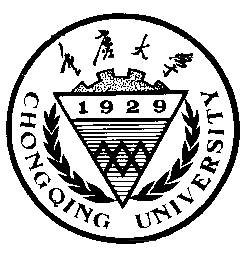
数据科学导论实验报告

实验一 python数据基础实践



学 生：严奕凡\_20220815

年 级：2022级

专 业：数据科学与大数据技术

重庆大学大数据与软件学院

2024年 5月 7日

## 实验目的：

为后续上机实验做准备，掌握python 基础操作：包括熟悉python 程序运行环

境及基本语法，理解python 基本数据类型，掌握python 的算术运算规则及表达

式的书写方法；掌握常用的python 基本数据函数分析库：pandas、numpy.

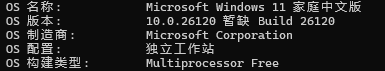
## 实验要求

1. Python基础操作
2. Python基本函数分析库

2.1、熟悉pandas操作

2.2、熟悉Numpy操作

## 开发环境：

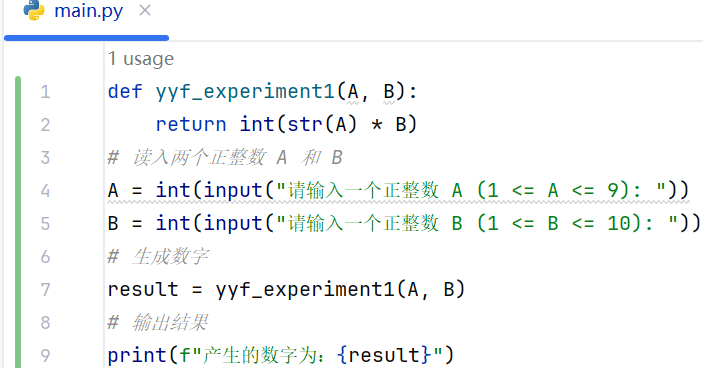


## 实验内容：

**Python 基础操作**

(1)读入2 个正整数A 和B，1<=A<=9, 1<=B<=10,产生数字AA…A,一共B

个A(例如输入2 和3，输出为222)



(2)将字符串str = "hElLo pYtHOn "做以下变换：

1)把所有字符中的小写字母转换成大写字母；

2)把所有字符中的大写字母转换成小写字母；

3)把第一个字母转化为大写字母，其余小写；

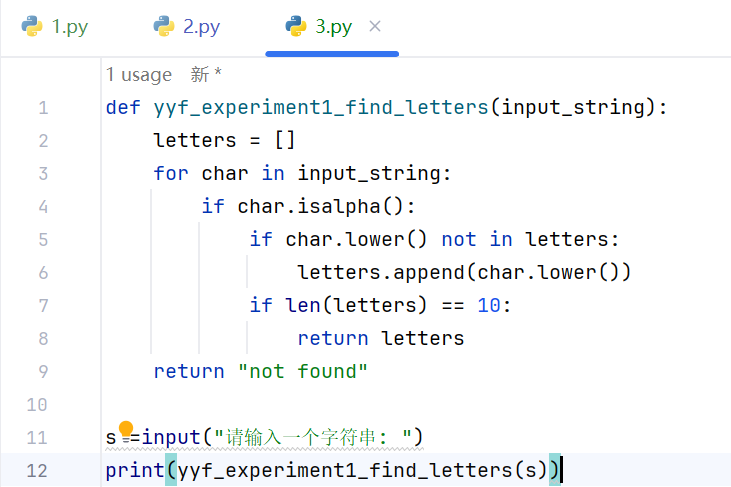
4)把每个单词的第一个字母转化为大写，其余小写



(3) 输出10 个不重复的英文字母(字符串转列表实现)：随机输入一个字符

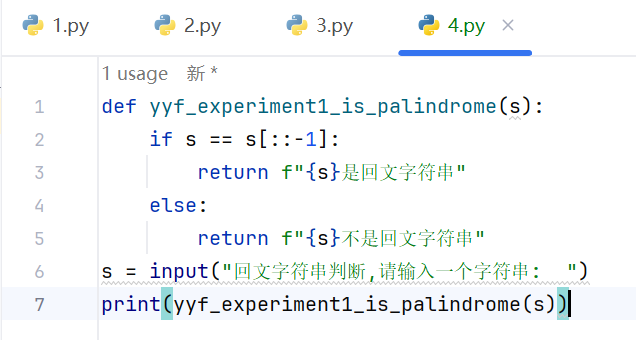
串，把最左边的10 个不重复的英文字母(不区分大小写)挑选出来。如

没有10 个英文字母，显示信息“not found”



(4)给定一个字符串，判断它是否是回文字符串(即类似于level, 12321 这样

的对称字符串)，如果是输出回文及YES，不是则输出原始字符串及NO。



(5)小明身高1.75，体重80.5kg。请根据BMI 公式(体重除以身高的平方)

