# 5数组实验

## 5.1实验目的

本次实验需要掌握数组的说明、初始化和使用；一维数组作为函数参数时实参和形参的用法。同时学会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。熟练使用基于分治策略的二分查找算法和选择法排序算法，同时能够实现相关算法。

## 5.2 必做题

### 5.2.1 源程序改错

**【题目】**

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。

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 }

**【错误原因分析及改错方案】**

源程序存在两处错误，分别是函数声明错误和逻辑错误，具体情况如下。

第5行错误，调用函数sort时第一个参数与函数声明不相符；正确形式应该为void sort(a,10)；

第18行错误，题目要求将数组中的元素按升序排列，而源程序为降序排序，故正确形式应该为if（b[j]>b[j+1]）。

**【修改后代码】**

#include<stdio.h>

void sort(int b[], int n);

int main(void)

{

int a[10] = { 27, 13, 5, 32, 23, 3, 17, 43, 55, 39 };

void sort(int[], int);

int i;

sort(a, 10);

for (i = 0; i < 10; i++)

printf("%6d", a[i]);

printf("\n");

return 0;

}

void sort(int b[], int n)

{

int i, j, t;

for (i = 0; i < n - 1; i++)

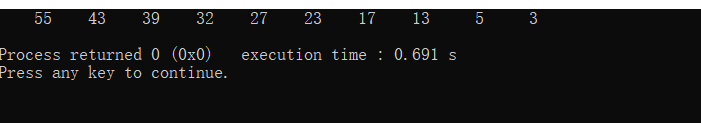
for (j = 0; j < n - i - 1; j++)

if (b[j] < b[j + 1])

t = b[j], b[j] = b[j + 1], b[j + 1] = t;

}

**【测试数据、程序测试理论结果及测试结果】**



### 5.2.2 源程序完善、修改、替换

【题目】

(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。

源程序如下，

#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? : ; /\* 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 \*/

;

}

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。

【答案】

（1） b[M-i] = j? a[j-1] : a[i-1] ;

for(k = --j; k < i; k++) a[k]=a[k+1] ;

（2）

【程序运行结果】

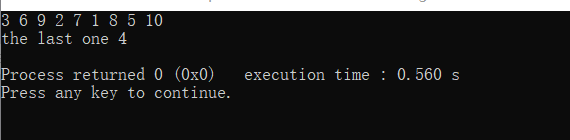


图5-3 实验五必做题5.2.2程序运行结果

### 5.2.3 跟踪调试

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。

#include<stdio.h>

void strncat(char[], char[], int);

int main(void)

{

char a[50] = "The adopted symbol is ", b[27] = "abcdefghijklmnopqrstuvwxyz";

strncat(a, b, 4);

printf("%s\n", a);

return 0;

}

void strncat(char s[], char t[], int n)

{

int i = 0, j;

while (s[i++]);

for (j = 0; j < n && t[j];)

s[i++] = t[j++];

s[i] = '\0';

}

**【调试过程】**

分别在进入函数strncat、for语句所在行及strncat函数块结束处设置断点。

当进入函数strncat中后，落到for语句所在行时，根据局部变量窗口观察i的值为23。如图5-4；当命中结束标志处的断点时，根据局部变量窗口，s为0x00aff944，指向字符串“The adopted symbol is ”，t为0x00aff920，指向字符串abcdefghijklmnopqrstuvwxyz，如图5-5

