

2019-2020 夏季学期

《计算机程序设计实训》课程安排

《计算机程序设计实训》课程组

(计算机工程与科学学院)

摘 要 根据课程目标,本文给出了《计算机程序设计实训》(0830A033、0869A001、08A6A006)课程的学习内容、实施方式以及课程要求等。同时,本文也作为模板展示了学生应该撰写的本课程小论文的格式要求。

关键词 程序设计实训;课程安排;文档格式模板

Arrangement of “Programming Training” in 2019-2020 Summer Semester

Teaching Team of “Programming Training”

(School of Computer Engineering and Science)

Abstract According to the aim of the course of “Programming Training”(course’s IDs: 0830A033, 0869A001 and 08A6A006), this paper gives the course arrangement, including the practical tasks, mode and some requirements. As an example, this paper also shows the document format of the course essay submitted by students.

Key words programming practical training; course arrangement; document format template

1 引言

本课程属于实践环节(必修),修课对象主要是2019级从理工大类分流至计算机科学与技术专业、智能科学与技术专业、网络空间安全专业的同学。本课程采用研究型学习方法,通过在线自动判题系统解题巩固C/C++程序设计基础,通过C/C++程序实验分析排序算法锻炼算法分析和数据处理能力,通过C-字符串处理练习强化训练指针操作技能。

在实验研究的基础上,以本文的格式为模板撰写课程小论文,并按时提交到学校“网上教学”平台本课程教学空间的“作业”中。该教学平台的“资料”中有部分源代码供参考、有两个C/C++程序集成开发环境供选用。通过本课程实训,使同学们进一步熟练掌握C/C++程序设计方法,熟练掌握数组、指针、函数以及标准I/O函数的使用,并初步掌握算法分析及基本的数据处理技巧。

2 实训内容

本课程的实训内容大体上包含3个方面。①在线自动判题系统解题典型I/O方法;②若干种排序算法的程序实验研究;③C-字符串处理算法与实现。

在线自动判题系统解题过程中，数据的输入与输出是一个重要环节。正确地读取测试数据、严格地按要求输出数据对“通过（Accepted）”解题至关重要。

排序是数据处理中的常用基本操作，它通过元素间的比较、交换或移动（多次赋值）实现数据元素按某种顺序进行重新排列。采取不同的比较、交换或移动策略形成了不同的排序算法。不同的排序算法中对数据元素进行的比较次数、赋值次数可能有较大的差别，影响整个排序操作的效率。排序算法实验研究是通过统计排序算法函数的运行时间、数组元素间比较次数、数组元素间赋值次数，用图、表等方式来展现并分析相应算法的时间复杂度、空间复杂度、排序的稳定性并得出一定的结论。

C-字符串是 C/C++ 程序设计中常用的数据。然而理解和掌握 C-字符串的基本概念、处理方法却涉及到“容器”、数组、指针等概念。因此，C-字符串处理是训练同学们掌握数组与指针及其在函数中传递等程序设计能力的极佳素材。

在上海大学“网上教学”平台^[1]本课程空间的“资料”中有部分源代码及 C/C++ 集成开发环境（Integrated Development Environment, IDE）供参考和选用。具体资料见表 1 所示。

表 1 上海大学“网上教学”平台本课程空间中的参考资料

序号	文件名	说明
1	2020 夏《计算机程序设计实训》 课程方案 20200630.docx	本文件。同时作为课程小论文格式模板。
2	OJs.zip	源代码。在线自动判题系统练习题集（前 5 个为例题，含测试数据）。
3	Sorts-C.zip 或 Sorts-CPP.zip*	源代码。包含三种基本排序算法，数据生成方法、运行时间测试方法等。其中 Sorts-C.zip 是纯 C 语言版的程序，Sorts-CPP.zip 是 C++ 版的程序。两种版本程序的功能相同。需在此源代码（二选一即可）基础上修改，以实现算法分析目标。 提示：请关注 Deubg 和 Release 文件夹中的批处理文件 run.bat。鼠标右键单击该文件并选择“编辑”可查看其内容。鼠标左键双击该文件，可运行 Sorts.exe 文件，并将测试结果保存在新创建的文本文件中。Open.bat 则用 Excel 打开测试结果文件，以便进行分析。
4	C-String.zip	源代码。C-字符串处理练习参考代码，待完成。
5	习题课.zip	习题课课件 PPT。
6*	SortShow 排序秀.zip	源代码。排序可视化示例。
7*	舞动的排序算法.zip	若干排序算法表演、排序可视化。
8	MinGWStudioFullSetup- 2.05r10.exe	C/C++ 集成开发环境（推荐使用）。
9*	Codeblocks-16.01mingw- setup.exe	C/C++ 集成开发环境。

