**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)第七行 sort(a[0],10);——>sort(a,10);

2)第十八行 if(b[j] < b[j+1])——> if(b[j] >b[j+1])

（2）错误修改后运行结果：

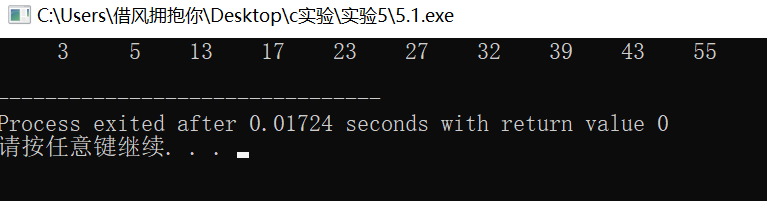


图5-2-1 改错题运行结果

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：略；

**解答：**

替换后的程序如下所示：

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. if(++j > i - 1) j = 0;
13. b[M-i] = j? a[j-1] :10 ;
14. if(j)
15. for(k = --j; k < i; k++)
16. a[k]=a[k+1];
17. }
18. for(i = 0;i<M-1; i++)
19. printf("%6d", b[i]);
20. printf("%6d\n", a[0]);
21. return 0;
22. }

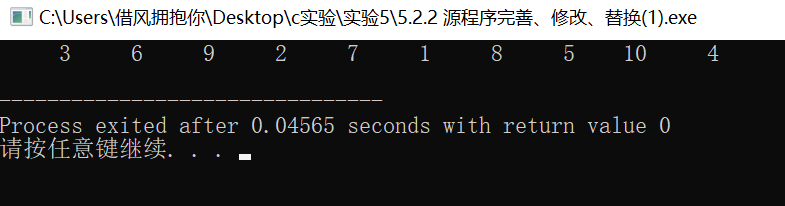
（2）错误修改后运行结果：

图5-2-2源程序完善、修改、替换运行结果

1. 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：**

替换后的程序如下所示：

1. #include<stdio.h>
2. #define M 10
3. #define N 3
4. int main(void)
5. {
6. int a[M]={0}, b[M];
7. int i, j, k;
8. for(i=M,j=0;i>0;i--)
9. {
10. for(k=1;k<=N;k++)
11. {
12. if(++j>M-1) j=0;
13. if(j==0&&a[9]==1||(j!=0&&a[j-1]==1))k--;
14. }
15. b[M-i]=j?j:M;
16. a[j-1]=1;
17. }
18. for(i=0;i<M;i++)
19. printf("%6d", b[i]);
20. return 0;
21. }

**主要思想：**做标记，遇到标记跳过；并且计数减少一；

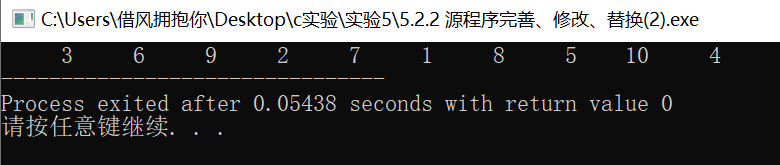
**需要注意的点：**当运行到边界时需注意在j=0时做判断。

图5-2-2-1源程序完善、修改、替换运行结果

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？

**解答：**如图所示

当光条落在for语句所在行时，i为23。

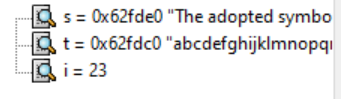


图5-2-3 (1) i的值

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

The adopted symbol is 和abcdefghijklmnopqrstuvwxyz

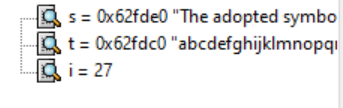


图5-2-3 (2) i的值

（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：略

**解答：**

分析：

while(s[i++]);这一条语句本来是统计s[i]的长度，但是如果是这样表示，当s[i]=‘\0’时，i的值仍然加了一，理论上i=26；但此时i=27;从而导致s[27]为‘\0’,printf()输出的时候，检测到‘\0’停止输出。故只需改为while(s[i])i++ ;即可。

