

```
# 1.统计文件数据中字母e出现的次数(不区分大小写)
# 文件内容: hello friend, can you speak English!
# 结果: 4
# 分析: 将文件内容读出, 然后统计读出的字符串中字符e的个数(字符串count功能)

# 2.统计文件数据中出现的的所有字符与该字符出现的个数(不区分大小写,标点与空格也算)
# 文件内容: hello friend, can you speak English!
# 结果:
{
    'h': 1,
    'e': 4,
    'l': 3,
    'o': 2,
    ' ': 5,
    ...
}
# 分析: 将文件内容读出, 然后统计读出的字符串中每个字符的个数, 形成字段(for遍历读取的字符串)

# 3.读取文件内容, 分析出所有的账号及对应的密码
# 文件内容: owen:123456|egon:123qwe|liuxx:000000
# 结果:
{
    'owen': '123456',
    'egon': '123qwe',
    'liuxx': '000000'
}
# 分析: 将文件内容读出, 然后按|拆分出 账号:密码 格式的子字符串, 再按:拆分成 账号及密码, 存放字典中

# 4.在题3的基础上, 账号密码已经被存储在文件中, 完成用户登录成功或失败(只做一次性判断)
# 文件内容: owen:123456|egon:123qwe|liuxx:000000
# 需求: 输入账号、密码, 然后进行登录判断, 账号密码均正确打印登录成功, 否则打印登录失败
# 分析: 先完成题3, 分析出账号密码字典, 然后拿输入的账号密码与字典中数据进行校验

# 5.在题3的基础上, 完成用户注册的功能(只做一次性判断)
# 文件内容: owen:123456|egon:123qwe|liuxx:000000
# 需求: 输入注册的账号、密码, 账号已存在的打印账号已存在, 注册失败, 反正打印注册成功, 并将新账号密码录入文件
# 结果: 如果输入mac、123123 => owen:123456|egon:123qwe|liuxx:000000|mac:123123
# 分析: 先完成题3, 分析出账号密码字典, 然后拿输入的注册账号与字典中数据进行校验, 如果校验没有新账号
# -- 1.采用 w 模式写文件, 可以在读取文件的内容后拼接 |mac:123123 字符串, 将拼接后的总字符串一次性写入
# -- 2.采用 a 模式写文件, 可以直接追加写入 |mac:123123 字符串
```

```
# -----
# 拓展1.统计文件中大写字母、小写字母、数字及其他字符出现的次数
# 文件内容: Abc123,-+XYZopq000.~/
# 结果:
{
    '大写字母': 4,
    '小写字母': 5,
    '数字': 6,
    '其他字符': 6
}
# 分析: 利用ASCII表, for循环遍历每一个字符value, eg: 'a' <= value <= 'z'就代表是小写字母

# 拓展2.完成登录注册系统(从空文件开始做)
# 需求分析:
'''
1.可以循环登录注册, 输入1代表选择登录功能, 输入2代表注册功能, 输入0代表退出, 其他输入代表输入有误, 重输
2.用户的账号密码信息存放在usr.txt文件中, 保证用户注册成功后, 重启系统, 用户信息仍然保存
3.登录在账号验证通过才输入密码验证登录, 账号验证三次失败自动进入注册功能, 登录三次验证失败自动退出系统
4.第一次注册, 文件写入 账号:密码 信息, 再次注册追加写入 |账号:密码 信息

分析过程: 略
'''
```