**天津理工大学**

**《计算思维与程序设计基础（Python语言）》课程**

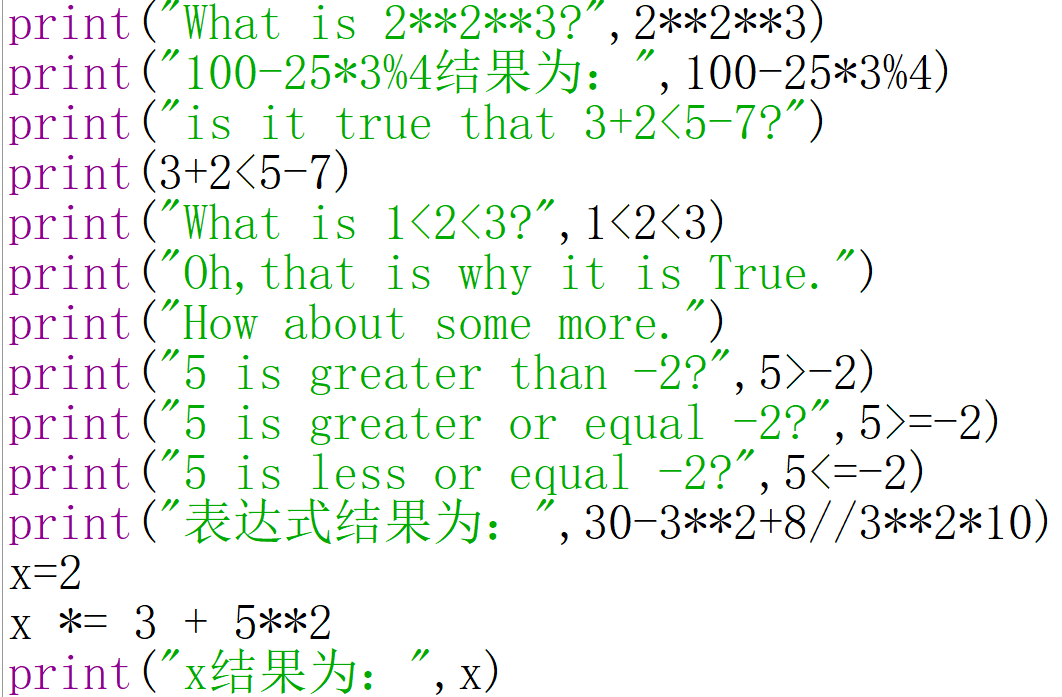
**实 验 报 告**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 学生姓名 | 王瀚祥 | 学号 | 20215212 | 班 级 | 04 |
| 实验名称 | 实验2：Python简单编程 | | | | |
| 实验编号 | 2290123-2 | | | 实验类型 | 验证性 |
| 实验评语 |  | | | 实验得分 |  |
| **（一）实验目的：**  **掌握解决计算问题的一般方法。**  **熟练基本语法，包括缩进、变量、命名等。**  **（二）实验步骤和要求：**  **1. 做实验之前，先创建文件夹，以自己的学号姓名方式命名，例如:20211234张三。**  **2. 仔细阅读实验内容中的题目，新建文件，将每道实验题的源程序按题目顺序依次命名为：sy2-01.py，sy2-02.py，……保存文件到自己的文件夹中。**  **3. 调试程序，如果有错误，将错误改正后，重新保存。得到正确的运行结果后，将程序代码复制粘贴到实验报告中每个实验后的相应位置，并将运行结果截图粘贴到相应位置（使用Windows自带的截图工具）**  **4. 所有实验都完成后，整理文件夹，同时也将本实验报告放入自己的文件夹中。压缩文件夹为\*.rar文件，发送到邮箱。**  收件人：279983512@qq.com  主 题：20221234张三实验2（修改为自己的）  附 件：是你的实验文件 | | | | | |

**（三）实验内容：**

【题目1】（sy2-01.py）：熟悉数字和数学计算。以下实验题目里有很多数学运算符号，熟悉下列每一个符号的名字，录入程序代码，运行程序。再回到列表，在括号中填写每个符号的作用。例如：+用来做加法运算。

