# 01

参考例2.1，由用户输入三角形的三条边的边长a、b和c，计算并输出三角形的面积s= 。其中，h为三角形周长的一半。

建议尝试使用2种输出格式控制方式。

#!/usr/bin/python3

# @File: 题目1.py

# --coding: utf-8--

# @Author: 孟德昊

# @Time: 2021年 10月 10日 19:26

# 说明:

import math

if \_\_name\_\_ == "\_\_main\_\_":

a = float(input("请输入一条边长："))

b = float(input("请输入一条边长："))

c = float(input("请输入一条边长："))

h = (a+b+c)/2

s = math.sqrt(h\*(h-a)\*(h-b)\*(h-c))

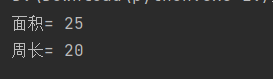
print(s)



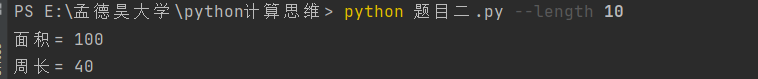
# 2

参考例6.2，解析命令行参数所输入的边长的值，计算并输出正方形的面积和周长

#!/usr/bin/python3  
# @File: 题目二.py  
# --coding: utf-8--  
# @Author: 孟德昊  
# @Time: 2021年 10月 10日 19:32  
# 说明:  
import argparse  
if \_\_name\_\_ == "\_\_main\_\_":  
 parser = argparse.ArgumentParser()  
 parser.add\_argument('--length',default=5,type=int,help="边长")  
 args = parser.parse\_args()  
 area = args.length \* args.length  
 print("面积=",area)  
 circu = args.length \* 4  
 print("周长=",circu)



下图为使用arggparse输入方式改变参数

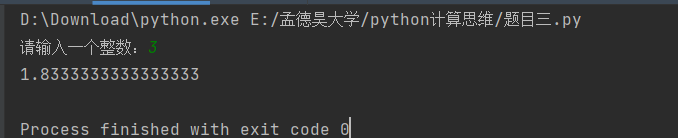


# 题目三

参考【例8.5】

先一个定义计算并返回第n阶调和数（1 + 1/2 + 1/3 + … + 1/n）的函数，然后在命令行中输入参数n，调用函数计算前n个调和数之和，并将每次函数调用的结果放入一个列表，最后输出该列表，以及列表中所有元素之和（精确到小数点后面一位）。

#!/usr/bin/python3  
# @File: 题目三.py  
# --coding: utf-8--  
# @Author: 孟德昊  
# @Time: 2021年 10月 10日 20:02  
# 说明:  
import sys  
def sum(n):  
 total = 0  
 for i in range(1,n+1):  
 total += 1/i  
 return total  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 n = int(input("请输入一个整数："))  
 total = sum(n)  
 print(total)



#!/usr/bin/python3  
# @File: 题目三.py  
# --coding: utf-8--  
# @Author: 孟德昊  
# @Time: 2021年 10月 10日 20:02  
# 说明:  
import sys  
def sum(n):  
 total = 0  
 for i in range(1,n+1):  
 total += 1/i  
 return total  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 n = int(sys.argv[1])   
 total = sum(n)  
 print(total)



# 题目四

(1) 打开该文件file2.txt，并将元组t = "spring", "summer", "autumn"写入到文件中，要求元组中每个元素单独放在一行中，建议使用for循环语句。

(2) 打开该文件并字符串" winter"追加到文件最后一行下面的一行，使得文件中的内容包括如下4行文字：

spring

summer

autumn

winter

(3) 打开该文件并将其中的最后两行文字显示到屏幕上。

(4) 思考：若一开始文件file2.txt不存在，一定要先创建该文件？若要在读取文件的过程中增加异常处理的代码，以便在文件不存在或打开失败的异常情况下，向用户提示“文件file2.txt打开失败！”，应该怎么修改代码》

### （1）

#!/usr/bin/python3

# @File: 题目四.py

# --coding: utf-8--

# @Author: 孟德昊

# @Time: 2021年 10月 10日 19:37

# 说明:

if \_\_name\_\_ == "\_\_main\_\_":

t = ("spring","summer", "autumn")

with open('./file2.txt','w') as fp:

for season in t:

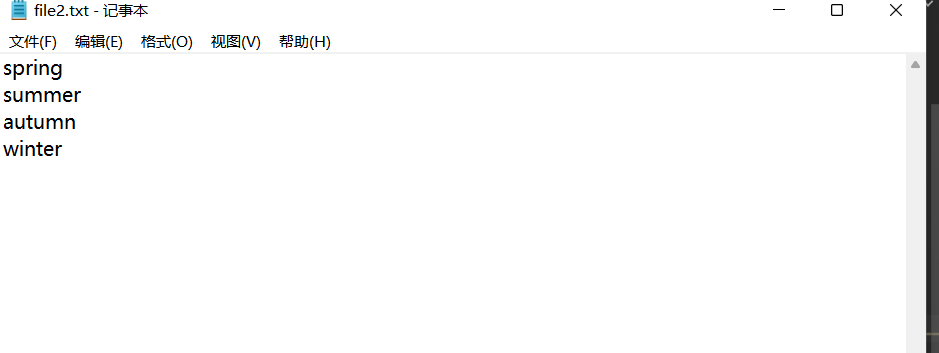
fp.write(season)

fp.write('\n')



