苏州大学实验报告

院、系	计算机学院	年级专业	计算机科学	姓名	柯骅	学号	2027405033
课程名称	Python 程序设计					成绩	
指导教师	李正华	同组实验者	无		实验日期		

实 验 名 称 实验八 函数

一. 实验目的

通过本次实验要达到如下目的：

1. 掌握 Python 函数的基本概念
2. 理解并掌握 Python 的定义和调用方法
3. 理解 Python 函数的参数传递原理

二. 实验内容

1. 编写一个函数，计算一个整数的所有因子之和，其中因子不包括整数本身，并编写测试程序，在测试程序中输入整数和输出整数的所有因子之和。例如：输入 8，调用该函数之后，得到结果为 7。
2. 编写一个函数，将一个整数的各位数字对调，并编写测试程序，在测试函数中输入整数和输出新的整数。例如：输入 123，调用该函数之后，得到结果为 321
3. (反素数)反素数指一个素数将其逆向拼写后也是一个素数的非回文数。例如：17 和 71 都是素数且都不是回文数，所以 17 和 71 都是反素数。请编写一个函数判断一个数是否是反素数？并编写测试程序找出前 30 个反素数输出到屏幕上，要求每行输出 8 个数，每个数占 5 列，右对齐。

4. (梅森素数)如果一个素数可以写成 2^p-1 形式，其中 p 是一个正整数，那么该数就称作梅森素数。请编写一个函数判断一个素数是否是梅森素数，如果是，则返回 p 的值，否则返回-1。并编写测试程序找出 1000 以内的所有梅森素数输出到屏幕上，要求输出格式如下：

   P(占 3 列右对齐) 2^p-1 (占 4 列右对齐) # 此行不需要输出

   2 3

   3 7

   5 31

5. 编写一个加密函数，实现对一个给定字符串中的字母转变为其后 n 个字符，如果遇到超过字母边界，则从最小字母继续计数，连续的数字字符作为一个整数扩大 n 倍之后替换到对应位置，其中 n 默认为 5。再编写一个解密函数实现对上述加密字符串进行解密。编写测试程序，在测试程序中输入字符串，并输出加密和解密后的字符串。

   例如：

   字符串 str1: avbV125av1, n 默认为 5

   则新的字符串 str2: fagA625fa5

6. 编写一个函数，将给定英文语句中的单词倒序。编写测试程序，从键盘输入英文语句，并输出倒序后的英语字符串。

例：给定"What a wonderful day!"，输出："day! wonderful a What"。

7. 编写一个函数，统计一个给定的英文语句中，某个指定位置的字符在字符串中出现的次数，统计时不区分字母的大小写，默认字符位置为 0。编写测试程序，在测试程序中输入英文语句，指定要查找的字符位置，并输出该字符在语句中出现的次数。例如：英文语句：This is a test example. 统计位置 0 的字符是 t，则在语句中出现的次数为：3。(3 次包括大写和小写的 t)
8. 编写一个递归函数，求解 Fibonacci 数列（兔子繁殖）问题的某项的值。编写测试程序，从键盘输入指定项，并输出 Fibonacci 数列指定项的值。
9. 编写一个函数实现冒泡排序。从键盘输入依次输入 10 个整数，分别按照从小到大、从大到小进行排序，并分别输出排序后的结果。
10. 编写一个函数实现选择排序。从键盘依次输入 10 个字母（如果有大小写，需要区分），按照字母的 ASCII 码值分别进行从小到大、从大到小的排序，并输出排序的结果。
11. 给定如下一段英文

    A major drawback of cross-network recommender solutions is that they can only be applied to users that are overlapped across networks. Thus, the non-overlapped users, which form the majority of users are ignored. As a solution, we propose CnGAN, a novel multi-task learning based recommend architecture。

    编写一个函数，要求实现以下功能：1）统计有多少个不同的单词；2）根据每个单词 ASCII 码值的和（单词 they ASCII 码值的和是：116+104+101+121=442）对单词进行从小到大的排序，重复出现的单词只算一次的和，按行输出单词及对应的和。

三. 实验步骤和结果（每一道题必须画流程图）

1. 第一题程序如下：

```
def func1(n):
    tot=0
    for i in range(1,n):#枚举 1-（n-1）的数
        if n%i==0:
            tot+=i
    return tot
print(func1(8))
print(func1(100))
#思路：用 for 循环找出所有因数，累加
```

