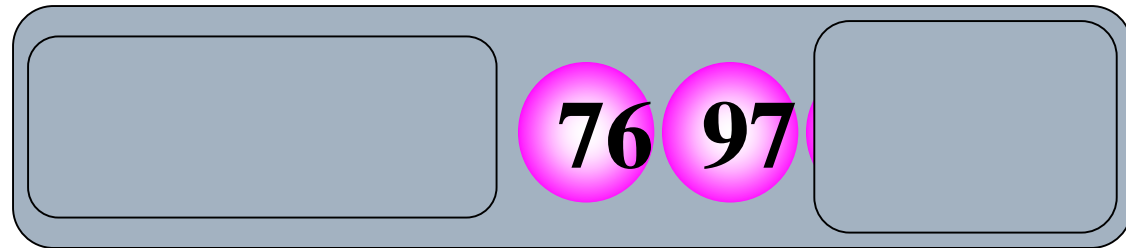
从小到大排序



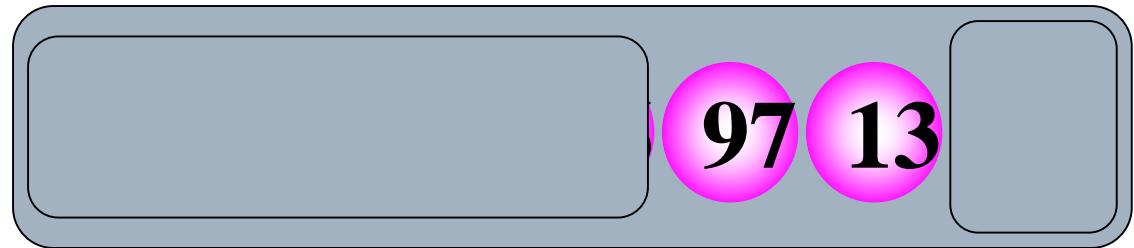
从小到大排序



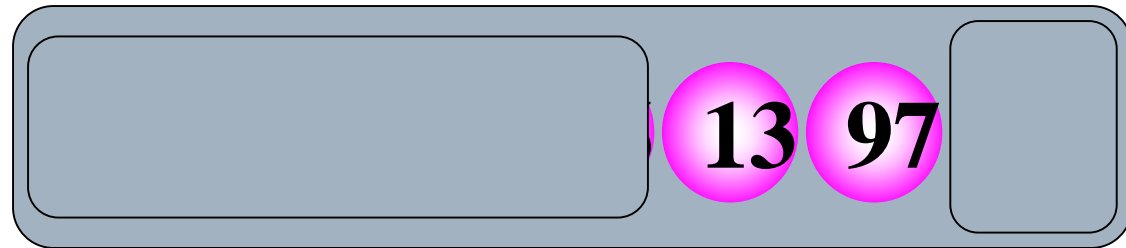
从小到大排序



从小到大排序



从小到大排序



从小到大排序



从小到大排序



从小到大排序



从小到大排序

38 49 65 76 13 27 97

这个过程就是冒泡

编程实现上述过程

一遍冒泡的程序

```
#include<stdio.h>
int main()
{
    int a[10]={2, 5, 3, 8, 6, 4, 1, 0, 9, 7};
    int i=0, j=0, max=0, maxIndex=0, t=0;
    for(i=0; i<9; i++)
    {
        if(a[i]>a[i+1])
        {
            t=a[i];
            a[i]=a[i+1];
            a[i+1]=t;
        }
    }

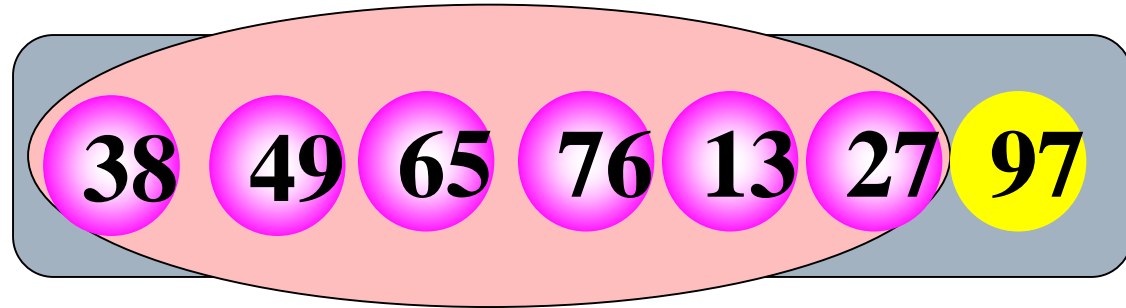
    for(i=0; i<10; i++)
    {
        printf("%d ", a[i]);
    }

    return 0;
}
```