## （2）

#!/usr/bin/python3  
# @File: 题目四.py  
# --coding: utf-8--  
# @Author: 孟德昊  
# @Time: 2021年 10月 10日 19:37  
# 说明:  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # t = ("spring","summer", "autumn")  
 # with open('./file2.txt','w') as fp:  
 # for season in t:  
 # fp.write(season)  
 # fp.write('\n')  
  
  
 with open('file2.txt','a') as fp:  
 fp.write("winter")



# （3）

#!/usr/bin/python3  
# @File: 题目四.py  
# --coding: utf-8--  
# @Author: 孟德昊  
# @Time: 2021年 10月 10日 19:37  
# 说明:  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 # t = ("spring","summer", "autumn")  
 # with open('./file2.txt','w') as fp:  
 # for season in t:  
 # fp.write(season)  
 # fp.write('\n')  
  
  
 # with open('file2.txt','a') as fp:  
 # fp.write("winter")  
 season = []  
 with open('file2.txt','r') as fp:  
 for i in fp:  
 season.append(i)  
 print(season[-2],season[-1])



思考：打开文件时需要先创建文件（仅当为r模式时）

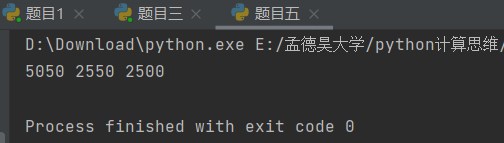
若打开失败使用trt except

try:  
 fp = open('file2.txt',)  
 # 输入其余代码  
except FileNotFoundError:  
 fp = open('file2.txt','a+/a/w/…')  
 # 输入其余代码

# 题目五

参考[例3.9]和[例3.10]，分别利用for循环和while循环求1~100中所有数之和，以及所有奇数的和、所有偶数的和。

#!/usr/bin/python3  
# @File: 题目五.py  
# --coding: utf-8--  
# @Author: 孟德昊  
# @Time: 2021年 10月 10日 20:14  
# 说明:  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 odd=even=total=0  
 for i in range(1,101):  
 total += i  
 if i%2 == 0 :  
 even += i  
 else :  
 odd += i  
  
 print(total,even,odd)



#!/usr/bin/python3

# @File: 题目五（2）.py

# --coding: utf-8--

# @Author: 孟德昊

# @Time: 2021年 10月 20日 19:25

# 说明:

if \_\_name\_\_ == "\_\_main\_\_":

odd = even = total = i = 0

while(i <= 100):

total += i

if (i % 2 == 0):

even += i

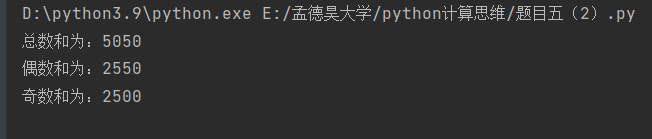
else: odd += i

i += 1

print("总数和为：{}".format(total))

print("偶数和为：{}".format(even))

print("奇数和为：{}".format(odd))



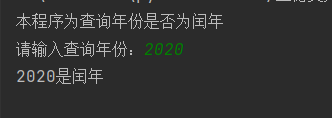
# 题目六

参考P44[例3.8]，至少使用2种方法编写一个函数来判断某一年是否为闰年。

年份数据由用户输入，调用该函数，函数调用完毕后，输出判断结果。

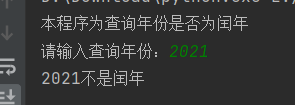
## （1）

#!/usr/bin/python3  
# @File: 题目六.py  
# --coding: utf-8--  
# @Author: 孟德昊  
# @Time: 2021年 10月 10日 20:23  
# 说明:  
def leap\_year(year):  
 if (year % 4 == 0 and year % 100 != 0) or y % 400==0:  
 print("{}是闰年".format(year))  
 else:  
 print("{}不是闰年".format(year))  
if \_\_name\_\_ == "\_\_main\_\_":  
 print("本程序为查询年份是否为闰年")  
 year = int(input("请输入查询年份："))  
 leap\_year(year)



## （2）

def leap\_year(year):  
 if year % 4 == 0:  
 print("{}是闰年".format(year))  
 else:  
 if year % 4 ==0:  
 if year % 100 == 0:  
 print("{}不是闰年".format(year))  
 else:  
 print("{}是闰年".format(year))  
 else:  
 print("{}不是闰年".format(year))  
if \_\_name\_\_ == "\_\_main\_\_":  
 print("本程序为查询年份是否为闰年")  
 year = int(input("请输入查询年份："))  
 leap\_year(year)



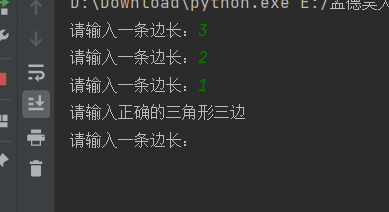
# 补充题

对于题1，若由用户输入三角形的三条边的边长（即a、b、c），则必须对三角形的边长小于或等于0和两边之和小于或等于第三边的情况识别出来，并提示用户。请思考下面两种解决方案：

1. 用while循环语句进行判断，出现上述错误会提示并要求用户重新输入三条边长，直到用户输入合法边长之后计算并输出三角形的面积。

## (1)

#!/usr/bin/python3  
# @File: 补充1.py  
# --coding: utf-8--  
# @Author: 孟德昊  
# @Time: 2021年 10月 10日 20:36  
# 说明:  
import math  
if \_\_name\_\_ == "\_\_main\_\_":  
 t = 1  
 while t:  
 a = float(input("请输入一条边长："))  
 b = float(input("请输入一条边长："))  
 c = float(input("请输入一条边长："))  
 if a <= 0 or b <= 0 or c <=0 or a+b <= c or a + c <= b or b + c <= a :  
 t=1  
 print("请输入正确的三角形三边")  
 else:  
 t = 0  
 break  
 h = (a+b+c)/2  
 s = math.sqrt(h\*(h-a)\*(h-b)\*(h-c))  
 print(s)