帮小明计算他的BMI 指数，并根据BMI 指数：

低于18.5： 过轻

18.5-25： 正常

25-28： 过重

28-32： 肥胖

高于32： 严重肥胖

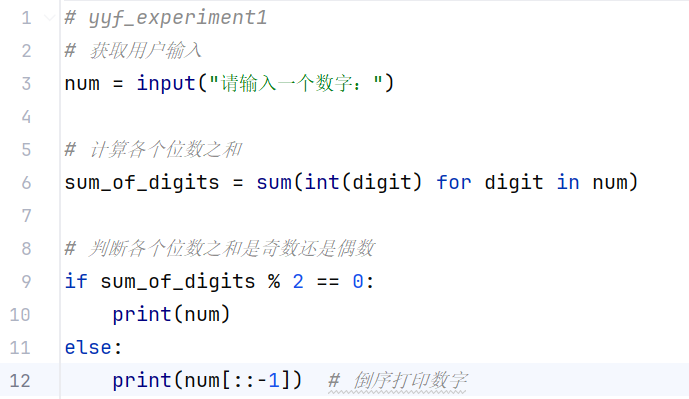
用if-elif 判断并打印结果：



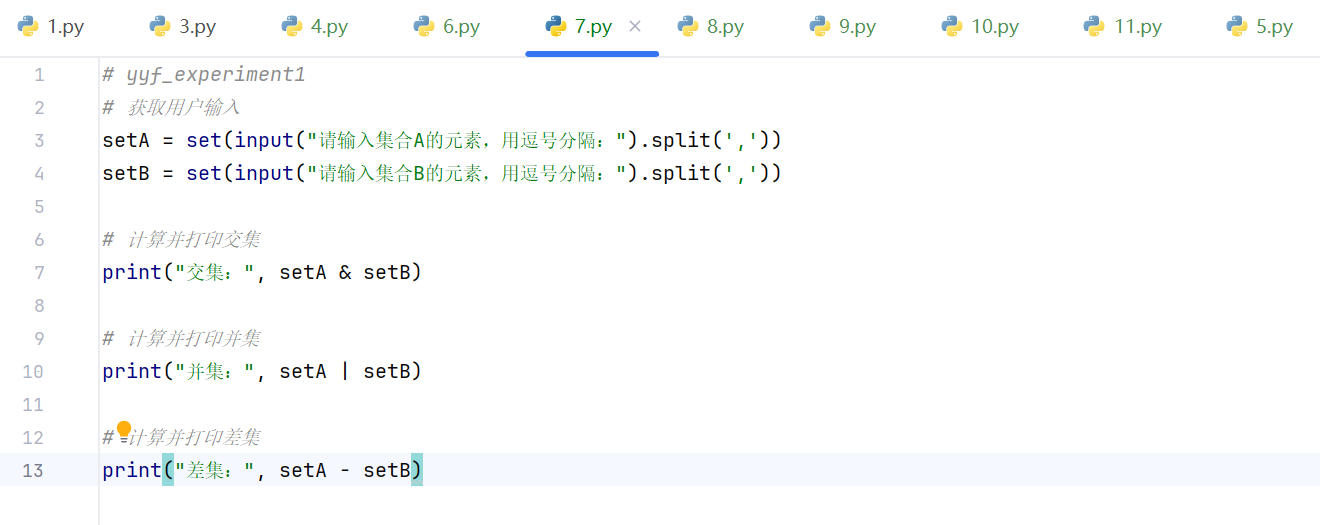
(6)输入一个数字：

1)若该数字各个位数之和为奇数，则将该数各个位数倒叙打印(如122，打印为221)。

2)若该数字个位数之和为偶数，则直接打印该数字。



(7)编写程序，输入两个集合setA 和setB,分别输出它们的交集，并集和差集

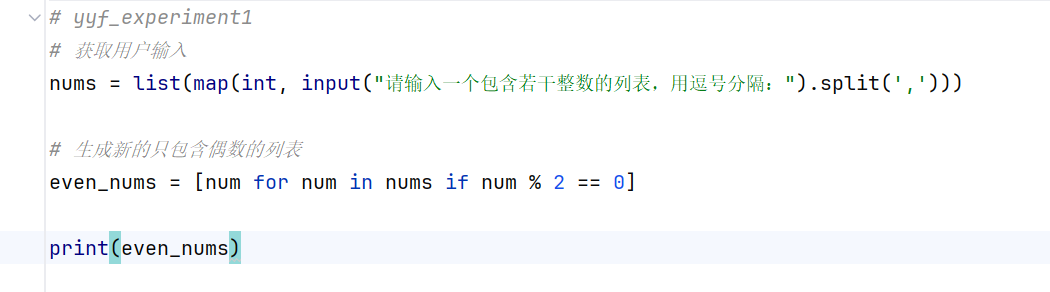


(8)编写程序：输入一个，输出它的二进制，八进制，十六进制表示形式



（9)编写程序：输入一个包含若干整数的列表，输出一个新列表，要求新列

表中只包含原列表的偶数(如请输入一个列表：[1,2,3,4,5,6]输出[2, 4, 6])



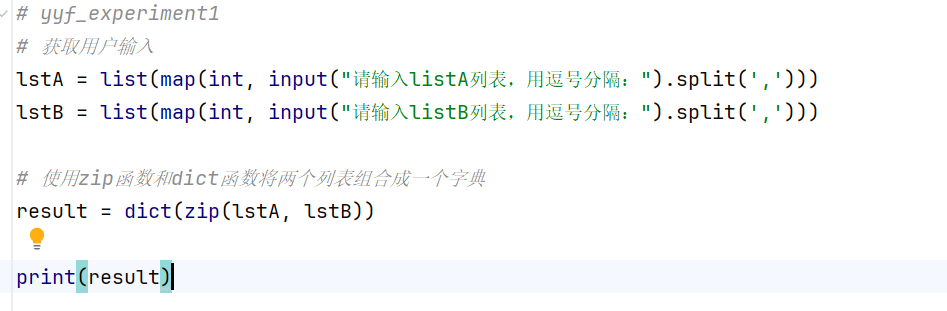
（10)编写程序，输入两个分别包含若干整数的列表lstA 和lstB,输出一个字

典，要求使用列表lstA 中的元素作为键，列表lstB 中的元素作为值，并且

最终字典中的元素数量取决于lstA 和lstB 中元素最少的列表的数量。（如

请输入listA 列表：[1,2,4,6]；请输入listB 列表：[3,5,6,8]输出{1: 3, 2: 5, 4:

6, 6: 8})

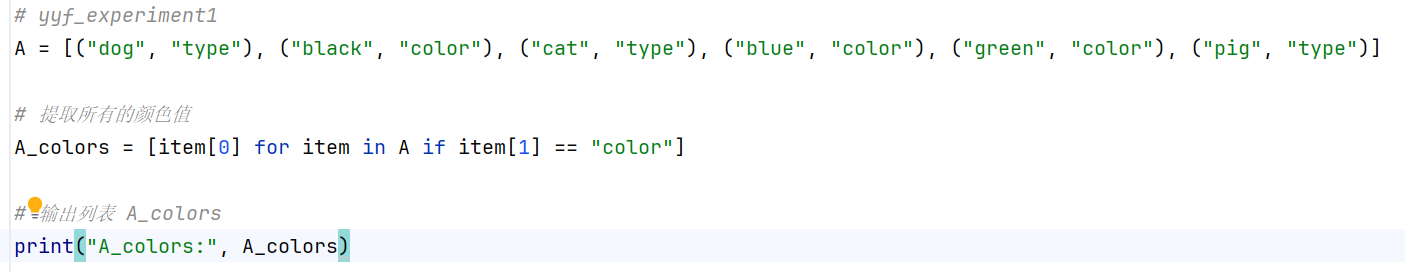


（11)假设已有列表A=[(“dog”,“type”),(“black”, “color”),(“cat”,“type”),(“blue”,

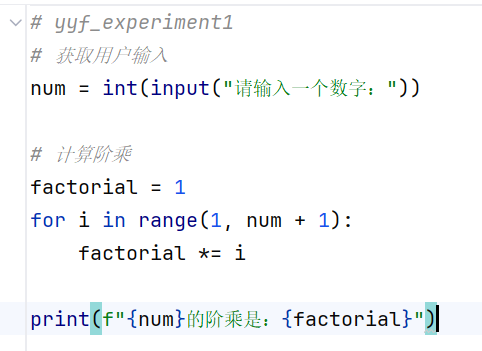
“color”),(“green”,“color”),(“pig”,“type”)]，其中每个元素都是一个元组。元

