Code worker 2.0

Version 2.0 update date: 2018.7.9

作者：[felix.zhang@alcatel-lucent.com](mailto:felix.zhang@alcatel-lucent.com)

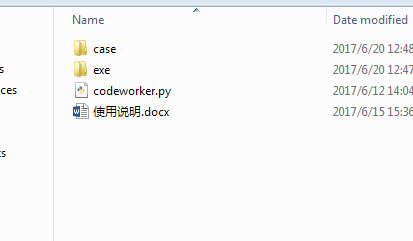
1.0作者[winter.wang@alcatel-lucent.com](mailto:winter.wang@alcatel-lucent.com)

概述

该python脚本用于批量执行源代码/可执行文件，根据多个测试用例给出运行结果报告。

使用方法

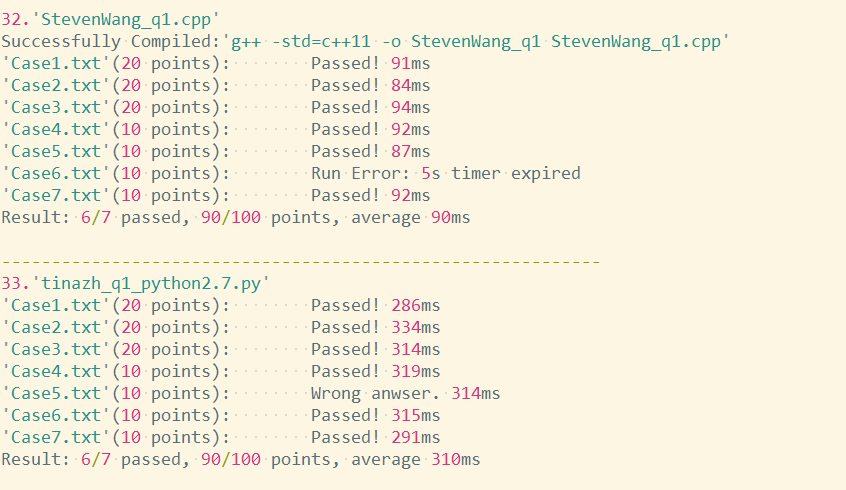
1. 测试用例放到case文件夹下，按照样例的格式编写测试case
2. 或者把测试用例写到‘testcases.xlsx’中。测试用例以testcases.xlsx中的为准。
3. 源程序放到exe文件夹下，支持C，C++，Python2，Python3，Java, bash和C/C++的exe文件。(如果提交答案用的是Microsoft的VC写的代码，gcc是无法编译运行的，所以直接把.exe文件放到exe文件夹下)



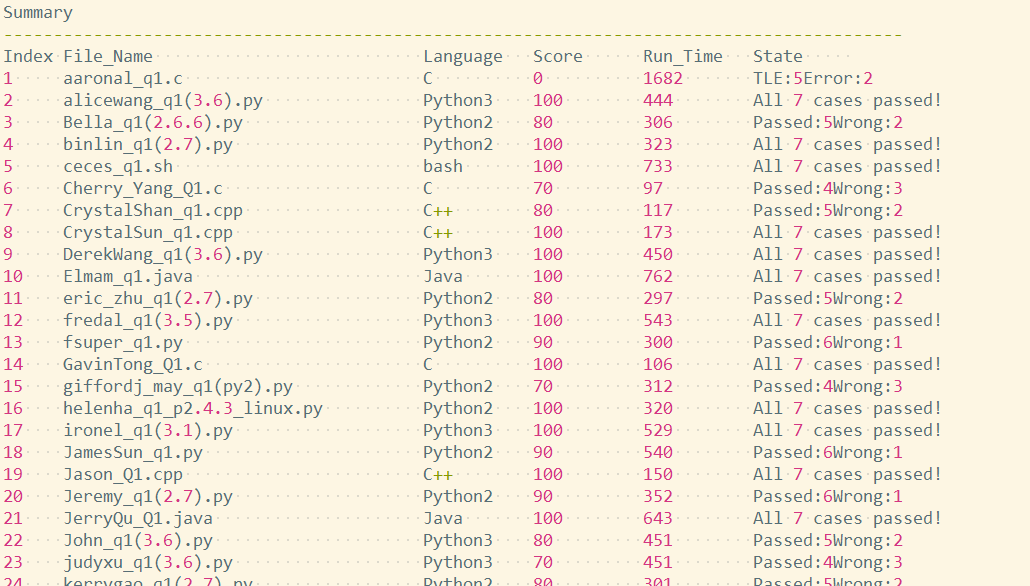
1. 用python2 运行程序codeworker.py，控制台将打印运行报告。
2. exe文件夹下将会生成‘result.xlsx’汇总的报告文件。

