

## Python 代码检查及白盒测试部分

执行本测试前，开发人员是否有提供本地/单元测试的测试结果或使用范例：

请提供相关的文档链接或在此处进行描述，无则可跳过：

代码检查部分			
测试项目	是否发现缺陷(通过:P, 不通过:F)	缺陷描述（如已上传至 GitLab,则仅提供对应的 GitLab Issue 号即可）	备注（请提供脚本文件名、有缺陷的 Function(Method)名和代码的具体位置,如 calculate.py 中第 15 行）
代码风格部分			
是否使用了 SonarQube Scanner 进行了扫描，且未出现严重问题（如不知道如何扫描可先标注为 F）			
方法的注释是否符合规范（参考 GitLab			

ProjectManagement/工作交付物/测试开发/python注释规范)			
方法中的变量名是否清晰易懂			
方法是否冗余（主要关注超过 100 行的方法）			
是否对常量命名（否则可能其他人不知道该常量的意义）			
安全性部分			
源代码中是否出现了数据库密码等敏感内容			
是否在连接后关闭了数据库连接			
是否有错误处理机制（如 except, finally 等）			
是否能够生成日志用于记录程序状态和错误排查			
运算、数据部分			
是否会出现 int, float 类型转换导致的误差			
运算中是否会出现除数为 0 的风险			
数组的下标值是否有超出界限的风险			
数组的下标值是否为整数			

控制流部分			
是否会出现死循环（无法停止的 <b>while/for</b> ）			
是否会出现无法进入的循环入口			
是否有 <b>if</b> 条件某一分支无法进入的情况			
输入/输出部分			
是否所有打开的文件在使用后都被关闭了			
如果程序涉及文件读取，程序能否处理“ <b>File Not Found</b> ”的问题？			
<b>Print</b> 出或存入 <b>log</b> 等其它文件的内容是否存在拼写/语法错误			

检查代码后，请测试人员对被测模块进行单元测试：

请填写相关测试用例（如用例过多无法填写在此表内，请单独上传测试用例描述文件，任意格式均可）：

用例目的及所用参数描述(传入函数的参数，或是修改了函数中什么变量的值)	用例状态（已执行：P/未执行：U）	预期结果（该用例下我们所期望的运行结果）	实际结果（方法/模块实际运行的结果，若方法运行时间较慢，请在此处注明）
例如：测试 <code>crssearch</code> 函数的功能，输入参数为： <code>Sub = 'ERG', Code='2050', sec=1, time=2019-02-02</code>	P	0.04%	0.04%

[illegible]

如进行了其它测试，请在此处说明：

