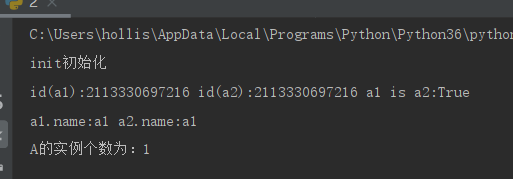
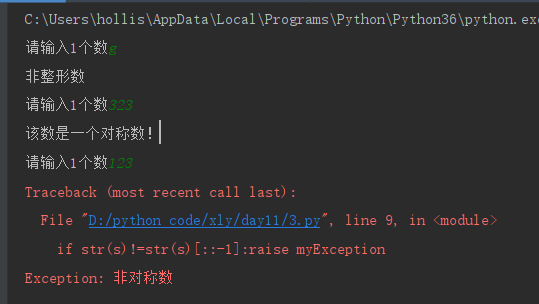
1. 实现单例模式

class A:  
 instance=None  
 first\_init=False  
  
 count=0 #显示实例数目  
 def \_\_new\_\_(cls, \*args, \*\*kwargs):  
 if not cls.instance:  
 cls.instance=super().\_\_new\_\_(cls)  
 cls.count+=1  
 return cls.instance  
  
 def \_\_init\_\_(self,name):  
 if not self.first\_init: #self可以拿到类属性  
 print('init初始化')  
 self.name=name  
 self.first\_init=True #实例属性覆盖类属性  
  
  
a1=A('a1') #第一次实例对象  
a2=A('a2') #init函数不会再次初始化，所传参数实际不起作用  
print('id(a1):{} id(a2):{} a1 is a2:{}'.format(id(a1),id(a2),a1 is a2))  
print('a1.name:{} a2.name:{}'.format(a1.name,a2.name))  
print('A的实例个数为：{}'.format(A.count))



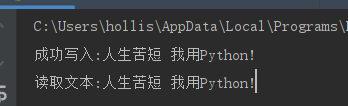
1. 通过try进行异常捕捉，确保输入的内容一定是一个整型数，然后判断该整型数是否是对称数，12321就是对称数，123321也是对称数，否则就不是

myException=Exception("非对称数")  
  
while True:  
 try:  
 s = int(input('请输入1个数'))  
 except ValueError :  
 print("非整形数")  
 else:  
 if str(s)!=str(s)[::-1]:raise myException  
 else:print('该数是一个对称数！')