组中的第一个元素表示值，第二个元素表示标签。试编写程序，将所有的

颜色值从列表A 中提取出来，存入列表A\_colors,并将该列表输出。



（12)输入数字计算阶乘：整数的阶乘（英语：factorial)是所有小于及等于

该数的正整数的积，0 的阶乘为1。即：n!=1×2×3×...×n。

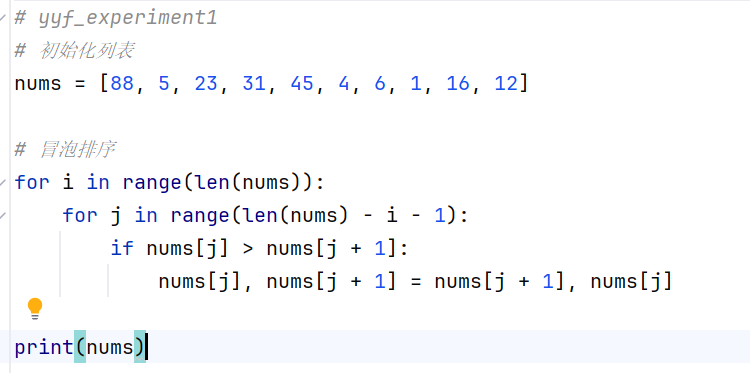
（13)编程实现：输入的数字是否为阿姆斯特朗数。（如果一个n 位正整数等

于其各位数字的n 次方之和,则称该数为阿姆斯特朗数。例如1^3 + 5^3 +

3^3 = 153。1000 以内的阿姆斯特朗数： 1, 2, 3, 4, 5, 6, 7, 8, 9, 153, 370, 371,

407。)

（14)对以下元素88,5,23,31,45,4,6,1,16,12 进行冒泡排序



（15)编程实现五人分鱼：A、B、C、D、E 五人在某天夜里合伙去捕鱼，到

第二天凌晨时都疲惫不堪，于是各自找地方睡觉。日上三杆，A 第一个醒

来，他将鱼分为五份，把多余的一条鱼扔掉，拿走自己的一份。B 第二个

醒来，也将鱼分为五份，把多余的一条鱼扔掉拿走自己的一份。C、D、

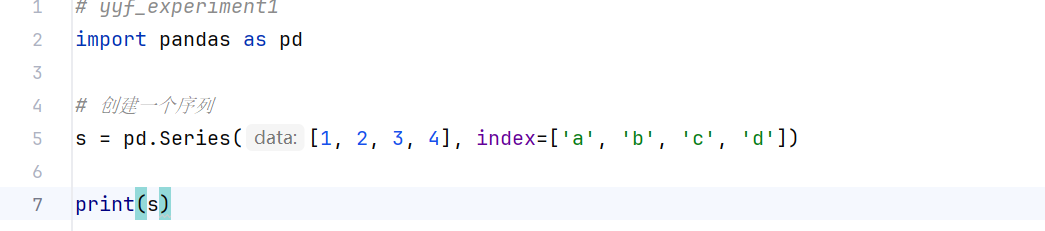
E 依次醒来，也按同样的方法拿鱼。问他们至少捕了多少条鱼?



**Python 基本函数分析库**

（一) 熟悉常用的pandas 操作

(1) 用pandas 中Series 创建一个序列s 为[1,2,3,4],标签为[a,b,c,d]

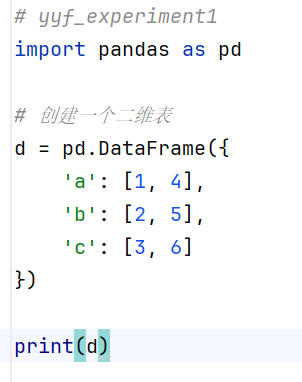


(2) 用DataFrame 创建一个二维表d

a b c

1 2 3

4 5 6



;也可用已有的序列来创建表格



(3) 用pandas 读取文件ocean\_temp.csv,命名为df，注意:文件的存储路径不能带有中文，否则读取可能出错; 读取文本格式的数据，一般用encoding指定编码utf-8。(4) 预览df 前5 行数据(5) 统计df 数据基本统计量(count, mean, std, min, quantiles, max)(6) 在df 的基础上创建一个从10 到50 行（包含10 行和50 行)的片段df\_slice，并输出。

（二)熟悉常用的Numpy 操作

(1) 从列表df 中创建一个名为tempArr 的NumPy 数组（ndarray),然后输出

tempArr 的大小和类型。



(2) 创建另一个与tempArr 大小相同的数组adjar，但这次所有元素都设置为20

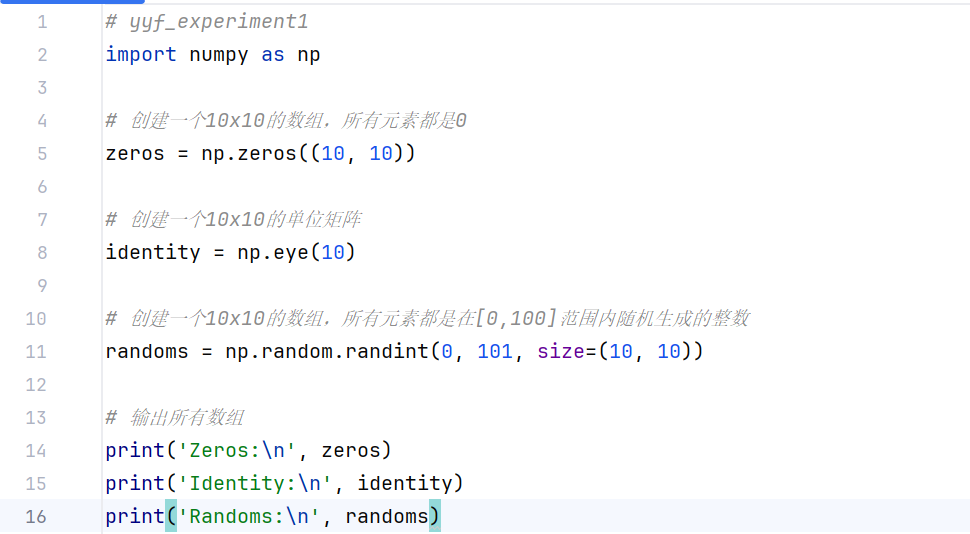
(3) 将上述两个数组添加到一个新的数组中，新数组命名为newTemp(4) 在newTemp 中的温度是华氏度。定义一个函数，将温度从华氏度转换为摄氏度，然后使用矢量化函数将newTemp 中的所有温度数据转换为摄氏度，

并存储在新数组cTemp 中。华氏度转摄氏度公式如下： C=(5/9)\*(F-32)(5) 显示上述所有数组的前50 个元素(6) 显示上述cTemp 数据的20 到50 行



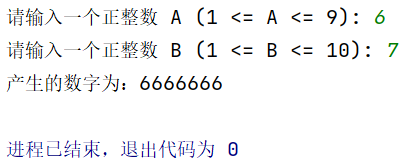
(7) 创建基本的10x10 数组：

1)所有元素都是0；2)I 矩阵（即对角线0 上的值1)；3)所有元素都是在[0,100]范围内随机生成的整数。输出所有数组。

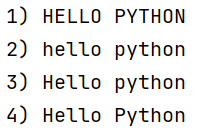


## 实验结果分析(重点)

**Python基本操作**

(1)

这段代码定义了一个函数 yyf\_experiment1(A, B)，该函数接受两个参数 A 和 B，并返回一个由 A 组成的 B 位数的整数。然后，代码从用户处接收两个正整数 A 和 B，并调用函数 yyf\_experiment1() 生成相应的数字，并将结果打印出来。

