

---

## 5 数组实验

### 5.1 实验目的

- (1) 掌握数组的说明、初始化和使用。
- (2) 掌握一维数组作为函数参数时实参和形参的用法。
- (3) 掌握字符串处理函数的设计，包括串操作函数及数字串与数之间转换函数实现算法。
- (4) 掌握基于分治策略的二分查找算法和选择法排序算法的思想，以及相关算法的实现。

### 5.2 实验内容及要求

#### 5.2.1 源程序改错

下面是用来将数组 a 中元素按升序排序后输出的源程序。分析源程序中存在的问题，并对源程序进行修改，使之能够正确完成任务。

源程序

```
1 #include<stdio.h>
2 int main(void)
3 {
4     int a[10] = {27, 13, 5, 32, 23, 3, 17, 43, 55, 39};
5     void sort(int [],int);
6     int i;
7     sort(a[0],10);
8     for(i = 0; i < 10; i++)
9         printf("%6d",a[i]);
10    printf("\n");
11    return 0;
12 }
13 void sort(int b[], int n)
14 {
15     int i, j, t;
16     for (i = 0; i < n - 1; i++)
```

---

```

17      for (j = 0; j < n - i - 1; j++)
18          if(b[j] < b[j+1])
19              t = b[j], b[j] = b[j+1], b[j+1] = t;
20 }

```

**解答：**

1) 错误修改：

- 1) 第 7 行 sort()函数第第一个参数错误，参数应为地址，a[0]是一个数，正确的形式为：sort(a, 10);
- 2) 第 18 行，语义错误，题目中的意思是降序排序，正确顶形式为：if(b[j] > b[j+1])

### 5.2.2 源程序完善、修改、替换

(1) 下面的源程序用于求解瑟夫问题：M 个人围成一圈，从第一个人开始依次从 1 至 N 循环报数，每当报数为 N 时报数人出圈，直到圈中只剩下一个人为止。请在源程序中的下划线处填写合适的代码来完善该程序。

源程序：

```

#include<stdio.h>

#define M 10

#define N 3

int main(void)
{
    int a[M], b[M]; /* 数组 a 存放圈中人的编号，数组 b 存放出圈人的编号 */
    int i, j, k;

    for(i = 0; i < M; i++)          /* 对圈中人按顺序编号 1—M */
        a[i] = i + 1;

    for(i = M, j = 0; i > 1; i--){
        /* i 表示圈中人个数，初始为 M 个，剩 1 个人时结束循环；j 表示当前报数人的位置 */

        for(k = 1; k <= N; k++)      /* 1 至 N 报数 */

            if(++j > i - 1) j = 0; /* 最后一个人报数后第一个人接着报，形成一个圈 */
    }
}

```

---

```
    b[M-i] = j? (b[M - i] = a[j - 1]) : (b[M-i]=a[i-1]); /* 将报数
为 N 的人的编号存入数组 b */
```

```
    if(j)
    for(k = --j; k < i; k++) /* 压缩数组 a, 使报数为 N 的人出圈 */
        a[k]=a[k+1];
    }

    for(i = 0; i < M - 1; i++) /* 按次序输出出圈人的编号 */
        printf("%6d", b[i]);

    printf("%6d\n", a[0]); /* 输出圈中最后一个人的编号 */

    return 0;
}
```

(2) 上面的程序中使用数组元素的值表示圈中人的编号, 故每当有人出圈时都要压缩数组, 这种算法不够精炼。如果采用做标记的办法, 即每当有人出圈时对相应数组元素做标记, 从而可省掉压缩数组的时间, 这样处理效率会更高一些。因此, 请采用做标记的办法修改 (1) 中的程序, 并使修改后的程序与 (1) 中的程序具有相同的功能。

