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 }

修改后代码：

#include <stdio.h>

int main()

{

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++)

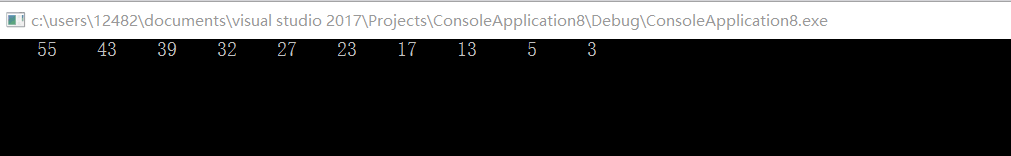
for ( j = 0; j < n - i - 1; j++)

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

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

}

运行截图：



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? : ; /\* 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;

}

修改结果：

#include<stdio.h>

#define M 10

#define N 3

int main()

{

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 \*/

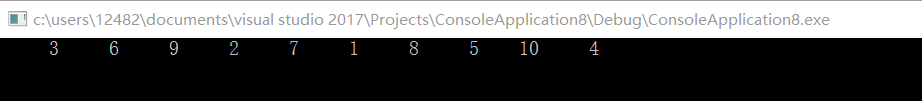
printf("%6d", b[i]);

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

return 0;

}

运行截图：



(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。

修改结果：

#include <stdio.h>

#define M 10

#define N 3

int main()

{

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

int i,j,k;

for(i=M,j=0;i>0;i--)

{

for(k=1;k<=N;k++)

{

if(++j>M-1)j=0;

if((j==0&&a[9]==1)||(j!=0&&a[j-1]==1))k=k-1;

}

b[M-i]=j?j:M;

a[j-1]=1;

}

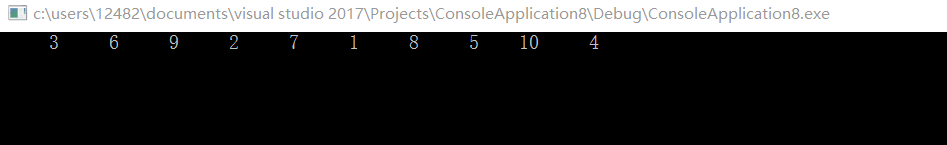
for(i=0;i<M;i++) printf("%6d",b[i]);

printf("\n");

return 0;

}

运行截图：

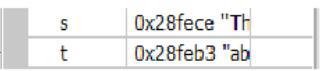


5.2.3 跟踪调试源程序

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';

}

修改后的程序：

#include <stdio.h>

#include <stdlib.h>

void strncat(char[],char[],int);

int main()

{

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

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。

程序代码：

#include <stdio.h>

int main()

{

int i, i0;

int array[3][4];

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

scanf("%d%d%d%d", &array[i][0], &array[i][1], &array[i][2], &array[i][3]);

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

printf("%5d%5d%5d%5d\n", array[i][0], array[i][1], array[i][2], array[i][3]);

printf("\n");

int array0[4][3];

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

{

for(i0=0;i0<3;i0++)

array0[i][i0]=array[i0][i];

}

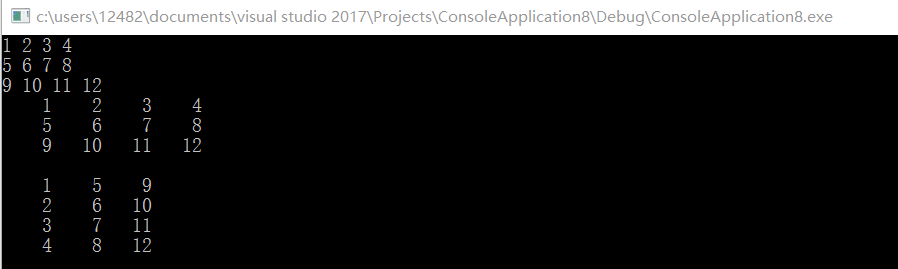
for(i=0;i<4;i++)

printf("%5d%5d%5d\n", array0[i][0], array0[i][1], array0[i][2]);

return 0;

}

运行截图：



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。

程序代码：

#include <stdio.h>

#define mask 1<<31

int main()

{

int i, N;

scanf("%d", &N);

int num[N];

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

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

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

{

char num\_s[32];

int i0;

for(i0=0;i0<32;i0++)

{

num\_s[i0]=((num[i]&mask)?'1':'0');

num[i]<<=1;

}

for(i0=0;i0<32;i0++) putchar(num\_s[i0]);

putchar('\n');

}

return 0;

}

运行截图：



1. 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!”。

程序代码：

#include <stdio.h>

int main()

{

int i, i0, i1, n, N;

scanf("%d", &n);

char name[n][100];

int grade[n];

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

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

int grade0;

char name0;

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

{

for(i0=n;i0>i;i0=i0-1)

{

if(grade[i0]>grade[i0-1])

{

grade0=grade[i0];grade[i0]=grade[i0-1];grade[i0-1]=grade0;

for(i1=0;i1<=99;i1++)

{

name0=name[i0][i1];name[i0][i1]=name[i0-1][i1];name[i0-1][i1]=name0;

}

}

}

}

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

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

printf("\n");

scanf("%d", &N);

int grades[N];

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

{

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

}

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

{

int r=0;

for(i0=1;i0<=n;i0++)

{

if(grades[i]==grade[i0]) {printf("%-20s %d\n", name[i0], grade[i0]);r=1;break;}

}

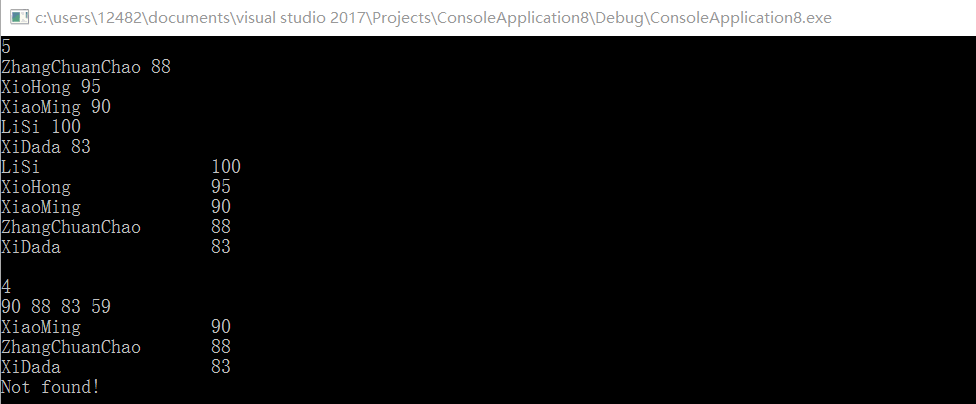
if(r==0) printf("Not found!\n");

}

return 0;

}

运行截图：



5.3 实验小结

源程序改错中，依赖于平时良好的输入习惯，在输入过程中已经可以检查出大多数语法错误，或者拼写错误。接下来再检查一下逻辑是否有误，无误则进行多组数据的编译检验，确认最终无误。

源程序修改与替换题锻炼我们将充分利用数组的能力，同时要注意数组下标值是从0开始的，时刻注意它和现实中的计数的不同，防止在引用过程中由于对应不同导致产生逻辑错误，严谨杜绝副作用。

程序设计题则是自主的设计实践，根据题目要求自主设计数组，增强对于所学知识的熟练度和理解程度，锻炼编写代码的能力以及并通过问题的进一步引申锻炼思维的灵活性。