(2) 用异常处理机制（raise语句和ValueError对象）来终止程序并提醒用户错误的原因。

## (2)

利用raise和Valueerror对象处理程序

#!/usr/bin/python3  
# @File: 补充2.py  
# --coding: utf-8--  
# @Author: 孟德昊  
# @Time: 2021年 10月 10日 20:41  
# 说明:  
import math  
if \_\_name\_\_ == "\_\_main\_\_":  
 try:  
 a = float(input("请输入一条边长："))  
 b = float(input("请输入一条边长："))  
 c = float(input("请输入一条边长："))  
 if a <= 0 or b <= 0 or c <=0 or a+b <= c or a + c <= b or b + c <= a :  
 raise ValueError  
 h = (a + b + c) / 2  
 s = math.sqrt(h \* (h - a) \* (h - b) \* (h - c))  
 print(s)  
 except ValueError:  
 print("请输入正确的三角形三边")

**题目1**

用折半查找法查33和58在列表[1,13,26,33,45,55,68,72,83,99]中的位置，

要求：

* 分别用递归和非递归的方法来实现。
* 列表[1,13,26,33,45,55,68,72,83,99]中各元素的数值从命令行参数中获取。
* 输出结果是：

关键字33在列表中的索引是：3

关键字58不在该列表中

#!/usr/bin/python3

# @File: 折半查找法1.py

# --coding: utf-8--

# @Author: 孟德昊

# @Time: 2021年 11月 10日 15:40

'''说明:

用折半查找法查33和58在列表[1,13,26,33,45,55,68,72,83,99]中的位置，

非递归的方法来实现

列表[1,13,26,33,45,55,68,72,83,99]中各元素的数值从命令行参数中获取。

'''

import sys

def compare(a,c,max,min,mid): #判断函数

for i in range(len(a)):

if ( c == a[min] ):

print("关键词{}在列表中的索引是：{}".format(c,min))

break

elif ( c == a[max] ):

print("关键词{}在列表中的索引是：{}".format(c,max))

break

elif ( c == a[mid] ):

print("关键词{}在列表中的索引是：{}".format(c,mid))

break

elif ( c > a[mid]) :

mid = int((mid + max + 1 ) / 2)

elif ( c < a[mid] ):

mid = int((mid + min ) / 2)

if(i == 9):

print("关键词{}不在该列表中".format(c))

if \_\_name\_\_ == "\_\_main\_\_":

a = [] #创建空列表

for i in range(1,11): #输入列表

b = int(sys.argv[i])

a.append(b)

min = 0 #初始最小值的索引

max = 9 #初始最大值的索引

mid = 4 #初始中值的索引

compare(a, 33, max, min, mid) #调用函数

compare(a, 58, max, min,mid)

#

# min = 0

# max = 9

# mid = int (( max + min + 1 ) / 2 )

# c = 58

# for i in range(len(a)):

# if ( c == a[min] ):

# print("关键词{}在列表中的索引是：{}".format(c,min))

# break

# elif ( c == a[max] ):

# print("关键词{}在列表中的索引是：{}".format(c,max))

# break

# elif ( c == a[mid] ):

# print("关键词{}在列表中的索引是：{}".format(c,mid))

# break

# elif ( c > a[mid]) :

# min = mid

# mid = int((mid + max + 1 ) / 2)

# elif ( c < a[mid] ):

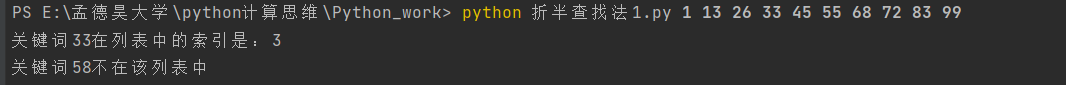
# max = mid

# mid = int((mid + min ) / 2)

#

# if(i == 9):

# print("关键词{}不在该列表中".format(c))



(2)

#!/usr/bin/python3

# @File: 折半查找法2.py

# --coding: utf-8--

# @Author: 孟德昊

# @Time: 2021年 11月 10日 15:50

'''说明:

用折半查找法查33和58在列表[1,13,26,33,45,55,68,72,83,99]中的位置，

用递归的方法来实现。

'''

import sys

def find\_num(a,c,min,max,mid):

if(max - mid >= 1):

if ( c == a[min] ):

print("关键词{}在列表中的索引是：{}".format(c,min))

elif ( c == a[max] ):

print("关键词{}在列表中的索引是：{}".format(c,max))

elif ( c == a[mid] ):

print("关键词{}在列表中的索引是：{}".format(c,mid))

elif ( c > a[mid]) :

min = mid

mid = int((mid + max + 1 ) / 2)

find\_num(a, c, min, max, mid)

elif ( c < a[mid] ):

max = mid

mid = int((mid + min ) / 2)

find\_num(a, c, min, max,mid)

else:

print("关键词{}不在该列表中".format(c))

if \_\_name\_\_ == "\_\_main\_\_":

a = []

for i in range(1, 11):

