

**课 程 实 验 报 告**

**课程名称： C语言程序设计实验**

**专业班级：**

**学 号：**

**姓 名：**

**指导教师：**

**报告日期： 2019.10.19**

**网络空间安全学院**

**目 录**

**[5. 数组实验 1](#_Toc404837920)**

[5.1 实验目的 1](#_Toc404837921)

[5.2 实验内容 1](#_Toc404837922)

[5.3 实验小结 1](#_Toc404837923)

# 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1行与第2行之间应加上sort的函数原型，正确形式为：

void sort(int [],int);

2) 第5行的函数声明应删去。

3）第7行调用该函数时，输入的应该是数组名（一个指针），正确形式为：

sort(a,10);

4）第18行的小于号应改为大于号，正确形式为：

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

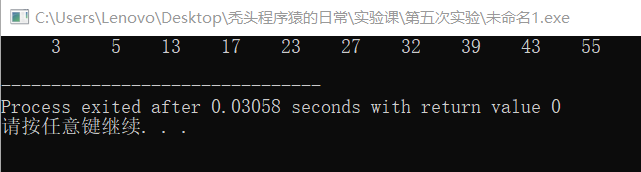
5）第18行与第19行之间应该加上花括号，否则无法进行循环，正确形式为：

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

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

}

（2）错误修改后运行结果：

****

**图5-1 改错题的运行结果**

**5.2.2 源程序修改替换**

(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? : ; /\* 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。

**解答：**

1. 填入的语句应为a[j-1] : a[i-1]
2. a[k]=k
3. 用标记的方法修改如下：

#include<stdio.h>

#define M 10

#define N 3

int main()

{

int a[M],b[M]={0};

int i,j=0,k=0;

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

a[i]=1;

i=0;

while(b[M-1]==0)

{

if(i==M) i=0;

if(a[i]==1) j++;

if(j==N){

b[k]=i+1,a[i]=0,j=0,k++;

}

i++;

}

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

{

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

}

printf("\n");

}

将所有被剔除的元素对应的数组下标所对应的内容变为0，并让循环自动忽略0.

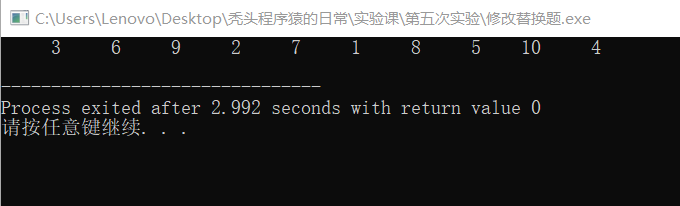


图5-2 程序修改替换题运行结果

**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：

#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';