(2) 

这段代码定义了四个函数，分别用于对输入字符串进行不同类型的大小写转换：

yyf\_experiment1\_to\_upper\_case(s): 将输入字符串中的所有小写字母转换为大写字母。

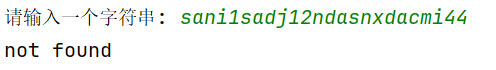
yyf\_experiment1\_to\_lower\_case(s): 将输入字符串中的所有大写字母转换为小写字母。

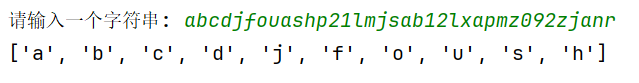
yyf\_experiment1\_capitalize\_first\_letter(s): 将输入字符串的第一个字母转换为大写字母，其余字母转换为小写。

yyf\_experiment1\_capitalize\_words(s): 将输入字符串中每个单词的第一个字母转换为大写，其余字母转换为小写。

然后，对字符串 "hElLo pYtHOn " 应用了这四个函数，并输出结果。

(3)





这段代码定义了一个名为yyf\_experiment1\_find\_letters的函数，该函数接受一个字符串作为输入参数。函数的目标是找到输入字符串中的前10个不重复的英文字母。

函数首先创建一个空列表letters，用于存储找到的不重复的字母。

然后，函数遍历输入字符串中的每个字符。对于每个字符，它首先检查该字符是否是字母（通过char.isalpha())。如果是字母，它会将该字母转换为小写（通过char.lower())，然后检查该字母是否已经在letters列表中。如果不在，它会将该字母添加到letters列表中。

这个过程会一直持续，直到letters列表中有10个字母，此时函数会返回这10个字母。如果遍历完整个字符串后letters列表中还没有10个字母，函数会返回字符串"not found"。

在函数定义之后，代码会提示用户输入一个字符串，然后调用yyf\_experiment1\_find\_letters函数处理这个字符串，并打印出结果。

(4) 这个输出信息更为直观些，虽然有点不符合题目要求





这段代码定义了一个名为yyf\_experiment1\_is\_palindrome的函数，该函数接受一个字符串s作为输入参数。函数内部通过比较字符串s和其反转字符串s[::-1]来判断s是否为回文字符串。

如果s等于其反转字符串（即s是回文字符串)，函数就返回一个字符串，表明s是回文字符串。否则，函数返回一个字符串，表明s不是回文字符串。

然后，代码通过input函数提示用户输入一个字符串，并将输入的字符串赋值给变量s。

最后，代码调用yyf\_experiment1\_is\_palindrome函数，将用户输入的字符串s作为参数，并打印出函数的返回结果。这样，用户就可以看到他们输入的字符串是否是回文字符串。

(5)



这段Python代码是用来计算并判断一个人的身体质量指数（BMI）。

首先，定义了两个变量height和weight，分别表示身高（单位：米）和体重（单位：千克）。

然后，使用公式bmi = weight / (height \*\* 2)计算BMI指数。这个公式是国际上常用的计算BMI的公式，其中体重是以千克为单位，身高是以米为单位。

接着，使用if-elif-else结构根据BMI指数的值判断并打印出身体状况。这个判断的标准是根据世界卫生组织的建议：

如果BMI小于18.5，那么打印"过轻"。

如果BMI在18.5到25之间，那么打印"正常"。

如果BMI在25到28之间，那么打印"过重"。

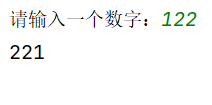
如果BMI在28到32之间，那么打印"肥胖"。

如果BMI大于等于32，那么打印"严重肥胖"。

这段代码的目的是根据一个人的身高和体重，判断他的身体状况是否健康。

(6)





这段Python代码首先获取用户输入的一个数字，然后计算这个数字各个位数之和，最后根据各位数之和是奇数还是偶数来决定如何打印这个数字。

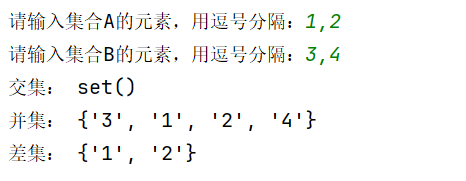
首先，使用input函数获取用户输入的一个数字，保存在变量num中。

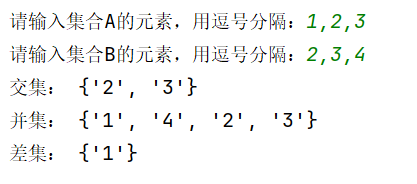
然后，使用列表推导式int(digit) for digit in num将数字的每一位转换为整数，然后使用sum函数计算这些整数的和，保存在变量sum\_of\_digits中。

接着，使用if-else结构判断sum\_of\_digits是奇数还是偶数。如果sum\_of\_digits除以2的余数是0，那么sum\_of\_digits是偶数，打印原数字；否则，sum\_of\_digits是奇数，打印倒序的数字。

这段代码的目的是根据用户输入的数字的各位数之和是奇数还是偶数，来决定如何打印这个数字。

(7)





这段Python代码首先获取用户输入的两个集合，然后计算并打印这两个集合的交集、并集和差集。

首先，使用input函数获取用户输入的两个集合的元素，元素之间用逗号分隔。然后使用split方法将输入的字符串按逗号分割成列表，再使用set函数将列表转换为集合，保存在变量setA和setB中。

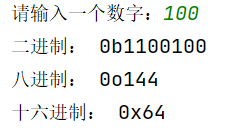
然后，使用&运算符计算两个集合的交集，即既在setA中又在setB中的元素。然后打印这个交集。

接着，使用|运算符计算两个集合的并集，即在setA中或在setB中的元素。然后打印这个并集。

最后，使用-运算符计算两个集合的差集，即在setA中但不在setB中的元素。然后打印这个差集。

这段代码的目的是根据用户输入的两个集合，计算并打印这两个集合的交集、并集和差集。

(8)



这段Python代码首先获取用户输入的一个整数，然后打印这个整数的二进制、八进制和十六进制表示。

首先，使用input函数获取用户输入的一个数字，然后使用int函数将输入的字符串转换为整数，保存在变量num中。

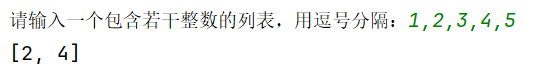
然后，使用bin函数将整数转换为二进制字符串，并打印。bin函数返回的字符串以"0b"开头，表示这是一个二进制数。

接着，使用oct函数将整数转换为八进制字符串，并打印。oct函数返回的字符串以"0o"开头，表示这是一个八进制数。

最后，使用hex函数将整数转换为十六进制字符串，并打印。hex函数返回的字符串以"0x"开头，表示这是一个十六进制数。

这段代码的目的是根据用户输入的一个整数，打印这个整数的二进制、八进制和十六进制表示。

(9)