* +：（加法运算）
* -：（……）
* /：（）
* \*：（）
* %：（）
* <：（）
* >：（）
* <=：（）
* >=：（）



【程序清单】

print("What is 2\*\*2\*\*3?",2\*\*2\*\*3)

print("100-25\*3%4的结果为:",100-25\*3%4)

print("is it true that 3+2<5-7?")

print(3+2<5-7)

print("What is 1<2<3?",1<2<3)

print("Oh,that is why it is True.")

print("How about some more.")

print("5 is greater than -2?",5>-2)

print("5 is greater or equal -2?",5>=-2)

print("5 is less or equal -2?",5<=-2)

print("表达式的结果为：",30-3\*\*2+8//3\*\*2\*10)

x=2

x\*=3+5\*\*2

print("x结果为:",x)

【运行结果】（截图）：



【题目2】（sy2-02.py）：编程题。编程实现汇率兑换：参考实验1【题目3】的设计思路，按照1美元=6.4元人民币的汇率编写一个美元和人民币的双向兑换程序。人民币用字母y结尾，美元用字母d结尾。其他变量自己定义即可。

【程序清单】

money=input("请输入带有符号的金额：")

while money[-1] not in ['N','n']:

if money[-1] in ['D','d']:

Y=6.4\*eval (money[0:-1])

print("折合人民币{:.2f}Y".format(Y))

elif money[-1] in ['Y','y']:

D=0.15625\*eval(money[0:-1])

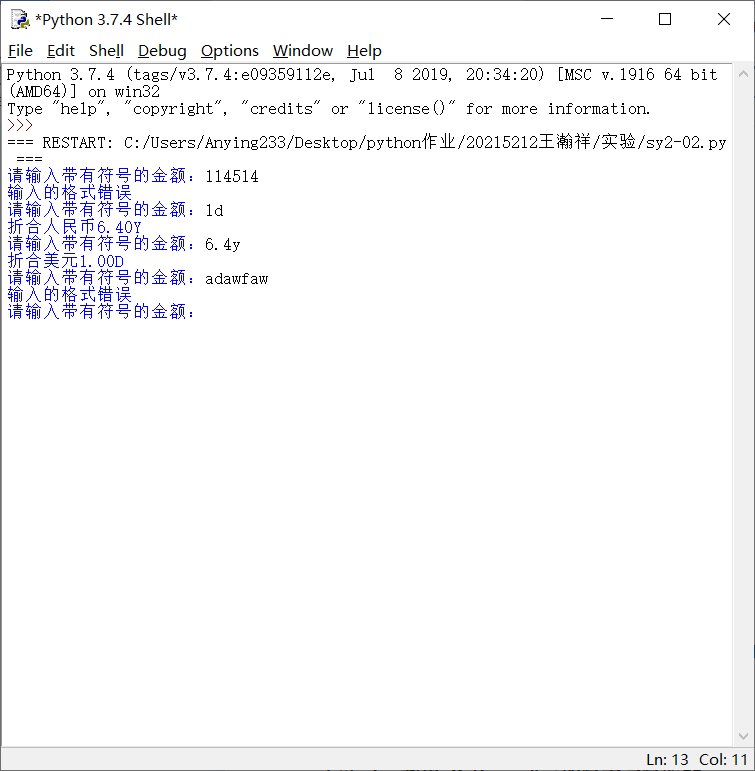
print("折合美元{:.2f}D".format(D))

else:

print("输入的格式错误")

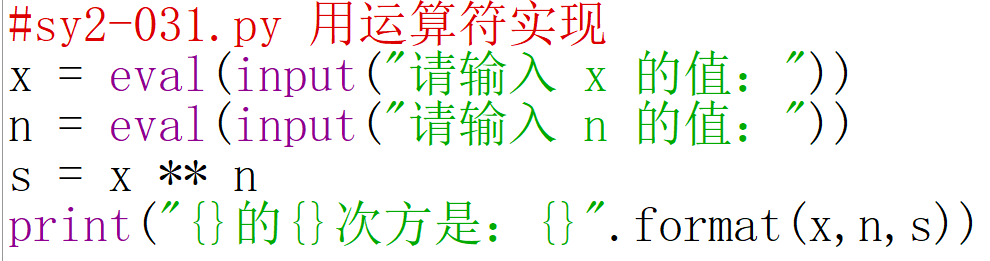
money=input("请输入带有符号的金额：")