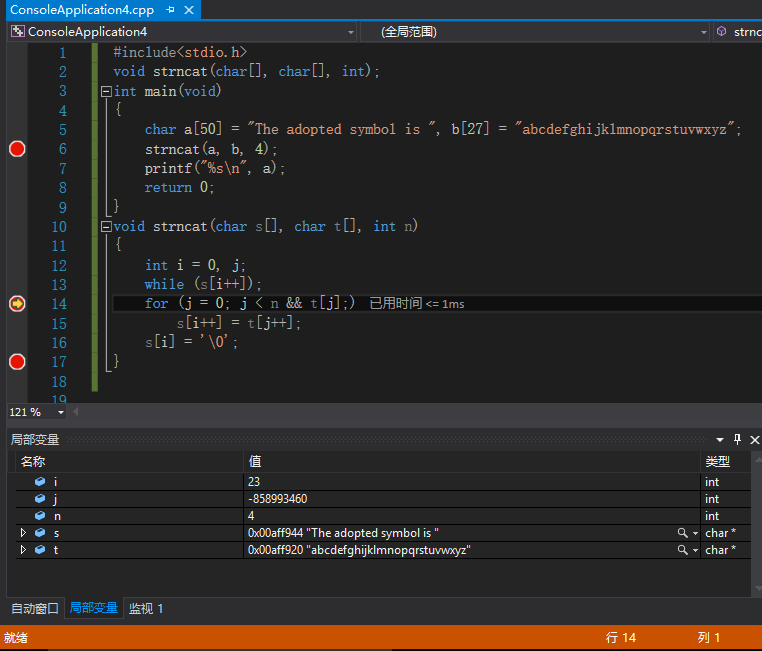


图5-4 实验五必做题5.2.3调试（1）

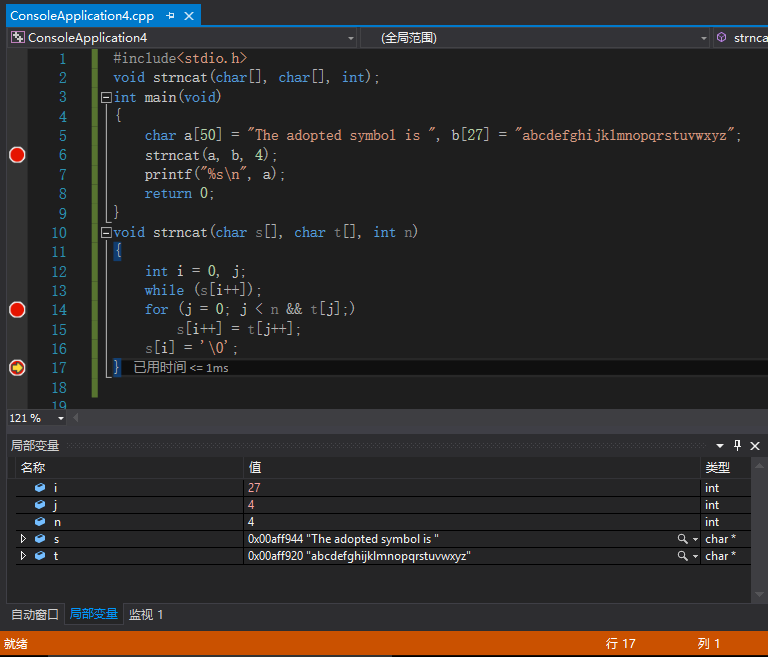


图5-5 实验五必做题5.2.3 调试（2）

**【错误分析及修改】**

根据调试过程可知，由于在while循环结束后i值为23，使得第一个字符串的a[22]即‘\0’没有被覆盖，所以输出时只有第一个字符串。

只需要将while的循环条件改为while（s[++i]!=’\0’）即可，修改后的输出结果如图5-6所示。

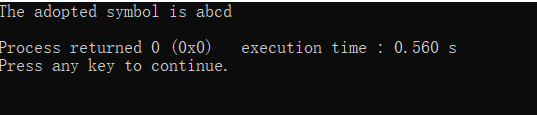


图5-6 实验五必做题5.2.3修改后运行结果

### 5.2.4 矩阵的运算

**【题目】**

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。

**【程序流程图】**

**如下所示**

**（在下一页，太长了，这页装不下）**

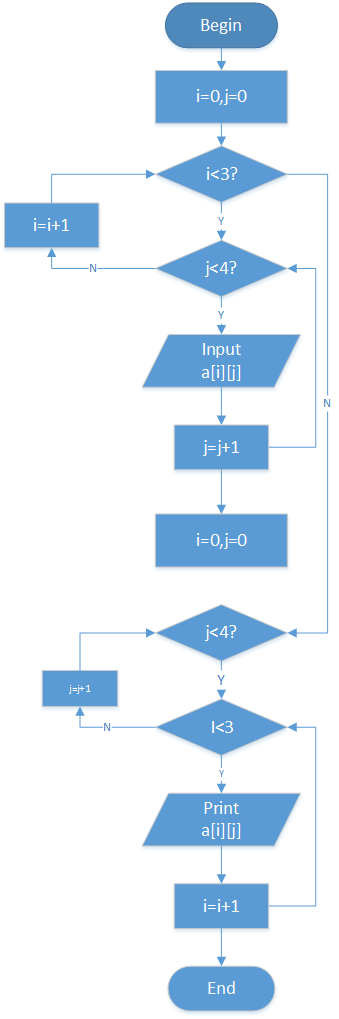


图5-7 实验五必做题5.2.4 算法流程图

**【程序清单】**

#include<stdio.h>

int main()