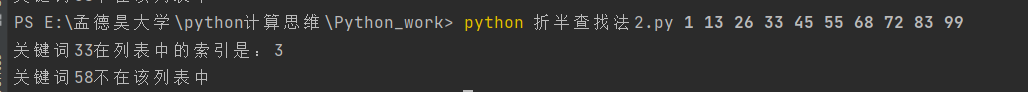
b = int(sys.argv[i])

a.append(b)

# print(a)

find\_num( a, 33, 0, 9, 4 )

find\_num( a, 58, 0, 9, 4 )



**题目2**

在例9.21的基础上，修改父类Dimension中的area函数，该函数就输出一行话“形状没定，无法计算面积！”。然后再增加一个三角形的类Triangle，在其中定义构造函数、计算面积的函数area。最后分别创建圆形、矩形、三角形的对象，然后输出它们的面积。

#!/usr/bin/python3

# @File: 输出面积.py

# --coding: utf-8--

# @Author: 孟德昊

# @Time: 2021年 11月 10日 17:15

'''说明:

在例9.21的基础上，修改父类Dimension中的area函数，该函数就输出一行话“形状没定，无法计算面积！”。

然后再增加一个三角形的类Triangle，在其中定义构造函数、计算面积的函数area。最后分别创建圆形、矩形、三角形的对象，然后输出它们的面积。

'''

from math import sqrt

class Dimension:

def \_\_init\_\_(self,x,y,z):

self.x = x

self.y = y

self.z = z

def area(self):

print("形状没定，无法计算面积！")

class Circle(Dimension):

def \_\_init\_\_(self,r):

Dimension.\_\_init\_\_(self,r,0,0)

def area(self):

return 3.14 \* self.x \* self.x

class Rectangle(Dimension):

def \_\_init\_\_(self,w,h):

Dimension.\_\_init\_\_(self,w,h,0)

def area(self):

return self.x \* self.y

class Triangle(Dimension):

def \_\_init\_\_(self,a,b,c):

Dimension.\_\_init\_\_(self,a,b,c)

def area(self):

p = (self.x + self.y + self.z)/2

return sqrt( p \* ( p - self.x ) \* ( p - self.y ) \* ( p - self.z ))

if \_\_name\_\_ == "\_\_main\_\_":

circle = Circle(1)

rectangle = Rectangle(1,2)

triangle = Triangle(2,2,2)

print(circle.area())

print(rectangle.area())

print(triangle.area())

文本

描述已自动生成

**题目3**

参照例9.23，编写运算符重载的程序

#!/usr/bin/python3

# @File: 编写运算符重载.py

# --coding: utf-8--

# @Author: 孟德昊

# @Time: 2021年 11月 10日 17:31

'''说明:参照例9.23，编写运算符重载的程序

'''

class Mylist:

def \_\_init\_\_(self,\*args):

self.\_\_mylist = []

for arg in args:

self.\_\_mylist.append(arg)

def \_\_add\_\_(self, n):

for i in range(0,len(self.\_\_mylist)):

self.\_\_mylist[i] += n

def \_\_sub\_\_(self, n):

for i in range(0,len(self.\_\_mylist)):

self.\_\_mylist[i] -= n

def \_\_mul\_\_(self, n):

for i in range(0, len(self.\_\_mylist)):

self.\_\_mylist[i] \*= n

def \_\_truediv\_\_(self, n):

for i in range(0, len(self.\_\_mylist)):

self.\_\_mylist[i] /= n

def \_\_len\_\_(self):

return (len(self.\_\_mylist))

def \_\_repr\_\_(self):

str1 = ''

for i in range(0,len(self.\_\_mylist)):

str1 += str(self.\_\_mylist) + ' '

return str1

if \_\_name\_\_ == "\_\_main\_\_":

m = Mylist(1,2,3,4,5)

m + 2;print(repr(m))

m - 2;print(repr(m))

m \* 2;print(repr(m))

m / 2;print(repr(m))

print(len(m))

电脑萤幕

描述已自动生成

**题目4**

建立学校信息管理系统：

* 建立Person类，包括：姓名name、性别sex、年龄age这3个数据成员，一个构造函数、一个输出信息的printInfo函数。
* 建立Student类，该类继承自Person类，并且新增数据成员：班级classes（用string类型）、学号studentID（用int类型）、成绩score（用字典类型，可以自己定义三门课程，比如语数外）、总人数count（用int类型，这是静态属性，即类属性），和一个构造函数、一个输出信息的printInfo函数。
* 建立Teacher类，该类也继承自Person类，并且新增：部门department（用string类型）、工号teacherID（用int类型）、讲授课程course（用列表类型）、薪水salary（用int类型，这是私有数据）这4个数据成员，和一个构造函数、一个输出信息的printInfo函数（这里不输出私有数据，但可以考虑一下如何提供私有数据的输出方法）。

然后创建2个学生对象和1个教师对象，然后分别调用这3个对象的printInfo函数输出各自的（非私有）数据信息。在学生的构造函数中让count自增，在析构函数中让count自减。

（说明：上面已经规定的内容在自己编程时也允许灵活调整，没有规定的更是可以自由发挥，主要是必须学会类的定义和使用）

#!/usr/bin/python3

# @File: 学校信息管理系统.py

# --coding: utf-8--

# @Author: 孟德昊

# @Time: 2021年 11月 10日 16:21

'''说明:建立学校信息管理系统：

建立Person类，包括：姓名name、性别sex、年龄age这3个数据成员，一个构造函数、一个输出信息的printInfo函数。

建立Student类，该类继承自Person类，并且新增数据成员：班级class（用string类型）、学号studentID（用int类型）、成绩score（用字典类型，可以自己定义三门课程，比如语数外）、总人数count（用int类型，这是静态属性，即类属性），和一个构造函数、一个输出信息的printInfo函数。