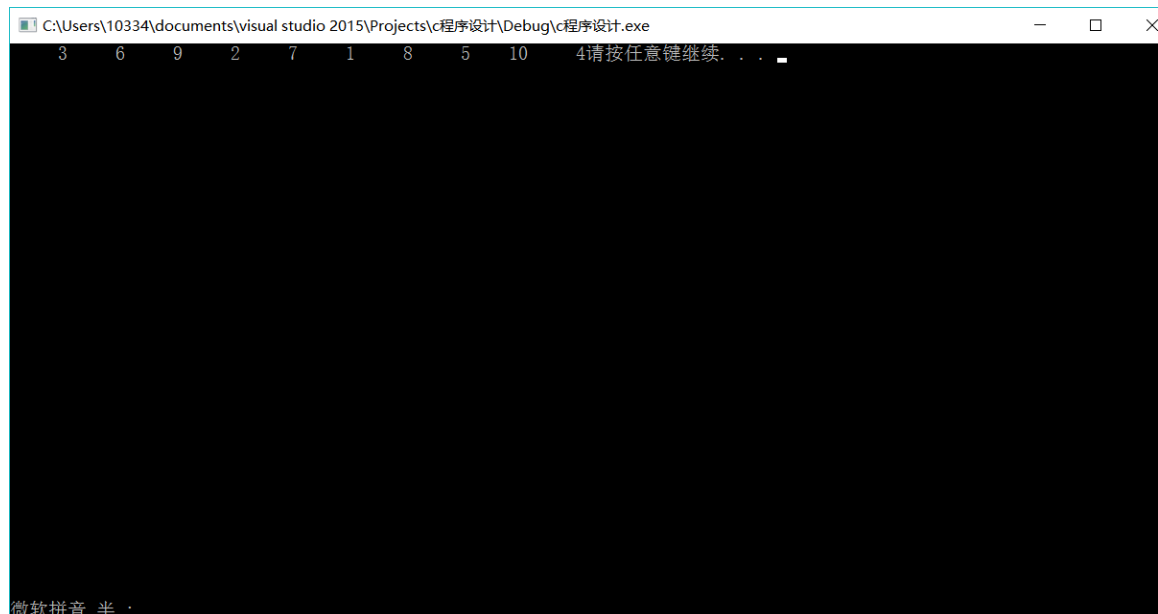
**解答:**

```
1.  #include<stdio.h>
2.  #define M 10
3.  #define N 3
4.  int main(void)
5.  {
6.      int a[M], b[M];
7.      int i, j, k;
8.      for (i = 0; i < M; i++)
9.          a[i] = i + 1;
10.     for (i = M, j = 0; i > 1; i--) {
11.         for (k = 1; k <= N; k++)
12.         {
```

---

```
13. while (!a[j])
    i.    ++j;
14. if (j++ >= M - 1) j = 0;
15. }
16. if (j)
17. {
18. b[M - i] = a[j - 1];
19. a[j - 1] = 0;
20. }
21. else
22. {
23. int t;
24. for (t = M - 1; a[t] == 0; t--);
25. b[M - i] = a[t];
26. a[t] = 0;
27. }
28. }
29. for (i = 0; i < M - 1; i++)
30. printf("%6d", b[i]);
31. for (i = 0; i < 10; i++)
32. if (a[i] != 0)
33. printf("%6d", a[i]);
34. system("pause");
35. return 0;
36. }
```

ii. 运行结果:



### 5.2.3 跟踪调试源程序

在下面所给的源程序中，函数 `strncat(s,t,n)` 本来应该将字符数组 `t` 的前 `n` 个字符连接到字符数组 `s` 中字符串的尾部。但函数 `strncat` 在定义时代码有误，不能实现上述功能。请按下面的要求进行操作，并回答问题和排除错误。

(1) 单步执行源程序。进入函数 `strncat` 后观察表达式 `s`、`t` 和 `i`。当光条落在 `for` 语句所在行时，`i` 为何值？当光条落在 `strncat` 函数块结束标记（右花括号 `}`）所在行时，`s`、`t` 分别为何值？

(2) 分析函数出错的原因，排除错误，使函数正确实现功能，最后写出程序的输出结果。

源程序：

- 1) `#include<stdio.h>`
- 2) `void strncat(char [],char [],int);`
- 3) `int main(void)`
- 4) {
  - i. `char a[50]="The adopted symbol is`  
`" ,b[27]="abcdefghijklmnopqrstuvwxyz";`
  - ii. `strncat(a, b, 4);`
- 5) `printf("%s\n",a);`
  - i. `return 0;`

---

```
6) }
7) void strncat(char s[],char t[], int n)
8) {
    i.    int i = 0, j;
9) while(s[i++] );
10) for(j = 0; j < n && t[j];)
11) s[i++] = t[j++];
    i.    s[i] = '\0';
12) }
```

**解答:**

- 1) i=23; s=0x00bcf940 ,t=0x00bcf91c
- 2) 第 10 行改为 for(j = 0,i--; j < n && t[j];), 此时 a[i]的值为'\0',之后拼接是将空字符覆盖。

#### 5.2.4 程序设计

编写并上机调试运行能实现以下功能的程序。

(1) 编写一个程序,从键盘读取数据,对一个 3×4 矩阵进行赋值,求其转置矩阵,然后输出原矩阵和转置矩阵。

(2) 编写一个程序,其功能要求是:输入一个整数,将它在内存中二进制表示的每一位转换成为对应的数字字符,存放到一个字符数组中,然后输出该整数的二进制表示。

(3) 编写一个程序,其功能要求是:输入 n 个学生的姓名和 C 语言课程的成绩,将成绩按从高到低的次序排序,姓名同时作相应调整,输出排序后学生的姓名和 C 语言课程的成绩。然后,输入一个 C 语言课程成绩值,用二分查找进行搜索。如果查找到有该成绩,输出该成绩同学的姓名和 C 语言课程的成绩;否则输出提示“not found!”。

**解答:**

(1).