{

int a[3][4];

int i,j;

for(i=0;i<3;i++)

{

for(j=0;j<4;j++)

scanf("%d",&a[i][j]);

}

printf("原矩阵为：\n");

for(i=0;i<3;i++)

{

for(j=0;j<4;j++)

printf("%d ",a[i][j]);

printf("\n");

}

printf("\n转置矩阵为：\n");

for(j=0;j<4;j++)

{

for(i=0;i<3;i++)

printf("%d ",a[i][j]);

printf("\n");

}

return 0;

}

**【测试数据、测试理论结果及实际结果】**

由于该程序不需要进行数据计算，而只是交换输出的顺序，故情况较为简单，仅设计了一组测试数据。

输入为

1 2 5 4

2 8 4 6

9 7 4 1

预测输出为

原矩阵为：

1 2 5 4

2 8 4 6

9 7 4 1

转置矩阵为：

1 2 9

2 8 7

5 4 4

4 6 1

实际输出结果如下图5-8，发现与预测输出一致，故程序正确。

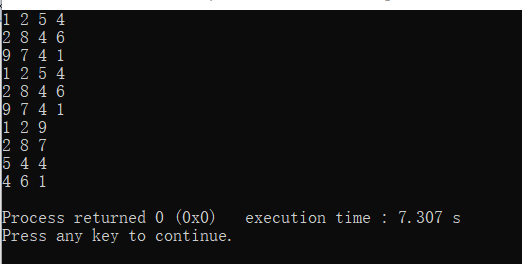


图5-8 实验五必做题5.2.4程序测试结果

### 5.2.5 整数二进制输出

**【题目】**

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。

**【解题思路流程图】**

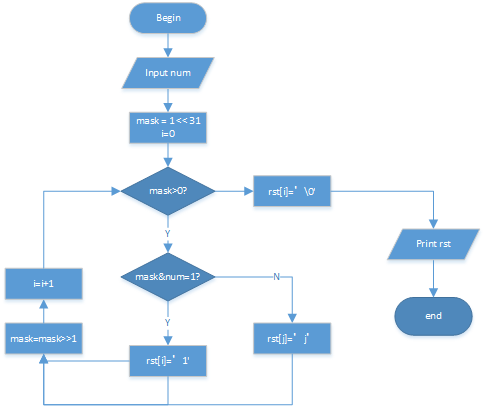


图5-9 实验五必做题5.2.5算法流程图

**【程序清单】**

#include<stdio.h>

int main()

{

unsigned int mask = 1 << 31;

int num;

char rst[40] = { 0 };

while (scanf("%d", &num) != EOF)

{

int i = 0;

for (mask = (1 << 31), i = 0; mask > 0; mask >>= 1, i++)

{

if (mask&num) //若此位置为1，则将rst中赋值为字符1

rst[i] = '1';

else rst[i] = '0';

}

rst[i] = '\0';//最后赋值结束符\0

printf("%s\n", rst);

}

return 0;

}

**【测试数据、理论结果及测试结果】**

表5-2 实验五必做题5.2.5测试数据、理论结果及测试结果

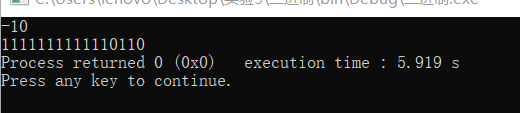


图5-10 实验五必做题5.2.5程序测试结果

### 5.2.6 C语言成绩排序查询

**【题目】**

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!”。

【程序清单】

#include<stdio.h>

#include<string.h>

#define n 3

#define max 20

int bin(float a[],int x,int t);

void sort(float score[],int index[],int m);

int main()

{

float score[n];

char name[n][max];

int i,index[n],x,s;

for(i=0;i<n;i++)

{

scanf("%s %f",&name[i][0],&score[i]);

}

for(i=0;i<n;i++)

{

index[i]=i;

}

sort(score,index,n);

for(i=0;i<n;i++)

{

printf("%s:%f\n",name[index[i]],score[index[i]]);

}

scanf("%d",&x);

s=bin(score,x,n);

if(s!=1)

printf("%s:%f\n",name[s],score[s]);

else printf("not found");

return 0;

}

void sort(float score[],int index[],int m)

{

int i,j,t;

for(i=0;i<m-1;i++)

for(j=0;j<m-1-i;j++)

{

if(score[index[j]]>score[index[j+1]])

{

t=score[index[j]];

score[index[j]]=score[index[j+1]];

score[index[j+1]]=t;

