

## 一、任务目标：

我们测试游戏时，良好的思维能力，丰富的游戏经验、善于主动发现和分享BUG的习惯，仔细认真以及对自我的高要求等，都能帮助我们有效做好测试工作，如果能进一步了解其实现原理，则能考虑到更多的测试点，测试更加全面。游戏是由代码实现的，要了解其实现原理，则需要具备一定的代码基础，对于初级测试工程师而言，如果能够看懂大部分代码，则可以大幅提升测试深度与测试效率，如果能够自己动手编写一些小脚本，则可进一步快速解决产品中遇到的问题。以下是一些python练习题，希望大家能够看懂其中的代码，并参照看懂的代码进一步完善。编码工具推荐vscode(Visual Studio Code)。

## 二、题目列表

### Q1:

阅读calculator.py，理解其中的python代码及其需要实现的功能

仿照calculator.py，给 base\_exercise.py 中的代码加上注释

最终提交文件命名为： base\_exercise\_q1.py

### Q2:

在base\_exercise.py的基础上，往下增加代码，完成以下功能：

1. name\_list增加了两位同学，Lily和Ruby，对应的score分别是80和90，请分别加到name\_list 和 score\_list 中，并更新score\_dict
2. Kitty离开了，请把Kitty及对应的score从 name\_list、score\_list 及 score\_dict 中删除
3. 找出最高的score及对应的同学

最终提交文件命名为： base\_exercise\_q2.py

### Q3:

仿照calculator.py，给 count\_words.py 中的代码加上注释

最终提交文件命名为： count\_words\_q3.py

#### Q4:

在count\_words.py的基础上，完成以下功能：

1. 新增函数，统计出现次数最少的单词，以及对应的出现次数，并统计这类单词有多少个（例如，如果统计结果是 learn 和 use，则是2个），返回结果
2. 新增函数，统计 python 这个单词的出现次数（要求 python 这个词是通过参数形式传入），返回结果
3. 新增函数，统计出现次数最多的字母，返回结果
4. 新增函数，要求统计读入文本的行数，返回结果
5. 修改函数 get\_localtime，要求返回的时间除了时分秒以外，还需要有年月日
6. 把以上功能得到的结果，追加写入到输出文件 test\_word\_result.txt 中，要求输出后的信息有较好的可读性

最终提交文件命名为： count\_words\_q4.py