- 1) 解体思路:利用数组存储矩阵, a[i][j],那么他的反置矩阵就是 a[j][i],利用嵌套 for 循环读取、交换和输出数组。
- 2) 程序清单:

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

---

```
2.  int main(void)
3.  {
4.  int matrix[3][4];
5.  int anti_matrix[4][3];
6.  int i, j, k;
7.  for (i = 0; i < 3; i++)
8.  for (j = 0; j < 4; j++)
9.  scanf(" %d", &matrix[i][j]);
10. for (i = 0; i < 3; i++)
11. for (j = 0; j < 4; j++)
12. anti_matrix[j][i] = matrix[i][j];
13. for (i = 0; i < 3; i++)
14. {
15. for (j = 0; j < 4; j++)
16. printf("%5d", matrix[i][j]);
17. putchar('\n');
18. }
19. putchar('\n');
20. for (i = 0; i < 4; i++)
21. {
22. for (j = 0; j < 3; j++)
23. printf("%5d", anti_matrix[i][j]);
24. putchar('\n');
25. }
26. system("pause");
27. }
```

```
1 2 3 4
5 6 7 8
9 10 11 12
    1    2    3    4
    5    6    7    8
    9   10   11   12

    1    5    9
    2    6   10
    3    7   11
    4    8   12
请按任意键继续. . .
```

3) 运行结果:

(2).

1)解题思路:使用字符数组存储字符, 利用 for 循环, 并且每移位一次读取并输出输入数的末位。

2)程序清单:

```
1. #include<stdio.h>
2. #include<string.h>
3. int main(void)
4. {
5.     int i,N,j,num,mask=1<<31;
6.     char bit[30][33];
7.     scanf("%d", &N);
8.     for (i = 0; i < N; i++)
9.     {
10.         bit[i][32] = '\0';
11.         scanf("%d", &num);
```



### 3)运行结果:

[illegible]

(3).

- 1) 解题思路: name 数组存储姓名, grade 数组存储成绩, 采用选择交换法进行排序, for 循环输出结果, 使用二分查找寻找成绩, 将查找结果存储在 search\_result 数组中, 找到 BinarySearch()函数为二分查找函数, 找到返回数组的下标, 未找到返回-1.使用 for 循环和 if 判断输出结果。

2) 程序清单: #include<stdio.h>

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

```
#define FD 100//第一维度
```

```
#define SD 20//第二维度
```

---

int Binary\_Search(int key, int a[], int n);//二分法查找，key 为查询目标，a[] 查找的范围，n 为 a[]的数组大小，找到返回 a[]的角标,找不到返回-1.

```
int main(void)
{
    char name[FD][SD];
    int grade[FD];
    int i, j, n; //i, j 控制循环

    scanf("%d", &n); //接下来有 n 组数据
    getchar();
    /*读取姓名和成绩*/
    for (i = 0; i < n; i++)
    {
        scanf(" %s", name[i]);
        scanf(" %d", &grade[i]);
        getchar();
    }
    /*选择交换法排序*/
    {
        int i, j, k;

        for (i = 0; i < n; i++)
        {
            k = i;
            for (j = i; j < n; j++)
            {
                if (grade[k] < grade[j])
                    k = j;
            }
        }
    }
}
```

---

```

        if (i != k)
        {
            int temp;
            temp = grade[i];
            grade[i] = grade[k];
            grade[k] = temp;//成绩排序
            char c[SD];
            strcpy(c, name[i]);
            strcpy(name[i], name[k]);
            strcpy(name[k], c);//姓名排序
        }
    }

    /*输出排序后的结果*/
    for (j = 0; j < n; j++)
        //printf("%d\n", grade[j]);
        printf("%-20s %d\n", name[j], grade[j]);
    putchar('\n');
    /*查询*/
    int N, search_result[FD];//进行 N 组查询,将查询的结果存在 search_result[]
    中。
    {
        scanf("%d", &N);
        getchar();
        for (j = 0; j < N; j++)
        {
            int key;//查询的目标数
            scanf("%d", &key);
            search_result[j]=Binary_Search(key, grade, n);//储存查询结果。
            //printf("*****%d\n",search_result[j]);

