**个人项目：WC**

实践是理论的基础和验证标准，希望读者贯彻“做中学”的思想，动手实现下面的项目，并和别人的成绩相比较，分析产生差距的原因。

1. 实现一个简单而完整的软件工具（源程序特征统计程序）。  
2. 进行单元测试、回归测试、效能测试，在实现上述程序的过程中使用相关的工具。  
3. 进行个人软件过程（PSP）的实践，逐步记录自己在每个软件工程环节花费的时间。  
 **1　WC 项目要求**

wc.exe 是一个常见的工具，它能统计文本文件的字符数、单词数和行数。这个项目要求写一个命令行程序，模仿已有wc.exe 的功能，并加以扩充，给出某程序设计语言源文件的字符数、单词数和行数。

实现一个统计程序，它能正确统计程序文件中的字符数、单词数、行数，以及还具备其他扩展功能，并能够快速地处理多个文件。  
具体功能要求：  
程序处理用户需求的模式为：

wc.exe [parameter] [file\_name]

基本功能列表：

wc.exe -c file.c //返回文件 file.c 的字符数

wc.exe -w file.c //返回文件 file.c 的词的数目

wc.exe -l file.c //返回文件 file.c 的行数

扩展功能：  
    -s   递归处理目录下符合条件的文件。  
    -a   返回更复杂的数据（代码行 / 空行 / 注释行）。

**空行：**本行全部是空格或格式控制字符，如果包括代码，则只有不超过一个可显示的字符，例如“{”。

**代码行：**本行包括多于一个字符的代码。

**注释行：**本行不是代码行，并且本行包括注释。一个有趣的例子是有些程序员会在单字符后面加注释：

    } //注释  
在这种情况下，这一行属于注释行。

[file\_name]: 文件或目录名，可以处理一般通配符。  
  
高级功能：

 -x 参数。这个参数单独使用。如果命令行有这个参数，则程序会显示图形界面，用户可以通过界面选取单个文件，程序就会显示文件的字符数、行数等全部统计信息。

需求举例：  
　　wc.exe -s -a \*.c

返回当前目录及子目录中所有\*.c 文件的代码行数、空行数、注释行数。

**2.　工作的细分**

正如谚语所说：不能一口吃成个胖子。罗马不是一天建成的。同样，一个功能完备的程序也不是一蹴而就的。在这里，我们讨论如何把大任务划分为可操作的小任务，以及任务的次序。  
读完项目的要求后，首先请估计完成整个项目需要多少时间？把这个[ 估计值 1] 记下来，并且写成PSP 的形式。其次，如何逐步分解一个项目的需求？在这个项目中，各种需求已经通过各种参数表达得比较清楚了。

基本功能  
扩展功能  
高级功能

详细地了解了需求后，我们再估计需要的时间并记录 [ 估计值2]。  
最后，列出各类功能下面的详细需求。

基本功能

支持 -c  
支持 -w  
支持 -l

扩展功能

支持 -s 参数  
支持 -a 参数  
支持各种文件的通配符（\*,?）

高级功能

基本的Windows GUI 程序操作  
支持通过图形界面选取文件  
支持通过图形界面展现文件的信息

请在这个时候把每一个详细功能所需的时间列出来，然后再把它们相加，得到[ 估计值 3]。  
同学相互之间比较估计值1、估计值2 和估计值3，看看差距是多少，有什么规律？

**3　如何保证质量— 回归测试**

如何保证在加入新功能的过程中，已有的功能可继续工作？这就要求我们有一套标准的测试来保证基本功能的正确性。  
写这样的程序，用项目本身的源文件来做测试应该是比较酷的，但是源文件本身在不断地变动，并不是一个很好的测试样本，我们要建立起一系列测试文件。如下：

空文件  
只有一个字符的文件  
只有一个词的文件  
只有一行的文件  
一个典型的源文件

如何为这个程序做有效的测试，有以下几种方法，自动化程度由低到高。

1. 手动测试，手工比较。  
2. 要做到不断地测试，可以把WC 的主要功能封装成一个类，然后测试程序调用这一个类的主要函数，得出结果并与标准作比较。  
3. 更进一步，把测试文件和正确的测试结果保存在文件中，测试驱动程序只要比较测试的输出和标准结果就能得出答案。  
4. 再进一步，把自动构建脚本和构建验证测试（Build Verification Test）结合起来。每一次构建之后，就自动运行测试，然后记录出现的Bug。  
了解测试的需求后，每人估计需要的时间并记录[ 估计值4]。

**4　标准测试集，正确性和速度评比**

既然大家的程序都写得差不多了，那就拉出来遛遛，看看是骡子是马。以子目录的形式把目前所有同学的程序都集中在一个总的测试目录下，作为测试集合。然后大家看看各自的程序要花  
多少时间才能正确并且较快地完成任务。 在这里，同学们要记下满足了标准测试集之后，每人实际花费的时间 [ 实际值]，并且按照本章PSP 的表格统计自己在软件开发的各个阶段所花费的时间。