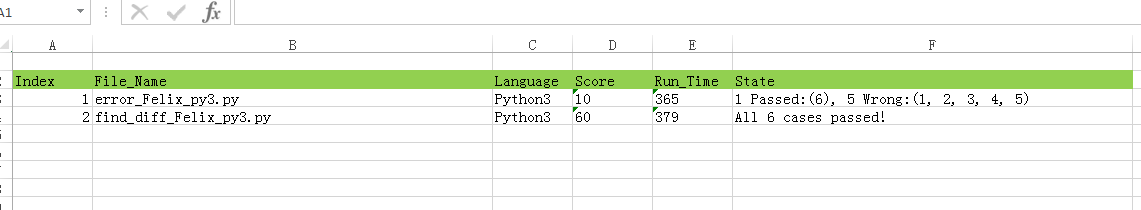
功能演示:

1. 每个程序执行结果:



1. 汇总结果:





2.0版本新功能:

1. 脚本文件夹可以放到任意位置执行

2. 输出每个提交答案每个测试用例的打分结果

3. 输出更多的信息: language, Run Time和详细的结果State

4. **如果提交答案运行时间过长或者死循环造成批改脚本无响应，添加5秒timer，超时后强制终止这个提交答案的运行，并且判错(Run Error: 5s timer expired)**

5. 输出一个简洁表格汇总最终的结果

6. 支持GNU .c和.cpp源文件的编译运行(使用gcc –o –std=C11/C++11编译)

7. 支持.java源文件的编译运行

8. 支持python2和python3的自动识别运行

9. 支持bash shell .sh文件的运行

10. 支持c/c++编译后的.exe文件的直接运行

11. 支持生成xlsx文件report

12. 支持测试用列从xlsx中读取

To Do Feature:

1. 显示代码行数

使用前配置

135.242.80.37已经配置完毕

**支持python2, python3, Java, C, C++和bash:**

1. **PYTHON:** Python2,3自动识别：安装python3 然后再装python2，这样python2将作为默认的使用版本。反之亦然。

更改python 3的python.exe 为 python3.exe

更改python 2的python.exe 为 python2.exe

这样就可以python2 xxx.py 执行python2程序，python3 xxx.py执行python3程序了

cmd中测试这两个命令。

有特殊的包，直接到python官网搜索，安装相应版本的.exe安装包即可. 如果是.whl文件用pip2 install xx.whl 或者pip3 install xx.whl安装对应版本。例如2017.5Q1,Q2有的答案用到了numpy模块。

1. **JAVA:** 安装最新的Java的JDK，然后命令行测试javac和java命令, 如果不能用，更新系统环境变量:

1) 创建CLASSPATH

值: .;%JAVA\_HOME%lib\dt.jar;%JAVA\_HOME%\lib\tools.jar

2)创建JAVA\_HOME

值: C:\Program Files\Java\jdk1.8.0\_131

3) Path 中加入: C:\Program Files\Java\jdk1.8.0\_131\bin;

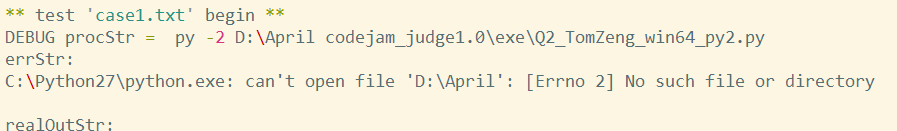
再在命令行测试javac和java命令，可以调用就OK.