建立Teacher类，该类也继承自Person类，并且新增：部门department（用string类型）、工号teacherID（用int类型）、讲授课程course（用列表类型）、薪水salary（用int类型，这是私有数据）这4个数据成员，和一个构造函数、一个输出信息的printInfo函数（这里不输出私有数据，但可以考虑一下如何提供私有数据的输出方法）。

然后创建2个学生对象和1个教师对象，然后分别调用这3个对象的printInfo函数输出各自的（非私有）数据信息。在学生的构造函数中让count自增，在析构函数中让count自减。

（说明：上面已经规定的内容在自己编程时也允许灵活调整，没有规定的更是可以自由发挥，主要是必须学会类的定义和使用）

'''

class Person:

def \_\_init\_\_(self,name,sex,age):

self.name = name

self.sex = sex

self.age = age

# Person.pirntInfo(self)

def pirntInfo(self):

print("姓名：{},性别：{},年龄：{}".format(self.name,self.sex,self.age))

class Student(Person):

count = 0

def \_\_init\_\_(self,name,sex,age,classes,studentId,score):

Person.\_\_init\_\_(self,name,sex,age)

self.classes = classes

self.studentId = studentId

self.score = {"语文":score[0],"数学":score[1],"英语":score[2]}

Student.count += 1

# Student.pirntInfo(self)

def \_\_del\_\_(self):

Student.count -= 1

print("{}已开除学籍".format(self.name))

print(Student.count)

def pirntInfo(self):

print("姓名：{},性别：{},年龄：{},班级：{},学号：{},成绩：{}".format(self.name,self.sex,self.age,self.classes,self.studentId,self.score))

print("学生人数:{}".format(Student.count))

class Teacher(Person):

def \_\_init\_\_(self, name, sex, age, department, teacherId, course,salary):

Person.\_\_init\_\_(self, name, sex, age)

self.department = department

self.teacherId = teacherId

self.course = course

self.\_\_salary = salary

# Teacher.pirntInfo(self)

def pirntInfo(self):

print("姓名：{},性别：{},年龄：{},部门：{},工号：{},讲授课程：{}".format(self.name, self.sex, self.age, self.department, self.teacherId,

self.course))

if \_\_name\_\_ == "\_\_main\_\_":

a = Person("孟德昊","男",18)

a.pirntInfo()

b = Student("孟德昊","男",19,2102,3201303014,[98,91,33])

b.pirntInfo()

c = Student("小q","男",18,2102,3201303015,[98,91,77])

c.pirntInfo()

q = Student("杨y","女",21,1902,3190606066,[99,100,100])

q.pirntInfo()

d = Teacher("老q", "男", 40, "计算机学院", 123456789,"python及计算思维","100000")

d.pirntInfo()

**文本

描述已自动生成**

调用私有数据在类中创建函数调用

**题目5**

随机产生一个0到100之间（包括0和100）的偶数，请用户猜测具体是哪个数，即：不断从标准输入读取用户的猜测值（用户每次输入一个偶数），并根据猜测值给出提示信息：“太大”、“太小”或“正确!”

#!/usr/bin/python3

# @File: 猜数.py

# --coding: utf-8--

# @Author: 孟德昊

# @Time: 2021年 11月 10日 19:14

'''说明:随机产生一个0到100之间（包括0和100）的偶数，请用户猜测具体是哪个数，即：不断从标准输入读取用户的猜测值（用户每次输入一个偶数），

并根据猜测值给出提示信息：“太大”、“太小”或“正确!”

'''

from random import randrange

def compare(a,s):

if ( a > s ):

too\_small()

elif( a < s ):

too\_big()

elif( a == s ):

right()

def right():

print("正确")

def too\_small():

print("太小")

def too\_big():

print("太大")

if \_\_name\_\_ == "\_\_main\_\_":

a = randrange(0,101,2)

# print(a)

while(1):

s = int(input("请输入一个偶数:"))

compare(a,s)

if(a == s): #当猜中时不在循环

break

文本

中度可信度描述已自动生成

**题目6**

编写一个程序，获取系统当前的时间等信息，输出格式如下面方框里所示。然后将这些信息写入一个文本文件，最后将文件中的内容读出来后显示在屏幕上。

说明：第一行是时间和日期，第二行是星期几，第三行是本月有几天。

|  |
| --- |
| 2021-11-09 21:19:43  Tuesday  There are 30 days in this month. |

#!/usr/bin/python3

# @File: 输出当前时间.py

# --coding: utf-8--

# @Author: 孟德昊

# @Time: 2021年 11月 10日 20:27

'''说明:编写一个程序，获取系统当前的时间等信息，输出格式如下面方框里所示。然后将这些信息写入一个文本文件，最后将文件中的内容读出来后显示在屏幕上。

第一行是时间和日期，第二行是星期几，第三行是本月有几天。

'''

from datetime import datetime

def days(month):

a = [1,3,5,7,8,10,12]

if month in a:

return 31

else:

return 30

if \_\_name\_\_ == "\_\_main\_\_":

date\_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

print(date\_time)

day = datetime.now().strftime("%A")

print(day)

month = datetime.now().strftime("%d")

x = days(month)

print("There are {} days in this month.".format(x))

with open("datetime.txt","w",encoding="utf-8") as dt :

dt.write(date\_time+'\n')

dt.write(day+'\n')

dt.write("There are {} days in this month.".format(x))