```

---

```

        }
    }
    /*输出查询结果*/
    for (j = 0; j < N; j++)
    {
        if (search_result[j] >= 0)
            printf("%-20s %d\n", name[search_result[j]],
grade[search_result[j]]);
        else printf("Not found!\n");
    }

    //system("pause");
}

int Binary_Search(int key, int a[], int n)
{
    int front = 0, back = n - 1, mid;
    while (back >= front)
    {
        mid = (front + back) / 2;
        if (a[mid] > key)
            front = mid + 1;
        else if (a[mid] < key)
            back = mid - 1;
        else return mid;
    }
    return -1;
}

```

### 3) 测试:

#### i. 测试数据:

5

ZhangChuanChao 88

XiaoHong 95

XiaoMing 90

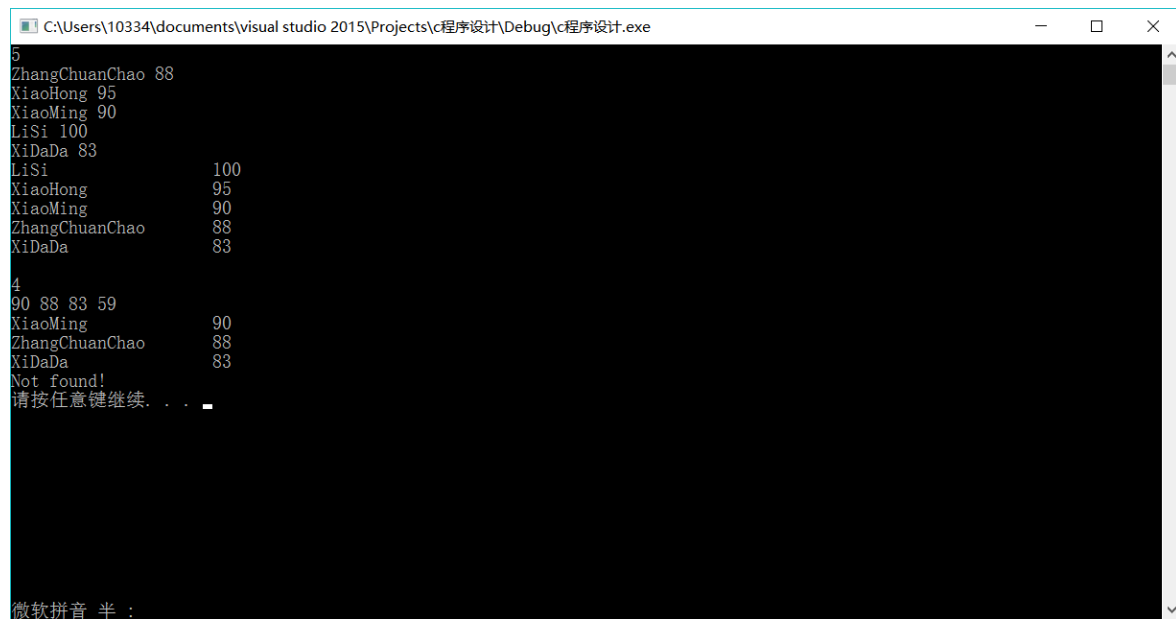
LiSi 100

XiDaDa 83

4

90 88 83 59

ii. 测试结果:



```
C:\Users\10334\documents\visual studio 2015\Projects\c程序设计\Debug\c程序设计.exe
5
ZhangChuanChao 88
XiaoHong 95
XiaoMing 90
LiSi 100
XiDaDa 83
LiSi          100
XiaoHong      95
XiaoMing      90
ZhangChuanChao 88
XiDaDa        83
4
90 88 83 59
XiaoMing      90
ZhangChuanChao 88
XiDaDa        83
Not found!
请按任意键继续. . .
```

### 5.3 实验总结:

本次实验使我初步掌握了掌握数组的说明、初始化和使用以及数组与指针的关系，可以通过一维数组作为函数参数进行函数之间的数据交换，掌握了字符串处理函数的设计，包括串操作函数及数字串与数之间转换函数实现算法，还学会了基于分治策略的二分查找算法和选择法排序算法的思想，以及相关算法的实现。

在实验过程中我也遇到了一些问题，对二维数组的理解不够透彻，比如在成绩处理的实验中，将二维数组作为参数传递给另一个函数的用法就没有很好的掌握，最后，通过百度查询和自行查阅资料（c 语言与程序设计，C primer plus）掌握了相关用法，在传递二维函数时，要将第二个维度明确下来，比如：

要处理 char 型数组 c[100][20]，对函数的声明可以是 L function(char a[][20],int n)或者 L function(char (\* a)[20],int n),引用时：function(c,100)，可见，数组与指针也有密切的联系。