**思路：**

1. 程序要实现顾客购买商品和管理员管理商品的两大功能。
2. 顾客购买商品需要实现查找商品，购买商品，打印购物清单，调整库存的功能。
3. 管理员管理商品需要实现登录，上架、下架商品，调整库存的功能。
4. 首先处理文档的数据
5. 第一次顾客输入查询种类 – 输出该种类下的所有商品名称和对应数量，可多次查询。
6. 第二次顾客输入{商品名称: 购买数量} – 输出购物清单
7. 管理员登录：先匹配用户名，再匹配密码，3次机会。
8. 提醒进货：读取文件后查找是否有库存已经为0的商品，如果有就提醒管理员应该进货了。
9. 上架/下架商品：按照每个种类来上架/下架，一次输入一个种类、多个商品。可以多次操作。
10. 调整库存：根据顾客或管理员的操作调整数据文件中的数据。

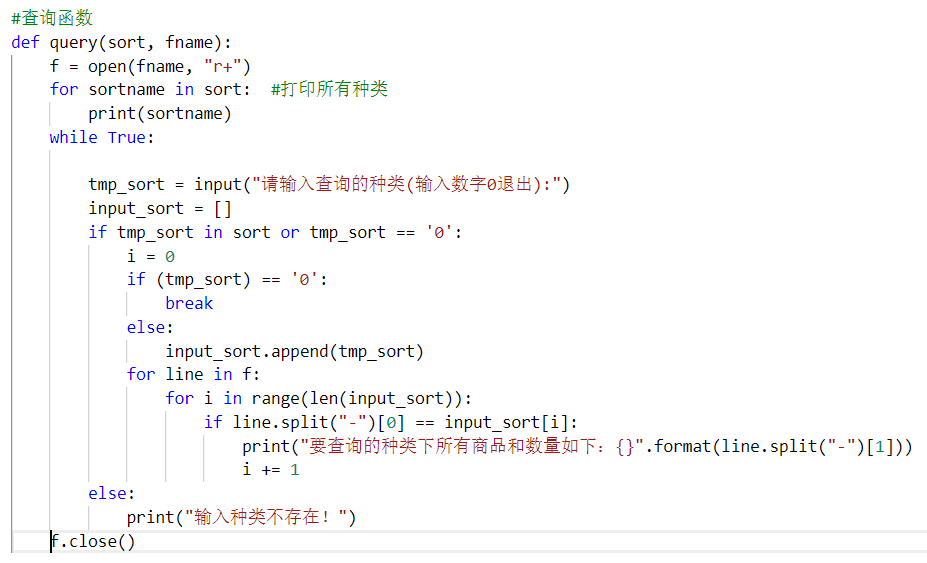
**程序结构：**

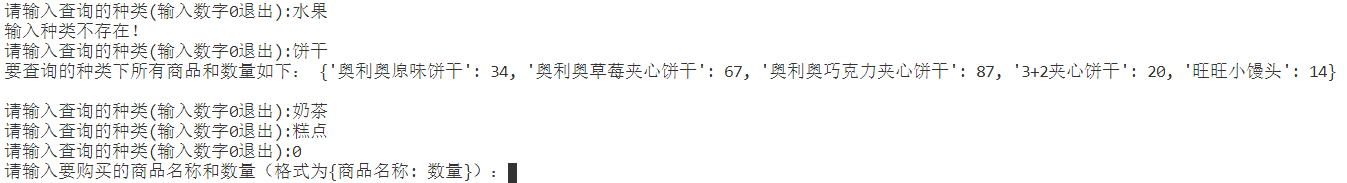
**fname = “data.txt”**

**finame = “details.txt”**

1. 处理details.txt文件 – 得到goodsnm，code，prices，date
2. 主函数 – main ( ) ：可以选择是顾客函数还是管理员函数，如果是管理员，判断是否登录成功
3. 得到种类名称函数 - getSort ( fname ) ：获得超市所有的商品种类生成种类表，返回sort
4. 查询函数 – query(sort, fname, goodsnm, code, prices, date) ：用户根据种类表输入要查询的种类，同时输出商品详细信息。一直循环至顾客输入数字0表示不再查询。如果输入的种类在文件中查找不到，输出"输入种类不存在！"
5. 输入购买清单 – inputBuyList ( )：顾客需要按照字典格式输入要购买的商品及数量，返回input\_buylist，如果格式不正确会输出"输入格式错误！"
6. 输出购物清单函数 – buyList(input\_buylist, prices, goodsnm, code)：输出顾客购买的商品数量、单价、总价
7. 调整库存函数 – inventory(input\_buylist, fname)：根据顾客的操作调整库存
8. 顾客函数 – customer(fname, finame)：用来查询商品种类，购买商品，打印购物小票
9. 上架商品函数 - add ( fname ) ：用于上架商品，循环至管理员输入数字0表示不再上架商品
10. 下架商品函数 – reduce ( fname ) ：用于下架商品循环至管理员输入数字0表示不再下架商品
11. 提醒函数 – notice(fname)：显示库存为0的商品，并提醒管理员上架
12. 管理员函数 – manager ( ) ：用来登录获取权限，获得提醒上架信息，管理商品，上架和下架商品

