# 一、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。

# 二、详细实现

1.

import collections

src = input("Please input the string\n")

def func(src):

''' srt is a string, the function is to calculate the letter in the string'''

ret = collections.Counter()

for i in src:

if i.isupper():

ret['Upper'] += 1

elif i.islower():

ret['Lower'] += 1

elif i.isdigit():

ret['Digit'] += 1

else:

ret['Other'] += 1

return tuple(ret.items())

print(func(src))

2.

def func(\*p):

print("The max is {}, and the sum is {}.".format(max(p), sum(p)))

ret = []

while True:

x = input("Please input the number.(input Q to quit)\n")

if x == 'Q':

break

ret.append(int(x))

func(\*ret)

3.

import copy

import inspect

import sys

def quick\_sort(A, start, end, key=None, reverse=True):

if start >= end:

return

left = start

right = end

mid = A[left]

if key is None:

if reverse is True:

while left < right:

while left < right and A[right] >= mid:

right -= 1

A[left] = A[right]

while left < right and A[left] < mid:

left += 1

A[right] = A[left]

A[left] = mid

else:

while left < right:

while left < right and A[right] < mid:

right -= 1

A[left] = A[right]

while left < right and A[left] >= mid:

left += 1

A[right] = A[left]

A[left] = mid

elif inspect.isfunction(key):

if reverse is True:

while left < right:

while left < right and key(A[right]) >= key(mid):

right -= 1

A[left] = A[right]

while left < right and key(A[left]) < key(mid):

left += 1

A[right] = A[left]

A[left] = mid

else:

while left < right:

while left < right and key(A[right]) < key(mid):

right -= 1

A[left] = A[right]

while left < right and key(A[left]) >= key(mid):

left += 1

A[right] = A[left]

A[left] = mid

else:

sys.exit("key is not a callable object.")

quick\_sort(A, start, left-1, key, reverse)

quick\_sort(A, left+1, end, key, reverse)

def Sorted(B, key=None, reverse=True):

'''quickly sort'''

A = copy.deepcopy(B)

quick\_sort(A, 0, len(A)-1, key, reverse)

return A

A = [2, 4, 1, 5]

print(Sorted(A, reverse=False))

A = [[2, 3], [5, 4], [4, 2]]

print(Sorted(A, key=lambda x:x[0], reverse=False))

print(Sorted(A, key=lambda x:x[1]))

Sorted(A, key=1)

4.

Book = dict()

while True:

indication = input("Please input the operation(Add, Del, Req)(Q to quit)\n")

if indication == 'Q':

break

name = input("Please input the name\n")

if indication == 'Add':

Book[name] = dict()

while True:

massage = input("Please input the massage, space to part(telephone 123)(Q to quit)\n")

if massage == 'Q':

break

t = massage.split(' ')

Book[name][t[0]] = t[1]

elif indication == 'Del':

del Book[name]

else:

print(Book[name])

for person in Book.keys():

print(Book[person])

5.

x = input("Please input the number to calculate.\n")

x = int(x)

gen = (i for i in range(2, x+1) if 0 not in (i % j for j in range(2, int(i\*\*0.5)+1)))

print(sum(gen))

# 三、实验结果

调试成功，通过几个简单测试点

# 四、心得体会

熟练使用了collections，掌握了序列解包和传入任意数据，自行实现了sorted有key和reverse，使用的快速排序，掌握了字典嵌套，也会了简单的generation