【注】标*号的文件可不必理会。

2.1 在线自动判题系统练习及测验

2.1.1 在线自动判题系统概述

程序设计在线自动判题系统采用“黑盒测试”方法判定解题源程序代码的正确性。本次练习基于“上海大学 Online Judge”平台（<http://acmoj.shu.edu.cn>）的“比赛”系统^[2]。本课程班级同学以自己的学号为用户名和初始密码登录（请登录后立即修改密码并牢记）。

第 173 号比赛“2019-2020 夏季学期《计算机程序设计实训》课程练习”为本课程练习。起止时间为 2020 年 7 月 27 日 00:00 至 2020 年 8 月 6 日 00:00。本练习的前 5 题为例题，其解题源代码和测试数据集全部公开（参见学校“网上教学”平台本课程空间^[1]或表 1）。这些练习的主要目的是解决在线自动判题程序设计中数据的读取方法问题，也为测验做准备。

2020 年 8 月 5 日（星期三）13:00-17:00，全体同学参加上海大学程序设计联赛夏季赛（无须另行报名）作为本课程的测验。参赛的相关网址等信息另行通知。

2.1.2 在线判题系统解题完整示例

以第一题“阶乘尾部 0 的个数”为例，详细说明题目、解题程序源代码、自测方法及提交等环节。

2.1.2.1 题目及其测试数据

问题描述	给定非负整数 n ，计算 n 的阶乘尾部 0 的个数。
输入	输入数据有若干行，每行上有一个非负整数 n ，对应一种情形。
输出	对于每一种情形，直接输出结果、换行。
输入样例	8 16 30
输出样例	1 3 7

图 1 公开的题目样式

在线自动判题系统的后台服务器上，有包含更多测试数据的文件（假设为 01in.txt 及由所谓的“标准程序”产生的标准结果文件 01out.txt），它们的内容如下（一般地，这些文件的内容是不公开的）。由图 2 可见：①实际测试数据与题目中的样例数据不同，且数量多；②实测数据中存在一些特殊情况（如：0 的阶乘等）。

01in.txt	01out.txt
0	0
1	0
4	0
5	1
6	1
24	4
25	6
26	6
124	28
125	31
126	31

图 2 实际测试数据文件及标准结果文件的内容

2.1.2.2 解题算法及程序实现

一般地，测试数据包含多种情形（即多种数据用例，具体数目解题者并不知晓），以考察解题程序的算法的正确性或适用面。测试数据的一种情形可能只占一行，也可能占多行；一行上的数据个数也可能确定或不定，故需要仔细阅读题目中的“输入样例”的说明。利用某种 C/C++ 程序集成开发环境，根据题目的要求，用标准 I/O 正确地读取输入数据、严格地按输出格式要求输入计算结果。请仔细阅读前 5 道例题的解题源程序，掌握其中有关测试数据的 I/O 处理方法。

仍以第一题为例，我们知道，用 `unsigned long` 类型或 `unsigned long long` 类型计算阶乘时，分别计算到 $13!$ 或 $21!$ 都会因溢出而出错。倘若使用 `double` 或 `long double` 类型，则因为它们有限的有效位而导致难以计算阶乘尾部 0 的个数（一般地，对于整数问题，不到万不得已不要借道到近似表示、近似计算的浮点型数据中去处理）。因此，需要另外设计算法解决该问题。

我们知道， n 阶乘是一系列整数的乘积 $1 \times 2 \times \cdots \times n$ 。设想已经将 $n!$ 进行质了因数分解 $2^{k_2} \times 3^{k_3} \times 5^{k_5} \times 7^{k_7} \times 11^{k_{11}} \times \cdots$ 。对 $n!$ 尾部出现 0 有“贡献”的是质因数 2 和质因数 5，显然有 $k_2 \geq k_5$ ，因此本题的解为 $\min(k_2, k_5) = k_5$ 。如下的程序中，`int zeros(int n)` 的功能便是对于给定的 n 计算 k_5 的函数。

```
// OJ01.c          阶乘尾部 0 的个数
#include <stdio.h>

int zeros(int n)
{
    int m=0, p=5;
    while(p<=n)
    {
        m += n/p;
        p *= 5;                // p 的变化规律为 5 的 1,2,3,...次方
    }
    return m;
}

int main()
{
    int n;
    while(scanf("%d", &n)==1)    // scanf 函数的返回值是其读取到的数据个数
        printf("%d\n", zeros(n));
    return 0;
}
```