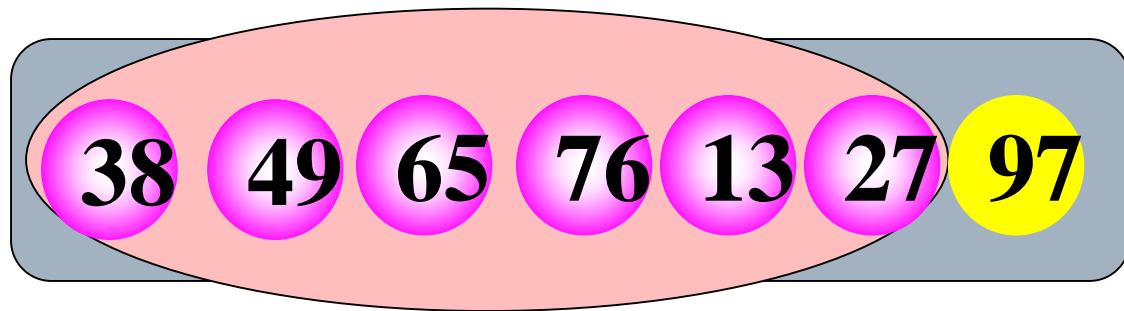
从小到大排序

38 49 65 76 13 27 97

从小到大排序



从小到大排序



编写完整的冒泡排序的程序

总结：冒泡排序算法（从小到大）

□ for ($i=0; i<N; i++$)

- 将 $1 \sim N-i$ 元素，相邻两个元素依次进行比较，如果前者大，则两者交换位置

练习

- 将一个数组中的值按照原始顺序的逆序重新存放，并输出逆序后的数组。 输入格式第一行有一个正整数 n ，表示原始的整数序列长度为 n （ n 不超过50）。第二行有 n 个整数，表示原始的整数序列。 第三行输出逆序的整数序列。每个整数后输出一个空格。请注意行尾输出换行。

练习

- 读入n名学生的成绩，将某一给定分数的学生人数输出。输入格式第1行：n（不超过100）第2行：n名学生的成绩，相邻两数字用一个空格间隔。第3行：给定一个分数。输出具有给定分数的学生人数，如果没有该分数则输出0。

一维字符数组

一维字符数组

□ 定义: `char cs[5];`

□ 使用: `cs[1] = 'a';`

一维字符数组的输入

```
for (i=0; i<10; i++)  
{  
    scanf ("%c", &cs[i]);  
}
```

一维字符数组的输出

```
for (i=0; i<10; i++)  
{  
    printf("%c", cs[i]);  
}
```

练习

- 输入一个单词，将其加密后输出。
- 要求：
 - 第一行输入一个单词，如 “beijing”
 - 第二行将输入的单词加密后输出，加密的方式是将每个字母变成字母表中它后面的那个字母，比如 ‘a’ 变成 ‘b’，‘z’ 变成 ‘a’。

练习【加密程序升级版】

- 输入一段文字，将其加密后输出。
- 要求：
 - 第一行输入一段文字，如 “I love BISTU.”
 - 第二行将输入的文字加密后输出，加密的方式是将每个字母变成字母表中它后面的那个字母，比如‘a’变成‘b’，‘z’变成‘a’。

练习【加密程序升级版】

- 输入一段文字，将其加密后输出。
- 要求：
 - 第一行输入一段文字
 - 第二行输入一个整数 n
 - 第三行将输入的文字加密后输出，加密的方式是将每个字母变成其后面的第 n 个字母。

二维数组

表格

	语文	数学	英语	程序设计基础
001	88	87	86	89
002	98	97	96	95
003	76	77	78	79

矩阵

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 22 & 33 & 44 & 55 \end{bmatrix}$$

二维数组的定义

□ 一般形式:

类型名 数组名[常量表达式][常量表达式];

数组元素类型 数组名[行数][列数];

□ 举例:

```
float score[5][5];
```

二维数组的使用

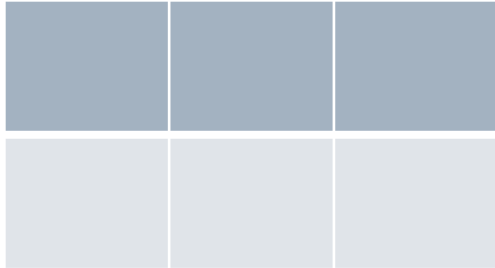
□ 二维数组元素的使用：

数组名[行下标][列下标]

□ 举例：

score[1][1]

左侧这个二维数组怎样定义？



□ `int a[2][3];`

□ `a[1][1]=3;` 对左侧这个二维数组的影响是什么？

	3	

□ $a[1][1]=3$; 对左侧这个二维数组的影响是什么?

