# 实验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

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5行函数声明应在main函数外。

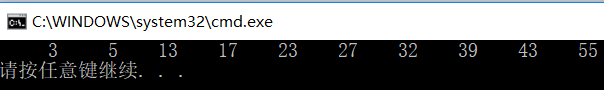
2) 第7行函数参数应为数组名，正确形式为：

sort(a,10);

3) 第18行<应为>号，正确形式为：

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

（2）错误修改后运行结果：



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：

#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? a[j - 1] : 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]; ;

}

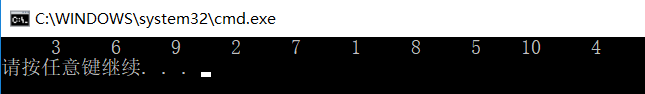
for(i = 0;i < M – 1; i++) /\* 按次序输出出圈人的编号 \*/

printf(“%6d”, b[i]);

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

return 0;

}



(2)使用标记法

#include<stdio.h>

#include<stdbool.h>

#define M 10

#define N 3

bool visited[M] = {0};

int main(void)

{

int i, j, k;

k = 0;

j = 0;

i = M;

do

{

++j;//圈中的所有位置

if (j > M)

j = 1;

if (!visited[j])

k++;

if (k == N)//当前报的数是M 重头开始

{

k = 0;

printf("% 6d", j);//输出被访问编号

visited[j] = true;//此处人已访问，设置为空

i--;

}

} while (i != 0);//直到所有人都访问过

}

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：

#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];)

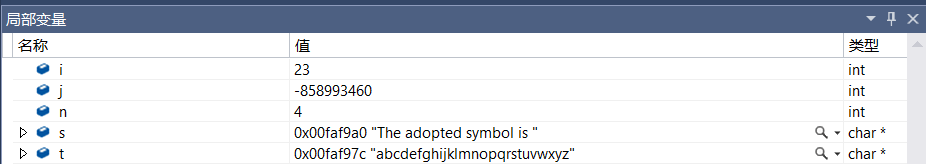
s[i++] = t[j++];

s[i] = '\0';

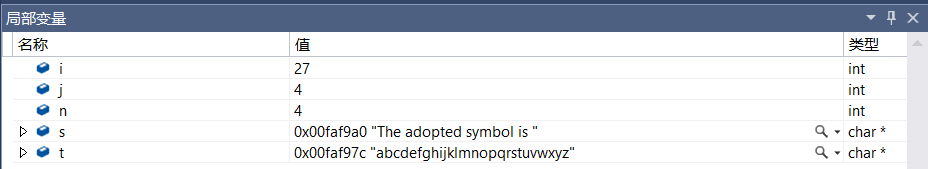
}

(1)单步调试

进入for循环：



**结束：**



(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++]);

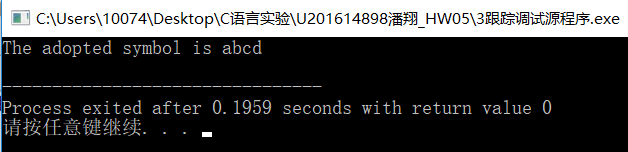
i--;

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

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

s[i] = '\0';

}



5.2.4 程序设计

（1）编写一个程序，其功能要求是：输入n个学生的姓名（不超过100个学生，姓名字符数不大于20）和C语言课程的成绩，将成绩按照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语言课程成绩；否则输出提示“Not found!”。

输入：第一行为正整数n，接下来的n行是n个学生和对应的成绩信息；然后是一个正整数N，表示需要查询N次，接下来的一行是N个需要查询的成绩。

输出：首先输出排序后的姓名和成绩信息，姓名和成绩用空格隔开，且姓名的宽度为20（左对齐，空格补齐）；输出一个空行；然后是N行查询结果，若查询到则输出其姓名和成绩信息，格式与前者一致，否则输出“Not found!”。

注意：为了方便查找，成绩为正整数，且无重复成绩。

**解答：**

1）算法流程如图5.1所示。



图5-1 编程题1的程序流程图

2）源程序清单

#include<stdio.h>

int Resort(int begin, int end);

int Inquire(int score, int begin, int end);

struct Student

{

char name[15];

int score;//成绩

};

struct Student stu[20];

int main()

{

int n;

scanf("%d", &n);

getchar();

int result;

for (int i = 1; i <= n; i++)

{

scanf("%s%d",stu[i].name,&stu[i].score);

getchar();

}

Resort(1, n);

for (int i = 1; i <= n; i++)

{

printf("%-20s %d\n", stu[i].name, stu[i].score);