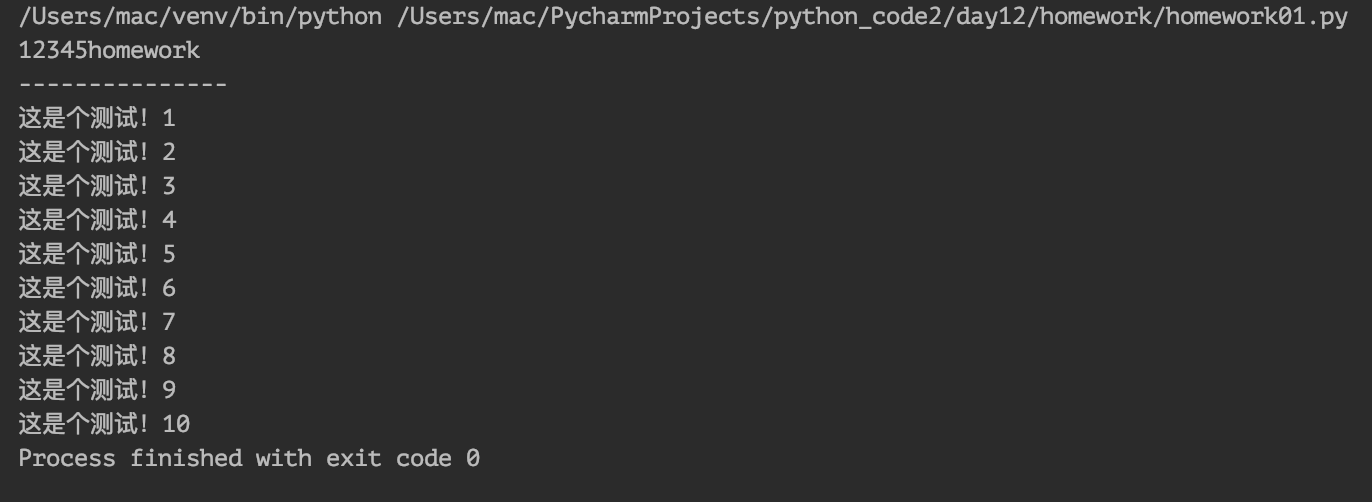
1. 以可读可写打开一个文件，然后写入“人生苦短 我用Python”，关闭文件

import os.path as p  
  
  
s='人生苦短 我用Python!'  
fileName='exercise5'  
if not p.exists(fileName):open(fileName,'w')  
  
with open(fileName,'r+',encoding='utf-8') as f:  
 f.write(s)  
 print('成功写入:{}'.format(s))  
  
with open(fileName,'r+',encoding='utf-8') as f:  
 print('读取文本:{}'.format(f.read()))



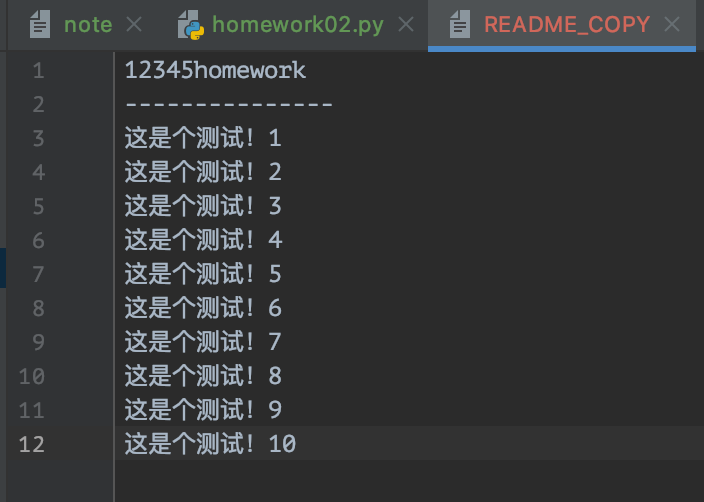
1. 通过readline依次读取文件每一行并打印

*#! usr/bin/bash  
# -\*- coding:utf-8 -\*-  
  
"""  
File Name: homework01.py  
Author: mok  
Creation Date: 20200304  
Description: 通过readline依次读取文件每一行并打印  
"""***def** main():  
 file = open(**"README"**, **'r'**, encoding=**"utf-8"**)  
 **while True**:  
 text = file.readline()  
 **if not** text:  
 **break** print(text, end=**""**)  
 file.close()  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()