图 3 解题源代码

2.1.2.3 自测方法及提交

解题源程序通过编译和链接生成本地的可执行文件后，需要进行自测。至少，所设计的解题程序对于题目中给出的输入样例数据进行试算，应该能够得到与输出样例相同的结果。可能的话，还需要进行更多的、包括边界数据等特殊情况数据的测试（当然，这些额外的测试用例，解题者应该知道“正确”的答案）。

初步通过自测后，便可以按在线自动判题系统的提示进行程序**源代码提交**（Submit）了（千万不要提交可执行程序文件）。源代码提交到判题系统后，将在该判题系统所在的服务器上重新编译生成该服务器上的操作系统（如：Linux）下的可执行文件，然后由判题系统利用标准 I/O 重新定向功能再启动该可执行文件

运行，得到该程序对测试数据的输出结果文件，通过比较该文件与“标准程序”生成的“标准结果文件”的对比情况反馈评判结果。**Accepted**（简记 **AC**）表示通过。不通过时有多种原因，包括 **Compile Error**（编译错。首先检查在本地计算机上能否通过编译，再检查是否有头文件没有包含，因为服务器上的编译系统与本地的编译系统有差别，而服务器上的编译系统更加严格）、**Runtime Error**（运行时错。需要检查数组下标是否越界、是否对空指针进行了操作、是否有可能除数为 0，等等）、**Time Limit Exceeded**（超时。需要检查程序中是否有无穷循环、算法是否应该优化以减少计算量）、**Wrong Answer**（答案错。需要检查算法是否正确、考虑问题是否周全。或者输出格式是否严格按照要求，如英文字母大小写、标点符号、空格等是否有误）。

一般的在线自动判题系统都会记录每次提交的情况，对不通过的提交会增加累计用时（俗称为“罚时”），直到该题通过。在线自动判题系统还有一个“排行榜”以所有答题者的 **AC** 数排序，**AC** 数相同时“用时+罚时”少者在前。

(1) 自测方法一

直接运行程序：从键盘上依次输入测试样例数据，则逐步在显示器上输出结果，观察这些结果是否与“输出样例”一致；最后输入组合键 **[Ctrl+Z]** 表示测试数据输入完毕，结束测试。

请注意：此时输入与输出交织在一起，属于正常现象。

(2) 自测方法二

在可执行文件所在的文件夹（例如 **Debug**）中，用 **Windows** 的“记事本”①建立文本文件（如文件命名为 **Input.txt**），将题目中的输入样例拷贝/粘贴到文件中并保存；②建立一个文本文件（如文件命名为 **run.bat**，先建立 **run.txt** 文件，然后将扩展名改成 **.bat**），其内容仅一行，如图 4 所示。

```
OJ01.exe < Input.txt > results.txt
```

图 4 批处理文件 **run.bat** 的内容

然后用鼠标双击 **run.bat** 文件，运行 **OJ01.exe** 文件。此时，操作系统对程序中的标准输入语句转向从 **Input.txt** 文件去读取数据，并将程序中的标准输出语句转而存入（新）文件 **results.txt** 中。其中的“<”为操作系统标准输入重新定向符，“>”为操作系统标准输出重新定向符。最后比较 **results.txt** 文件的内容与输出样例是否相同。

2.2 排序算法实验研究

2.2.1 三种排序算法分析

对于所涉及的排序算法，本课程的重点在于实验分析或实验验证，即：不要求进行严密地理论分析，需要时直接引用相关的理论结果。

首先研究基本的冒泡排序算法、选择排序算法和快速排序算法（如参见表 1 提到的 **Sorts-C.zip** 文件）测试三种排序算法的时间。

其次，利用 **Score** 结构体数组讨论排序算法的稳定性。所谓排序的**稳定性**是指：假定在待排序的记录序列中，存在多个具有相同的关键字的记录，若经过排序，这些记录的**相对次序保持不变**，即在原序列中，若 $x[i]$ 与 $x[j]$ 相等，且 $x[i]$ 在 $x[j]$ 之前，而在排序后的序列中， $x[i]$ 仍在 $x[j]$ 之前，则称这种排序算法是稳定的；否则称为不稳定的^[3]。对于不稳定的排序算法只要举出一个实例即可。而对于稳定的排序算法，则不展开深入的理论研究（给出结论即可）。