这段Python代码首先获取用户输入的一个包含若干整数的列表，然后生成一个新的只包含偶数的列表，并打印这个新列表。

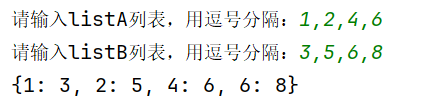
首先，使用input函数获取用户输入的一个包含若干整数的列表，元素之间用逗号分隔。然后使用split方法将输入的字符串按逗号分割成列表，再使用map函数和int函数将列表中的每个元素转换为整数，最后使用list函数将结果转换为列表，保存在变量nums中。

然后，使用列表推导式[num for num in nums if num % 2 == 0]生成一个新的只包含偶数的列表。这个列表推导式的意思是：对于nums中的每个元素num，如果num除以2的余数是0（即num是偶数），那么将num添加到新列表中。这个新列表保存在变量even\_nums中。

最后，打印even\_nums。

这段代码的目的是根据用户输入的一个包含若干整数的列表，生成一个新的只包含偶数的列表，并打印这个新列表。

(10)

  
这段Python代码首先获取用户输入的两个列表，然后使用zip函数和dict函数将这两个列表组合成一个字典，并打印这个字典。

首先，使用input函数获取用户输入的两个列表，元素之间用逗号分隔。然后使用split方法将输入的字符串按逗号分割成列表，再使用map函数和int函数将列表中的每个元素转换为整数，最后使用list函数将结果转换为列表，保存在变量lstA和lstB中。

然后，使用zip函数将lstA和lstB中的元素按照相同的索引配对，生成一个新的迭代器。然后使用dict函数将这个迭代器转换为字典，保存在变量result中。在这个字典中，lstA中的元素作为键，lstB中的元素作为值。

最后，打印result。

这段代码的目的是根据用户输入的两个列表，生成一个新的字典，并打印这个字典。

(11)



这段Python代码首先定义了一个包含多个元组的列表A，然后从这个列表中提取所有的颜色值，并打印这些颜色值。

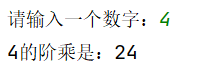
首先，定义了一个列表A，列表中的每个元素都是一个元组，元组的第一个元素是一个字符串，元组的第二个元素是一个字符串，表示第一个元素的类型。

然后，使用列表推导式[item[0] for item in A if item[1] == "color"]提取所有的颜色值。这个列表推导式的意思是：对于A中的每个元素item，如果item的第二个元素（即item[1]）等于"color"，那么将item的第一个元素（即item[0]）添加到新列表中。这个新列表保存在变量A\_colors中。

最后，打印A\_colors。

这段代码的目的是从一个包含多个元组的列表中提取所有的颜色值，并打印这些颜色值。

(12)

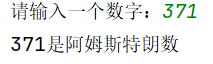


这段Python代码首先获取用户输入的一个整数，然后计算这个整数的阶乘，并打印结果。

首先，使用input函数获取用户输入的一个数字，然后使用int函数将输入的字符串转换为整数，保存在变量num中。

然后，定义一个变量factorial，并初始化为1。这个变量用来保存阶乘的结果。

(13) 



这段Python代码首先获取用户输入的一个整数，然后检查这个整数是否是阿姆斯特朗数，并打印结果。

首先，使用input函数获取用户输入的一个数字，然后使用int函数将输入的字符串转换为整数，保存在变量num中。

然后，使用str函数将num转换为字符串，保存在变量str\_num中。这样可以方便地获取num的位数，即字符串的长度，保存在变量n中。

接着，定义一个变量sum，并初始化为0。这个变量用来保存各位数字的n次方之和。

然后，使用for循环遍历str\_num中的每个字符（即num的每一位数字）。在每次循环中，将当前的字符转换为整数，然后计算这个整数的n次方，并累加到sum中。

最后，使用if语句检查sum是否等于num。如果等于，那么num就是阿姆斯特朗数，打印相应的结果；否则，num不是阿姆斯特朗数，打印相应的结果。

这段代码的目的是根据用户输入的一个整数，检查这个整数是否是阿姆斯特朗数，并打印结果。

(14)

  
这段Python代码首先初始化一个包含若干整数的列表，然后使用冒泡排序算法对这个列表进行排序，并打印排序后的列表。

首先，定义了一个列表nums，包含了10个整数。

然后，使用冒泡排序算法对nums进行排序。冒泡排序是一种简单的排序算法，它重复地遍历列表，比较每对相邻的元素，如果它们的顺序错误就把它们交换过来。遍历列表的工作是重复地进行直到没有再需要交换，也就是说该列表已经排序完成。

在这段代码中，使用了两层for循环来实现冒泡排序。外层循环控制遍历的次数，内层循环控制每次遍历时的比较和交换。在内层循环中，如果当前元素nums[j]大于下一个元素nums[j + 1]，那么就交换这两个元素的位置。

最后，打印排序后的列表nums。

这段代码的目的是对一个包含若干整数的列表进行排序，并打印排序后的列表。

(15)

这段Python代码是用来解决一个经典的数学问题：五个人一起去捕鱼，每人都捕到了一些鱼。晚上，他们决定睡觉，第二天早上再一起分鱼。但是，晚上有一个人醒来，他把所有的鱼分成五份，发现多了一条，就把这条鱼扔掉，然后拿走了自己的一份。后来，又醒来一个人，他也把剩下的鱼分成五份，发现也多了一条，就把这条鱼扔掉，然后拿走了自己的一份。后来，其他的人都这样做了。问题是，他们至少捕了多少条鱼？

首先，初始化鱼的数量为5。

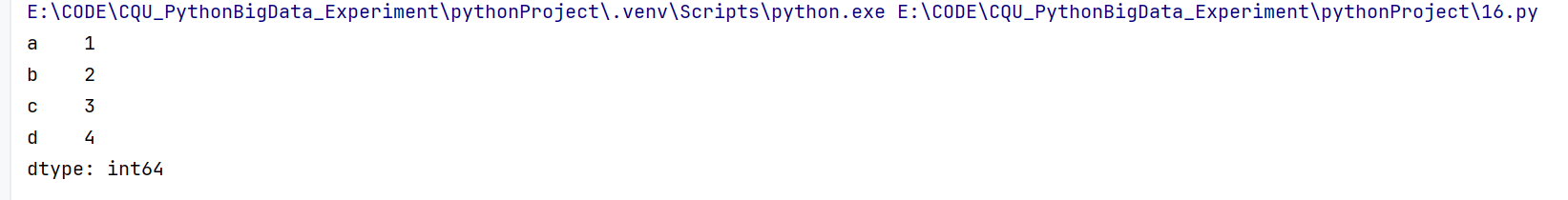
然后，使用一个无限循环来尝试不同的鱼的数量。在每次循环中，首先假设总的鱼的数量为当前尝试的数量。然后，对每个人进行操作：如果总的鱼的数量不能被5整除，那么就跳出循环；否则，扔掉一条鱼，然后分给一个人。如果所有的人都满足条件，那么就找到了答案，打印答案并跳出无限循环。