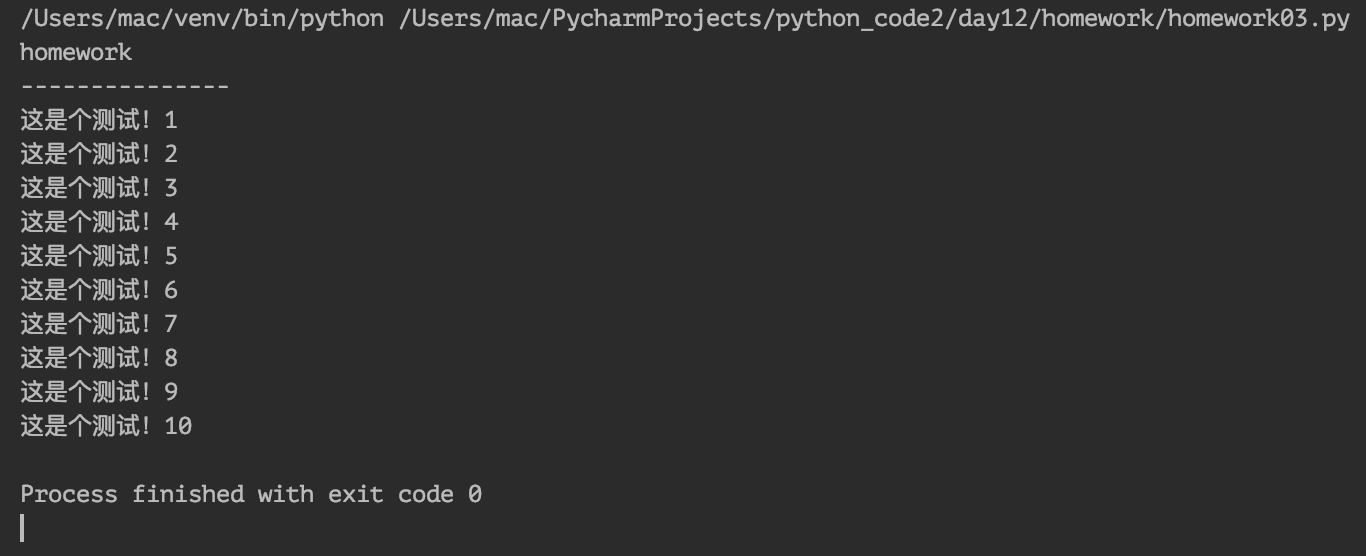
1. 读取readme文件内容，复制到readme1中

*#! usr/bin/bash  
# -\*- coding:utf-8 -\*-  
  
"""  
File Name: homework02.py  
Author: mok  
Creation Date: 20200304  
Description: 读取readme文件内容，复制到readme1中  
"""***def** main():  
 file\_read = open(**"README"**, **"r"**, encoding=**"utf-8"**)  
 file\_write = open(**"README\_COPY"**, **"w"**, encoding=**"utf-8"**)  
  
 **while True**:  
 text = file\_read.readline()  
 **if not** text:  
 **break** file\_write.write(text)  
  
 file\_read.close()  
 file\_write.close()  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()