最后，对 **double** 型数组的 3 个排序函数进行修改，在每个函数中增加 2 个无符号扩展的长整型指针形参（**unsigned long long ***），分别用于间接“返回”相关函数执行数组元素间的比较次数、数组元

素间赋值次数（不统计辅助操作的次数，即不统计下标比较、下标增减等操作次数）。根据统计结果，对不同排序算法进行对比分析。

2.2.2 测试用例设计

(1) 数据类型

测试数据已经包括了基本数据类型 `int` 和 `double`，以及如下结构体类型的数组。

```
struct Score
{
    char Id[9]; // 学号，最多存放8个字符
    int Chinese, Math, English, Physics, Chemistry, Total; // 5门课程成绩及总分
};
```

图5 数据的组织形式描述

(2) 数据生成

对于结构体数组各元素，学号字符串从"00000001"起顺序编排，各科成绩的分数按 Box-Muller 算法产生服从正态分布（即高斯分布） $N(\mu, \sigma^2)$ 的随机数，然后计算总分。例如：可取均值 $\mu = 72$ ，方差 $\sigma^2 = 36$ ，即根方差 $\sigma = 6$ 。根据著名的“ 3σ 原则”，数值分布在区间 $(\mu - 3\sigma, \mu + 3\sigma) = (54, 90)$ 中的概率为 99.74%，亦即，只有不到 3% 的数值分布在该区间之外。

对于基本数据类型（`int` 或 `double`）数组，各元素分别按 3 种方式生成。

- ① 随机生成服从区间[50,100]上均匀分布的随机数（`int` 型则采用四舍五入取整）。
- ② 随机生成服从均值 $\mu = 72$ ，方差 $\sigma^2 = 36$ 的正态分布的随机数（`int` 型则采用四舍五入取整）。
- ③ 特殊构型的数据，如完全顺序的数据、完全逆序的数据。

扩展研究内容（可选）：如何度量给定数组的有序程度？如何产生具有指定有序度的数组？（该问题目前没有很好的答案）

2.2.3 对比测试与分析

(1) 执行时间测试

如下自定义函数可用于测试程序中某段代码的执行时间（参见 Sorts-C.zip 中的源代码）。

```
double gettime(int restart) // 参数非零表示设置新的计时起点
{
    const double c = 1.0/CLOCKS_PER_SEC; // 毫秒转换至秒
    static clock_t t = 0; // t 为静态局部变量，存储计时起点
    if(restart || t==0) t = clock(); // 根据实参决定是否重新确定计时起点
    return c*(clock()-t); // 从上一计时点到现在所经历的时间差
}
```

图6 获取代码段执行所需时间的函数

(2) 排序算法对数组元素间比较和赋值次数统计

由于测试数据的随机性，排序算法函数在每次执行时所花费的时间、统计得到的元素间操作次数均可能不同。为了使实验结果具有可比性，有如下建议。

- ① 数据规模从 1024, 2048, ..., 65536;

② 不同算法所用的随机数据保持相同（用一个数组存放所谓的“原始数据”，将原始数据拷贝到另一个数组进行排序操作）；

③ 测试时关闭网络以及一些无关的程序；

④ 尝试测试调试版（Debug）和发布版（Release）的可执行程序。以 MinGW Developer Studio 集成开发环境为例，在 Build 的下拉菜单中选 Build All Configurations。在 Debug 和 Release 文件夹中建立批处理文件（如：run.bat），其内容与下面的类似。

可执行文件名 >> 文本文件名.txt

图 7 批处理文件的内容

它利用操作系统（如 Windows）的标准输出重新定向功能，将程序的输出存入的指定的文本文件中，以利于进一步分析。

⑤ 尝试执行其中的批处理文件 Open.bat，它利用 Excel 软件打开程序输出所产生的测试结果数据文件。在 Excel 软件中，选择一定单元格的数据可方便地制作数据分析用的柱状图、折线图和表格等。

