**R8-1 csv文件基本操作 (30 分)**

附件: [scores.xls](https://images.ptausercontent.com/6a7ebc7d-d46d-4b58-883f-2690c18a5da5.xls)

请在D盘用自己的真实学号建立文件夹，如：d:\2019001\，并将附件下载到该文件夹内，然后使用该文件生成对应的CSV格式数据文件：学号.csv (用自己的学号，以下简称csv文件)，完成下面操作：

**1.编写代码一：** 请将下方数据添加到你生成的文件.csv尾部(不含标题行)，使用input逐行录入数据；然后将文件结果打印到窗口，观察文件内容是否正确；

程序代码提示： #=======导入csv类模块=============

#=======打开文件，准备写入数据=============

#=======创建csv对象=============

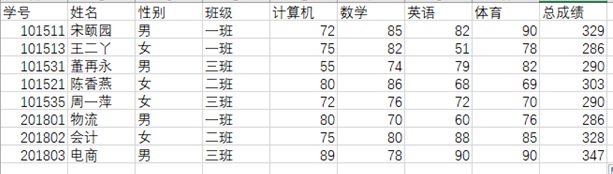
#=======利用input获得一行或多行数据=============

#=======写入一行或多行数据=============

#=======将光标移至文件开始或者重新打开文件进行读取并输出结果=============

**2.编写代码二**（在上面代码后面添加下方功能）：

编写程序实现功能：读取上一步操作成功后的csv文件内容，写入到一个新建的scoreresult.csv文件中，并且在每个学生最后增加一列“总成绩”，计算出每个学生的成绩总分。参考如下图示：



参考程序算法： #=======打开学号.cvs文件和新建scoreresult.csv=============

#=======生成csv.eader对象,并读取表标题=============

#=======将上面标题内容读取后，尾部增加‘总成绩’，并写入到scoreresult.csv=============

#=======利用循环读取或一次性读取所有数据=============

#=======获取每行后面四项数据，进行求和后，添加到每行末端，写入文件scoreresult.csv===

#=======屏幕打印文件scoreresult.csv内容，检查是否与图示雷同=============

import csv

with open(r"D:\2020263055\2020263055.csv","a+",newline="") as f1:

fwrite=csv.writer(f1)

for i in range (3):

x = input().split(",")

fwrite.writerow(x)

f1.close()

with open(r"D:\2020263055\2020263055.csv","r") as f2,open(r"D:\2020263055\scoreresult.csv","w") as f3:

fread2=csv.reader(f2)

fwrite3=csv.writer(f3)

fread2=list(fread2)

x = fread2[0]+["总成绩"]

fwrite3.writeow(x)

for j in fread2[1:]:

sumx=int(j[4])+int(j[5])+int(j[6])+int(j[7])

j.append(int(sumx))

fwite3.writerow(j)

**R8-2 文件编程7-4 (10 分)**

7-4. 用水量文件"water.txt"的第一列为账号，下面是每个月的用水量（后一个数-前一个数），共十二个月。每立方米需付1.05元。编程计算每户一年的水费。@ 。程序压缩后（zip格式）以文件形式上传！

0000359333 772 789 806 847 880 901 950 991 1022 1043 1064 1089 1114

0000359305 121 132 145 156 168 179 192 206 219 230 246 258 273

0000359708 1008 1046 1102 1167 1209 1255 1311 1362 1407 1453 1512 1563 1604

0000359504 541 567 590 622 651 689 701 732 758 775 796 814 847

0000359209 401 412 441 466 479 490 508 522 541 572 603 637 666。

score=open(&apos;water.txt&apos;,&apos;r&apos;)  
  
for i in score.readlines():  
    i=i[:-1]  
    i=i.split()  
    pay=float(&apos;%.2f&apos;%((int(i[-1])-int(i[1]))\*1.05))   
    print(&apos;{}:{:.2f}&apos;.format(i[0],pay))  
  
score.close()

**R8-1 复制文件 (30 分)**

[三国演义1-10.docx](https://images.ptausercontent.com/efe433a2-9234-4655-a692-50fbc044f6dc.docx)

将上面附件【三国演义1-10.docx】下载到本地硬盘后将内容复制到一个新建文本文件中（文件名:sanguo.txt），并将该文本文件作为源文件，请设计三个函数操作该文本文件实现后面提示功能：

函数1：funsta（文件名） #统计指定文件信息函数2：findname(人物名,文件名) #统计指定人物出场次数函数3：funcopy（源文件，目标文件）#能将源文件全部内容读取后，实现复制到目标文件的功能：【为防止编码出错，打开时文件统一使用utf8编码格式：open(文件名,读或写模式,encoding='UTF-8')】

1.**函数1**：统计出源文件的三项数据：行数、字符数量（计空格等）和字符数量（不计空格和换行符）；（完成后可用word统计功能进行核实） def funsta (filename):

**return** columnx,charx,chary

2.**函数2**：统计某人物（如：张飞）在文件中出现的次数； （提示：名字位置可能出现：一个部分位于某行尾部和剩余部分在下一行头部的情况：如张\n 飞，也需要统计）@ def findname (name,filename):

**return** statisticname

3.**函数3**：将源文件读取后写入到到目标文件中。@

（两个文件的路径及文件名由主程序通过input（）函数输入,如：d:\user\sanguotext.txt (仅供参考)） def funcopy (sourcefile，destfile):

print ("filecopy success!")