【运行结果】（截图）：

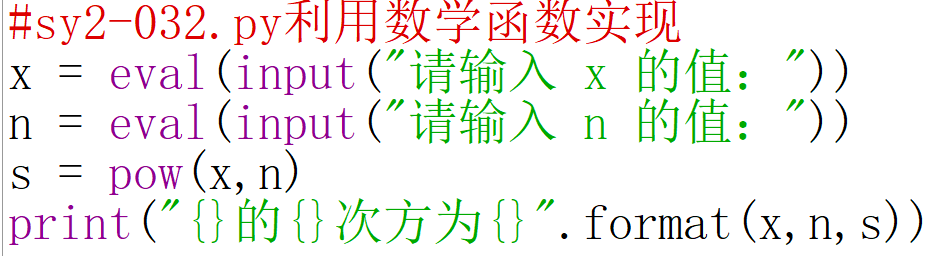


【题目3】验证题。从键盘输入整数x和n，求x的n次幂。用四种方法实现。录入代码，分别保存为sy2-031.py、sy2-032.py、sy2-033.py、sy2-034.py。并分别给出运行结果（对每一种方法给出不同的值）

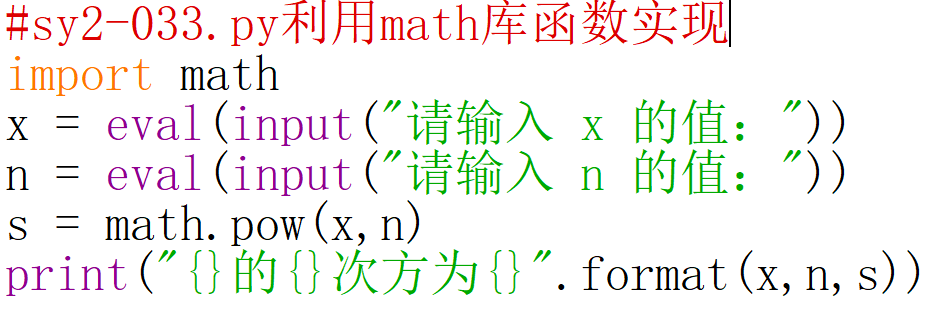
方法 1：利用 Python 语言的算术运算符\*\*。



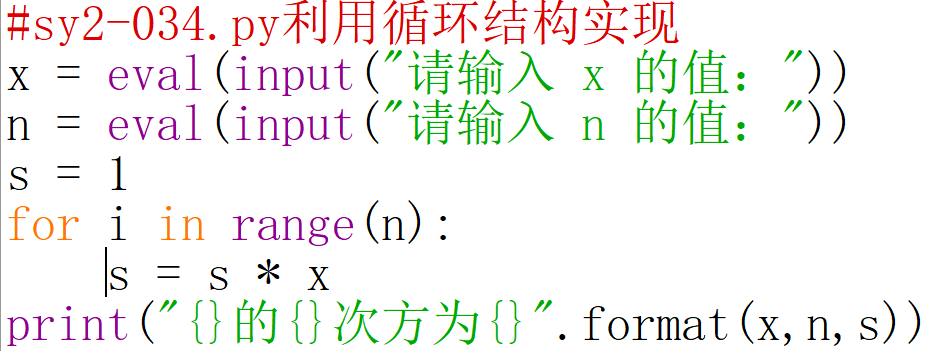
方法 2：利用 Python 语言的内置函数 pow()。



方法 3：利用 math 库的 pow()函数。



方法 4：利用循环结构。



【方法1程序清单】

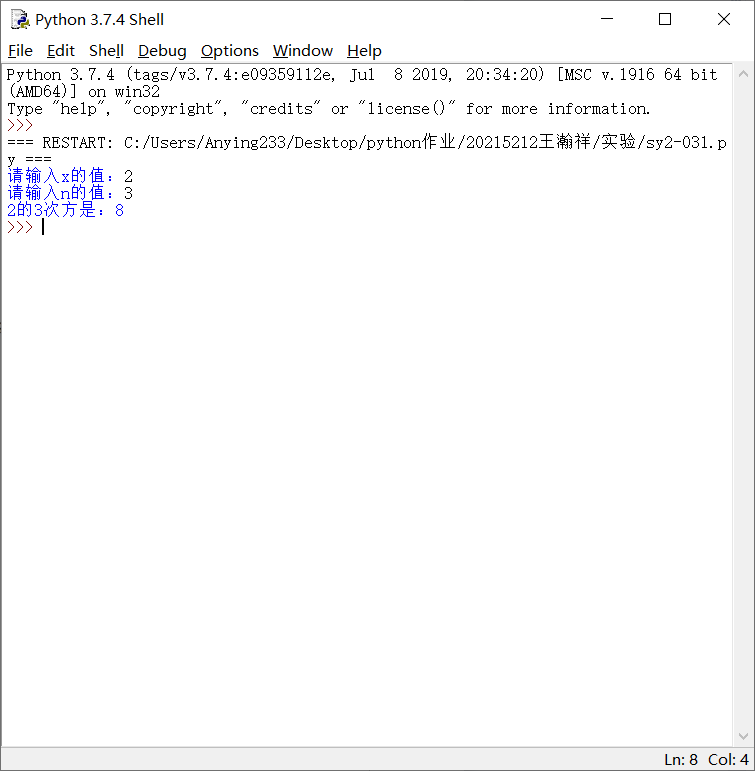