1. **C/C++:** 安装MinGW(可以下载安装CodeBlocks编译器，其带有MinGW). 命令行测试g++和gcc命令，可以调用就OK. 注意只支持GNU gcc/g++编译。不支持微软的visualstdio编译，需要直接运行.exe文件.
2. **.sh文件:** 安装Cygwin，然后环境变量添加C:\cygwin64\bin. 在调用.sh文件之前，如果有换行的错误，先用vi 打开，然后:set fileformat=unix 保存退出(防止/r错误)。命令行直接就可以bash xx.sh了。

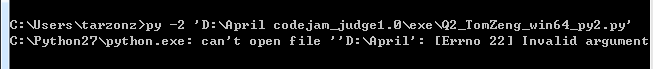
更新记录:

**Bug:**

1. subprocess 在执行命令的时候，如果文件路径中有空格，那么会认为空格之前的就是要处理的对象，空格之后的不会处理，这样就造成了识别文件错误:



直接在cmd中做测试:



解决办法是把工作目录带到这个目录下，那么就只输入文件名行了！

os.chdir(path) #用于改变当前工作目录到指定的路径

更优的解决办法是command用list的形式，这样就支持带空格的名字了。

1. 结果输出有回车换行，根据平台不同windows是’\n\r’, linux 是’\n’

解决办法使用os.linesep

os.linesep字符串给出当前平台使用的行终止符。例如，Windows使用'\r\n'，Linux使用'\n'而Mac使用'\r

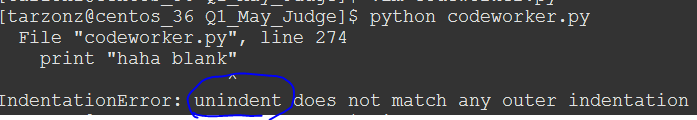
1. 解决java 无法运行.class文件。使用java –cp . xxx。 需要加-cp . 参数
2. 解决编译C/C++的错误，不使用-O2参数，如果使用命令为，g++ -O2 –o xx xx.cpp

正确的编译命令为 gcc –o xxx xxx.c 和 g++ -o xxx xxx.cpp

1. C文件编译失败，提示error:for循环中不能有初始化，可以用c99 c11编译，解决办法是gcc命令加参数-std=c11而且要写在-o前面。C++的也要改成-std=c++11
2. Subprocess 调用中命令’py -2 xx.py’ 可以shell=True也可以False。但是’python xxx.py’必须使用shell=True
3. 通过py -2 和 py -3 运行的python程序不能被tskill终止，原因是这两个进程会继续派生python.exe进程，所以还是要用python.exe 或者改名的python3.exe来执行python程序.
4. 有的c程序报warning, 将其忽略.
5. Subprocess输入参数建议用sequence，也就是列表形式，不要用字符串，容易有bug。用sequence的形式每一个参数都要作为列表的一个元素。例如’java –cp . xxx’这个命令写成sequence形式就是[‘java’,’-cp’,’.’,’xxx’]
6. c/c++build完成后，跑第一个case总是用时很多，解决办法是空跑一下第一个case然后正式开始跑所有的case
7. Windows 栈大小限制，遇到一个c程序，在栈申请了4MB内存，然后运行就挂掉了

Int f[100][100][100] 解决办法，放到静态全局区，就好了

1. 程序在windows下编辑完后，拿到linux下用vim编辑，每次o换行就提示不对齐错误:



解决办法：主要原因是空格和tab的混淆。Linux下vim设定了4空格代替tab，所以每次o了会有四个空格，但是windows下用的sublime默认是tab。在vim中用:set list可以看到^I就是tab。所以用撤销vimrc中的的set expandtab就可以了