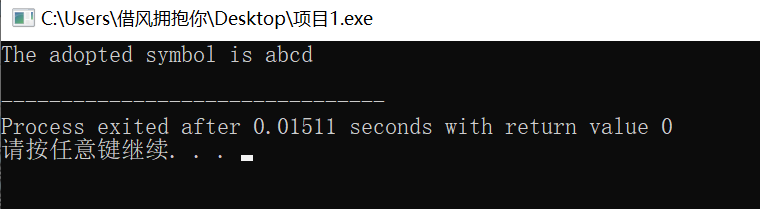
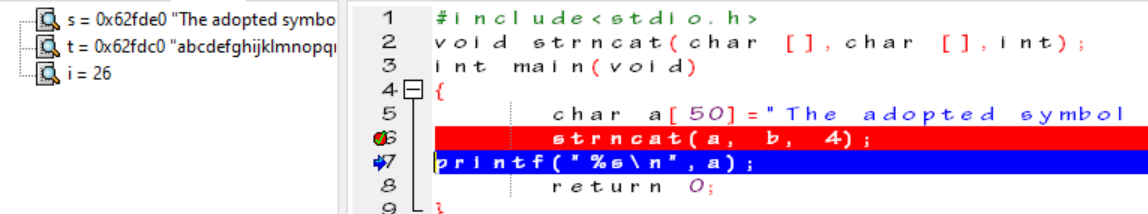
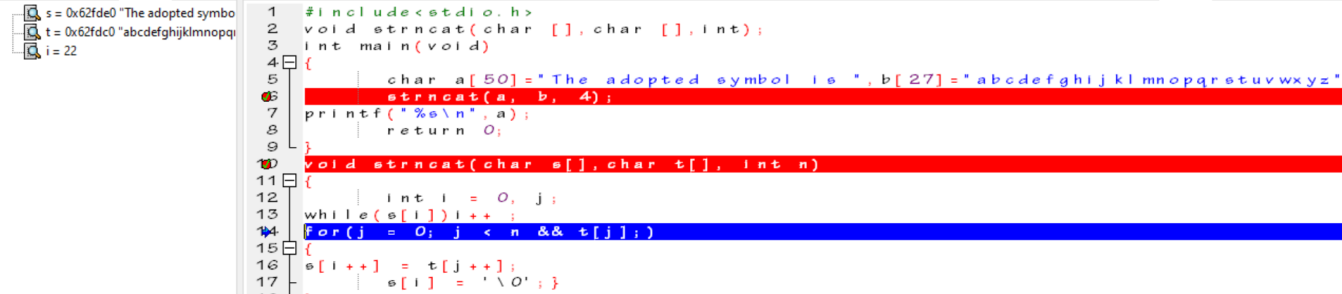