}

**解答：**

1. 落在for语句上时，i的值如图所示

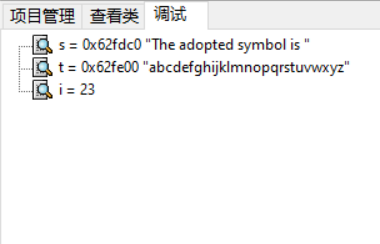
****

图5-3 调试题的观察结果1

1. 落在结束标记时，s、t的值如图所示：

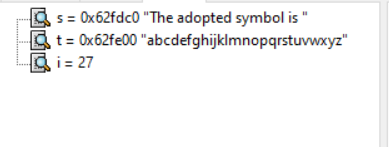


图5-4 调试题的观察结果2

1. 程序出错的原因是while后少了i--，会导致无法覆盖前一字符串的结束标志‘\0’，而导致其单独成串。

改正后程序的运行结果如图

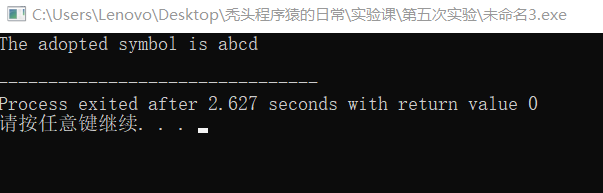


图5-5 调试题的观察结果

**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。

**解答：**1） 算法流程如图1.1所示。

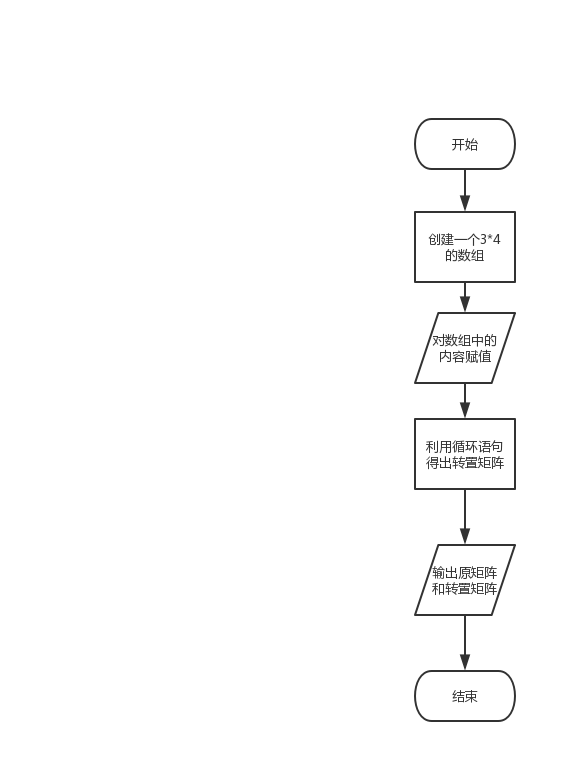


图5-6 编程题1的程序流程图

2）源程序清单

#include<stdio.h>

int main(void)

{

int a,b,i;

int c[3][4];

int d[4][3];

printf("请输入3组整数，每组包括4个整数");

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

{for(a=0;a<4;a++)

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

}

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

{for(a=0;a<3;a++)

d[i][a]=c[a][i];

}

printf("原矩阵为：\n");

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

{for(a=0;a<4;a++)

printf("%3d",c[i][a]);

printf("\n");

}

printf("转置矩阵为：\n");

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

{for(a=0;a<3;a++)

printf("%4d",d[i][a]);

printf("\n");

}

}

3）测试

（a） 测试数据：

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

（b） 对应测试数据的运行结果截图

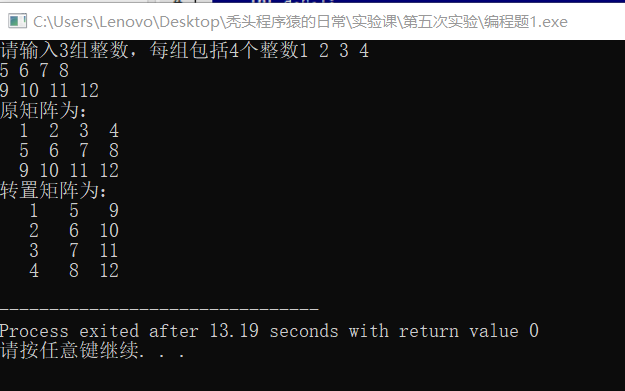


图5-7 编程题1的运行结果实例1

（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。

**解答：**

1. 解题思路：

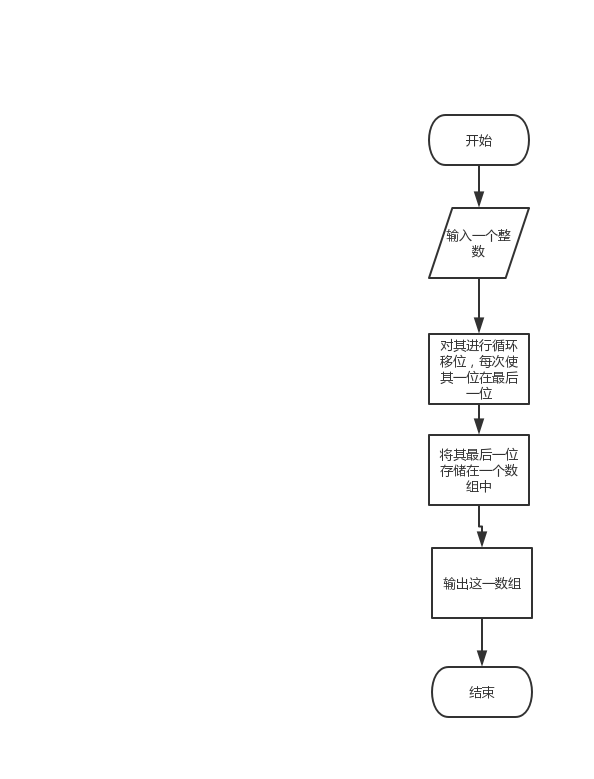


图5-8 编程题1的程序流程图

2）程序清单：

#include<stdio.h>

int main(void)