图形用户界面, 文本

中度可信度描述已自动生成

图形用户界面, 文本

描述已自动生成

**（1）判断回文**

编写程序，判断用户输入的一个字符串是否为回文。如果是则输出True，否则输出Flase

#!/usr/bin/python3

# @File: 判断回文.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 23日 19:58

'''

说明:编写程序，判断用户输入的一个字符串是否为回文。如果是则输出True，否则输出Flase

'''

if \_\_name\_\_ == "\_\_main\_\_":

while(True):

print("输入#号表示结束")

a = list(input()) #输入一个字符串并将其每个元素分别存入到一个列表中

)

for i in range(int(len(a)/2)): #按输入列表的长度的一半查询是否对称

if(a[i] == a[len(a) - i - 1]): #相等即为回文，否则不为回文。

flag = 1

else :

flag = 0

break

if(1 == flag) :

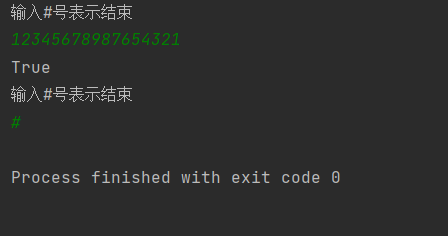
print("True")

else :

print("False")

else:

break

****

**（2）集合及其关系程序**

随机生成10个从0~10区间内（包含0和10）的整数，分别组成集合A、集合B，输出A和B的内容、长度、最大值、最小值，以及A和B的并集、交集和差集

#!/usr/bin/python3

# @File: 集合及其关系程序.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 23日 20:26

'''说明:

'''

import random

if \_\_name\_\_ == "\_\_main\_\_":

A = set()

B = set()

for i in range(10):

A.add(random.randint(0, 11))

for i in range(10):

B.add(random.randint(0, 11))

print(A)

print(len(A))

print(max(A))

print(min(A))

print(B)

print(len(B))

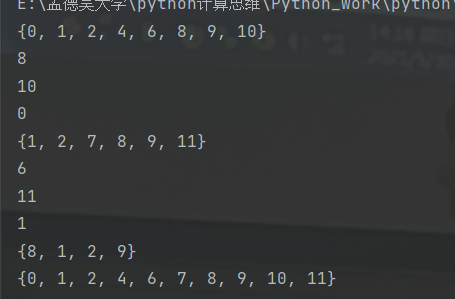
print(max(B))

print(min(B))

print(A&B)

print(A|B)

print(A.difference(B))



**（3）分别用迭代器和生成器函数来输出斐波那契数列的第10项~第20项**

#!/usr/bin/python3

# @File: 斐波那契数列\_迭代法.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 23日 21:06

'''

说明:

'''

if \_\_name\_\_ == "\_\_main\_\_":

li = [1,1]

for i in range(2,100):

a = li[i-1] + li[i-2]

li.append(a)

for j in range(9,20):

print(li[j])

日历

描述已自动生成

#!/usr/bin/python3

# @File: 斐波那契数列\_递归.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 23日 21:21

'''

说明:

'''

def fib(f,l):

li = [1,1]

for i in range(2,l+1):

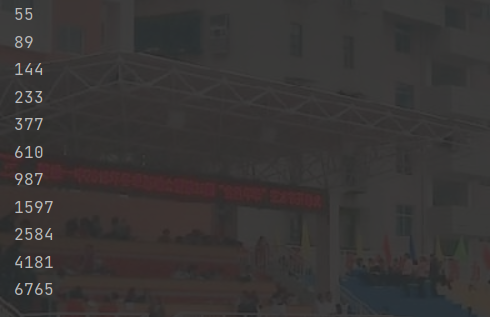
a = li[i - 1] + li[i - 2]

li.append(a)

for i in range(f-1,l):

print(li[i])

if \_\_name\_\_ == "\_\_main\_\_":

fib(10,20) 

**（4）二进制（随机）文件的读写**

1、打开一个空的随机文件my.dat，往里面写入下面2行字节数据：

Xiaoming

student

2、然后读入该文件的后7个字节并输出到屏幕上。

#!/usr/bin/python3

# @File: 二进制（随机）文件的读写.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 23日 22:08

'''

说明:

'''

if \_\_name\_\_ == "\_\_main\_\_":

with open('my.dat','wb') as fp:

fp.write(b'Xiaoming\n')

fp.write(b'student')

with open('my.dat','rb') as fp:

str = fp.read()

print(str[-7:])

图形用户界面, 应用程序

描述已自动生成

图形用户界面, 网站

描述已自动生成

**（5）对文本文件进行统计**

参考例15.14编写一个文本统计程序，利用正则表达式对文本文件“abstract.txt”中读取字符串序列，统计文本文件“abstract.txt”中包含的段落数、行数、句数、单词数，以及统计各单词出现的频率。abstract.txt文件的内容如下面框中所示，要事先创建该文件。

|  |
| --- |
| Cloud services are exploding, and organizations are converging their data centers in order to take advantage of the predictability, continuity, and quality of service delivered by virtualization technologies.  In parallel, energy-efficient and high-security networking is of increasing importance. Network operators, and service and product providers require a new network solution to efficiently tackle the increasing demands of this changing network landscape. Software-defined networking has emerged as an efficient network technology capable of supporting the dynamic nature of future network functions and intelligent applications while lowering operating costs through simplified hardware, software, and management.  In this article, the question of how to achieve a successful carrier grade network with software-defined networking is raised. Specific focus is placed on the challenges of network performance, scalability, security, and interoperability with the proposal of potential solution directions. |

