实验报告六

姓名：白宇 学号：118010100327 专业班级：18无非3班

一、实验名称：组合数据类型

二、实验目的：了解列表、字典、集合三种组合数据类型

三、实验要求：

1、能运用集合的特性对数据去重。

2、掌握运用字典概念处理复杂数据信息，及字典的排序。

3、能运用列表构建数据结构。

4、综合运用组合数据类型进行文本词频统计。

四、实验内容：

1. 程序练习题6.1。随机密码生成。编写程序，在26个字母大小写和9个数字组成的列表中随机生成10个8位密码。

（提示：合法的密码字符共有62个（26个大写英文字母，26个小写英文字母，10个数字），我们要用这62个字符构建一个列表，然后从这个列表中随机抽取8个字符就可以构成一个密码。字符列表的构建可以用最笨的方法，那就是依次写出这62个字符，如:

ch=[‘A’, ‘B’, …., ‘Z’, ‘a’, ‘b’, …, ‘z’, ‘0’, ‘1’, …, ‘9’]（注意程序中不能出现省略号，此处仅是示意），这样做太辛苦。另一种方法就是用程序生成，代码片段如下：

ch = []

c = 'A'

while c <= 'Z':

ch.append(c)

c = chr(ord(c) + 1)

c = 'a'

while c <= 'z':

ch.append(c)

c = chr(ord(c) + 1)

c = '1'

while c <= '9':

ch.append(c)

c = chr(ord(c) + 1)

代码中用到的append()函数是列表中的一个方法，用来在列表的最后追加一个元素；ord()函数是计算字符的的ASCII码值，chr()是把一个整数转换为其对应的字符。

总的程序结构应是一个双重循环，外重循环10次，内重循环8次。）

2. 程序练习题6.2。重复元素判定。 编写一个函数，接受列表作为参数，如果一个元素在列表中出现了不止一次，则返回True,但不要改变原来列表的值。同时编写调用这个函数和测试结果的程序。

（提示：参看教材156页表6.1中的s.count(x)说明，使用此函数。）

3. 程序练习题6.3。重复元素判定续。利用集合的无重复性改编程序练习题6.2的程序，获得 学一个更快更简洁的版本。

（提示：参看教材159页代码片段，使用集合set可以去重。）

4. 程序练习题6.4。文本字符分析。 编写程序接收字符串，按字符出现频率的降序打印字母，分别尝试录入一些中英文文章片段， 比较不同语言之间字符频率的差别。

（提示：参考教材173页代码10~14行。）

5、程序练习题6.5。生日悖论分析。生日悼论指如果一个房间里有23人或以上，那么至少有两个人生日相同的概率大于50%。编写程序，输出在不同随机样本数最下，23个人中至少两个人生日相同的概率。

提示：参考代码如下：

from datetime import datetime

from random import \*

# 随机生成n个人的生日，返回一个列表，列表中每一个元素的形如（月，日）

def generateSamples1(n:int):

birthdays = []

days = [31,29,31,30,31,30,31,31,30,31,30,31]

for i in range(n): # 随机生成n个人的生日（月，日）

month = randint(1,12)

day = randint(1,days[month-1])

someday = (month,day)

birthdays.append(someday)

return birthdays

# 随机生成n个人的生日，返回一个列表，列表中每一个元素的形如（月，日）

def generateSamples2(n:int):

birthdays = []

days = [31,28,31,30,31,30,31,31,30,31,30,31]

year = randint(1950,2000)

for i in range(n): # 随机生成n个人的生日（月，日）

month = randint(1,12)

if (year%400==0) and (year%4==0 and year%100 != 0):

days[1] = 29

else:

days[1] = 28

day = randint(1,days[month-1])

someday = (month,day)

birthdays.append(someday)

return birthdays

# 计算在给定的样本列表birthdays中，23个人中至少有两人生日相同的概率

# 在函数中随机取23个人，计算是否有两人生日相同，重复n次来计算概率

# param birthdays -- 样本列表，列表中存有若干个人的生日数据

# param n -- 计算概率时，事件的重复次数。n越大，计算的概率越接近真实值

def calSameBirthdayProb(birthdays:list, n:int):

num = 0

for i in range(n):

people = sample(birthdays,23)

pset = set(people)

if len(pset) != len(people):

num += 1

return num/n

def main():

while True:

n = int(input("输入一个整数：")) # n是人群的数量

# 如果输入的人群数量少于23则结束

if n < 23:

break

birthdays = generateSamples2(n)

print("{}个随机样本数量下，23个人中至少有两人生日相同的概率是：{}".format(n, calSameBirthdayProb(birthdays, 100000)))

main()

6. 程序练习题6.6。《红楼梦》人物统计。编写程序统计《红楼梦》中前20位出场最多的人物。

（提示：参考教材实例10实现。

拓展：统计《红楼梦》中金陵十二钗的出场次数。金陵十二钗是指：宝钗、黛玉、凤姐、元春、探春、湘云、妙玉、迎春、惜春、巧姐、李纨、可卿。）

1. 实验小结：

主要学习了运用集合的特性对数据去重。组合数据类型中元组、数组、列表和字典等类型及基本操作，运用字典概念处理复杂数据信息及字典的排序。讲解了如何组合数据类型进行文本词频统计。