x=eval(input("请输入x的值："))

n=eval(input("请输入n的值："))

s = x\*\*n

print("{}的{}次方是：{}".format(x,n,s))

【运行结果】（截图）：



【方法2程序清单】

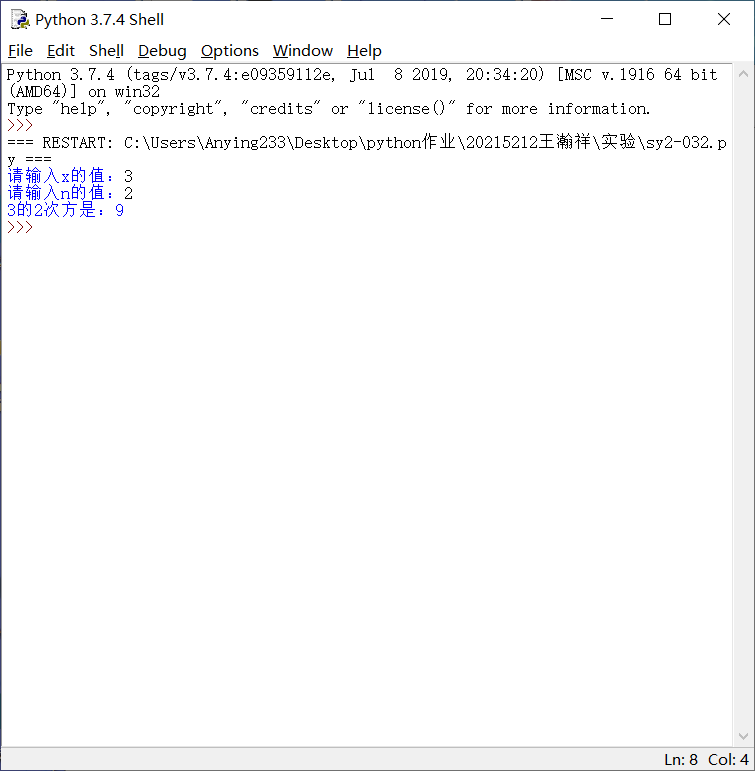
x=eval(input("请输入x的值："))

n=eval(input("请输入n的值："))

s = pow(x,n)

print("{}的{}次方是：{}".format(x,n,s))

【运行结果】（截图）：



【方法3程序清单】

import math

x=eval(input("请输入x的值："))

n=eval(input("请输入n的值："))

s=math.pow(x,n)

print("{}的{}次方是：{}".format(x,n,s))

