# W1：

寻找第6个默尼森数

经典程序设计问题：找第n个默尼森数。P是素数且M也是素数，并且满足等式M=2\*\*P-1，则称M为默尼森数。例如，P=5，M=2\*\*P-1=31，5和31都是素数，因此31是默尼森数。

提交方式直接将答案（**M的值**）写在txt文件中通过网络提交。

# W3：

3.1.编程练习：用字典创建一个平台的用户信息（包含用户名和密码）管理系统，新用户可以用与现有系统帐号不冲突的用户名创建帐号，已存在的老用户则可以用用户名和密码登陆重返系统。建议基本框架为：

def newusers():

enter a name

if the name is used in the system:

enter again

else:

set the password

… …

def oldusers():

Enter the username and password

if password is right:

print(name, 'welcome back ')

else:

print('login incorrect')

… …

def login():

option = '''

(N)ew User Login

(O)ld User Login

(E)xit

'''

Enter the option

… …

if \_\_name\_\_ == '\_\_main\_\_':

login()

3.2.编程练习：创建DataFrame的编程小练习：已知有一个列表中存放了一组音乐数据：

music\_data = [("the rolling stones","Satisfaction"),("Beatles","Let It Be"),("Guns N' Roses","Don't Cry"),("Metallica","Nothing Else Matters")]

请根据这组数据创建一个如下的DataFrame：

singer song\_name

1 the rolling stones Satisfaction

2 Beatles Let It Be

3 Guns N' Roses Don't Cry

4 Metallica Nothing Else Matters

【参考代码】

# -\*- coding: utf-8 -\*-

"""

create DataFrame object

@author: Dazhuang

"""

import pandas as pd

music\_data = [("the rolling stones","Satisfaction"),("Beatles","Let It Be"),("Guns N' Roses","Don't Cry"),("Metallica","Nothing Else Matters")]

music\_table = pd.DataFrame(music\_data)

music\_table.index = range(1, 5)

music\_table.columns = ['singer', 'song\_name']

print(music\_table)

3.2.编程练习：利用字典进行词频统计：对于一个已分词的中文句子：

"我 是 一个 测试 句子，大家 赶快 来 统计 我 吧 ， 大家 赶快 来 统计 我 吧 ， 家 赶快 来 统计 我 吧 ， 重要 事情 说 三遍 ！"

可以用collections模块中的Counter()函数方便地统计词频，例如可用如下代码：

import collections

import copy

s = "我 是 一个 测试 句子 ， 大家 赶快 来 统计 我 吧 ， 大家 赶快 来 统计 我 吧，大家 赶快 来 统计 我 吧 ， 重要 事情 说 三遍 ！"

s\_list = s.split()

s\_list\_backup = copy.deepcopy(s\_list)

[s\_list.remove(item) for item in s\_list\_backup if item in '，。！”“']

print(collections.Counter(s\_list))

这个问题也可以通过字典来解决，请编写用字典解决本问题的程序。

【参考程序】

s\_dict = {}

s = "我 是 一个 测试 句子 ， 大家 赶快 来 统计 我 吧 ， 大家 赶快 来 统计 我 吧 ， 大家 赶快 来 统计 我 吧，重要 事情 说 三遍！"

s\_list=s.split()

for item in s\_list:

if item.strip() not in "，。！“”":

if item not in s\_dict:

s\_dict[item] = 1

else:

s\_dict[item] += 1

print(s\_dict)

进一步，可以很容易地统计文件中的词频。

# W4. 编程作业: 男女电影评分差异分析编程(计分)

计算MovieLens 100k数据集中男性女性用户评分的标准差并输出。

数据集下载http://files.grouplens.org/datasets/movielens/ml-100k.zip

其中u.data 表示100k条评分记录，每一列的数值含义是：

user id | item id | rating | timestamp

u.user表示用户的信息，每一列的数值含义是：

user id | age | gender | occupation | zip code

u.item文件表示电影的相关信息，每一列的数值含义是：

movie id | movie title | release date | video release date |IMDb URL | unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy |Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi |Thriller | War | Western |

可能会用到的相关函数：

pandas.read\_table(filepath\_or\_buffer, sep='\t', names=None)

pandas.pivot\_table(data, values=None, columns=None, aggfunc='mean')

pandas.merge(left, right, how='inner')

更详尽的API文档请参考http://pandas.pydata.org/pandas-docs/stable/。

输出结果：

Gender

F **\***

M **\***

结论：标准差高的评分差异大

**注意**：先分别计算每个人电影评分的平均分再按性别求标准差

请将**男女**评分（**保留2位小数**）组合后放到一个txt文件中通过网络提交，例如：如果两者评分的标准差分别是

**男：0.32**

**女：0.35**

**则提交：**

**3235**