**Python实验报告2**

**1.实验内容**

1）编写函数，接收一个字符串，分别统计大写字母、小写字母、数字、其他字符的个数，并以元组的形式返回结果。

2）编写函数，可以接收任意多个整数并输出其中的最大值和所有整数之和。

3）编写函数，模拟内置函数sorted()。

4）用字典建立一个通讯录，向字典中添加和删除通讯人（名字、电话、邮箱、工作单位等），查询某个人的信息，然后输出通讯录中所有人的信息。

5）用生成器的方式计算任意起止范围内质数的和。质数,又称素数,是大于1的自然数，除了1和它本身外，不能被其他自然数整除。

**2.详细实现**

**1）**

import re

def count\_str(s):

count = (len(re.findall('[A-Z]', x)), len(re.findall('[a-z]', x)), len(re.findall('[0-9]', x)), len(re.findall('[\W\_]', x)))#通过正则表达式进行对应的匹配，统计所给字符串中大写字母，小写字母，数字，和其他特殊字符的个数

return count

x = input("请输入待统计的字符串")

print(count\_str(x))

**2）**

def fun(\*x):

#输入多个整数，打印最大值和它们的和

print(sum(x))

print(max(x))

number = list(map(int, input('请输入待操作的数字序列,数字之间以空格分隔').split()))#从键盘读入多个整数并将其转换为列表。

fun(\*number)

**3）**

def new\_sorted(x, key=None, reverse=False):

#通过使用sort模拟sorted，结果返回一个新的排序好的对象

new\_list = x

if reverse:

new\_list.sort(key=key, reverse=True)

return new\_list

else:

new\_list.sort(key=key)

return new\_list

print(new\_sorted("This is a py test string".split(), key=str.lower))

print(new\_sorted("This is a py test string".split(), key=str.lower, reverse=True))

**4）**

def add\_person():

#从键盘读入需要添加到通讯录中的人的姓名及其相关信息

person\_name = input("请输入添加的通讯人姓名")

address\_book[person\_name] = input("请输入该通讯人的电话、邮箱、工作单位，数据之间以空格隔开").split()

def find\_person():

#从键盘读入需要查找的通讯人姓名，若查找成功则打印其相关信息，否则则提示没有该用户，函数等待下一个新的通讯人姓名的输入进行查找或是得到输入“退出”以结束查找。

x = input("请输入要查找的用户名")

if x in address\_book.keys():

print(address\_book[x])

else:

print("通讯录中没有该用户，请检查并重新输入用户名，或者输入退出则放弃查找")

x = input("请输入要查找的通讯人姓名")

while x not in address\_book.keys() and x != '退出':

print("通讯录中没有该用户，请重新输入用户名，输入退出则放弃删除")

x = input("请输入要查找的通讯人姓名")

if x in address\_book.keys():

print(address\_book[x])

def delete\_person():

#从键盘读入需要删除的通讯人姓名，若删除成功则打印删除成功的提示，否则则提示没有该用户，函数等待下一个新的通讯人姓名的输入进行删除或是得到输入“退出”以结束删除。

x = input("请输入要删除的通讯人姓名")

if x in address\_book.keys():

del address\_book[x]

print('删除成功')

else:

print("通讯录中没有该用户，请重新输入用户名，输入退出则放弃删除")

x = input("请输入要删除的通讯人姓名")

while x not in address\_book.keys() and x != '退出':

print("通讯录中没有该用户，请重新输入用户名，输入退出则放弃删除")

x = input("请输入要删除的通讯人姓名")

if x in address\_book.keys():

del address\_book[x]

print("删除成功")

address\_book = {}

add\_person()

add\_person()

delete\_person()

find\_person()

print(address\_book)#主程序最后打印当前整个通讯录

**5）**

import math

def is\_prime(x):

#输入一个大于一的自然数，判断其是否为质数，是则返回1，否则返回0

if x <= 1:

print("请输入一个大于一的自然数")

return

i = 2

while i <= math.sqrt(x):

if x % i == 0:

return 0

i += 1

return 1

def get\_prime(x, y):

#建立一个生成器，存储给定区间[x,y]之间的所有质数

while x <= y:

if is\_prime(x) == 1:

yield x

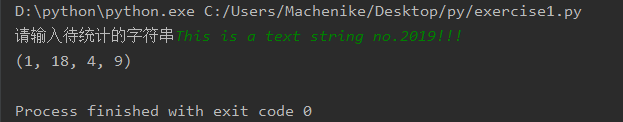
x += 1

x, y = int(input("请输入计算下限")), int(input("请输入计算上限"))

print(sum([n for n in get\_prime(x, y)]))#取出生成器中所有的值对其求和得到结果。

1. **实验结果**

**1）**



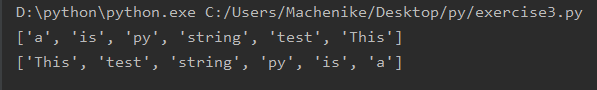
题目一通过调试。

**2）**



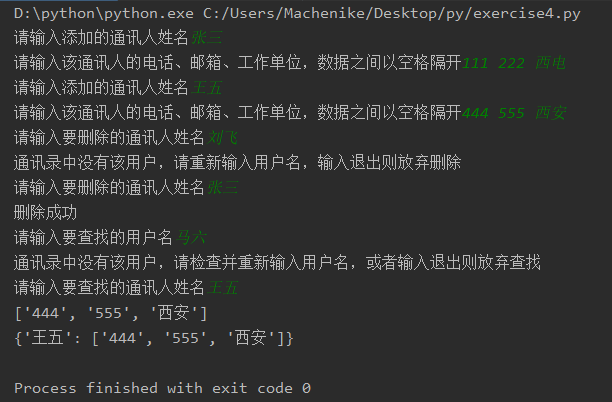
题目二通过调试。

**3）**



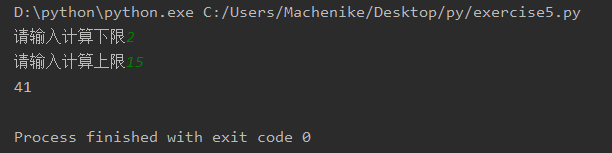
题目三通过调试。

**4）**



题目4通过调试。

**5）**



题目5通过调试。

**4、心得体会**

通过这次py实验，使得我对于函数的理解又加深了不少，特别是对于可变参数的使用。但此次实验也有不足之处，就在于对于生成器的理解与使用还不是特别明白和熟练。虽然程序运行无误，我也大致学会了生成器的一些简单使用方法，但对于生成器的优势与意义以及对应的适用范围还是有些一知半解。查阅相关资料以后大部分对于生成器优势的描述都是节省内存，但由于此时的我对于内存这一块的知识掌握得还比较差，因此也不能立即明白这之中深沉次的意义与价值，需要学习的内容看来还有很多。

何剑萍 18130500172