{

int b,i;

unsigned int a;

printf("请输入一个整数：");

scanf("%d",&b);

char c[33];

if(b>=0)

{a=b;

for(i=0;i<32;i++)

c[i]=((a<<i)>>31)+48;

c[i]='\0';

puts(c);

}

if(b<0)

{a=-b;

for(i=0;i<32;i++)

c[i]=((a<<i)>>31)+48;

c[i]='\0';

c[0]='41';

puts(c);

}

} 3）测试

（a） 测试数据：

测试数据的方法如表1-1所示

表5-1 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 |
| x |  |  |
| 用例1 | 5 | 00000000000000000000000000000101 | 00000000000000000000000000000101 |
| 用例2 | 10 | 00000000000000000000000000001010 | 00000000000000000000000000001010 |
| 用例3 | 20 | 00000000000000000000000000010100 | 00000000000000000000000000010100 |

（b）运行结果：

对应测试测试用例1的运行结果如图所示。

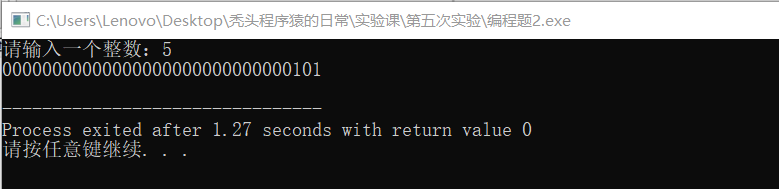


图5-9 编程题2的运行结果实例1

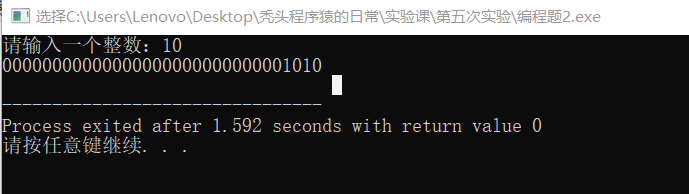
对应测试测试用例2的运行结果如图1-3所示。

图5-10 编程题2的运行结果实例2

对应测试测试用例3的运行结果如图1-4所示。

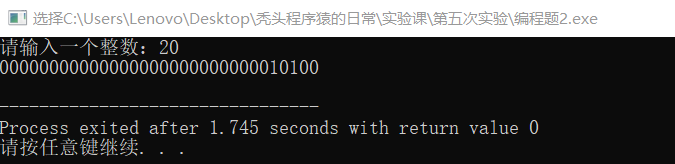


图5-11 编程题2的运行结果实例3

说明上述的运行结果与理论分析吻合，验证了程序的正确性。

**（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. 算法流程如图所示。

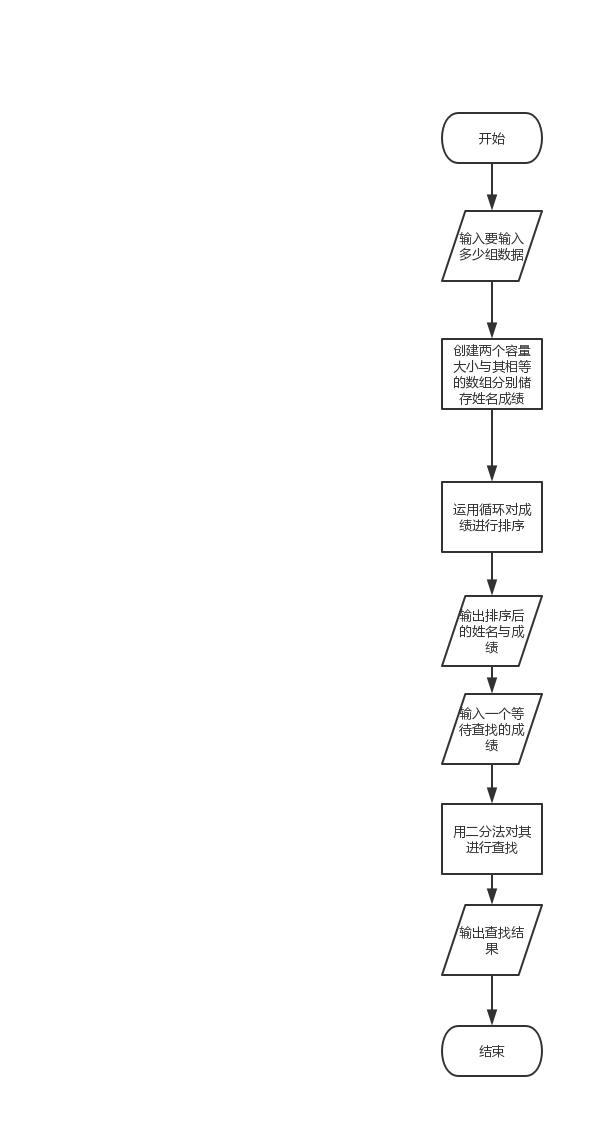


图5-12 编程题3的程序流程图

1. 源程序清单

#include<stdio.h>

#include<string.h>

int Search(int a[],int x,int n)

{int front=0,back=n-1,middle;

while(front<=back)

{middle=(front+back)/2;

if(x<a[middle])

back=middle-1;

else if(x>a[middle])

front=middle+1;

else

return (middle);

}

return -1;

}