图5-2-3 (3) 跟踪调试源程序程序输出

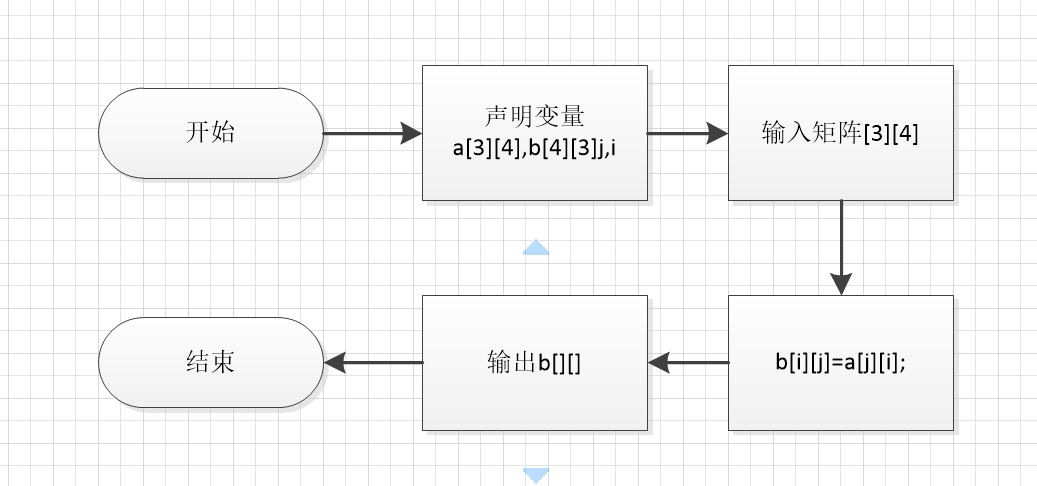


5.2.4 程序设计

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。

1) 解题思路：

图5-2-4-1 编程题1的程序流程图

1. 程序清单
2. #include<stdio.h>
3. int main(void)
4. {int a[3][4],b[4][3],j,i;//------------------------声明
5. for(i=0;i<3;i++)
6. for(j=0;j<4;j++)
7. scanf("%d",&a[i][j]); //------------------------输入
8. for(i=0;i<3;i++)
9. {for(j=0;j<4;j++)
10. printf("%6d",a[i][j]); //------------------------输出原矩阵
11. printf("\n");}
12. for(i=0;i<4;i++) //------------------------行
13. for(j=0;j<3;j++)
14. b[i][j]=a[j][i]; //------------------------转换
15. for(i=0;i<4;i++){
16. for(j=0;j<3;j++)
17. printf("%6d",b[i][j]); //------------------------输出转置矩阵
18. printf("\n");
19. }
20. }

3）测试

（a） 测试数据：

表5-2-4-1 编程题1的测试数据

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例 | 程 序 输 入 | 理论结果 | 运行结果 |
| 用例1 | a[3][4] | 略 | 如图 |
| 1 2 3 4 5 6 7 8 9 10 11 12 |