4.**主程序调用以上三个函数实现对应功能** 【提交代码时，请点击提交框上方的”插入代码</>“按钮，将代码置于按钮提示范围内提交】

import csv

def funsta (filename):

f1 = open("D:\Python平时学习\sanguo.txt","r", encoding='UTF-8')

columnx = 0

for line in f1:

line = line.strip('\n')

if len(line) == 0:

continue

columnx += 1

f1s = str.f1

charx = []

for i in f1s:

for j in charx:

if i != j:

charx.append(i)

list\_f1= [line.strip('\n') for line in f1]

chary = []

for line in list\_f1:

list\_split = line.split()

list\_str = ''.join(list\_split)

chary.append(list\_str)

return columnx,charx,chary

**R8-2 文本文件-合并两个文本文件内容到新文件 (10 分)**

假定后面附件文件：

1. a.txt和b.txt的行数完全相同;
2. 设计程序将文件a.txt的第1行作为新建文件c.txt的第1行，文件b.txt的第1行作为新建文件c.txt的第2行，文件a.txt的第2行作为新建文件c.txt的第3行，文件b.txt的第2行作为新建文件c.txt的第4行，依此规律完成两个文件内容的合并。
3. 最后打开文件c.txt读取内容并打印

提示：先下载文件a.docx和b.docx到本地磁盘，分别打开后复制内容到新建的a.txt和b.txt文件中；再设计Python程序进行文件的合并及打印结果到屏幕。

[a.docx](https://images.ptausercontent.com/64ae51e5-8d99-4993-91a4-7bd1787b682d.docx)

[b.docx](https://images.ptausercontent.com/c8759e8b-03f6-48e7-b7c2-82a6ebb91998.docx)

import csv

with open(r"D:\Python平时学习\a.txt","r",encoding="UTF-8")as f1,open(r"D:\Python平时学习\b.txt","r",encoding="UTF-8")as f2,open(r"D:\Python平时学习\c.txt","a",encoding="UTF-8")as f3:

fread1 = csv.reader(f1)

fread2 = csv.reader(f2)

fwrite = csv.writer(f3)

fread1 = list(fread1)

fread2 = list(fread2)

fwrite.writerow(fread1[0])

fwrite.writerow(fread2[0])

fwrite.writerow(fread1[1])

fwrite.writerow(fread2[1])

fwrite.writerow(fread1[2])

fwrite.writerow(fread2[2])

with open(r"D:\Python平时学习\c.txt","r",encoding="UTF-8")as f4:

chall=f4.read()

print(chall)

**R8-1 文件读取 (10 分)**

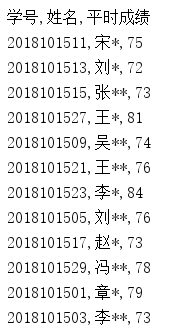
**请用你的真实学号建立：学号.py ，以实现以下功能：**

下方附件-文件1是存放的某班数学课平时作业记录，最后一项为平时总成绩，请下载到本地系统D盘，将下载的文件内容复制到新建的记事本文件：文件1.txt 文件中，然后程序中实现：

将文件1.txt内容读出，为了便于给学生发布成绩，仅仅保留学号,姓名,平时成绩三项写入到新建的文件2中，应当在文件2中和屏幕输出都达到下方输出图示效果；

[文件1.docx](https://images.ptausercontent.com/c0d59d39-1203-4d93-b29a-c84ca056bb14.docx)

请参见输出样例（文件2存储的内容）



with open(r'D:\文件1.txt','r+',encoding='UTF-8')as f1,open(r'D:\文件2.txt','w+',encoding='UTF-8')as f2:

x=f1.readlines()

for i in x :

i=i.split(',')

n=i[0]+','+i[1]+','+i[-1]

print(n,file=f2)

print(n)