【运行结果】（截图）：



【方法4程序清单】

x=eval(input("请输入x的值："))

n=eval(input("请输入n的值："))

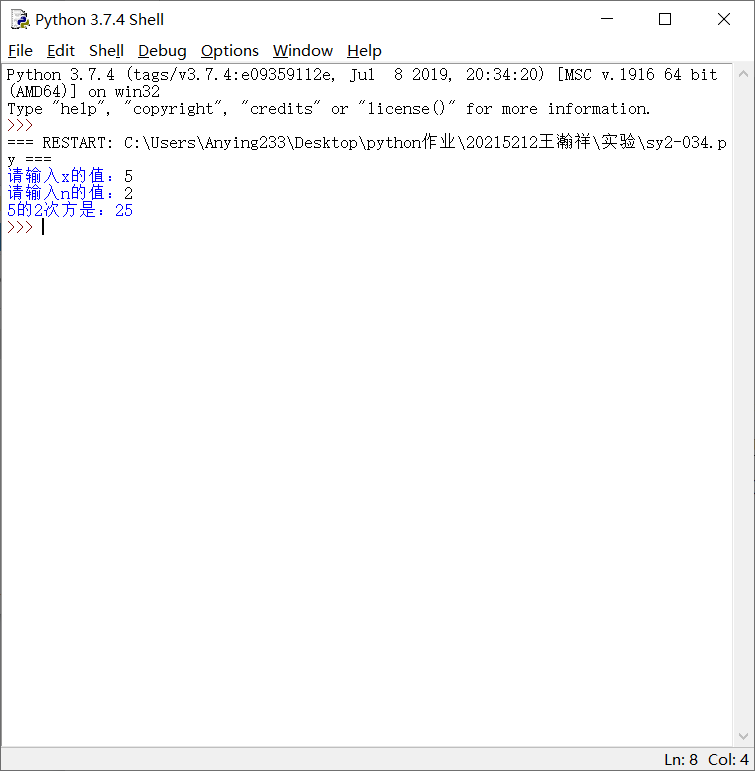
s=1

for i in range(n):

s=s\*x

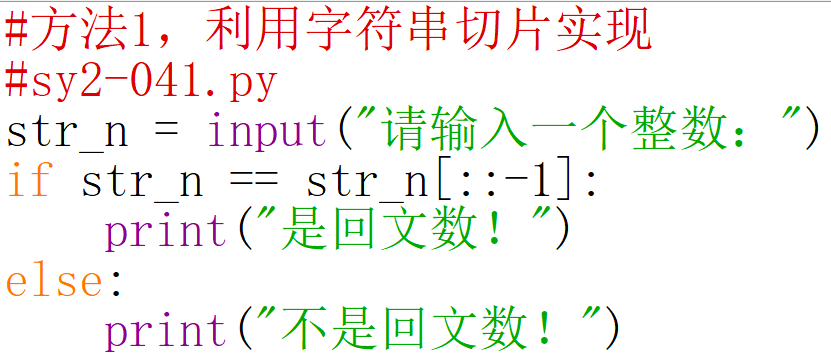
print("{}的{}次方是：{}".format(x,n,s))