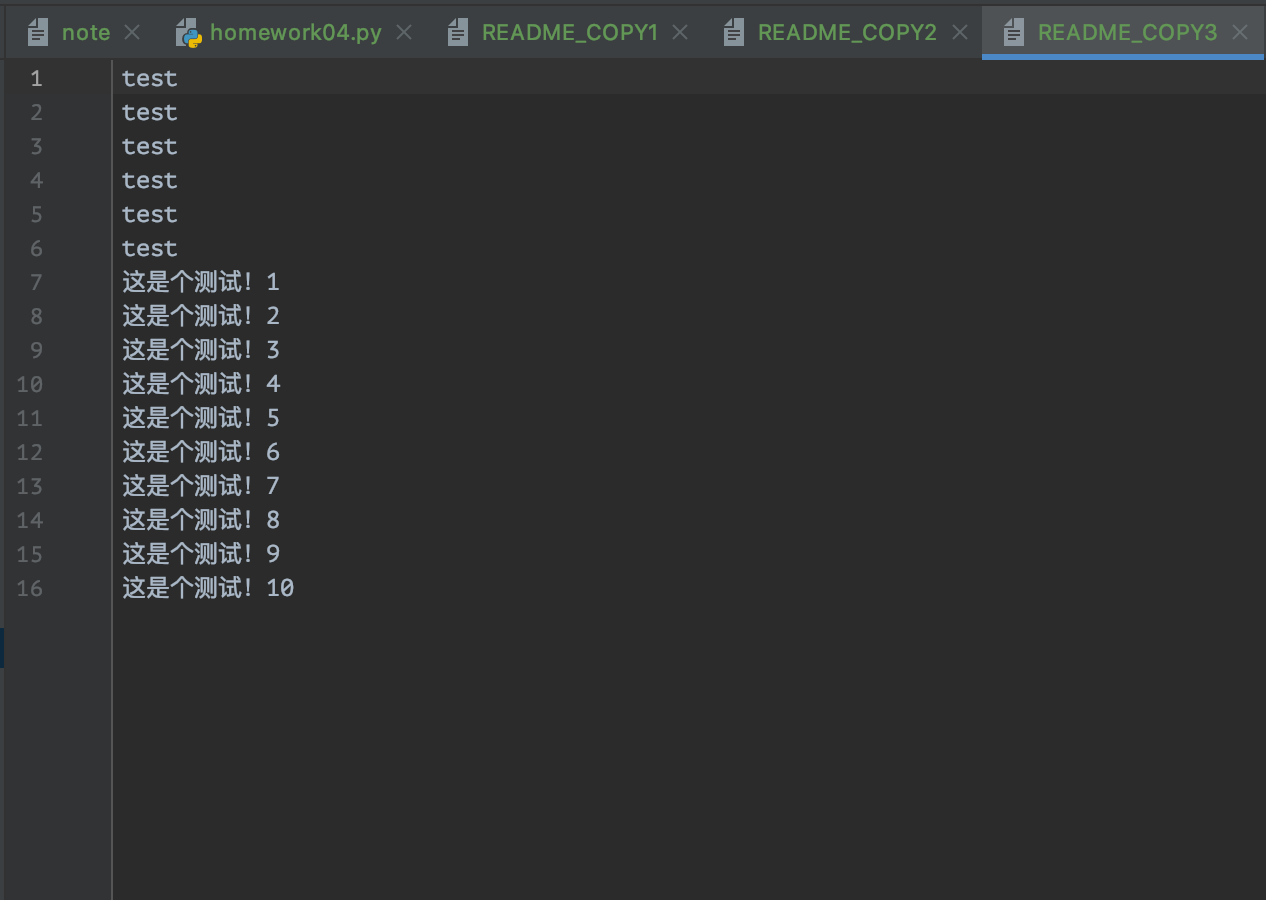
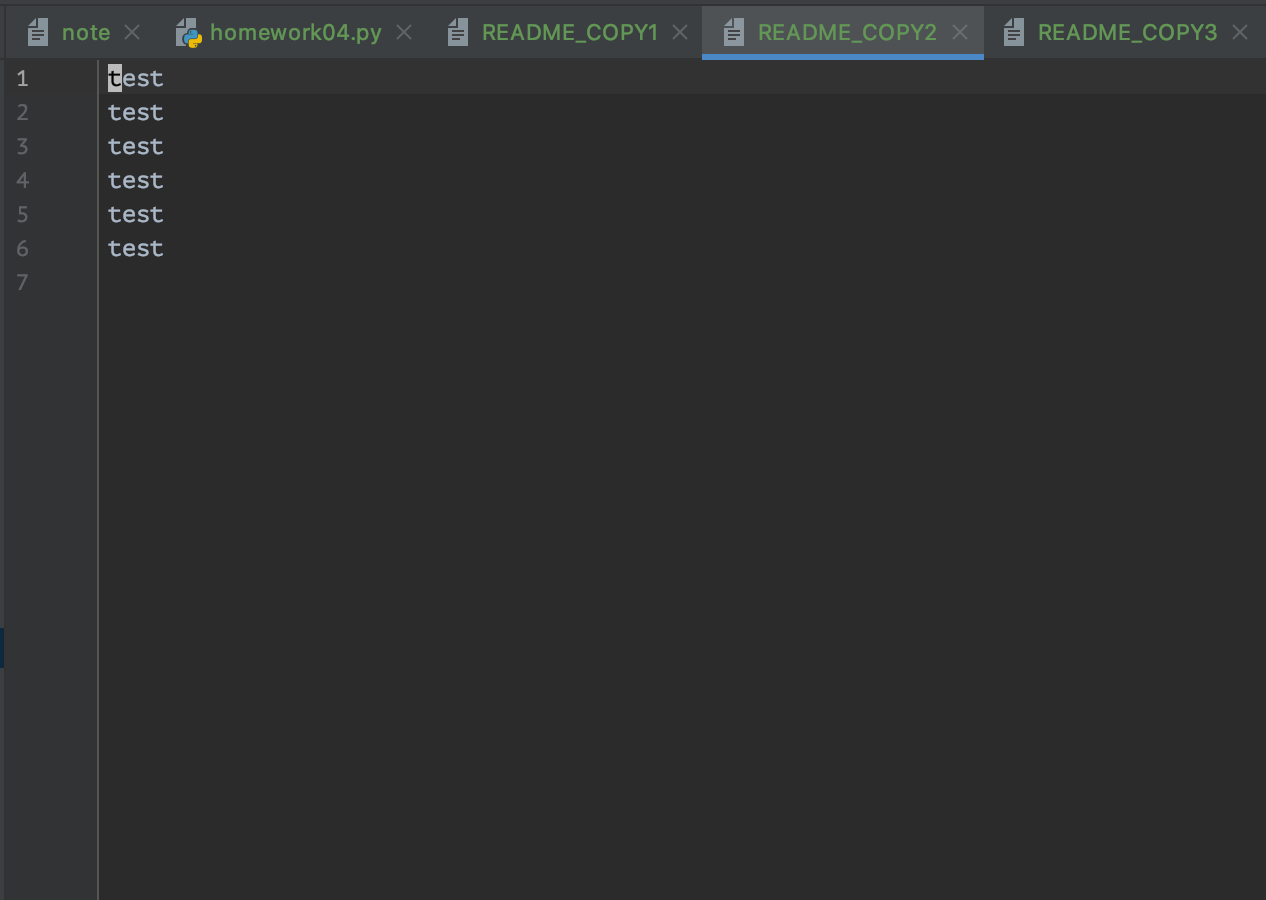