int main(void)

{

int a,b,i,x,y,z;

printf("请输入有多少组学生成绩：");

scanf("%d",&a);

char c[a][15];

char f[15];

int d[a];

printf("请分别输入学生的姓名、成绩，用空格隔开");

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

scanf("%s%d",c[i],&d[i]);

for(x=0;x<a-1;x++)

{

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

{

if(d[i]<d[i+1])

{b=d[i];

d[i]=d[i+1];

d[i+1]=b;

strcpy(f,c[i]);

strcpy(c[i],c[i+1]);

strcpy(c[i+1],f);

}

}

}

printf("\n");

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

{strtok(c[i], "\n");

printf("%s %d\n",c[i],d[i]);

}

printf("\n请输入你要查找的成绩：");

scanf("%d",&y);

z=Search(d,y,a);

if(z==-1)

printf("Not found!");

else

printf("%s %d",c[z],d[z]);

}

1. }}测试
2. 测试数据：

a 70

b 80

c 99

查找 80

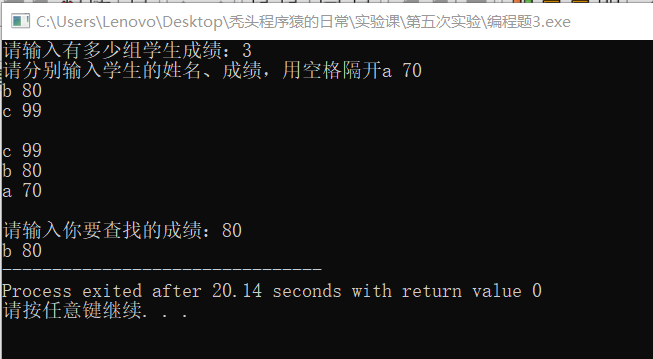
1. 对应测试测试用程序的运行结果如图所示。

图5-13 编程题3的运行结果实例

说明上述的运行结果与理论分析吻合，验证了程序的正确性。

**5.2.5 选做题程序设计**

编写并上机调试运行能实现以下功能的函数和程序。

(1) 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。

1）源程序清单

#include<stdio.h>

#include<string.h>

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

{int i,a,b;

a=strlen(s);

char c[100];

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

c[i-n]=s[i];

for(i=0;i<(strlen(t)-1);i++)

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

for(i=0;c[i]!='\0';i++)

s[i+n+strlen(t)-1]=c[i];

s[i+n+strlen(t)-1]='\0';

