结对编程小组:熊泽廷 王路庆

1．概要部分

（1）代码能符合需求和规格说明么？
答：能

（2）代码设计是否有周全的考虑？
答：是

（3）代码可读性如何？
答：能读懂

（4）代码容易维护么？
答：能运行出来

（5）代码的每一行都执行并检查过了吗？
答：检查了

2．设计规范部分

（1）设计是否遵从已知的设计模式或项目中常用的模式？
答：是
（2）有没有硬编码或字符串/数字等存在？

答：没有

（3）代码有没有依赖于某一平台，是否会影响将来的移植（如Win32到Win64）？

答：不会影响的

（4）有没有无用的代码可以清除？

答：也许有无用的代码，但是没有删除。因为删除后，又会出现错误。

3．代码规范部分

（1）修改的部分符合代码标准和风格么?

答：符合。

4．具体代码部分

（1）有没有对错误进行处理？对于调用的外部函数，是否检查了返回值或处理了异常？

答：对错误有处理，并改正。而且改正后有检查了。

（2）参数传递有无错误，字符串的长度是字节的长度还是字符（可能是单/双字节）的长度，是以0开始计数还是以1开始计数？

答：中级计算中并未大量运用字符串.

（3）边界条件是如何处理的？Switch语句的Default是如何处理的？循环有没有可能出现死循环？

答：多次测试确保满足条件.

（4）有没有使用断言（Assert）来保证我们认为不变的条件真的满足？

答：没有。

（5）对资源的利用，是在哪里申请，在哪里释放的？有没有可能导致资源泄露（内存、文件、各种GUI资源、数据库访问的连接，等等）？有没有可能优化？

答：用的都是c语言的标准库，无需优化

（6）数据结构中是否有无用的元素？
答：没有

5．效能

（1）代码的效能（Performance）如何？最坏的情况是怎样的？
答：代码比较完整，程序可以运行出来，目前还未有运行不出来的情况。

（2）代码中，特别是循环中是否有明显可优化的部分

答：可以，repeat部分难以设计出来.

（3）对于系统和网络调用是否会超时？如何处理?

答：不会超时，出现问题就通过网络，老师，图书等途径解决。

6．可读性
代码可读性如何？有没有足够的注释？

答：有必须的注释。

7．可测试性
代码是否需要更新或创建新的单元测试？
还可以有针对特定领域开发（如数据库、网页、多线程等）的核查表。

答：不需要。