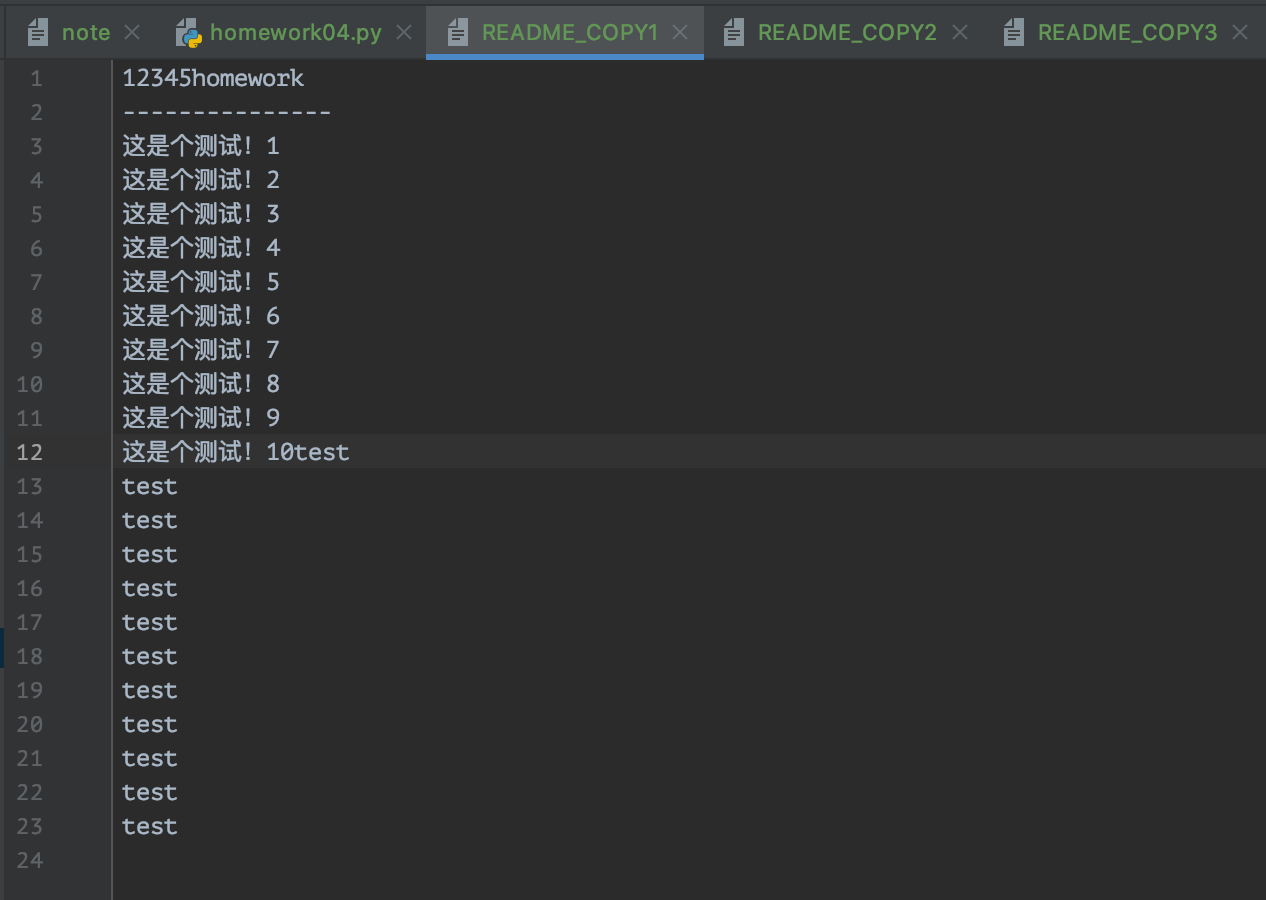
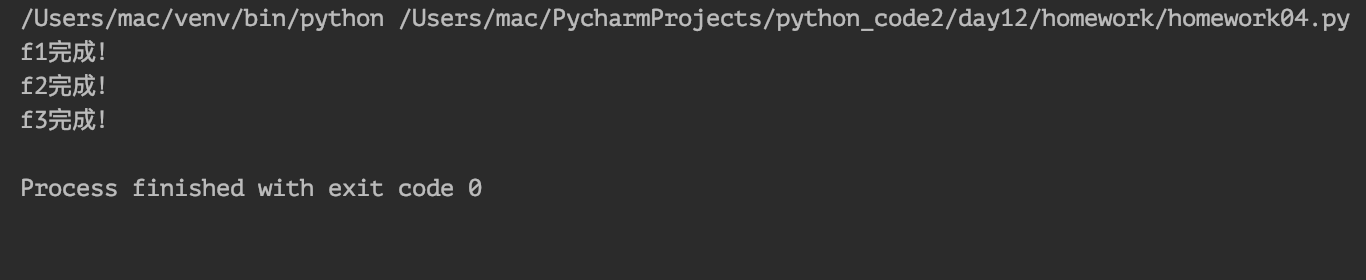
1. 通过seek从文件开头偏移5个字节，然后把文件剩余内容读取