#!/usr/bin/python3

# @File: 对文本文件进行统计.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 23日 22:20

'''

说明:

'''

import re

import collections

def analyze\_text(text):

paragh = re.split("\n\n", text)

paragh\_count = len(paragh)

print("段落数：{}".format(paragh\_count))

lines = re.split("\n", text)

lines\_count = len(lines)

print("行数：{}".format(lines\_count))

sentences = re.split("[.?!]", text)

sentences\_count = len(sentences)

print("句子数：{}".format(sentences\_count))

words = re.split(r"\W+", text)

words\_count = len(words)

print("单词数：{}".format(words\_count))

freqs = collections.Counter(words)

for w,m in freqs.items():

if(w != ''):

print(w,m)

if \_\_name\_\_ == "\_\_main\_\_":

filename = "abstract.txt"

with open(filename, 'r') as fp:

test = fp.read()

analyze\_text(test.strip())

图形用户界面, 应用程序

中度可信度描述已自动生成

**（6）编写程序，分别输入3个字符串，利用正则表达式依次验证其是否为有效的中国国内的电话号码、中国国内的邮政编码，以及正确的网站网址格式。**

输入输出格式自己定义，比如：

请输入中国电话号码：

0511-88789657

这是有效的电话号码格式

#!/usr/bin/python3

# @File: 验证号码格式.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 23日 22:41

'''

说明:

'''

import re

def china\_mobile\_phone(mobile\_num):

if(re.findall("^(\(\d{3}\)|\d{3}-)?\d{8}$",mobile\_num,flags=0) != None):

print("这是有效的电话号码格式")

else:

print("这不是有效的电话号码格式")

def china\_post\_num(post\_num):

if (re.findall("^\d{6}$", post\_num, flags=0) != None):

print("这是有效的邮政编码格式")

else:

print("这不是有效的邮政编码格式")

def web\_num(url\_num):

if (re.findall("^https?://\w+(?:\.[^\.]+)+(?:/.+)\*$", url\_num, flags=0) != None):

print("这是有效的网站网址格式")

else:

print("这不是有效的网站网址格式")

if \_\_name\_\_ == "\_\_main\_\_":

print("请输入要验证的号码格式")

print("1.中国电话号码格式")

print("1.邮政编码")

print("3.网站网址格式")

pro = int(input())

if(1 == pro):

mobile\_num = input("请输入中国电话号码：")

china\_mobile\_phone(mobile\_num)

elif(2 == pro):

post\_num = input("请输入中国国内的邮政编码：")

china\_post\_num(post\_num)

elif(3 == pro):

url\_num = input("请输入网址：")

web\_num(url\_num)

盒子上写着字

低可信度描述已自动生成

**（7）用2个列表来生成字典**

编写程序，输入2个列表，以第一个列表中的元素为“键”，以第二个列表中的元素为“值”创建字典。若两个列表长度不等，则以短的为准而丢弃较长列表中后面的元素。最后输出字典。

#!/usr/bin/python3

# @File: 用2个列表来生成字典.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 23日 23:10

'''

说明:

'''

if \_\_name\_\_ == "\_\_main\_\_":

a = list(input().split(','))

b = list(input().split(','))

c = {}

if(len(a) >= len(b)):

for i in range(len(b)):

c[a[i]] = b[i]

else:

for i in range(len(a)):

c[a[i]] = b[i]

print(c)



1. 编写程序，创建类MyMath，计算圆的周长和面积以及球的表面积和体积，半径等参数由用户输入，编写测试代码，结果均保留两位小数

#!/usr/bin/python3

# @File: 计算圆的周长和面积以及球的表面积和体积.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 30日 18:32

'''

说明:编写程序，创建类MyMath，计算圆的周长和面积以及球的表面积和体积，半径等参数由用户输入，编写测试代码，结果均保留两位小数

'''

class MyMath(object):

def \_\_init\_\_(self,r):

'''

初始化类

'''

self.r = r

self.PI = 3.14

def printcircularperimeter(self):

'''

圆的周长

'''

print("{:.2f}".format(2 \* self.PI \* self.r))

def printcirculararea(self):

'''

圆的面积

'''

print("{:.2f}".format(self.r \* self.r \* self.PI))

def printballSurface\_area(self):

'''

球的表面积

'''

print("{:.2f}".format(4 \* self.PI \* self.r \* self.r))

def printballvolume(self):

'''

球的体积

'''

print("{:.2f}".format(4 \* self.PI \* self.r \* self.r \* self.r / 3))

if \_\_name\_\_ == "\_\_main\_\_":

r = int(input()) #用户输入半径

mymath = MyMath(r) #创建实例化对象

'''

调用类中的函数

'''

mymath.printcircularperimeter()

mymath.printcirculararea()

mymath.printballSurface\_area()

mymath.printballvolume()

模糊的照片

描述已自动生成

2． 定义计算和返回第n阶调和数（1+1/2+1/3+…+1/n）的函数，输出前n个调和数，n由用户输入。结果保留3位小数

#!/usr/bin/python3

# @File: 计算和返回第n阶调和数.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 30日 18:54

'''