**调试过程：**

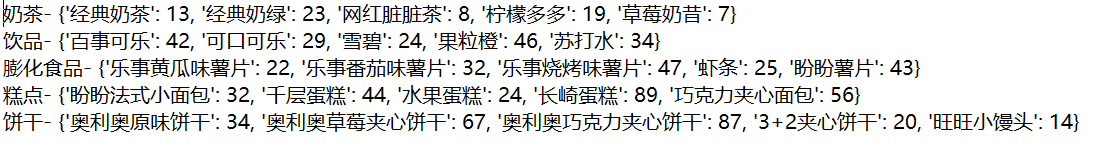


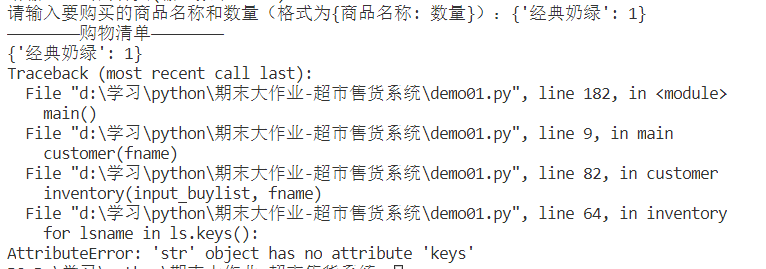
****

1. 与预期执行结果有偏差，输入之后的两个查询种类后并没有输出种类的信息。

错误原因：文件的打开与关闭应放在while True的循环里，否则之后输入的种类都只是在一个已经读完了的文件里查找，这样是找不到对应信息的。

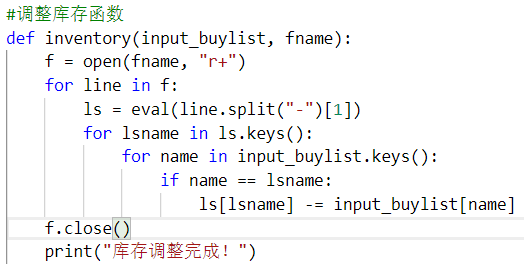


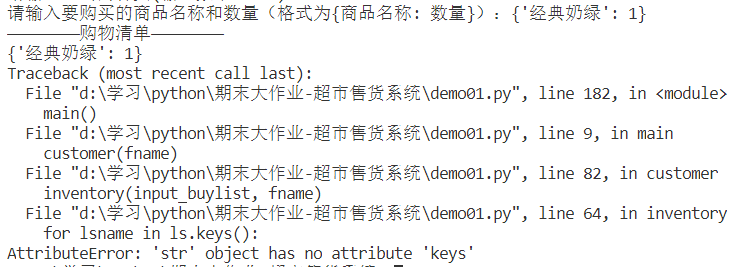
****

****

1. 调整库存函数出错，在顾客买完之后无法调整库存

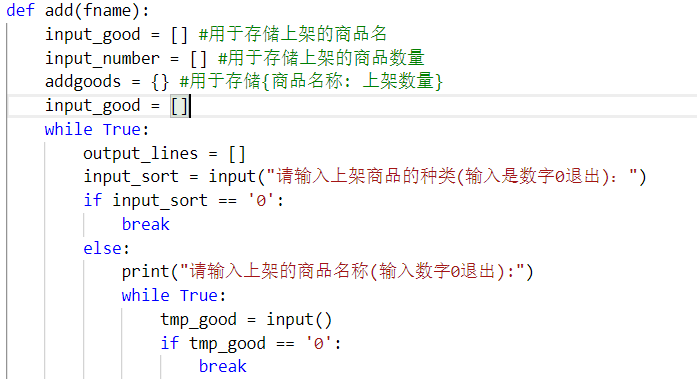
错误原因：ls = line.split("-")[1]，这个ls是字符串，而不能直接用作字典。应为ls = eval(line.split("-")[1])。

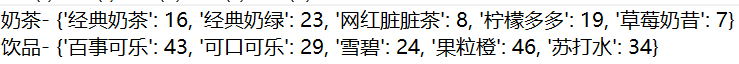
****

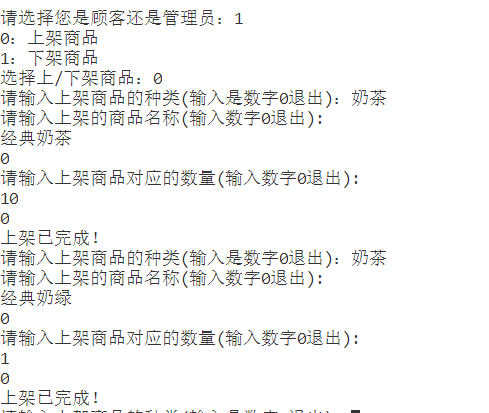
****

1. 调整库存函数无法使用

错误原因：错误地认为可以直接在文件中修改数据，实际上从文件中读取的数据都储存在内存中，修改的只是内存中的数据，如果要修改文件中的数据应该把内存中修改后的数据重新写入文件。

