

西 安 邮 电 大 学  
(计算机学院)

课内实验报告

实验名称: 实验 4

专 业: 网络工程

班 级: 2002

姓 名: 李玲

学 号: 04192065

日 期: 2021 年 4 月 22 日

## 一. 问题描述

### ● 基础类

- (1) 从键盘输入 10 个学生的成绩存储在数组中，求成绩最高者的序号和成绩，以及平均成绩。
- (2) 对整型数组进行逆置，即：将数组中的所有元素镜像对调。
- (3) 编写程序，对从键盘输入 6 个字符串进行排序，并将排序后的结果输出。
- (4) 将 2 个按升序排列的数列，仍按升序合并存放到另一个数组中，要求每个数都一次到位，不得在新数组中重新排序。

### ● 提升类

- (1) 对于任意的  $M \times M$  方阵，编程按照下面的规律对其初始化并输出。如  $7 \times 7$  的矩阵。

1	2	3	4	5	6	7
<b>24</b>	25	26	27	28	29	8
23	40	41	42	43	30	9
22	39	48	49	44	31	10
21	38	47	46	45	32	11
20	37	36	35	34	33	12
19	18	17	16	15	14	13

- (2) 学生的序号输入  $n$  名学生的成绩，按照分数由高到低的顺序输出学生的名次、序号和成绩，要求成绩相同的学生具有相同的名次。

输入： 学生人数和  $n$  名学生的成绩

输出： 按分数由高到低的顺序输出学生的名次、序号和成绩

输入： 5

22 42 92 82 92

输出： 1:3, 92

1:5, 92

3:4, 82

4:2, 42

4:1, 22

## ● 综合类

**题目：**特大整数的精确相加。特大整数用长整型也存不下，如果用双精度实型存储则会造成误差，可以用字符数组存储所有位，再按十进制由低到高逐位相加，同时考虑进位。

### 算法分析：

① 初始化：将两个特大整数输入两个字符数组，将两个字符数组的各元素右移，使最低位的元素位置对齐，高位补 0，为了存储最高位的进位，位数多的数最高位前也应补一个 0。

② 从最低位对应的数组元素开始将数字字符转换为整型数据相加，因为数字字符 '0' 对应的 ASCII 值是 48，则：整型数据  $1+2$ 。相当于  $(\text{'1'}-48)+(\text{'2'}-48)$ ，即  $\text{'1'}+\text{'2'}-96$ 。

③ 将和整除以 10，余数就是该位的结果，并转换为字符存入该位，商就是进位数。

④ 再对高一位对应的数组元素操作，将该位数字字符转换为整型相加，并与低位的进位数相加，将和整除以 10，余数就是该位的结果，商就是本位的进位数。

⑤ 重复④直到最高位。

⑥ 如果最高位相加时进位数大于 0 则将此进位数转换为字符存入最高位上 1 位。

测试数据 1:

123456789<回车>

23456789<回车>

146913578

测试数据 2 (最高位有进位):

999999999<回车>

999999999<回车>

1999999998

## 二. 代码实现

### ● 基础类

```
shuzuqihe.cpp
1  #include<stdio.h>
2  int main()
3  {
4      int a[10];
5      int i,max,k;
6      float sum,ever;
7
8      for(i=0;i<10;i++){
9          scanf("%d",&a[i]);
10     }
11
12     for(i=0;i<10;i++){
13         if(a[i]>a[i+1]){
14             max=a[i];
15             k=i;
16         }
17         sum+=a[i];
18     }
19     printf("%d %d\n",k,max );
20     ever = sum/10;
21     printf("%f",ever);
22     return 0;
23 }
```

(1)

```

1-2.cpp
#include<stdio.h>
int main() {
    int n,i,j,temp;
    scanf("%d",&n);
    int arr[n]= {0};

    for(i = 0; i < n; i++) {
        scanf("%d",&arr[i]);
    }
    printf("\n");
    for(i = 0; i < n/2; i++) {
        j = n-1-i;
        temp = arr[i];
        arr[i] = arr[j];
        arr[j] = temp;
        j--;
    }

    printf("逆置后的数组为: \n");
    for(i = 0; i < n; i++) {
        printf("%d ",arr[i]);
    }
    return 0;
}

```

(2)

(3)

```

13.cpp
1  #include<stdio.h>
2  #include<string.h>
3  int main(void)
4  {
5      printf("请输入六个字符串\n");
6      char str[6][10] = {0};
7      for (int i = 0; i < 6; i++) {
8          scanf("%s",str[i]);
9      }
10     for (int i = 0; i < 6 - 1; i++) {
11         for (int j = 0; j < 6 - 1 - i; j++) {
12             if (strcmp(str[j], str[j + 1]) > 0) {
13                 char strtemp[10] = {0};
14                 strcpy(strtemp, str[j]);
15                 strcpy(str[j], str[j + 1]);
16                 strcpy(str[j + 1], strtemp);
17             }
18         }
19     }
20     printf("排好序后字符串顺序为:\n");
21     for (int i = 0; i < 6; i++) {
22         printf("%s\n",str[i]);
23     }
24     return 0;
25 }