运行结果如下：

```
7
117
```

2. 第二题程序如下：

```
def func2(n):
    return int(str(n)[::-1])#转化成字符串再转化成整数

print(func2(650000))
print(func2(123456789))
#思路：转化成字符串，用反向切片
```

运行结果如下：

```
56
987654321
```

3. 第三题程序如下：

```
def jud(n):#判断是否为反素数
    t=0
    for i in range(2,int(n**0.5)+1):
        if n%i==0:
            t=1
    n=int(str(n)[::-1])
    for i in range(2,int(n**0.5)+1):
        if n%i==0:
            t=1
    if t:
        return False
    else:
        return True

c=0
i=1
while c<30:#计数，只输出 30 个
    i+=1
    if jud(i):
        c+=1
        print('%5d'%i,end='')
        if c%8==0:
            print()#换行
#思路：编写一个判断反素数的函数
#      只需要在判断素数的基础上在判断他的反数即可
#      在主程序中用 while 控制个数
```

运行结果如下：

```
    2    3    5    7   11   13   17   31
   37   71   73   79   97  101  107  113
  131  149  151  157  167  179  181  191
  199  311  313  337  347  353
```

4．第四题程序如下：

```
def func4(n):
    t=0
    #判断是否是素数
    for i in range(2,int(n**0.5)+1):
        if n%i==0:
            return False
    n=n+1
    tot=0
    #计算 p
    while n!=1:
        if n%2!=0:
            return False
        else:
            n=n//2
            tot+=1
    return tot

for i in range(2,1000):
    if func4(i):#返回值只有 False 和 p 两种情况
        print('%3d%4d'%(func4(i),i))
```

#思路：梅森素数只需要判断是否是素数和是否是 2 的 p 次方-1 即可

运行结果如下：

```
  2   3
  3   7
  5  31
  7 127
```

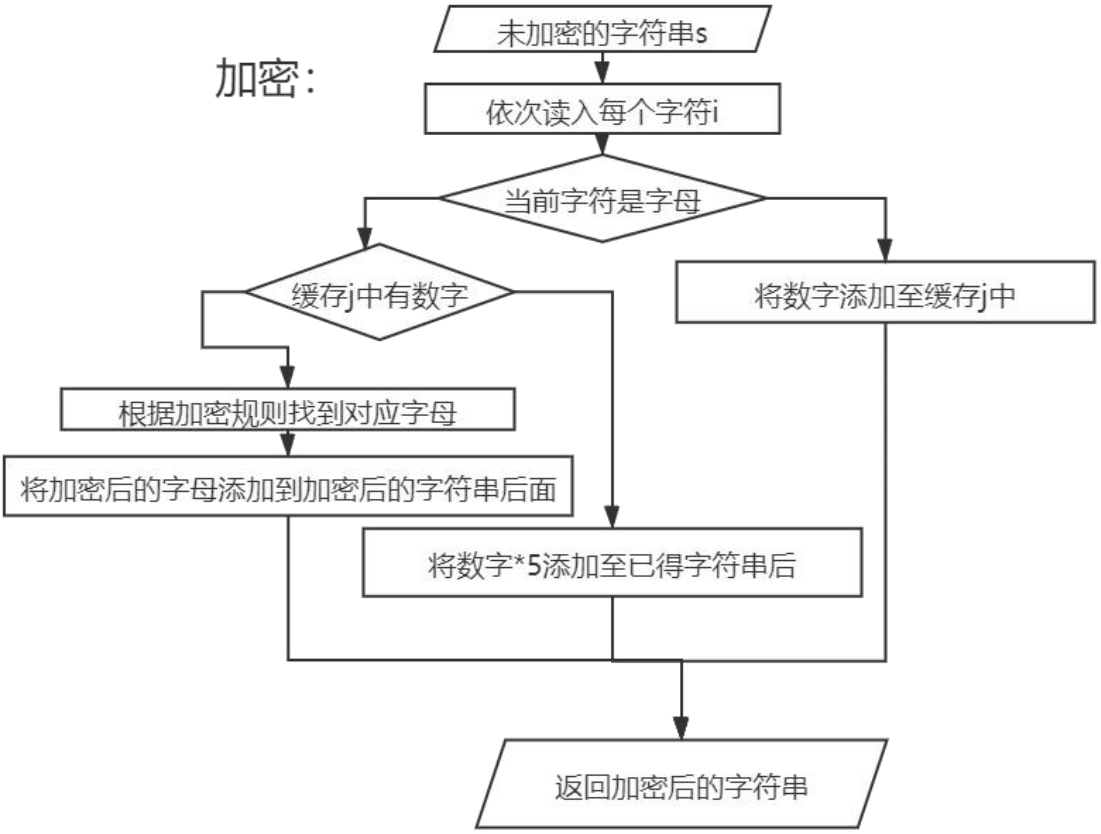
5. 第五题程序如下：

```
def jia(s,n=5):#加密
    s = s + ' '
    ss = ''
    j = 0
    for i in s:
        o = ord(i)
        if o in range(48, 58):#判断并联系连在一起的整数
            j = j * 10 + o - 48
        elif j != 0:
            ss = ss + str(j * 5)
            j = 0
        if o in range(97, 123):
            ss = ss + chr((o +n-96) % 26 + 96)
        if o in range(65, 91):
            ss = ss + chr((o +n-64) % 26 + 64)
    return ss

def jie(s,n=5):#解密
    s = s + ' '
    ss = ''
    j = 0
    for i in s:
        o = ord(i)
        if o in range(48, 58):#判断并联系连在一起的整数
            j = j * 10 + o - 48
        elif j != 0:
            ss = ss + str(j // 5)
            j = 0
        if o in range(97, 123):
            ss = ss + chr((o-96+26-n) % 26+96)
        if o in range(65, 91):
            ss = ss + chr((o-64+26-n) % 26+64)
    return ss

s=input('please input a string:')
n=int(input('please input n:'))
print(jia(s,n))
print(jie(s,n))
#思路：枚举所有的 s 中的字符
#      利用其 ord（）进行加密和解密
```