}

}

}

int bin(float a[],int x,int p)

{

int begin=0,end=p-1,mid;

while(begin<end)

{

mid=(begin+end)/2;

if(x<a[mid])

end=mid-1;

else if(x>a[mid])

begin=mid+1;

else return mid;

}

return 0;

}

**测试数据及测试预测结果】**

此程序再第一次输入阶段尽量保证分数的区分度，而在第二次输入分数查找时，分为搜索中间、最大、最小进行检验，故设计以下6个学生样例：

A 79

B 88

C 72.5

D 90

E 84

F 81

在搜索阶段，输入72.5,90,84,81,79，88六个数进行测试，应分别输出Not Found！，C，D，E，F，A，B四个结果，实际输出如图5-12，完全符合预测结果。

**【测试结果】**



图5-12 实验五必做题5.2.6测试结果

## 5.3 选做题

**【题目】**

编写函数strnins(s,t,n),其功能是：可将字符数组 t中的字符串插入到字符数组 s中字符串的第n个字符的后面。

**【算法流程图】**

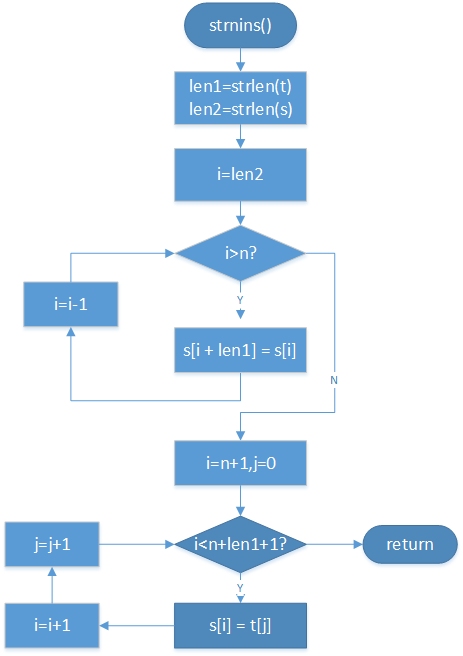


图5-13 实验五选做题算法流程图

**【程序清单】**

#include<stdio.h>

#include<string.h>

void strnins(char s[], char t[], int n);

int main()

{

char s[100], t[100];

int n;

printf("Input s:\n");

fgets(s, 100, stdin);

s[strlen(s) - 1] = s[strlen(s)];

printf("Input t:\n");

fgets(t, 100, stdin);

s[strlen(t) - 1] = s[strlen(t)];

printf("Input n:\n");

scanf("%d", &n);

strnins(s, t, n);

printf("%s", s);

return 0;

}

void strnins(char s[], char t[], int n)

{

int len1 = strlen(t);

int len2 = strlen(s);

int i = 0, j = 0;

//移位空出位置

for (i = len2; i > n; i--)

s[i + len1] = s[i];

//插入字符串

for (i = n + 1, j = 0; i < n + len1 + 1; i++, j++)

s[i] = t[j];

}

## 5.4 自设题

**【题目】**

从键盘输入一个年月日，计算该日期在一年中是第几天。（用数组）

**【程序清单】**

#include<stdio.h>

int main()

{

int year,month,day,m,i;

m=0;

scanf("%d %d %d",&year,&month,&day);

int a[13]={0,31,28,31,30,31,30,31,31,30,31,30,31};

int b[13]={0,31,29,31,30,31,30,31,31,30,31,30,31};

if((year%4==0&&year%100!=0||year%400==0))

{

for(i=1;i<month;i++)

m+=b[i];

m+=day;

}

else

{

for(i=1;i<year;i++)

m+=a[i];

m+=day;

}

printf("the day is the year of %d day",m);

return 0;

}

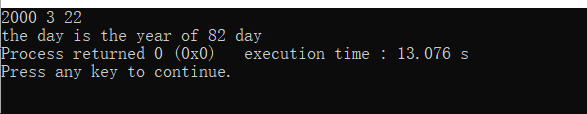


图5-14 实验五 自设题样例测试结果

## 5.5小结

本次实验中收获最大的两个实验题就是N人报错和C语言成绩排序两个题目，前者考验了我们的思维，让我对三目判断语句有了深刻的理解，还学会了用标记法来代替一次次压缩数组的方法，提高了算法的效率。后者是一个综合性很强的程序，运用了二分法查找和递归算法这两个重要的算法，同时对数组的运用熟练程度有一定的要求。