}

printf("\n");

int N;

int temp\_score;

scanf("%d", &N);

for (int i = 1; i <= N; i++)

{

scanf("%d", &temp\_score);

Inquire(temp\_score,1,n);

}

return 0;

}

int Resort(int begin, int end)

{

if (begin == end)

{

return 0;

}

int maxi=begin;

struct Student temp;

for (int i = begin; i <= end; i++)

{

if (stu[i].score > stu[maxi].score)

{

maxi = i;

}

}

temp = stu[begin];

stu[begin] = stu[maxi];

stu[maxi] = temp;

return Resort(begin+1, end);

}

int Inquire(int score,int begin,int end)

{

int flag=0;

int half = (begin + end) / 2;

if (begin >= end || stu[half].score == score)

{

if (score == stu[half].score)

{

printf("%-20s %d\n", stu[half].name, stu[half].score);

return half;

}

else

{

printf("Not found!\n");

return -1;

}

}

if (begin < end)

{

if (stu[half].score>score)

{

return Inquire(score, half + 1, end);

}

else

{

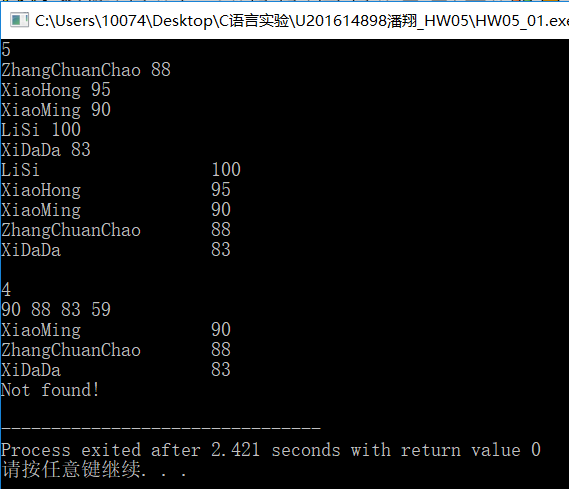
return Inquire(score, begin, half);

}

}

}

3）测试



（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。  输入：第一行为正整数N，表示有N组输入；第二行为N个int型整数，即N组输入。  输出：对每组输入的整数，输出其二进制表示，每组一行 |

**解答：**

1） 算法流程如图5.2所示。



图5-2 编程题2的程序流程图

2）源程序清单

#include<stdio.h>

void ToBin(int n);

int N;

int i;

int BIN[100][33];

int main()

{

int n = 0;

scanf("%d", &n);

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

{

scanf("%d", &N);

ToBin(N);

}

return 0;

}

void ToBin(int N)

{

int t;

t = 1;

t = t << 31;

for (int j = 1; j <= 32; j++)

{

BIN[i][j]=( N&t ? '1' : '0');

printf("%c",BIN[i][j]);

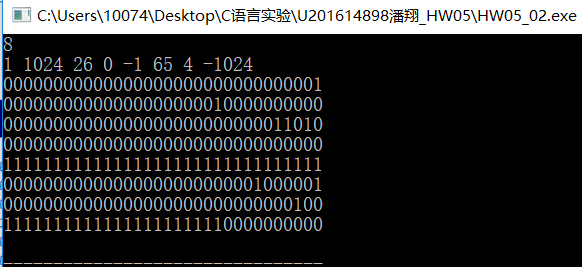
N = N<< 1;

}

printf("\n");

}

3）测试



（3）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。

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，然后分别输出原矩阵和转置矩阵。

输入：一个3\*4的整数矩阵。

输出：输出原矩阵和转置矩阵，每个数的宽度为5（右对齐，空格补齐），原矩阵和转置矩阵用空行隔开

1. 算法流程如图下所示。
   1. 取数组a[4][5]从a[1][1]开始存取先行后列
   2. 输出数组先列后行
2. 源程序清单

#include<stdio.h>

void transpose();

int a[4][5];

int main()

{

int i, j;

for (i = 1; i <=3; i++)

{

for (j = 1; j <=4; j++)

{

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

}

}

for (i = 1; i <=3; i++)

{

for (j = 1; j<=4; j++)

{

printf("%5d",a[i][j]);

}

printf("\n");

}

printf("\n");

transpose();

return 0;

}

void transpose()

{

int i, j;

for (j = 1; j <= 4; j++)

{

for (i = 1; i <= 3; i++)

{

printf("%5d", a[i][j]);