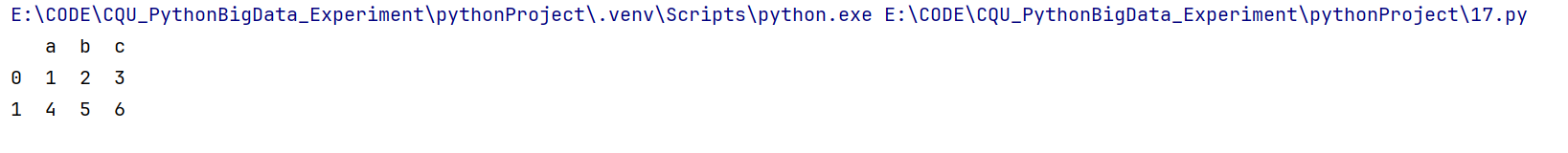
如果没有找到答案，那么就增加鱼的数量，然后继续下一次循环。

这段代码的目的是找到满足题目条件的最小的鱼的数量。

**Python 基本函数分析库**

（一) 熟悉常用的pandas 操作

(1) 

(2) 

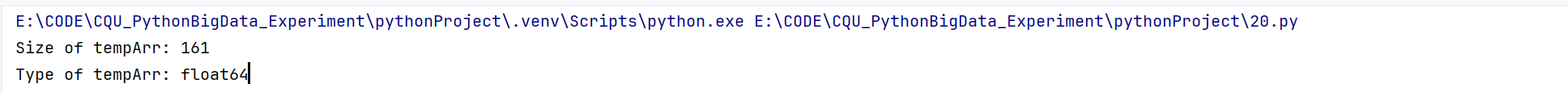
(3) (4) (5)(6)





这里显然是把第一行的数据当成“数据名”处理了，再正常情况下 第一行一般要说明这列数据是什么意思，很合理，比如说-0.07 一般来说是要给属性

（二)熟悉常用的Numpy 操作

(1) 

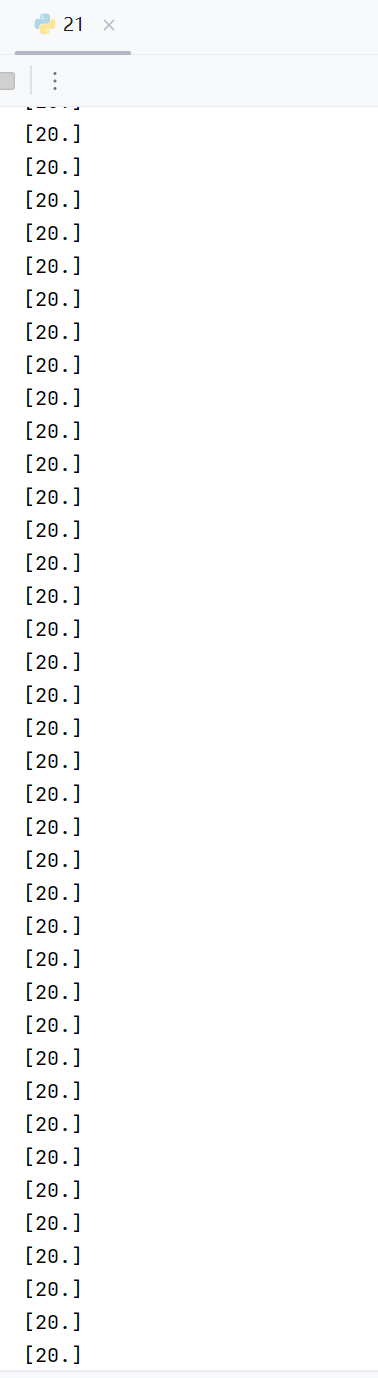
这段Python代码首先导入pandas库，然后创建一个pandas的Series对象，并打印这个对象。

首先，使用import语句导入pandas库，这是一个提供数据分析功能的Python库。

然后，使用pd.Series函数创建一个Series对象。Series是pandas库中的一种数据结构，它是一种类似于一维数组的对象，它由一组数据（各种NumPy数据类型）以及一组与之相关的数据标签（即索引）组成。在这个例子中，创建的Series对象的数据是[1, 2, 3, 4]，索引是['a', 'b', 'c', 'd']。

最后，使用print函数打印这个Series对象。

这段代码的目的是创建一个pandas的Series对象，并打印这个对象。

(2) 

这段Python代码首先导入pandas库，然后创建三个pandas的Series对象，并使用这些Series对象创建一个pandas的DataFrame对象，最后打印这个DataFrame对象。

首先，使用import语句导入pandas库，这是一个提供数据分析功能的Python库。

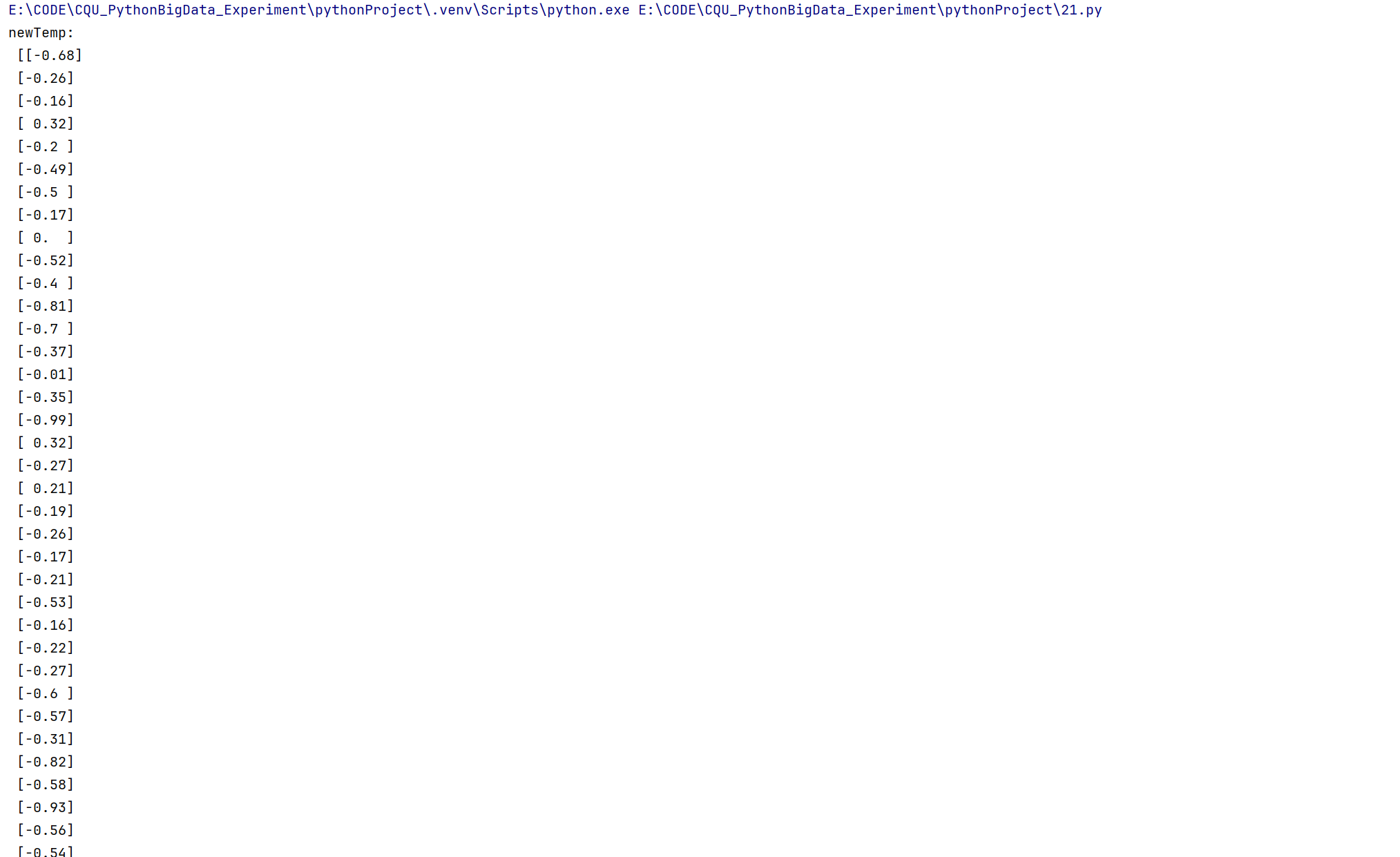
然后，使用pd.Series函数创建三个Series对象，每个对象的数据和名称分别是[1, 4]和'a'，[2, 5]和'b'，[3, 6]和'c'。