**R8-2 文件创建 (10 分)**

\*\* 文件的创建和使用\*\*

1.创建一个文本文件，文件名为学号，按下方格式通过input()读取个人真实信息并将内容写入文件中，具体样式请参照样例图片：

请输入学号

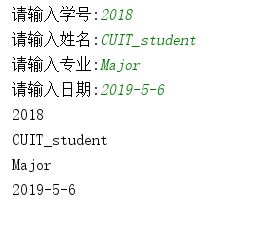
请输入姓名

请输入专业

请输入日期(日期格式：年-月-日)

2.再次打开这个文件，读取文件的所有内容并在屏幕显示后退出。

程序运行过程及结果参照下图：



with open(r'D:\2020263055.txt','r+',encoding='UTF-8') as file:

x = input("请输入学号:")

y = input("请输入姓名:")

m = input("请输入专业:")

n = input("请输入日期:")

print(x)

print(y)

print(m)

print(n)

file.close()

**R7-1 GPA计算 (10 分)**

根据全球学校通用的学分绩点（即GPA，Grade Point Average）规则,请创建一个程序统计该学生的GPA，设计思路：

1.设计一个类student，使其能具备学生基本信息和方法，数据如下：

学生基本属性（成员变量）应具有三个：学号，姓名(均为字符类型)，gpa（为浮点数）；

设计的成员方法：能返回学生姓名的getName()、获得学生基本信息的getInfo()、计算gpa的getGPA()等三个基本方法（仅供参考）

2.定义一个函数用于实现输入一系列课程的课程信息（chengji 和xuefen）；

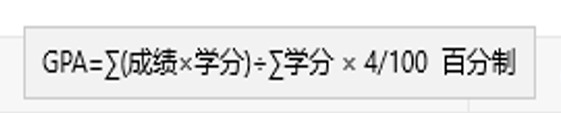
3.你的主程序应该创建一个新的学生对象，具有0 个积分点；

4.然后在主程序中调用对象的方法或你设计的函数统计课程信息并打印出该学生最终得到的GPA(精确到小数点2位)

参考下面格式存储一个学生对象的数据，注意成绩是整数，学分为浮点数：

[[学号，姓名],[课程1成绩，课程1学分],[课程2成绩，课程2学分],[课程3成绩，课程3学分]...]

GPA计算公式参考全球高校采纳的通用公式：



。 请复制下方代码，根据题目要求和下方信息提示，在空白处填写代码实现程序功能。

**class** **student**:

**def** **\_\_init\_\_**(self, stuid, name, gpa): *#类的构造函数，请根据前后代码提示在后面完善该部分代码*

self.stuid = stuid

**def** **getName**(self):

**return** self.name

**def** **getgpa**(self, courselist): *#请根据题目GPA公式，空白处计算GPA，也可以删掉这部分代码自行设计*

self.gpa = a / b \* 4 / 100

**def** **makeStudent**(x, y): *#该函数用于实现某学生对象的课程信息录入，不属于student类的方法（函数）*

*# 循环依次录入每门课程的分数及学分，一行输入一门课程的信息，用一个空格隔开；输入"-1 0"时，结束课程录入*

strcourse = [[x, y]]

**return** strcourse

idx, namex = input().split(' ')

*#此处定义一个新的学生对象*

list1 = makeStudent(idx, namex)

studentx.getgpa(list1)

print('The student is:{} {}'.format(studentx.stuid, studentx.name))

print('The GPA is:{:0.2f}'.format(studentx.gpa))

输入格式:

第一行输入学生学号及姓名注意使用一个空格隔开。 然后直接输入课程分数和学分，也是用一个空格隔开两个值（一行是一个课程的信息） ..... 直到准备结束录入课程信息时，就输入：-1 0

输出格式:

第一行输出学生对象信息学号和姓名。 第二行输出GPA分数。 务必参考样例

输入样例:

在这里给出一组输入。例如：

2018001 zhangsan

90 4

80 3.5

70 2

-1 0

输出样例:

上面的输入应得到：

The student is:2018001 zhangsan

The GPA is:3.28

class student:

def \_\_init\_\_(self, stuid, name, gpa):

self.stuid = stuid

self.name=name

self.gpa=gpa

def getname(self):

return self.name

def getName(self):

return self.stuid,self.name

def getgpa(self, courselist):

a=b=0

for i in courselist:

a+=(eval(i[0])\*eval(i[1]))

b+=eval(i[1])

self.gpa = a / b \* 4 / 100

return self.gpa

def makeStudent(x, y):

strcourse=[]

while True:

x,y=input().split()

if x=='-1'and y=='0':

break

else:

strcourse += [[x, y]]

return strcourse

idx, namex = input().split(' ')

studentx=student(idx,namex,0.0)

list1 = makeStudent(idx, namex)

studentx.getgpa(list1)

print('The student is:{} {}'.format(studentx.stuid, studentx.name))

print('The GPA is:{:0.2f}'.format(studentx.gpa))