*#! usr/bin/bash  
# -\*- coding:utf-8 -\*-  
  
"""  
File Name: homework03.py  
Author: mok  
Creation Date: 20200304  
Description: 通过seek从文件开头偏移5个字节，然后把文件剩余内容读取  
"""***def** main():  
 file = open(**"README"**, **"r"**, encoding=**"utf-8"**)  
 file.seek(5)  
 text = file.read()  
 print(text)  
 file.close()  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()



1. 练习open的a+，w+模式，感受他们与r+的区别

*#! usr/bin/bash  
# -\*- coding:utf-8 -\*-  
  
"""  
File Name: homework04.py  
Author: mok  
Creation Date: 20200304  
Description: 练习open的a+，w+模式，感受他们与r+的区别  
"""***def** main():  
 text = **"test\n"** \* 6  
  
 f1 = open(**"README\_COPY1"**, **"a+"**, encoding=**"utf-8"**)  
 f1.write(text)  
 print(**"f1完成！"**)  
 f1.close()  
  
 f2 = open(**"README\_COPY2"**, **"w+"**, encoding=**"utf-8"**)  
 f2.write(text)  
 print(**"f2完成！"**)  
 f2.close()  
  
 f3 = open(**"README\_COPY3"**, **"r+"**, encoding=**"utf-8"**)  
 f3.write(text)  
 print(**"f3完成！"**)  
 f3.close()  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()