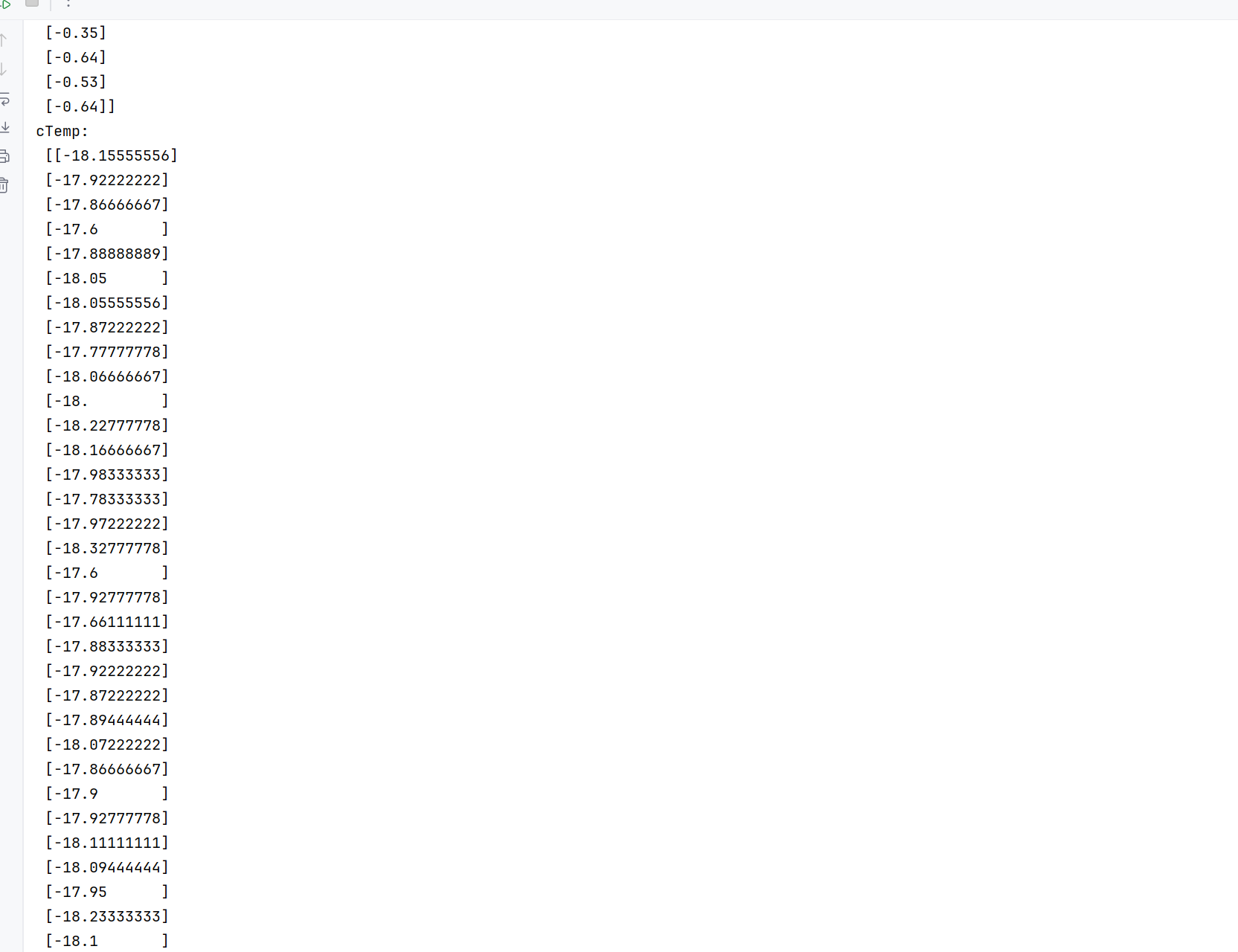
接着，使用pd.concat函数将这三个Series对象沿着列方向（axis=1）拼接起来，创建一个DataFrame对象。在这个DataFrame对象中，每个Series对象就是一列，Series对象的名称就是列的名称。

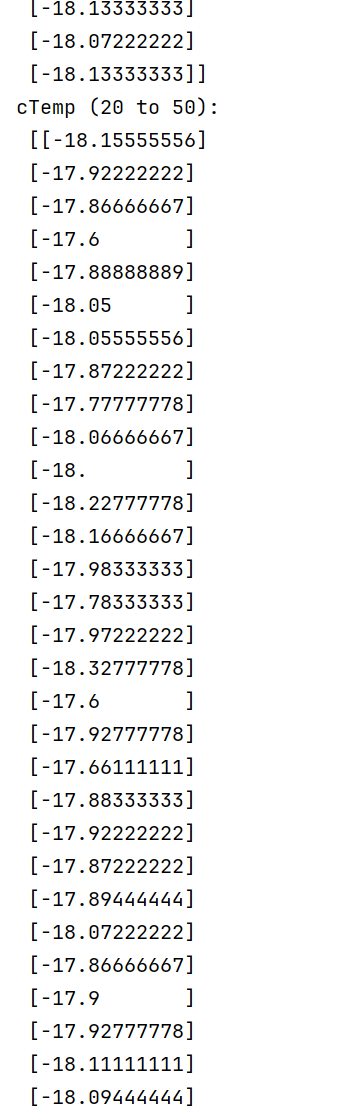
最后，使用print函数打印这个DataFrame对象。

这段代码的目的是创建一个pandas的DataFrame对象，并打印这个对象。

(3)(4) (5) (6)







数据过多 仅简要展示

这段Python代码首先导入pandas和numpy库，然后读取一个CSV文件到一个pandas的DataFrame对象，预览前5行数据，统计数据的基本统计量，创建一个从第10行到第50行的数据片段，并打印这些结果。