从不同算法之间、不同数据类型之间、不同测试数据构型、不同的编译结果（Debug 和 Release）多个维度展开讨论，进行归纳提炼，给出一定的实验研究结论，并思考可能的进一步改进方向。

2.3 C-字符串处理

字符串是程序设计中常用的一种数据。然而 C 语言中的字符串（常称为 C-字符串）的概念、操作却涉及到数组、指针、函数等概念。也正是这个原因，使得有关 C-字符串的练习成为理解和掌握数组、指针、函数操作的极佳素材。

C-字符串按字符数组的形式存放在计算内存中，并且以特殊字符 '\0' 作为串结束标志。其中的字符数组被称为字符串的**容器**，一般数组、动态数组（即堆数组）或常量数组都可能成为 C-字符串的容器。C-字符串的容器常常需要程序员花额外的精力进行管理，特别是需要保证有“足够大”的容量。

由于 C-字符串在内存中连续存放，又有串结束标志字符，并且认为其容器已经设置妥当且容量“足够大”，在这样的前提下，在 C 语言程序中，常将 `char*` 或 `const char*` 类型理解为 C-字符串（变量）及 C-字符串常量。

本课程将 C-字符串的练习内容集中到同一个程序中（参见表 1 中提到的 C-String.zip 文件），分成如下 3 个部分。请运行“C-String 可执行文件参考.exe”，它是完成该练习后应有的结果。

2.3.1 自定义函数实现 C-字符串基本操作

C 语言中，对 C-字符串的操作体现在若干标准函数，如 `strlen`、`strcpy`、`strncpy`、`strcat`、`strcmp`、`atoi`、`atof`、`gets`、`puts`、`sprintf`、`sscanf` 等。

本部分练习要求实现如下 5 个函数，函数原型已经给定（参见 C-String.h 文件）。

<code>unsigned int StrLen(const char *str);</code>	// 计算字符串的长度
<code>char *StrCpy(char *dest, const char *source);</code>	// 字符串复制（拷贝）
<code>char *StrCat(char *dest, const char *source);</code>	// 字符串拼接
<code>int StrCmp(const char *str1, const char *str2);</code>	// 字符串比较（按 ASCII 序）
<code>int Atoi(const char *str);</code>	// 将字符串转换成整数

图 8 自定义版 C-字符串处理函数的原型

其中的函数名按包含大小写字母的“驼峰”方式命名。要求实现与之对应于的全小写的标准函数具有的功能。请在 C-String.c 文件中完成 5 个函数的定义，测试这些函数的数据已给定（参见 test.c 文件）。

2.3.2 交换 C-字符串

要求完成 4 个函数的设计，包括函数原型设计、函数定义、测试代码中函数调用。请思考 SWAP1 函数和 SWAP2（或 SWAP3）函数应用场合能否互换，其原因是什么。要求函数原型设计时尽量做到精准。

2.3.3 C-字符串数组排序

一系列字符串的集合称为字符串数组。字符串数组有两种不同的存储形式：①二维数组；②字符指针数组。已经设计了两个不同的函数 Show1 和 Show2 分别用于输出两种形式的字符串数组个元素的值。对于这两种形式，请分别设计函数对它们进行排序。要求任意选择一种排序算法（例如冒泡排序），编写如下 4 个排序函数。

表 2 字符串数组的四个排序函数名设计表

排序规则 数组形式	字典序 (即 ASCII 序)	长度序(长度相等 时按字典序)
二维数组	Sort1A	Sort1L
字符指针数组	Sort2A	Sort2L

3 时间安排和要求

3.1 课程表

实训时长为 2 周，在线习题课、在线研讨或答疑安排如下表。其余时间由同学们自行安排。

表 3 课程表

星期 日期	星期一	星期二	星期三	星期四	星期五
第三周 7.27-7.31	习题课 1 (OJ 解题方法) 3-4 节 0830A033 (计科) 7-8 节 0869A001 (智能)、 08A6A006 (网安)			习题课 2 (排序及字符串) 3-4 节 0830A033 (计科) 7-8 节 0869A001 (智能)、 08A6A006 (网安)	
第四周 8.3-8.7	研讨或答疑 3-4 节 0830A033 (计科) 7-8 节 0869A001 (智能)、 08A6A006 (网安)		13:00-17:00 测验 (上海大学 程序设计联赛)	研讨或答疑 3-4 节 0830A033 (计科) 7-8 节 0869A001 (智能)、 08A6A006 (网安)	