教务处制



运行结果如下：

```
please input a string:avbV125av1
please input n:5
fagA625fa5
vqwQ25vq0

please input a string:fagA625fa5
please input n:5
kflF3125kf25
avbV125av1
```

6. 第六题程序如下：

```
def func6(s):
    return ' '.join(list(s.split())[::-1])
print(func6(input('please input a sentence:')))
#思路：字符串和列表转换的熟练应用
```

运行结果如下：

```
please input a sentence:What a wonderful day!
day! wonderful a What
```

7. 第七题程序如下：

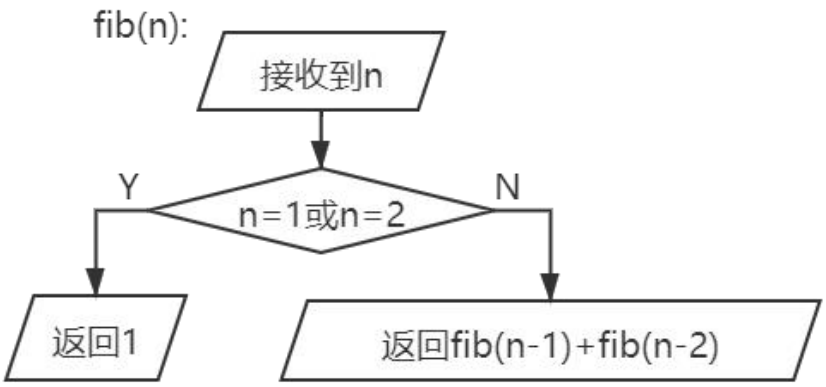
```
def func7(s,n=0):
    d,s={},s.upper()#对大小写不敏感
    for i in s:
        d[i]=d.get(i,0)#放止报错
        d[i]+=1
    return(d[s[n]])
print(func7('This is a test example.'))
#思路：用字典统计每种字母出现的次数
```

运行结果如下：

```
3
```

8. 第八题程序如下：

```
def fib(n):
    if n==1 or n==2:#初值/终止条件
        return 1
    else:
        return fib(n-1)+fib(n-2)#自身调用自身
print(fib(int(input('please input n:'))))
#思路：根据初值决定终止条件，再用递归不断调用自身
```