二维数组的初始化

```
int  
    a[2][3]={ {1, 3, 5}, {2, 4, 6} } ;
```

1	3	5
2	4	6

怎样输出二维数组的所有数组元素？

二维数组元素的输出

```
for (i=0; i<HANGSHU; i++)  
{  
    for (j=0; j<LIESHU; j++)  
    {  
        printf("%d ", score[i][j]);  
    }  
    printf("\n");  
}
```

怎样输入二维数组的所有数组元素？

二维数组元素的输入

```
for (i=0; i<HANGSHU; i++)  
{  
    for (j=0; j<LIESHU; j++)  
    {  
        scanf ("%d", &score[i][j]);  
    }  
}
```

练习

已知一个矩阵和一个数，求他们的乘积

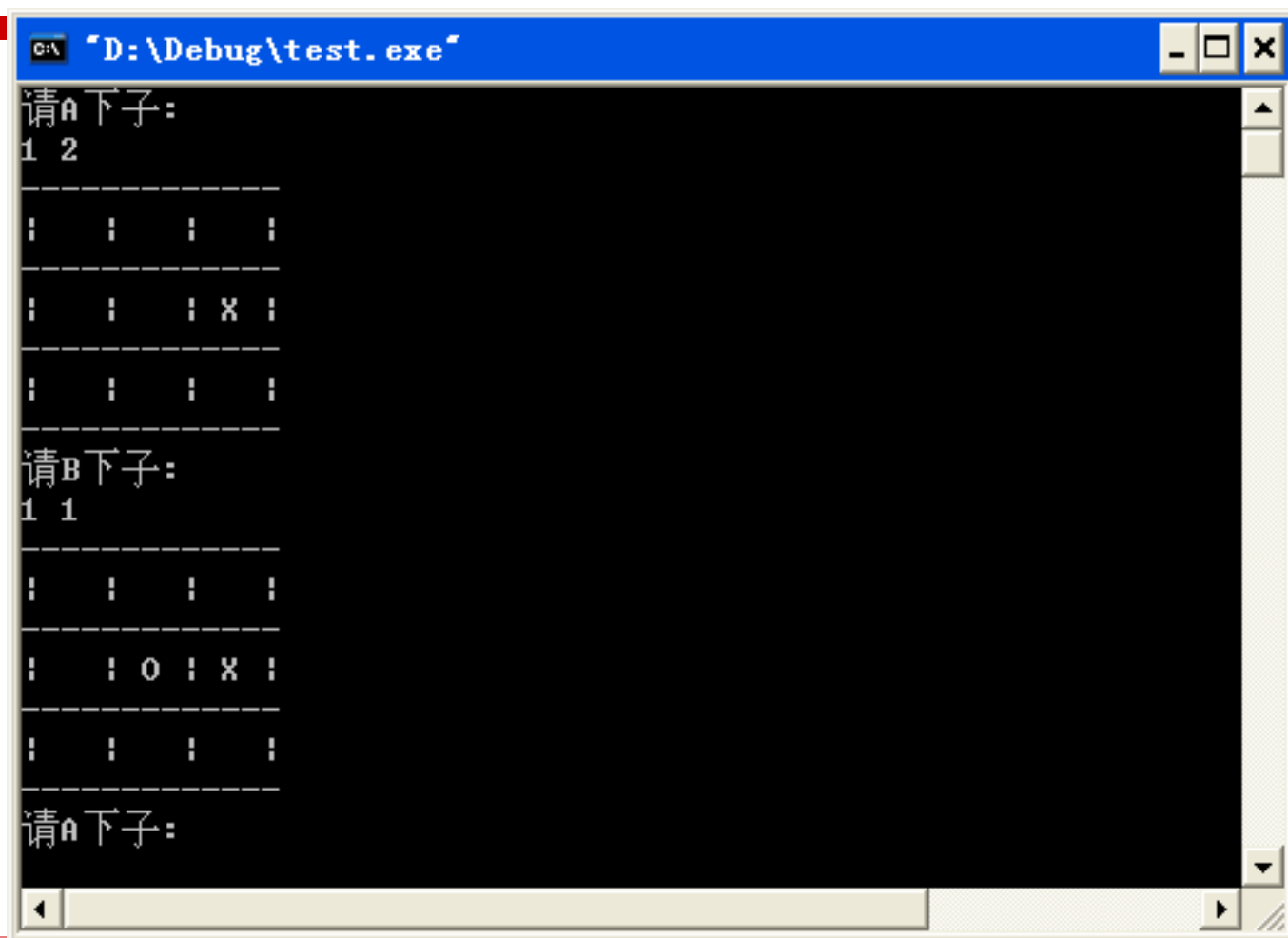
$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times 5 = \begin{bmatrix} 5 & 10 & 15 \\ 20 & 25 & 30 \end{bmatrix}$$