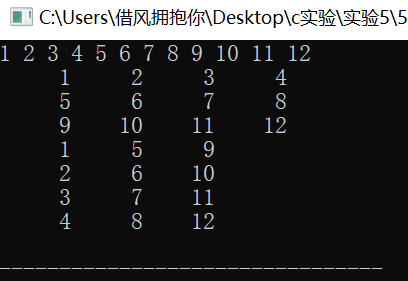


图5-2-4-1 编程题1的测试用例一的运行结果

1. 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。

1) 解题思路：

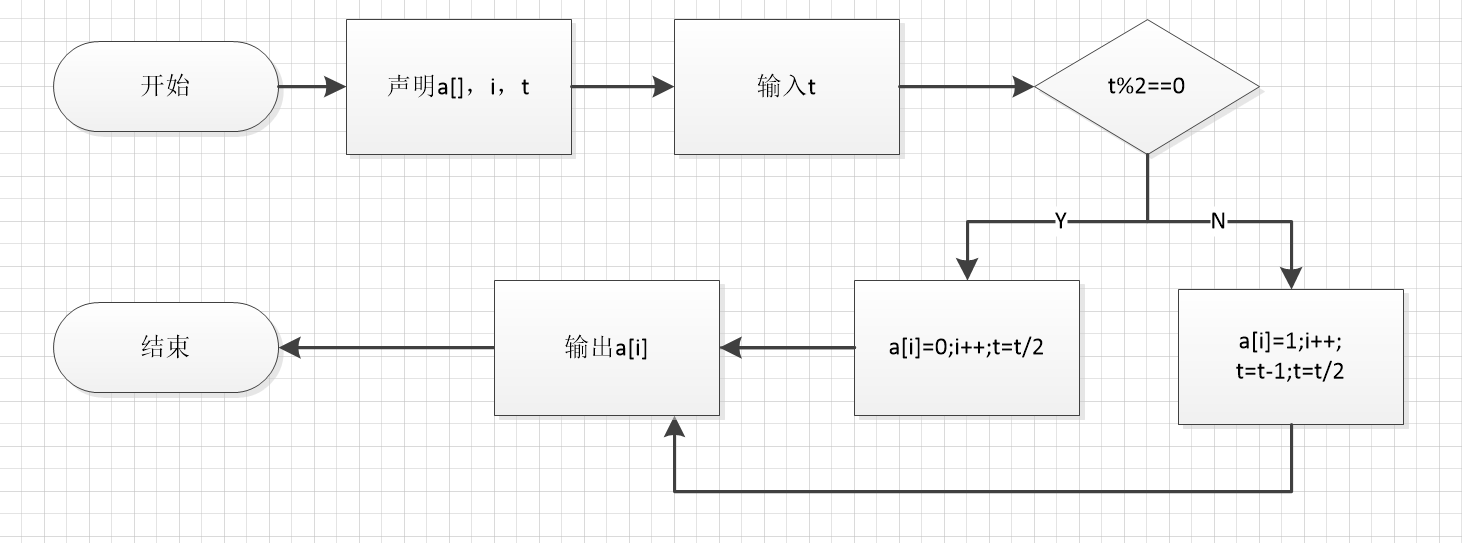


图5-2-4-2 编程题2的程序流程图

2）程序清单

1. #include<stdio.h>
2. int main(void)
3. {int a[32]={0},t,i=0;
4. scanf("%d",&t);
5. while(t!=0){
6. if(t%2==0)
7. {a[i]=0;i++;t=t/2; }
8. else{a[i]=1;i++;t=t-1;t=t/2;}}
9. while(i<=32) {printf("%d",a[32-i]);i++;}
10. }

3）测试

（a） 测试数据：

表5-2-4-2 编程题2的测试数据

|  |  |  |  |
| --- | --- | --- | --- |
| 测试用例 | 程 序 输 入 | 理论结果 | 运行结果 |
| 用例1 | t | 000000000100100101001 | 如图 |
| 255 |
| 用例2 | 2345 | 0000000000000000011111111 | 如图 |