}

int main(void)

{char s[100],t[100];

int n;

fgets(s,100,stdin);

fgets(t,100,stdin);

scanf("%d",&n);

strnins(s,t,n);

puts(s);

}

2）测试

A）测试数据：aaaa

11

2

B）运行结果

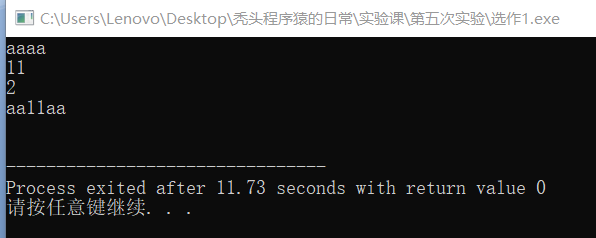


图5-14 选做题1的运行结果实例

(2) 编写一个实现八皇后问题的程序，即：在8\*8方格国际象棋盘上放置8个皇后，任意两个皇后不能位于同一行、同一列或同一斜线（正斜线或反斜线）上，并输出所有可能的放法。

1）源程序清单：

#include<stdio.h>

#define N 8

int a[N]={0};

int s=0;

int flag(int m1,int n1,int m2,int n2)

{

if(m1==m2||n1==n2||m1-m2==n1-n2||m1-m2==n2-n1)

return 0;

else

return 1;

}

void queen(int m)

{

int i,j;

if(m>=N)

{

printf("%d\n",++s);

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

{

for(j=0;j<a[i];j++)

printf("0");

printf("\*");

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

printf("0");

printf("\n");

}

return;

}

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

{

int t=1;

for(j=m-1;j>=0;j--)

{

if(flag(m,i,j,a[j])==0)

{

t=0;

break;

}

}

if(t==1)

{

a[m]=i;

queen(m+1);

}

}

return;

}

int main()

{

queen(0);

return 0;

}

2）运行结果：

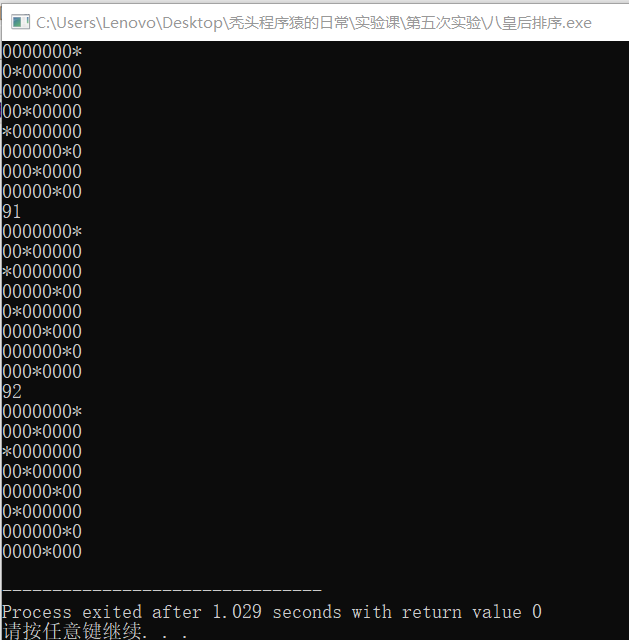


图5-15 选做题2的运行结果实例

5.3 实验小结

主要叙述实验过程中遇到的问题，如何解决的，通过分析、结果问题后的体会。  
 在这次实验的过程中，我在做八皇后问题时遇到了困难。最终我通过上网和查阅相关书籍解决了这一问题。

通过本次实验课，我深刻地体会到了“纸上得来终觉浅，绝知此事要躬行”这句话所蕴含的深刻道理。也许课上听完讲后我觉得自己已经掌握了相关的知识，但直到自己亲自动手编程是才发现自己的能力并没有想象中那么出色。希望我以后可以在这种活动中收获更多！

参考文献

[1] 曹计昌,卢萍,李开. C语言程序设计,北京： 科学出版社,2013

[2] 李开,卢萍,曹计昌. C语言实验与课程设计, 北京：科学出版社,2011