练习

求两个矩阵的和，如：

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} + \begin{bmatrix} 6 & 3 & 4 \\ 5 & 1 & 2 \end{bmatrix} = \begin{bmatrix} 7 & 5 & 7 \\ 9 & 6 & 8 \end{bmatrix}$$

井字棋



```
C:\> "D:\Debug\test.exe"

请A下子:
1 2

-----
|   |   |   |
-----
|   |   | X |
-----
|   |   |   |
-----

请B下子:
1 1

-----
|   | O | X |
-----
|   |   |   |
-----
|   |   |   |
-----

请A下子:
```

字符串

字符串

- ❑ 字符串也存储在字符数组中
- ❑ 字符串的最后一个字符是 ‘\0’
- ❑ 字符串 “Hello” 占几个字符？
- ❑ `char name[10]="jiangyuru";`

字符串的输入和输出

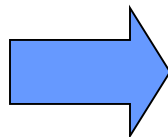
方式一

逐个字符输入/输出。（用格式符**%c**）

方式二

将整个字符串一次输入或输出。
（在**scanf/printf**函数中使用格式符**%s**）

```
char c[10];  
scanf("%s",c);  
printf("%s",c);
```



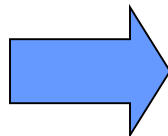
happy↵
happy

good bye↵
good

方式三

将整个字符串一次输入或输出。
（使用**gets/puts**函数）

```
char c[10];  
gets(c);  
puts(c);
```



happy↵
happy

good bye↵
good bye

练习

□ 输入一个人的姓名，并打印输出

练习

- 输入十个人的姓名，并按序输出。
- 输入十个学生的姓名，并按照逆序输出。
- 每个人的名字不超过20个字符。

字符串函数

```
#include <string.h>
```

1、字符串连接

strcat(字符串数组1, 字符串数组2);

将字符串**2**连接到字符串**1**之后，结果放在字符串数组**1**中。函数调用后的返回值为字符串数组**1**的地址。

字符串函数

```
#include <string.h>
```

2、字符串复制

```
strcpy(字符数组1, 字符串2);
```

例如:

```
char str1[10],str2[] = {"china"};  
strcpy(str1,str2);
```

字符串函数

<string.h>

#include

3、字符串比较

strcmp(字符串1, 字符串2);

比较的规则与其他语言中的规则相同:

逐个比较每个字符（按**ASCII**码值比较）,直到出现不同的 字符或遇到 ‘\0’ 为止。

比较结果有函数值带回。

- (1) 若字符串**1**=字符串**2**, 函数值为**0**;
- (2) 若字符串**1**>字符串**2**, 函数值为一个正整数;
- (3) 若字符串**1**<字符串**2**, 函数值为一个负整数;

练习

□ 输入十个人的姓名，并按升序输出

练习

求一个矩阵的转置矩阵

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$



$$\begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

练习

- 一个班共有6名学生，已知每个学生有5门课程的成绩
- 请输出每个学生的平均成绩
- 请输出每门课程的平均成绩
- 请输出学生的最高平均成绩和序号
- 请输出课程的最高平均成绩和序号

练习

求两个矩阵的积，如：

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 & 5 \\ 5 & 3 \\ 8 & 1 \end{bmatrix} = \begin{bmatrix} 35 & 14 \\ 77 & 41 \end{bmatrix}$$

练习

- 求一个 $n \times n$ 矩阵的对角线元素之和，其中 n 在1到100之间。
- 输入第一行为一个正整数 n
- 输入第二行为 $n \times n$ 个数，用空格分隔
- 输出为正对角线元素之和

练习

- 求一个 $n \times n$ 矩阵的对角线元素之和，其中 n 在1到100之间。
- 输入第一行为一个正整数 n
- 输入第二行为 $n \times n$ 个数，用空格分隔
- 输出为副对角线元素之和

$$\begin{bmatrix} 35 & 14 \\ 77 & 41 \end{bmatrix}$$

练习

- 求矩阵的鞍点。鞍点位置上的元素在该行中值最大，在该列中值最小。如右上角矩阵的鞍点为35.
- 输入：前两个数为行数和列数，其后是矩阵中的元素
- 输出：如果没有鞍点，输出N0；如果有输出格式为：[值1，行下标1，列下标1]， [值2，行下标2，列下标2]……