【运行结果】（截图）：

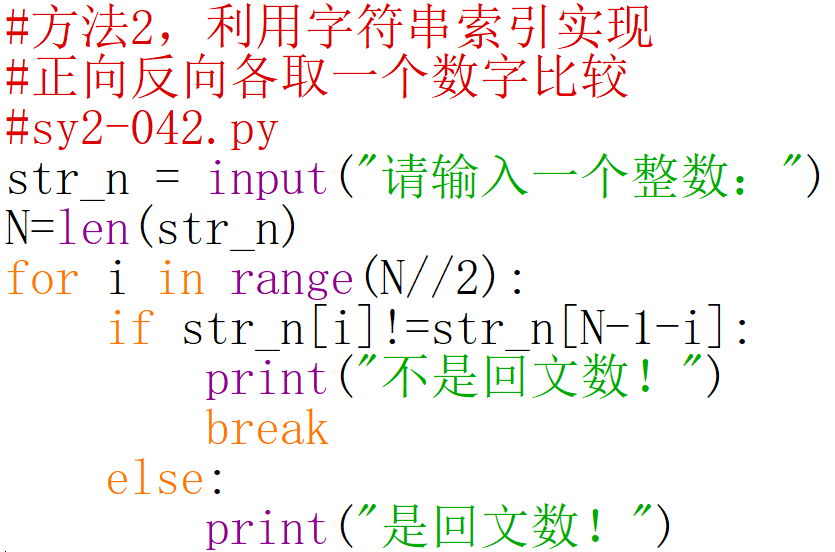


【题目4】验证题。设n是一个任意自然数，如果n的各位数字反向排列所得自然数与n相等，则n被称为回文数。下面代码实现回文数判断。用三种方式实现。录入代码，分别保存为：sy2-041.py、sy2-042.py、sy2-043.py，并分别给出运行结果（对每一种方法给出不同的值）。

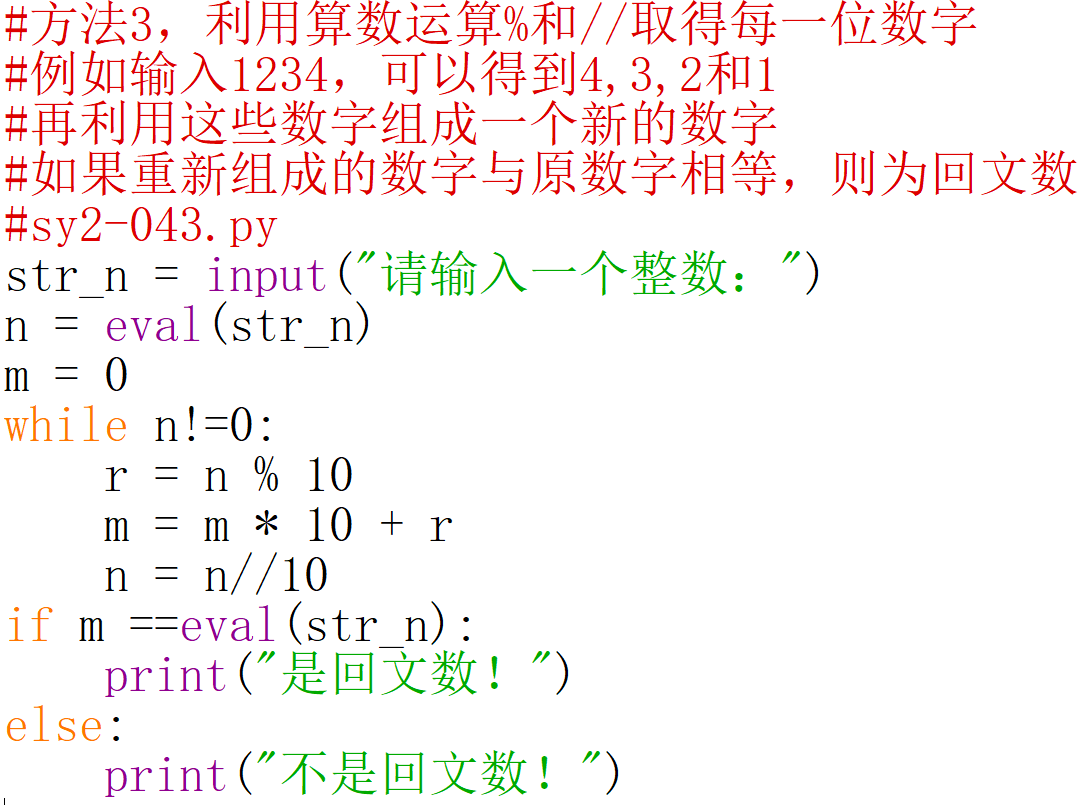
方法1：



方法2：



方法3：



【方法1程序清单】

str\_n=input("请输入一个整数：")