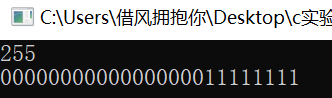


图5-2-4-2（1）编程题2的测试用例一的运行结果

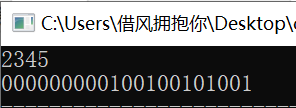


图5-2-4-2（1）编程题2的测试用例一的运行结果

（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!”。

1) 解题思路：

第一部分：

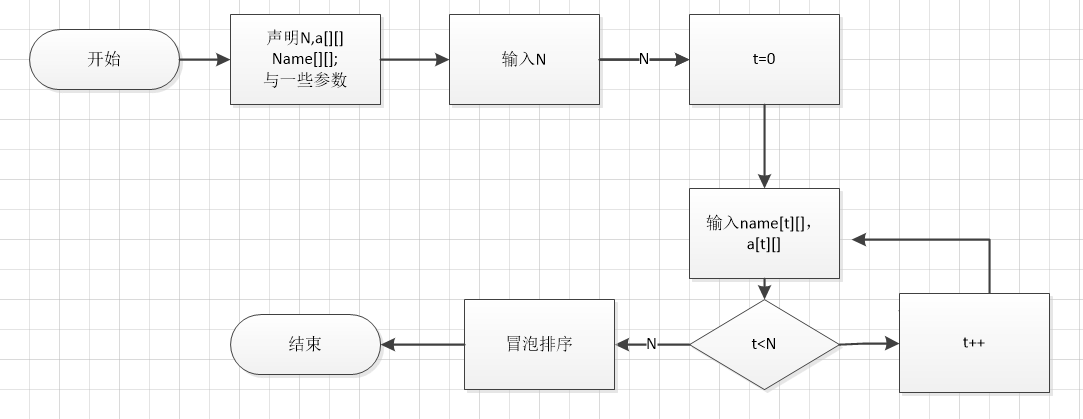


图5-2-4-3（1） 编程题1的程序流程图

第二部分：

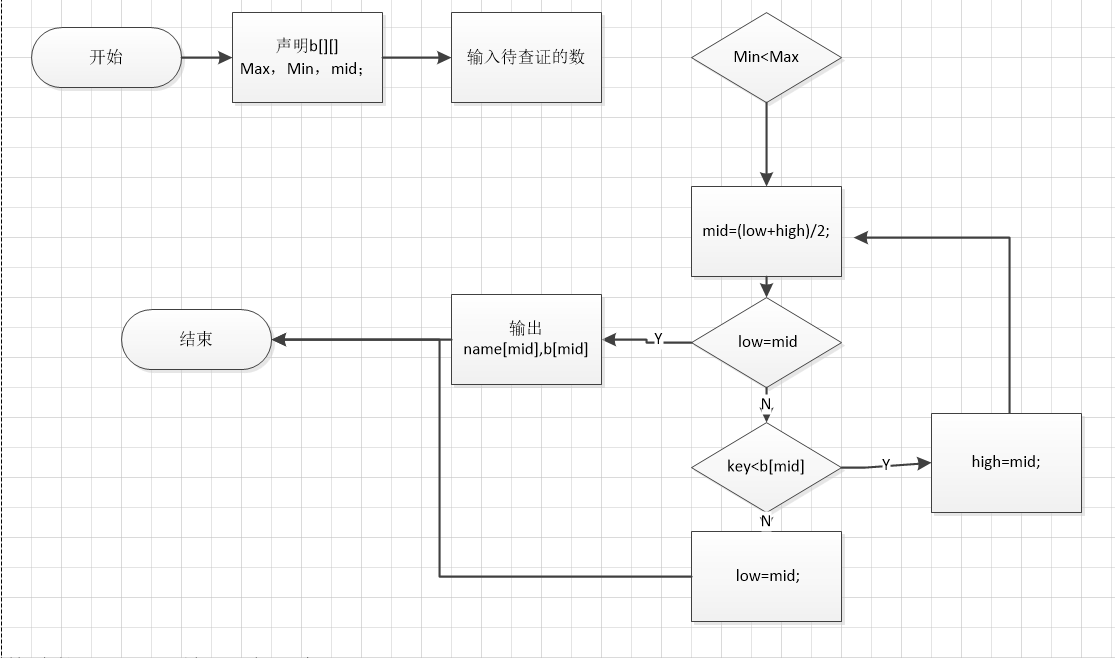


图5-2-4-3（2） 编程题1的程序流程图

2）程序清单

1. #include<stdio.h>
2. void swap(char \*a,char \*b) ;
3. int main(void)
4. {
5. char name[20][20],temp;
6. int b[20],N;
7. int i,a,t;
8. int y,key;
9. int g=0;
10. scanf("%d",&N);
11. /\*数据输入\*/
12. for(t=0;t<N;t++)
13. scanf("%s %d",&name[t],&b[t]);
14. /\* 冒泡排序\*/
15. for(i=0;i<N-1;i++)
16. for(a=0;a<N-1;a++){
17. if(b[a]>b[a+1])
18. {
19. y=b[a];
20. b[a] = b[a+1];
21. b[a+1] =y;
22. swap(name[a],name[a+1]);
23. }}
24. /\*输出\*/
25. for(g=0;g<N;g++) printf("\n%s:%d",name[g],b[g]) ;
26. /\*二分法\*/
27. scanf("%d",&key) ;
28. int low,high,mid;
29. low=0;
30. high=N-1;
31. while(low<=high)
32. {
33. mid=(low+high)/2;
34. if(key<b[mid])
35. high=mid;
36. if(key>b[mid])
37. low=mid;
38. else key=b[mid];
39. }
40. printf("%s: %4d\n",name[mid],b[mid]);
41. return 0;
42. }

3）测试

（a） 测试数据：

表5-2-4-3 编程题3的测试数据

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 测试用例 | 程 序 输 入 | | | | | 理论结果 | 运行结果 |
| 用例1 | a[20][20] | Name[20][20] | | | N |  |  |
|  | 234 | | Q | 3 | | A:23  Z:145  Q:234 | 如图 |
| 23 | | A |
| 145 | | Z |

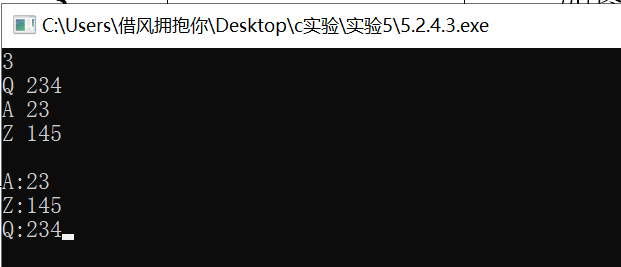


图5-2-4-3 编程题3的测试用例一的运行结果

## 5.3 实验小结

1.程序优化的时候发现一直无法输出，后来发现是while和if的区别，在判断循环条件时，若括号中时等式，if只运行一次而while是一个循环，导致一开始无法输出，陷入死循环。

2.字符串的交换：一开始在编程题第三题的时候数字交换了但是名字无法输出，后来通过搜索知道可以用指针来进行交换。

3.八皇后的那一题，一开始只能输出一种方法，而且还输出错误。后来加入flag作为标识，才可以进行返回赋值。