```

```

13.cpp 14.cpp
1  #include<stdio.h>
2  #include<string.h>
3  #include<math.h>
4
5  void bin(int n,int a[],int m,int b[]) {
6      int c[200];
7      int i,j;
8
9      for(i=0; i<n; i++) {
10         c[i]=a[i];
11     }
12     for(i=n; i<n+m; i++) {
13         c[i]=b[i-n];
14     }
15     for(i=0; i<m+n; i++) {
16         for(j=0; j<m+n; j++) {
17             if(c[i]<c[j]) {
18                 int t;
19                 t=c[i];
20                 c[i]=c[j];
21                 c[j]=t;
22             }
23         }
24     }
25     for(i=0; i<n+m; i++) {
26         printf("%4d",c[i]);
27     }
28 }

```

(4)

● 提升类

(1)

```

30 int main() {
31     int a[100],b[100];
32     int m,n;
33
34     printf("请输入数组a元素个数 n: ");
35     scanf("%d",&n);
36     printf("请输入 %d 个数组元素: ",n);
37
38     int i;
39
40     for(i=0; i<n; i++) {
41         scanf("%d",&a[i]);
42     }
43     printf("请输入数组b元素个数 m: ");
44     scanf("%d",&m);
45     printf("请输入 %d 个数组元素: ",m);
46     for(i=0; i<m; i++) {
47         scanf("%d",&b[i]);
48     }
49     bin(n,a,m,b);
50     return 0;
51 }
52

```

```

13.cpp [1] 21.cpp
1  #include<stdio.h>
2  #include<stdlib.h>
3  #include<string.h>
4
5  int mun[20][20];
6  int main() {
7      int i,j,n;
8      scanf("%d",&n);
9      int k = 1;
10     int num[20][20]={0};
11     for(i = 0; i <= n/2; i++) {
12         for(j = i; j <= n-i-1; j++) {
13             num[i][j] = k++;
14         }
15         for(j = i+1; j <= n-i-1; j++) {
16             num[j][n-i-1] = k++;
17         }
18         for(j = n-i-2; j >= i; j--) {
19             num[n-i-1][j] = k++;
20         }
21         for(j = n-i-2; j > i; j--) {
22             num[j][i] = k++;
23         }
24     }
25     for(i = 0; i < n; i++) {
26         for(j = 0; j < n; j++) {
27             printf("%4d",num[i][j]);
28         }
29         printf("\n");
30         printf("\n");
31     }
32     return 0;
33 }

```

(2)

```

22.cpp
1  #include<stdio.h>
2  int main() {
3      int i,j,n,max,v,w=0,m=0,a[100]= {0};
4
5      scanf("%d",&n);
6      for(i=0; i<n; i++) {
7          scanf("%d",&a[i]);
8      }
9      for(i=0; i<n; i++) {
10         max=0;
11         for(j=0; j<n; j++) {
12             if(a[j]>max) {
13                 max=a[j];
14                 v=j+1;
15             }
16         }
17         if(m!=max) {
18             w++;
19         }
20         m=max;
21         printf("%d:%d %d",w,v,max);
22         printf("\n");
23         a[v-1]=0;
24     }
25     return 0;
26 }

```

### 三.测试

- 基础类

(1)

```
D:\C\1-2.exe
7
2 4 1 5 3 6 8

逆置后的数组为:
8 6 3 5 1 4 2

-----
Process exited after 11.26 seconds with return value 0
请按任意键继续. . .
```

(2)

```
D:\C\shuzuqiuh.exe
1 3 2 4 5 8 7 5 3 3
9 3
4.100000

-----
Process exited after 15.62 seconds with return value 0
请按任意键继续. . .
```

(3)

```
D:\C\13.exe
请输入六个字符串
aaaaa
ccccc
fffff
bbbbbb
ddd
ssss
排好序后字符串顺序为:
aaaaa
bbbbbb
ccccc
ddd
fffff
ssss

-----
Process exited after 31.4 seconds with return value 0
请按任意键继续. . .
```



(4)

```
D:\C\14.exe
请输入数组a元素个数 n: 6
请输入 6 个数组元素: 1 4 5 6 7 8
请输入数组b元素个数 m: 3
请输入 3 个数组元素: 8 9 14
1 4 5 6 7 8 8 9 14
-----
Process exited after 25.88 seconds with return value 0
请按任意键继续. . .
```

● 提升类

(1)

```
D:\C\21.exe
7
1 2 3 4 5 6 7
24 25 26 27 28 29 8
23 40 41 42 43 30 9
22 39 48 49 44 31 10
21 38 47 46 45 32 11
20 37 36 35 34 33 12
19 18 17 16 15 14 13
-----
Process exited after 1.836 seconds with
请按任意键继续. . .
```

(2)

```
D:\C\22.exe
6
87 80 67 82 90 86
1:5 90
2:1 87
3:6 86
4:4 82
5:2 80
6:3 67
-----
Process exited after 14.31 seconds with return value 0
请按任意键继续. . .
```

四. 总结

1. 实验过程中遇到的问题及解决办法;

对指针不太熟练，多看多练多问

2. 对设计及调试过程的心得体会。  
感受到了指针的乐趣，写代码也更有心得