****

****

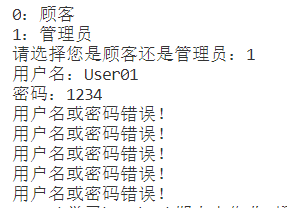
****

****

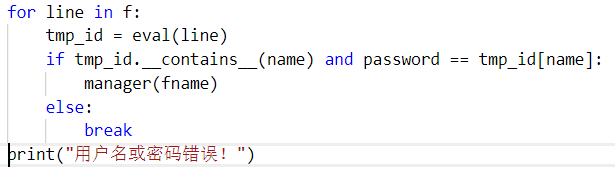
1. 应该经典奶茶+10，经典奶绿+1，但是现在两者都+10了

错误原因：input\_good = [ ]放在了循环外，导致每一次输入都没有更新，应该把input\_good = [ ]放在循环内，每一次输入更新，防止调整库存时把上一次的商家啊商品再次上架。

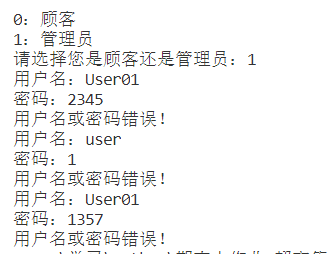
****

****

1. 输入密码错误时，反复输出“用户名或密码错误”，应该只输出一遍

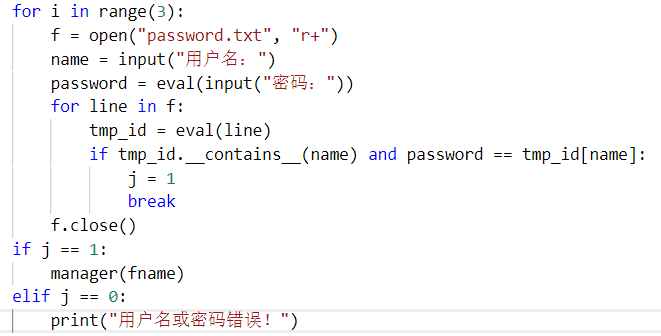
错误原因：把print(“用户名或密码错误！”)放在了for line in f这个循环中，应该是

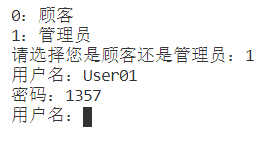
****

****

1. 当最后一次输入正确时仍然报错

错误原因：文件在第一次时已经被读完而没有从头开始，应该把文件的开关放在循环中

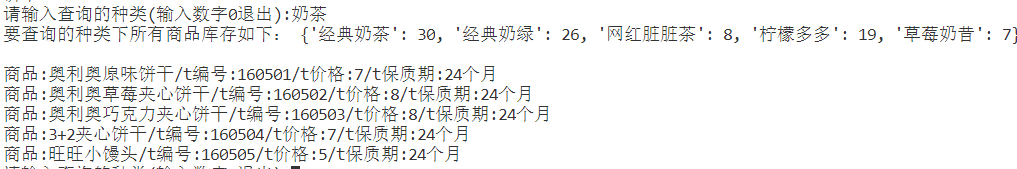
****

****

1. 输入正确时也没有退出循环

错误原因：把判断是否正确的条件即j的值的判断放在了3次循环外，应该放在循环里，每输入一次就判断一次。

****

****

1. 无论查询的时哪个种类，输出的都是最后一个种类的详细信息。

错误原因：k += 1的位置放错了循环，应为

1. 在新增商品详细信息时，原来的函数总是因为缺少参数而报错，原本把处理商品详细信息的代码放在了自定义函数里，但后来发现因为处理之后需要以原本形式返回多个列表，所以封装在函数里面的方法是行不通的，只好把处理商品详细信息的代码放在最开始，把处理好的存储信息的列表作为全局变量使用，解决报错的问题。
2. 在优化各个函数，使其可以循环操作直到用户输入数字0表示退出的时候，一些数据的操作需要注意：是放在循环内还是循环外；文件的操作也需要关注是否在操作前已经读取完了，或是在再次读取前已经关闭。
3. 在判断管理员是否登录成功时，应注意在登录成功后使用break退出循环，否则在操作后还会显示登陆界面。
4. 在写下架函数时，逻辑出现问题，程序无法正常运行，后调整读取数据和判断是否进行操作的顺序。

**心得体会：**

1. 学会使用VS CODE的调试功能，使程序的调试更加方便、清楚。
2. 在读取数据或者处理数据时选择恰当的在内存中的处理方式。
3. 写各个函数时注意各函数之间的联系，是否返回数据、是否需要参数传入等。
4. 注意循环和数据初始化、数据处理、判断条件之间的关系。
5. 读取文件时注意，在一次读取完之后要重新读取的话，只能先关闭第一次打开的文件，再重新打开，重新读取。
6. 设计要人性化，给予用户足够多的提示，输出的格式也会相应的调整。
7. 注意数据输入的容错处理。