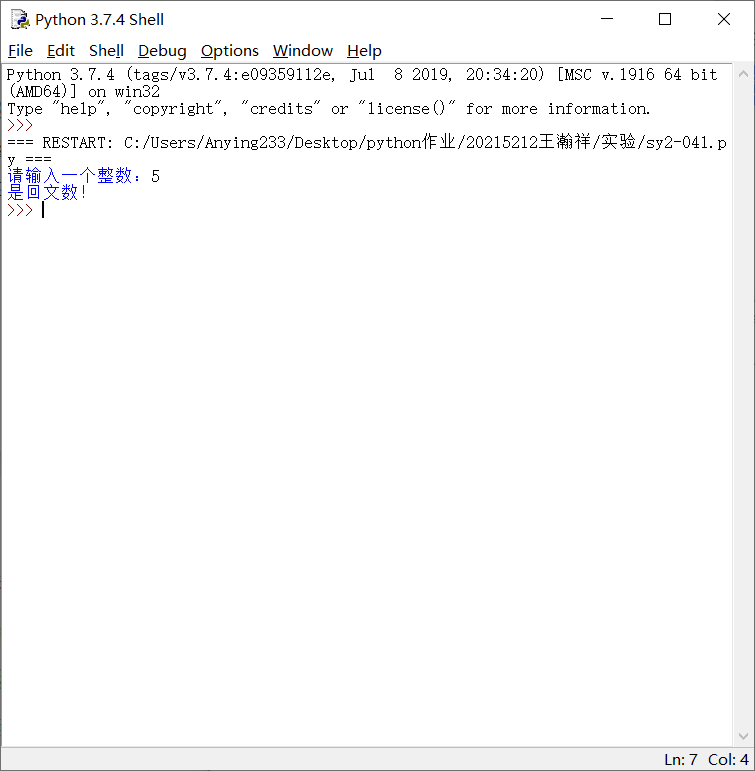
if str\_n==str\_n[::-1]:

print("是回文数！")

else:

print("不是回文数！")

【运行结果】（截图）：



【方法2程序清单】

str\_n=input("请输入一个整数：")

N=len(str\_n)

for i in range(N//2):

if str\_n[i]!=str\_n[N-1-i]:

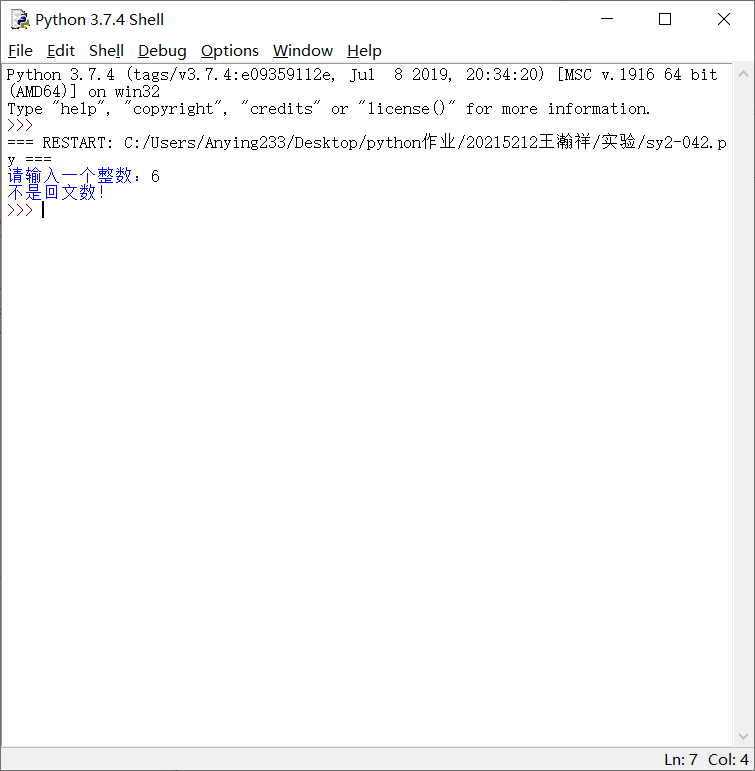
print("是回文数！")

break

else:

print("不是回文数！")

【运行结果】（截图）：



【方法3程序清单】

str\_n=input("请输入一个整数：")

n=eval(str\_n)

m=0

while n!=0:

r=n%10

m=m\*10+r

n=n//10

if m ==eval(str\_n):

print("是回文数！")

else:

print("不是回文数！")

【运行结果】（截图）：

