Python实验内容备份

实验一

（1）编写一个输出小于N的所有素数

# -\*- coding: utf-8 -\*-

# ----------------begin---------------------#

num1='da'

while True:

num1=input("N=")

if num1.isnumeric() and int(num1)>2:

break

print("error!input again:")

def functionname( num ):

lis=[]

while(num>1):

i=num-1

boo=True

while i>1:

if num%i==0:

boo=False

break

i=i-1

if boo or num==2:

lis.insert(0,num)

num=num-1

return lis

lis=functionname(int(num1))

num1=1

he=0

avge=0

for x in lis:

print(x, end=" ")

if num1%8==0:

print('')

num1=num1+1

he=he+x

print('')

avge=he/len(lis)

print("sum=%d,average=%.2f" % (he ,avge))

（2）本关任务：输出小于任意给定的正整数N(N>20)的所有完美数

所谓完美数是：若大于0且小于正整数M的所有能整除M的因子的和等于M，则M为完美数，例如：6=1+2+3，因此，6是完美数

N=int(input('N='))

#----------begin----------

i=2

st=[]

for i in range(2,N):

sum=0

st=[]

for j in range(1,int(pow(i,0.5)+1) ):

if i%j==0:

st.append(j)

st.append(int(i/j))

sum=sum+j+int(i/j)

if(sum>i\*2):break

st.sort()

del st[len(st)-1]

if(sum==i\*2):

print (i,end='=')

print("+".join(str(j) for j in st))

（3）本关任务：学校举行校园歌手大赛，请了N个评委给选手打分，你负责设计一个程序：依次输入N个评委的评分，去掉1个最高分和1个最低分，以其余平均分作为选手最终得分。

# -\*- coding: utf-8 -\*-

#----------begin----------

import re

j=0

st=[]

st2=[]

score=0

while(j<=5):

st=re.split(',| |，',input("input score:"))

st2=[]

j=0

for i in st:

if not i.isalpha():

if float(i)>0 and float(i)<=10:

j=j+1

st2.append(float(i))

print("valid scores",end=':')

print(st2)

max1=max(st2)

min1=min(st2)

sum1=0

for i in st2:

sum1=sum1+i

score=(sum1-max1-min1)/(len(st2)-2)

print('max',end='=')

print(max1,end=',')

print('min',end='=')

print(min1,end=',')

print("score=%.3f" % score)

实验二：

给定一个整数N（N>=10）,从1开始，输出元素个数为N的斐波那契数列

# -\*- coding: utf-8 -\*-

#----------请在下面给出主程序调用的几个函数的定义--------

def getint():

while(True):

data=input("input a positive int:")

if(data.isnumeric() and int(data)>0):

return int(data)

break

def fibdata(numb=10):

list1=[]

i=1

j=1

while(numb>0):

list1.append(i)

t=i+j

i=j

j=t

numb=numb-1

return list1

def outputdata(data,numb2=5):

num=len(data)#列表总个数

for i in range(num):

print(data[i],end=" ")

if (i+1)%numb2==0:

print("")

文件：

import os

import re

import csv

class fileLine:

def \_\_init\_\_(self, path, rl):

fileList = self.printPath(path, rl)

self.findLine(path, fileList)

def printPath(self, path, rl):

fileList = []

files = os.listdir(path)

for f in files:

if os.path.isfile(path + '/' + f):

for i in rl:

i = ".\*" + i

for j in re.findall(i, f):

if re.findall(j, f) != "":

fileList.append(f)

break

return fileList

def findLine(self, path, fileList):

csvFile = open("C:\yuybl\kaifa\python\wenjian\pythonstudy\pyt.csv", 'w', encoding='utf-8') #创建csv文件

writer = csv.writer(csvFile) #创建写的对象

#先写入columns\_name

#writer.writerow(["index","a\_name","b\_name"]) #写入列的名称

#写入多行用writerows #写入多行

#writer.writerows([[1,a,b],[2,c,d],[3,d,e]])

with open("C:\yuybl\kaifa\python\wenjian\pythonstudy\pyt.csv", 'w', encoding='utf-8')as fp\_w:

for i in fileList:

i = path+"\\"+i

fh = open(i, encoding='UTF-8')

read\_fh = fh.readlines()

fh.close()

number\_code = 0

number\_empty = 0

number\_note = 0

pattern = '.\*#' # 正则匹配模式

for x in read\_fh:

if '#' in x: # 计算注释数目

if re.findall(pattern, x)[0][:-1].isspace() or re.findall(pattern, x)[0][:-1] == '':

number\_note += 1

else:

number\_code += 1

elif x.isspace():

number\_empty += 1

else:

number\_code += 1

writer.writerow([str(i),str(number\_code+ number\_note)])#'{0}，{1}\n'.format(i, (number\_code+ number\_note))

print(i, "，" , (number\_code+ number\_note))

csvFile.close()

if \_\_name\_\_ == '\_\_main\_\_':

# url = str(input("请输入路径"))

url = "C:\yuybl\kaifa\python\wenjian\pythonstudy"

rl = ['py','txt']

# rl = list(input("请输入文件扩展名").split(" "))

fileLine(path=url, rl=rl)

第2关：类的定义与使用：一元二次方程求解

# -\*- coding:utf-8 -\*-

#----------在下面添加类定义代码-----------

class QuadraticEquation:

def \_\_init\_\_(self,a=1,b=1,c=1):

self.a=a

self.b=b

self.c=c

def getroot(self):

dict1={}

panduan=self.b\*\*2-4\*self.a\*self.c

dict1['x1']=(-self.b+panduan\*\*0.5)/(2\*self.a)

dict1['x2']=(-self.b-panduan\*\*0.5)/(2\*self.a)

if panduan >0:

dict1['state']=1

elif panduan==0:

dict1['state']=0

else:

dict1['state']=2

return dict1

def outputresult(self,result):

print("{}x+{}x\*\*2+{}=0".format(self.a,self.b,self.c),end=" ")

if result['state']==0:

print("has two equal roots x1=x2=%.3f" %result['x1'])

elif result['state']==1:

print("has two inequal real roots:x1=%.3f,x2=%.3f" %(result['x1'],result['x2']))

elif result['state']==2:

print("has two complex roots x1={:.3f},x2={:.3f}".format(result['x1'],result['x2']))

### 实验四：第1关：数学库numpy的使用

编写一个程序：输入任意一个矩阵，求它的转置矩阵，并判断其是否为奇异矩阵，若不是，则输出其行列式和逆矩阵。

import numpy as np

#a = np.array([[1,-1,3.4],[4.5,6,1],[-10,3,2]])[[1,-1,3.4],[4.5,6,1],[-10,3,2]]

import re

a=input("请输入一个N×N矩阵:")

a=a.split("[[")[1].split("]]")[0]

a=a.split("],[")

c=[]

for i in a:

i=re.split(",", i)

d=[]

for j in i:

if(float(j)%1==0):

d.append(int(j))

else:

d.append(float(j))

c.append(d)

c=np.array(c)

print("该矩阵的转置矩阵为:")

print(c.T)

b=np.linalg.det(c)

if(b==0):

print("这是奇异矩阵")

else:

print("该矩阵的行列式为:%.2f" % b)

print("该矩阵的逆矩阵为：")

print(np.linalg.inv(c))