# 课程报告（样例）

学号\_\_\_\_\_\_\_\_\_\_\_ 姓名\_\_\_\_\_\_\_\_\_\_\_\_\_ 时间\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |
| --- | --- | --- |
| 教师评分 | **评分点1：能用图、表等记录实验结果，并撰写文档报告（2分：A:2.0 B:1.0 C:0.0 ）**  **【重点评价2、3部分】**  A：能用图、表、文字等完整地记录、描述实验过程及结果，报告格式规范，图表制作清晰，行文流畅  B：能用图、表、文字等记录、描述实验过程及结果，报告格式、图表等规范性不足，行文流畅性不足  C：未提交报告本部分为0 可上下浮动0.5分 |  |
| **评分点2：分析及表达个人见解（3分：A:3 B:1.5 C:0.0）【重点评价4、5部分】**  A：对测试中的问题，以及解题的重点、难点及关键问题有分析，阐述个人见解  B：对问题分析不足 C：未提交报告本部分为0 可上下浮动1分 |  |
| **评分点3：有参考文献及引用（0 ~ -3分：A:0 B:-1 C:-3 D:0）**  A：文末列举了相关参考文献，并在文中有引用：不扣分  B：文末列举了参考文献，但在文中引用不足：扣1分  C:未列举参考文献：减3分 D：未提交报告本部分为0 |  |
| 总分 |  |

标题，五号宋体加粗，正文，五号宋体不加粗，全文行距1.2倍，正文的段落首行缩进2字符。请用A4纸双面打印装订提交，注意控制表格、图等大小和格式，页数不要超过A4纸2张。(打印时去掉)

**0 题目**

案例二 排序问题

**1 问题描述**

本问题要求从键盘任意输入10个分数成绩（百分制，正整数），对这10个成绩按从高到低顺序排序，然后输出排序后的成绩。根据问题描述，本题假设输入数据一定为[0,100]之间的整数。

**2 问题分析**

输入：10个整型数据。由于输入数据为批量、同类型数据，故本题采用数组来存储数据。

输入数据描述：int scores[10];

输出：10个，整型数据。输出为排好序的scores数据元素值。

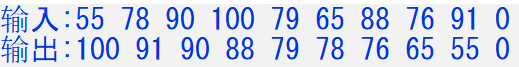
**3 算法描述**

此处绘制流程图。略

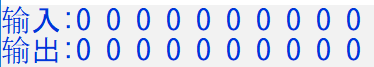
**4 实验结果记录**

设计测试数据一共5组。由于问题中假设输入数据是[0,100]之间的整数，故未对负数进行测试。

（1）测试数据1: 一组无顺的随机数



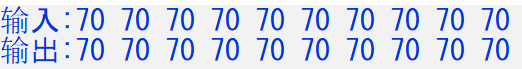
（2）测试数据2: 全部相同的数据



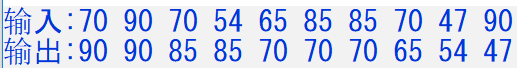
（3）测试数据3：……



（4）测试数据4：……



（5）测试数据5：部分相同的数据



**5 实验分析及结论**

**5.1 实验心得**

经过反复测试和调试，最终达到了问题要求。总结本次实验中遇到的问题，主要有三方面。

（1）在输入数据后出现程序不响应的情况。

经测试发现调用scanf函数时未使用地址符。将scanf(“%d”,scores[i]);

修改为：scanf(“%d”,&scores[i]);解决了该问题。

（2）输入测试数据，发现未进行排序。

编码时，误将 if(scores[j]<=scores[j+1]) 写成 if(scores[i]<=scores[j+1])，导致内循环中参加比较的数据不正确。这种失误主要还是由于编码前对算法分析得不够所致。

（3）需要特别注意数组越界的问题。

本次在编码时特别注意了检查数组边界，因此并未出现失误。在今后的练习中，还是应注意检查数组越界的问题。

**5.2 实验结论**

本实验是对任意10个[0,100]之间的整数排序。从7组测试数据的结果看，程序符合预期，全部测试结果均符合题目要求。本题采用了冒泡排序算法进行排序，但排序算法还可以采用改进的冒泡、选择排序、改进的选择排序等方法，这些算法在排序的思路、时间、空间性能方面存在差异。在后续学习过程中，将进一步对各种排序算法进行研究和比较。

**6 讨论一下**

回应案例的“讨论一下”中的问题。

（1）你还查阅了哪些排序算法？

我还查看了快速排序[1]、选择排序[2]、改进的选择排序[3]算法。

（2）你认为使用数组存在什么局限？

在C语言中，数组必须预定义大小。但在实际问题中，如数据规模不能预先确定，那么就必须预定义足够大的数组。比如，预计重庆理工大学本科生不超过3万人，所以数组长度定义为30000，但大多数时候，数组空间没有全部用完，这将降低内存使用效率。另外，数组长度一经定义，就无法扩展[4]大小，这也令程序数据不够灵活。

**7 程序代码**

#include <stdio.h>

int main()

{

int scores[10];

int i,j,temp;

for(i=0;i<10;i++) //输入分数

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

for(i=0;i<9;i++) //冒泡排序

{

for(j=0;j<9-i;j++)

{

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

{

temp=scores[j];scores[j]=scores[j+1];scores[j+1]=temp;

}

}

}

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

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

}

**参考文献：**

[1] [凉夏y](https://blog.csdn.net/LiangXiay). 快速排序详解. https://blog.csdn.net/LiangXiay/article/details/121421920

[2] [Reveralotf](https://www.zhihu.com/people/bai-ma-fei-ma-92-95). 排序算法学习分享（一）选择排序. https://zhuanlan.zhihu.com/p/153405930

[3] [yal49](https://blog.csdn.net/yangle4695). 选择排序及改进方法. https://blog.csdn.net/yangle4695/article/details/52214180

[4] [牛帅兵](https://blog.csdn.net/qq_36574108). 数组的使用及其不足. https://blog.csdn.net/qq\_36574108/article/details/122942494