首先，使用import语句导入pandas和numpy库，这两个都是提供数据分析功能的Python库。

然后，使用pd.read\_csv函数读取一个CSV文件到一个DataFrame对象。在这个例子中，读取的文件名是'ocean\_temp.csv'，文件的编码格式是'utf-8'。

接着，使用print和df.head函数预览DataFrame对象的前5行数据。

然后，使用print和df.describe函数统计DataFrame对象的数据的基本统计量，包括计数、平均值、标准差、最小值、25%分位数、中位数、75%分位数和最大值。

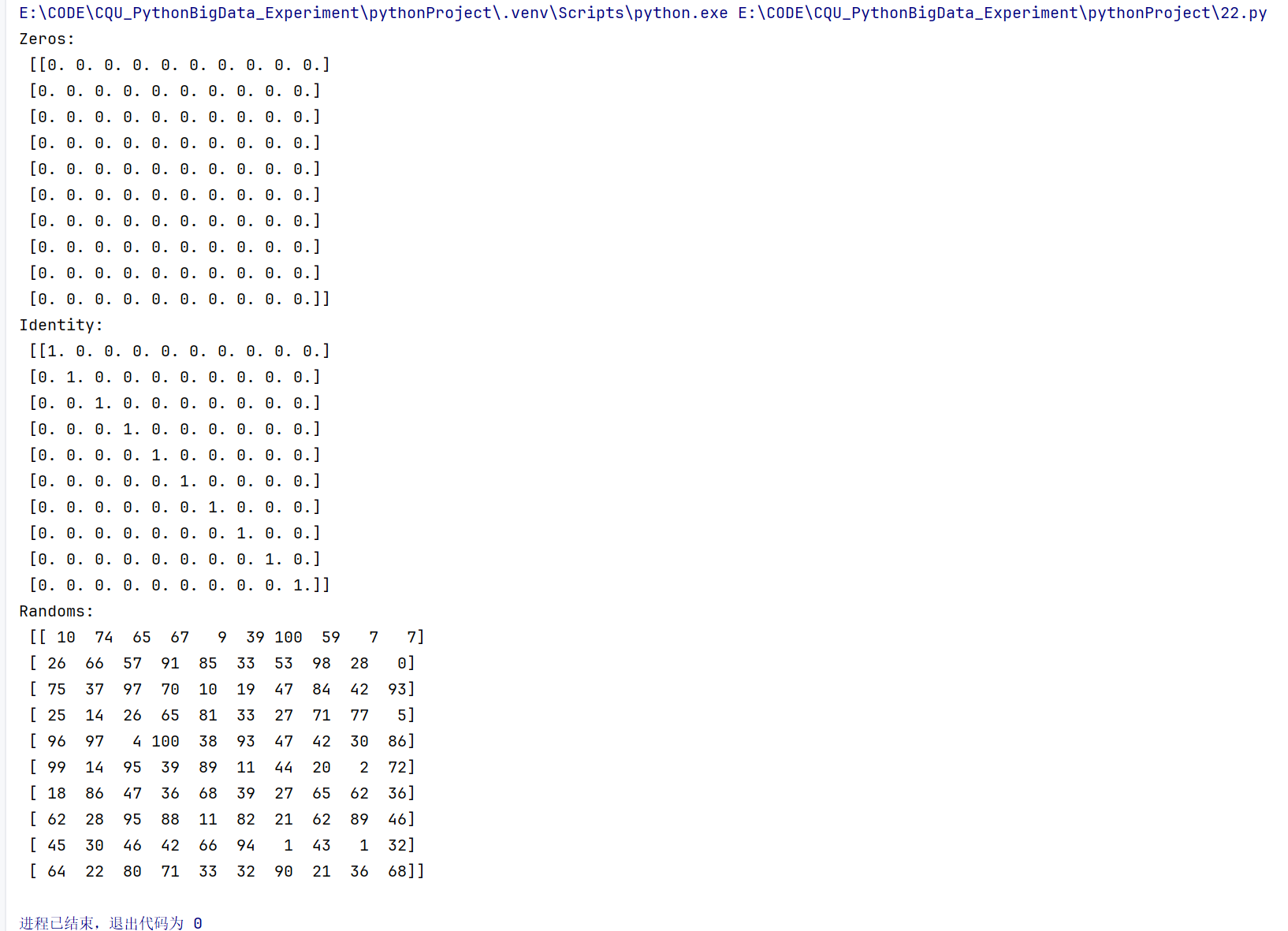
接着，使用df.iloc函数创建一个从第10行到第50行的数据片段。在Python中，索引是从0开始的，所以第10行对应的索引是9，第50行对应的索引是49。但是，iloc函数的结束索引是不包含的，所以这里的结束索引应该是50。

最后，使用print函数打印这个数据片段。

这段代码的目的是读取一个CSV文件到一个DataFrame对象，预览前5行数据，统计数据的基本统计量，创建一个从第10行到第50行的数据片段，并打印这些结果。

(7)

1)2)3)

****

这段Python代码首先导入pandas和numpy库，然后读取一个CSV文件到一个pandas的DataFrame对象，预览前5行数据，统计数据的基本统计量，创建一个从第10行到第50行的数据片段，并打印这些结果。

首先，使用import语句导入pandas和numpy库，这两个都是提供数据分析功能的Python库。

然后，使用pd.read\_csv函数读取一个CSV文件到一个DataFrame对象。在这个例子中，读取的文件名是'ocean\_temp.csv'，文件的编码格式是'utf-8'。

接着，使用print和df.head函数预览DataFrame对象的前5行数据。

然后，使用print和df.describe函数统计DataFrame对象的数据的基本统计量，包括计数、平均值、标准差、最小值、25%分位数、中位数、75%分位数和最大值。

接着，使用df.iloc函数创建一个从第10行到第50行的数据片段。在Python中，索引是从0开始的，所以第10行对应的索引是9，第50行对应的索引是49。但是，iloc函数的结束索引是不包含的，所以这里的结束索引应该是50。

最后，使用print函数打印这个数据片段。

这段代码的目的是读取一个CSV文件到一个DataFrame对象，预览前5行数据，统计数据的基本统计量，创建一个从第10行到第50行的数据片段，并打印这些结果。

## 问题和解决

我想把这些代码保存到代码仓库备忘，但是太久没用git了，所以去查了下代码

git remote add AA BBBBB

其中AA为代码仓库的别称 BBBBB为地址

例如

git remote add py <https://github.com/mozhongzhou/CQU_Python_experiment>

以及git push 相关的各种代码 复习了一遍

但是操作完后才发现pycharm也能像vscode一样快捷使用源代码管理器，甚至pycharm与git的联动更多，更为方便。 图形化界面确实比在cmd里敲代码来得舒服

例如



在左上角可以快速找到源代码管理器，这点类似于vscode，可能是由于pycharm在安装时关联了vscode，获取了个性化设置。

我的代码仓库地址：[mozhongzhou/CQU\_Python\_experiment (github.com)](https://github.com/mozhongzhou/CQU_Python_experiment)