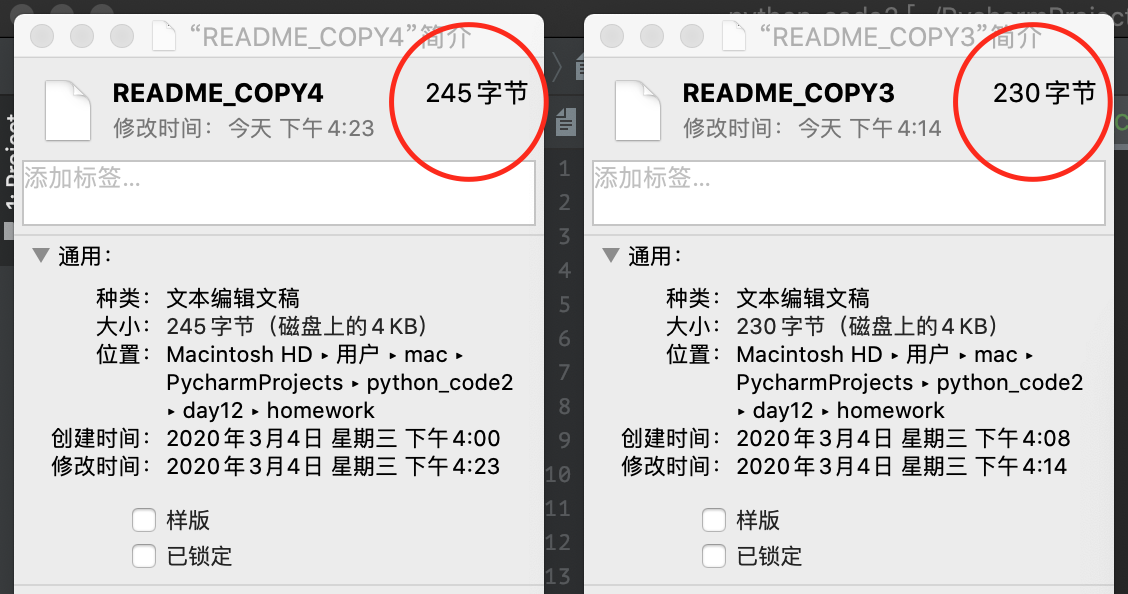
a+：在文件末尾指针处追加内容

w+：原文件内容删除，添加新内容

r+：在文件头部追加内容，覆盖原文件内容

1. 练习rb+模式，感受与r+模式的区别

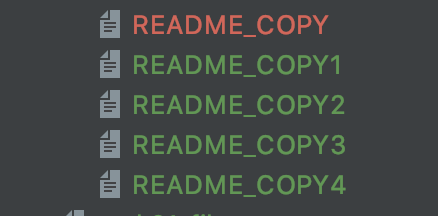
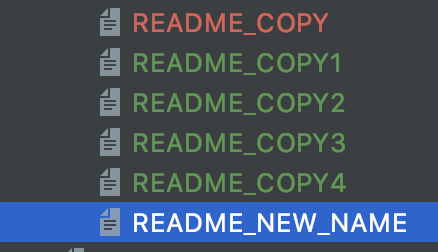
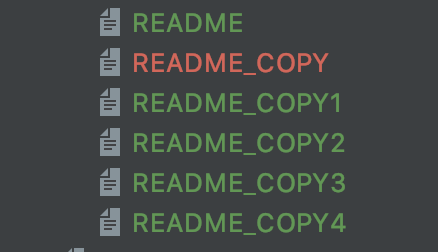
*#! usr/bin/bash  
# -\*- coding:utf-8 -\*-  
  
"""  
File Name: homework05.py  
Author: mok  
Creation Date: 20200304  
Description: 练习rb+模式，感受与r+模式的区别  
"""***def** main():  
 text = **b"test\n"** \* 6  
 f4 = open(**"README\_COPY4"**, **"rb+"**)  
 f4.write(text)  
 print(**"f4完成！"**)  
 f4.close()  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()