教务处制

运行结果如下：

```
please input n:5
5
```

```
please input n:10
55
```

9．第九题程序如下：

```
def mao1(a):#从小到大排序
    l=len(a)
    for i in range(l):#每次找到剩余数中最大数
        for j in range(l-i-1):
            if a[j]>a[j+1]:
                temp=a[j]
                a[j]=a[j+1]
                a[j+1]=temp
    return a

def mao2(a):#从大到小排序
    l=len(a)
    for i in range(l):#每次找到剩余数中最小数
        for j in range(l-i-1):
            if a[j]<a[j+1]:
                temp=a[j]
                a[j]=a[j+1]
                a[j+1]=temp
    return a

l=list(map(int,input('please input 10 numbers:').split()))
print(mao1(l))
print(mao2(l))
#思路：从数组的第一个数开始，依次和后面的数相比
#      若前者大则交换顺序，直到所有大的数冒到最后，最后按照从小到大排序。
```

运行结果如下：

```
please input 10 numbers:6 5 8 9 7 4 2 3 0 1
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9]
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]
```

10. 第十题程序如下：

```
def sor1(l):#正序
    return ''.join(sorted(l,key=lambda x:ord(x)))
def sor2(l):#逆序
    return ''.join(sorted(l,key=lambda x:-ord(x)))
l=list(input('please input a string:'))
print(sor1(l))
print(sor2(l))
#思路：用 key=lambda 进行排序
#      逆序排序只需要加上负号即可
```

运行结果如下：

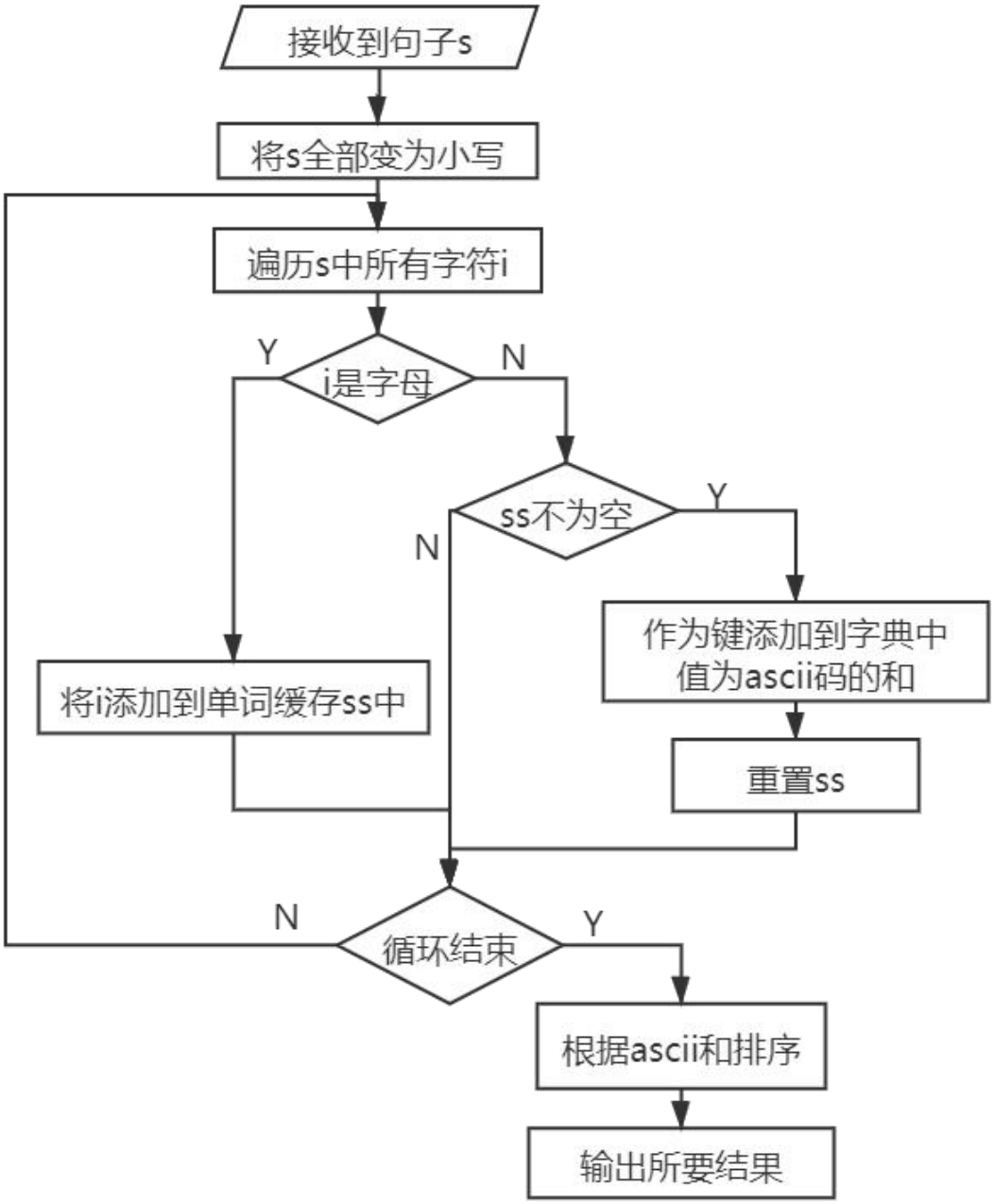
```
please input a string:FGhdgaIpOm
FGIOadghmp
pmhgdaOIGF
```