说明:定义计算和返回第n阶调和数（1+1/2+1/3+…+1/n）的函数，输出前n个调和数，n由用户输入。结果保留3位小数

'''

sum = 0

def Harmonic\_number(n):

global sum

if n != 1 :

sum += 1 / n

Harmonic\_number(n-1)

elif n == 1:

sum += 1

print("{:.3f}".format(sum))

if \_\_name\_\_ == "\_\_main\_\_":

n = int(input())

Harmonic\_number(n)

图片包含 标志, 仪表, 街道, 钟表

描述已自动生成

3. 随机产生10个两位的正整数，存入列表ls中，然后列表中的数进行排序（由小到大），最后分别输出：排序后的列表，这10个数的平均数，以及高于平均值的数的个数。

#!/usr/bin/python3

# @File: 排序均值.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 30日 19:02

'''

说明: 随机产生10个两位的正整数，存入列表ls中，然后列表中的数进行排序（由小到大），最后分别输出：排序后的列表，这10个数的平均数，以及高于平均值的数的个数。

'''

# 导入随机包

from random import randint

ls = [] # 创建全局变量

def makerandommath():

"""

随机生成十个二位正整数放入列表中

"""

global ls

for i in range(10):

a = randint(10, 99)

ls.append(a)

def sortedlist():

"""

排序列表内的数值

"""

global ls

ls.sort(reverse=False)

# print(ls)

def makeaverage():

"""

求列表内数的平均数

"""

sum = 0

for i in range(len(ls)):

sum += ls[i]

average = sum / (i + 1)

print(average)

return average

def countbigger(average):

"""

求列表中大于平均数的数的个数

"""

count = 0

for i in range(len(ls)):

if average < ls[i]:

count += 1

print(count)

if \_\_name\_\_ == "\_\_main\_\_":

makerandommath()

sortedlist()

average = makeaverage()

countbigger(average)

手机屏幕的截图

中度可信度描述已自动生成

4. 编写一个程序，接收一系列单个空格分隔的单词作为输入，在删除所有重复的单词并按字母升序排序后打印这些单词。

假设向程序提供以下输入:

hello world and practice makes perfect and hello world again

则输出为:

again and hello makes perfect practice world

#!/usr/bin/python3

# @File: 排序打印单词.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 30日 19:22

"""

说明:

"""

def setup\_letter\_list(ls1):

"""

设置列表内不重复且不包含空格的字母

"""

ls2 = []

for i in ls1:

if i not in ls2 and i != ' ':

ls2.append(i)

return ls2

def sort\_list(ls4):

"""

排序并输出

"""

ls4.sort()

for i in ls4 :

print("{}".format(i),end=" ")

if \_\_name\_\_ == "\_\_main\_\_":

# 输入列表以空格分开每个单词

ls = list(input().split(" "))

ls3 = setup\_letter\_list(ls)

sort\_list(ls3)

图形用户界面, 文本

描述已自动生成

5. 企业发放的奖金根据利润提成。利润(I)低于或等于10万元时，奖金可提10%；利润高于10万元，低于20万元时，低于10万元的部分按10%提成，高于10万元的部分，可可提成7.5%；20万到40万之间时，高于20万元的部分，可提成5%；40万到60万之间时高于40万元的部分，可提成3%；60万到100万之间时，高于60万元的部分，可提成1.5%，高于100万元时，超过100万元的部分按1%提成，从键盘输入当月利润I，求应发放奖金总数？

if \_\_name\_\_ == "\_\_main\_\_":

prince = 0

i = float(input())

if i <= 100000:

prince += i \* 0.1

elif i > 100000 and i <= 200000:

prince = (i - 100000) \* 0.075 + 100000 \* 0.1

elif i > 200000 and i <= 400000:

prince = (i - 200000) \* 0.05 + 100000 \* 0.075 + 100000 \* 0.1

elif i > 400000 and i <= 600000:

prince = (i - 400000) \* 0.03 + 200000 \* 0.05 + 100000 \* 0.075 + 100000 \* 0.1

elif i > 600000 and i <= 1000000:

prince = (i - 600000) \* 0.015 + 200000 \* 0.03 + 200000 \* 0.05 + 100000 \* 0.075 + 100000 \* 0.1

else:

prince = (i - 1000000) \* 0.01 + 400000 \* 0.015 + 200000 \* 0.03 + 200000 \* 0.05 + 100000 \* 0.075 + 100000 \* 0.1

print(prince)

张着嘴

中度可信度描述已自动生成

6. 求s=a+aa+aaa+aaaa+aa...a的值，其中a是一个数字。例如2+22+222+2222+22222(此时共有5个数相加)，几个数相加由键盘控制。

#!/usr/bin/python3

# @File: 求值.py

# --coding: utf-8--

# @Author: 昊昊反思

# @Time: 2021年 11月 30日 19:46

"""

说明:

"""

from math import pow

if \_\_name\_\_ == "\_\_main\_\_":

"""

输入a值和n值

"""

a = int(input())

n = int(input())

ls = [] # 创建空列表

sum = 0 # 和为0

for i in range(0,n):

math = 0 # 每个值初值为0

for j in range(0,i+1): # 循环是math = a,aa,aaa,aaaa,……

math = math + pow(10,j) \* a

# print(math)

ls.append(math) # 将值加入列表

for i in ls:

sum += i # 求和

print(sum)

图形用户界面

中度可信度描述已自动生成