文件内容一样，存储换行不一样。r+将\n存储为\r\n，rb+存储为\n，故CRLF与LF文件的大小会不一样

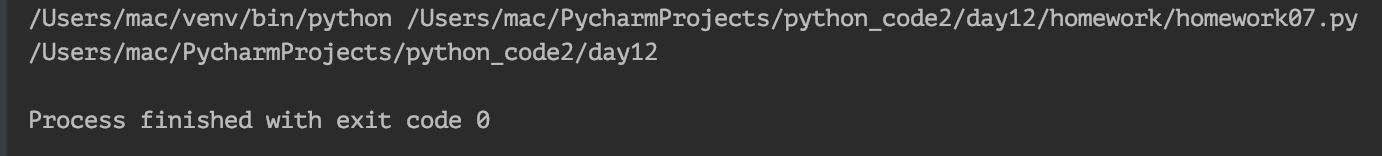
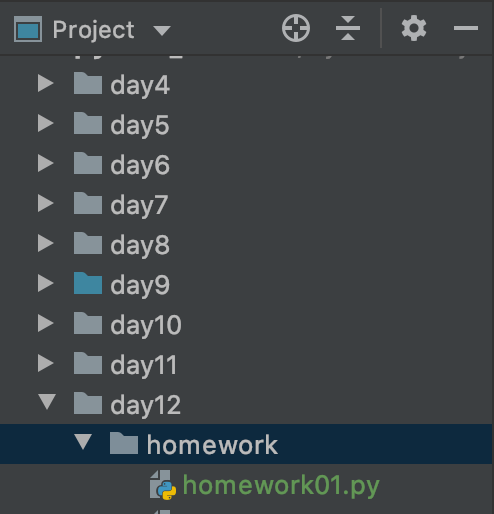
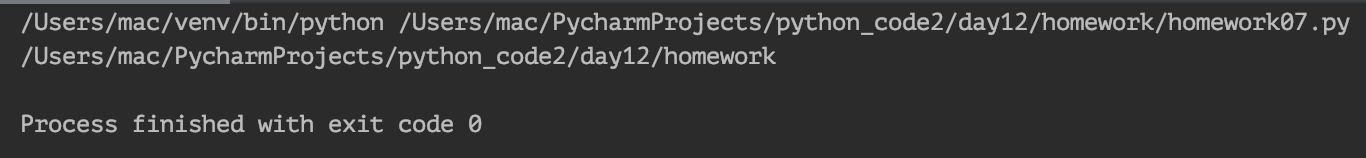
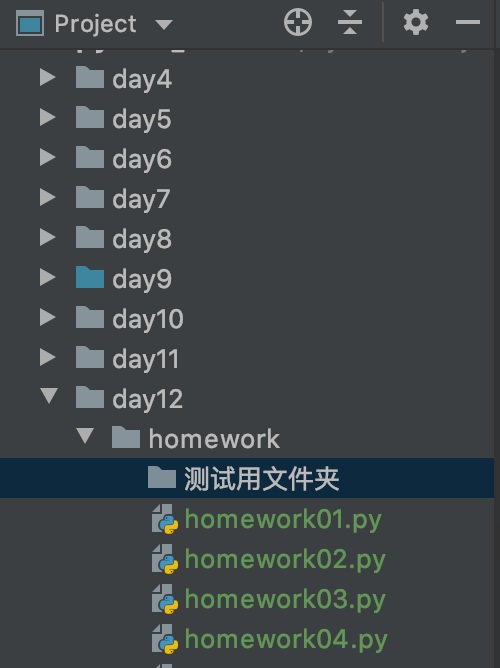
1. 对文件进行改名并删除文件

*#! usr/bin/bash  
# -\*- coding:utf-8 -\*-  
  
"""  
File Name: homework06.py  
Author: mok  
Creation Date: 20200304  
Description: 对文件进行改名并删除文件  
"""***import** os  
  
  
**def** main():  
 *# os.rename("README", "README\_NEW\_NAME")* os.remove(**"./README\_NEW\_NAME"**)  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()



1. 创建文件夹，删除文件夹（均使用相对路径），改变程序执行路径，获取程序当前路径

*#! usr/bin/bash  
# -\*- coding:utf-8 -\*-  
  
"""  
File Name: homework07.py  
Author: mok  
Creation Date: 20200304  
Description: 创建文件夹，删除文件夹（均使用相对路径），改变程序执行路径，获取程序当前路径  
"""***import** os  
  
  
**def** main():  
 *# os.mkdir("./测试用文件夹")  
 # print(os.getcwd())  
 # os.rmdir("./测试用文件夹")* os.chdir(**"../"**)  
 print(os.getcwd())  
  
  
**if** \_\_name\_\_ == **'\_\_main\_\_'**:  
 main()



难度作业

1. 有时间的同学预习明天的内容，可以提前自行写一下深度优先遍历

from os import listdir  
import os.path as p  
  
  
def find\_all\_file(path):  
 if not p.isdir(path):  
 print(path)  
 return  
  
 for x in listdir(path):  
 x=p.join(path,x)  
 find\_all\_file(x)  
  
  
path=p.abspath(input('输入路径'))  
find\_all\_file(path)