- 【注】 1. 欢迎同学们踊跃参加在线研讨：提出问题、讨论解决方案、报告研究进展等。
2. 在线习题课、研讨或答疑均同时使用超星（签到）和 zoom 会议（具体的 zoom 会议信息参见 <http://ncov2019.shu.edu.cn/>，及“网上教学”本课程平台的“通知”）。

3.2 课程小论文要求

围绕本文指定内容开展研习、计算机编程实验，并对结果进行分析。在上述研习的基础上，主要针对排序算法实验研究、C-字符串处理，每位同学根据自己的研究成果、心得和体会进行归纳提炼，自拟课程小论文的题目。要求题目文法合理、达意，切忌笼统含糊。小论文的各种要件不要缺少，仿照本文格式撰写（由

于 Word 版本不同可能造成格式上有所差异将被认可)。图表必须有说明、公式必须有编号。正文可列出部分关键程序代码,所列的程序代码需要有足够的注释。课程小论文中还需要撰写课程心得体会,以及对课程的意见和建议。并按表 4 的要求在规定的时间内(课程结束后一周)内将规定的材料上传至“网上教学”平台本课程空间的“作业”中。

4 结语

综上所述,本课程的具体任务和要求如表 4 所示。

表 4 实训任务汇总

序号	任务	说明
1	在线自动判题系统解题、测验(程序设计竞赛)	① OJ 解题(5 道例题、10 道练习题,都需要提交); ② 上海大学程序设计联赛夏季赛,相关网址等信息另行通知。
2	排序算法的程序实验研究	① 通过运行时间、数组元素间比较次数和赋值次数的统计,分析算法的效率; ② 利用 Score 结构体数组实验讨论几种排序算法的稳定性; ③ *可选内容。讨论数组有序程度的度量方法、生成方法(参见第 2.2.2 小节)。
3	C-字符串处理	① 基本的 C-字符串处理函数(完成 5 个函数的定义); ② C-字符串交换(完成 4 个函数,并调用测试); ③ C-字符串数组排序(完成 4 个函数,并调用测试)。
4	撰写课程小论文	主要针对排序算法实验研究、C-字符串处理两部分内容,概述相关算法原理、结果分析或阐述关键技术等,取得的成果、收获体会等。 小论文题目自拟,正文部分要尽可能图文并茂(如柱状图、折线图,实测结果表等)、语言流畅、表达准确。按本文的格式,各种要件(中英文标题、作者、单位、摘要、关键词,参考文献)不要缺少。
5	提交材料要求	将所完成的全部源代码、实验结果数据文件、课程小论文打包压缩成一个文件,文件名按“学号-姓名-计算机程序设计实训”为主名,于 2020 年 8 月 15 日 0 时 0 分前上传至上海大学“网上教学”平台本课程教学空间的“作业”中。

另外,本文作为格式模板主要设置了“样式^①”、“段落”中的行距选项。采用的中文字体主要是黑体、宋体,英文字体主要有 Times New Roman 和 Courier New。这样,便于更好地控制版面。

由于疫情的影响,本次课程无法在实验室机房里与同学们面对面地交流。采取线上教学虽是无奈之举,却能更加锻炼同学们的自学能力、创新研究能力。本课程开展计算机程序实验研究,是践行“在研究学习和成长”的本科研究型教学模式的一种探索。可望在同学们的努力下取得好的效果。

致谢 虽然有“疫情”的干扰,相信在大家的共同努力下,本课程的教学工作将顺利完成。希望并感谢同学们积极主动地开展线上、线下的研讨活动。

参 考 文 献

- [1] 上海大学“网上教学平台” <http://www.elearning.shu.edu.cn/portal> (需要登录并进入本课程教学空间)
- [2] 上海大学 Online Judge 在线自动判题系统 <http://acmoj.shu.edu.cn>
- [3] 百度百科 <https://baike.baidu.com/item/排序算法稳定性/9763250>

^①小提示: 在本文件中输入一行文字,将其“样式”设置成“大标题”,然后换行。在新的一行中自动按作者姓名格式等待用户输入作者姓名。再换行输入作者单位、再换行……(惊喜不惊喜😊)。