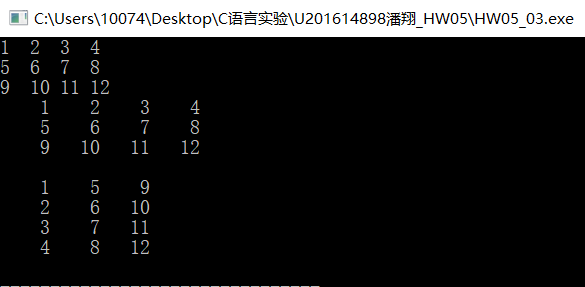
}

printf("\n");

}

}

1. 测试



## 5.3 选做题

（1）字符串中间插入函数

1）源程序清单

#include<stdio.h>

#include<string.h>

#include<stdbool.h>

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

int main()

{

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

strnins(a, b, 5);

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

return 0;

}

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

{

char temp[100];

if (n < 0 || n>strlen(s))

{

printf("Error!\n");

return false;

}

strcpy(temp, s + n);

strcpy(s+n, t);//t+n之后添加到s结尾处

strcat(s, temp);

return true;

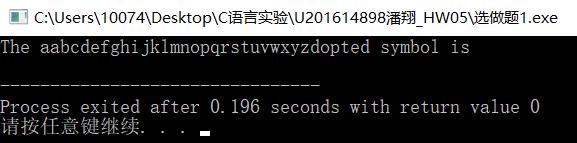
}

2）测试用例：

The adopted symbol is

abcdefghijklmnopqrstuvwxyz

3）测试结果:

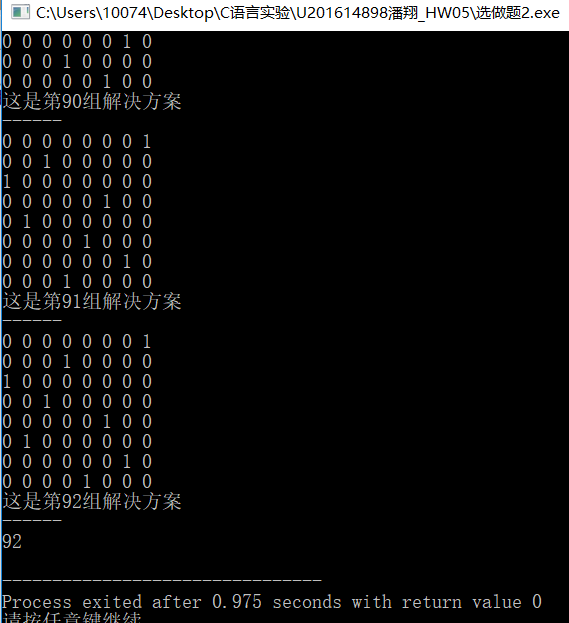


(2)八皇后

1）源程序清单：

2）测试用例：无

3）测试结果：



## 5.4 自设题

1. 自设实验题目

描述：

输入一个n维矩阵 求它的行列式

输入：

维数为n的数组

输出：

它的行列式

1. 实验目的：熟悉数组的使用方法，考虑矩阵运算库的编写
2. 实验程序：

#include<stdio.h>

#include<math.h>

int countA(int a[100][100], int n);

int main()

{

int a[100][100];

int i, j;

int n;

scanf("%d", &n);

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

{

for (j = 1; j <= n; j++)

{

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

}

}

printf("%d\n", countA(a,n));

return 0;

}

int countA(int a[100][100], int n)

{

int b[100][100] = { { 0 } };

int i = 0, j = 0, sum = 0; //i,j为行与列,sum为行列式的值

int x = 0, c = 0, p = 0; //用x判断加与减符号

if (n == 1)

{

return a[0][0];

}

for (i = 0; i < n; i++) //余子式存储

{

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

{

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

{

if (c < i) //判断行列式的移动方向

{

p = 0; //当p=0时,行列式只向左移,即消去对应的第一列的数

}

else

{ //否则行列式左移后再上移

p = 1;

}

b[c][j] = a[c + p][j + 1];

}

}

if (i % 2 == 0) { //i+j（此时j=0）为偶数,加法

x = 1;

}

else //i+j为奇数,减法

{

x = (-1);

}

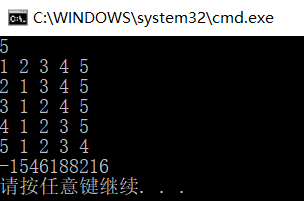
sum += a[i][0] \* countA(b,n-1) \* x; //计算行列式的值

}

return sum;

}

1. 实验用例：



1. 实验结论：

编写考虑输入判错，并使接口清晰

## 5.5 实验小结

**（1）**注意数组下标的起始位置，考虑从0开始符合人类思考习惯

**（2）**对多层for语句的输出逻辑清楚

**（3）**数组指针间接访问的掌握