11. 第十一题程序如下：

```
def func(s):
    s = s.lower()#对大小写不敏感
    ss,l='',{}#用字典储存，键是字母，值是 ascii 码的和
    for i in s:
        o = ord(i)
        if o in range(97, 123):
            ss = ss + i
        else:
            if ss:#添加单词至字典
                l[ss]=l.get(ss,sum(ord(ss[i]) for i in range(len(
ss))))
                ss=''
    l=list(l.items())
    l.sort(key=lambda x:x[1])#用关键字排序
    print(len(l))
    for i in l:
        print(i[0],i[1])

s='A major drawback of cross-network recommender solutions is tha
t they can only be applied to users that are overlapped across ne
tworks. Thus, the non-overlapped users, which form the majority o
f users are ignored. As a solution, we propose CnGAN, a novel mul
ti-task learning based recommend architecture。'
func(s)
```

#思路：（1）因为就算大写也只能算是一个单词，所以先全部变成小写
#　　（2）先采用字典储存，键是单词，值是 ascii 的和
#　　（3）排序用关键字排序即可



运行结果如下：

```
40
a 97
be 199
as 212
of 213
is 220
we 220
to 227
can 306
are 312
the 321
non 331
that 433
task 435
form 436
they 442
only 450
thus 452
based 511
cngan 519
which 531
major 537
novel 548
cross 554
multi 555
users 562
across 651
applied 735
ignored 744
propose 776
network 778
drawback 831
learning 848
majority 879
networks 893
solution 893
recommend 954
solutions 1008
overlapped 1074
recommender 1169
architecture 1283
```

拓展题：

（1）汉诺塔

程序如下：

```
def func(n,a,b,c):
    if n==1: #当 n=1 是直接移动即可
        print(a,'to',c)  # 打印 func 为字符串,a 和 c 是参数
        return
    func(n-1, a, c, b)  # 移动上面的 n-1 个盘子
    func(1,a,b,c)  # 最后一个盘子
    func(n-1,b,a,c)  # b 到 a 再到 c

func(int(input('please input the number of plates:')),'A','B','C')
```

思路：（1）先把 n-1 个圆盘从 A 移动到柱子 B。
（2）剩下的 1 个盘移动到柱子 C
（3）将 B 上的 n-1 个移动到 A 再移动到 C
而第一，第三步就相当于移动 n-1 个盘子，从而形成递归

运行结果如下：

```
please input the number of plates:3
A to C
A to B
C to B
A to C
B to A
B to C
A to C


please input the number of plates:4
A to B
A to C
B to C
A to B
C to A
C to B
A to B
A to C
B to C
B to A
C to A
B to C
A to B
A to C
B to C
```

教务处制

（2）辗转相除法求最小公倍数

程序如下：

```
x,y=list(map(int,input("请输入两个正整数:").split()))
a,b=max(x,y),min(x,y)#a 是较大值，b 是较小值
while a%b!=0:
    temp=a%b
    a=b
    b=temp#交替过程
print("最小公倍数数是:",x*y//b)
```

思路：辗转相除法最后得到 x*y//b 是最小公倍数

运行结果如下：

```
请输入两个正整数:8 12
最小公倍数数是: 24

请输入两个正整数:5 6
最小公倍数数是: 30
```

四. 实验总结（包括对老师的建议）

掌握了 Python 函数的基本概念

理解并掌握了 Python 的定义和调用方法

理解了 Python 函数的参数传递原理

Python 中函数是可以嵌套的

定义函数时有正常参数，变长参数，默认值参数，变长关键字参数四种参数可以定义

并且变长参数后的正常参数，必须使用关键字调用

变长参数调用时，将传 0-无穷个参数，函数内会以元组方式来表示