



**程序设计实验报告4**

学 院

专 业

年级班别

学 号

学生姓名

指导教师

2018年 10 月

# 实验4

## 实验目的

1. 学会使用多文件多函数的编程；
2. 了解常见的排序算法；
3. 能够理解和使用指针变量；
4. 能够设计通过指针引用数组和字符串以及指向函数的指针的程序。

## 实验工具

Tubro C 2.0/3.0

## 编写要求

对每个实验题目，需要提供如下内容：

1. 程序流程图（应采用NS图或传统流程图）；
2. 具体程序代码（具有良好的程序设计风格，如程序模块化、加注释、缩进的书写格式）；
3. 实验运行结果（对比测试用例）；
4. 对运行情况所做的分析以及调试程序所取得的经验收获。

## 排版要求

1. 标题四号宋体，正文小四宋体，段间距1.5倍；
2. 除封面以外，其他内容双面打印。

## 实验题目

1． 多文件多函数编程

sort.prj -->sort.exe

有两个整型数组，一个长度为8，一个长度为3,将两个数组合并,存放在数组c中，合并和存放规则如下

s1.c:功能：先采用选择法将a数组按照从大到小进行排序，输出a数组的所有元素。函数原型：void sortA(int a[],int asize);

s2.c:功能：再采用冒泡法将b数组按照从小到大进行排序，输出b数组的所有元素。函数原型：void sortB(int b[],int bsize);

s3.c:功能：将a、b数组的元素进行冒泡法合并排序，使得a、b数组的元素是从小到大排序，最后将a，b数组元素依次拷贝到数组c当中。并将数组c的数据输出。函数原型：void sortAnB2C(int \*a,int asize,int \*b,int bsize,int \*c,int csize);

s4.c功能：主调函数

void main()

{

int a[8],b[3],c[11];

/\*输入a数组\*/；/\*输入b数组\*/；

sortA(a,8); sortB(b,3); sortAnB2C(a,8,b,3,c,11);

}

比如：a {8,3,1,5,4,9,-1,10} b {6,3,2}

(1)排序 a {10,9,8,5,4,3,1,-1} b {2,3,6}

(2)合并排序 a {-1,1,2,3,3,4,5,6} b {8,9,10}

(3) C {-1,1,2,3,3,4,5,6,8,9,10}

测试用例：（1） a:{8,3,1,5,4,9,-1,10} b:{6,3,2}

合并后结果 a:{-1,1,2,3,3,4,5,6} b:{8,9,10}

C:{-1,1,2,3,3,4,5,6,8,9,10}

（2） a:{11,8,7,6,4,3,2,1} b:{5,3,2}

合并后结果 a:{1,2,2,3,3,4,5,6,} b:{7,8,11}

**s1.c---NS图**

|  |  |  |  |
| --- | --- | --- | --- |
| int i, j, k, temp; | | | |
|  | for (i = 0; i < asize - 1; i++) | | |
| k=i | | |
| for (j = i + 1; j < asize; j++) | | |
|  | a[k]<a[j]  Y | N |
| k=j |  |
|  | k != i  N  Y | |
| temp=a[i]  a[i]=a[k]  a[k]=temp | |  |
| 输出数组a | | | |

**S2.c—NS图**

|  |  |  |  |
| --- | --- | --- | --- |
| int i, j, temp; | | | |
|  | for (i = 0; i < bsize - 1; i++) | | |
| for (j = i; j < bsize – 1 -i; j++) | | |
|  | b[j]>b[j+1]  Y | N |
| temp=b[j]  b[j]=b[j+1]  b[j+1]=temp |  |
| 输出数组b | | | |

**S3.c—NS图**

|  |  |  |
| --- | --- | --- |
| int i, j, k**;** | | |
| sort(a, asize); | | |
|  | for (i = 0, j = 0, k = 0; k < csize; k++) | |
| N  Y  j == bsize | |
| c[k] = a[i++]; |  |
| continue; |
| N  Y  i == asize | |
| c[k] = b[j++]; |  |
|  | continue; |
| c[k] = (a[i] < b[j]) ? a[i++] : b[j++]; | |
| 输出数组c | | |

**void sort(int a[], int n);**

|  |  |  |  |
| --- | --- | --- | --- |
| int i, j, temp; | | | |
|  | for (i = 0; i < n - 1; i++) | | |
| for (j = 0; j < n – 1 - i; j++) | | |
|  | a[j]>a[j+1]  Y | N |
|  | temp=a[j]  a[j]=a[j+1]  a[j+1]=temp |  |

**s4.c---NS图**

|  |
| --- |
| int a[8], b[3], c[11]; |
| 输入a[8], b[3]的元素 |
| sortA（a, 8） |
| sortB(b, 3) |
| sortAnB2C(a, 8, b, 3, c, 11) |

**代码：**

**s1.c**

void sortA(int a[], int asize)

{

int i, j, k, temp;

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

{

k = i;

for (j = i + 1; j < asize; j++)

if (a[k] < a[j])

k = j;

if (k != i)

{

temp = a[k];

a[k] = a[i];

a[i] = temp;

}

}

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

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

printf("\n");

}

**s2.c**

void sortB(int b[],int bsize)

{

int i, j, temp;

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

for (j = 0; j < bsize – i - 1; j++)

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

{

temp = b[j];

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

b[j + 1] = temp;

}

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

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

printf("\n");

}

**s3.c**

void sortAnB2C(int \*a, int asize, int \*b,int bsize,

int \*c, int csize)

{

int i, j, k;

void sort(int [], int);

sort(a, asize);

for (i = 0, j = 0, k = 0; k < csize; k++)

{//使用归并排序使两个数组合并

if (i == asize)

{

c[k] = b[j++];

continue;

}

if (j == bsize)

{

c[k] = a[i++];

continue;

}

c[k] = (a[i] < b[j]) ? a[i++] : b[j++];

}

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

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

printf("\n");

}

void sort(int a[],int n)

{

int i, j, temp;

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

for (j = 0; j < n – i - 1; j++)

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

{

temp = a[j];

a[j] = a[j + 1];

a[j + 1] = temp;

}

}

**s4.c**

#include <stdio.h>

int main()

{

extern void sortA(int a[], int asize);

extern void sortB(int b[], int bsize);

extern void sortAnB2C(int \*a, int asize, int \*b,

int bsize, int \*c, int csize);

int a[8], b[3];

int c[11];

int i;

for (i = 0; i < 8; i++)

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

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

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

sortA(a, 8);

sortB(b, 3);

sortAnB2C(a, 8, b, 3, c, 11);

return 0;

}

**sort.prj**

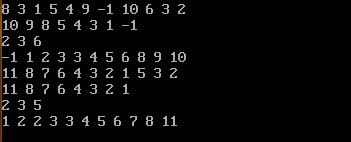
s1.c

s2.c

s3.c

s4.c

**运行结果：**



**总结：**s4.c中把a，b两个数组合并到c用的是归并排序，因为之前a数组是从大排到小，c数组要求是从小排到大，所以在函数中又写了一个冒泡来让a数组从小排到大，从而实现a，b的合并。