**第 1 章 初识 Python**

**实验 1：完成教材例题 ex0101~ex0104.py。**

要求：掌握 IDLE 环境下程序的建立和执行过程，并能简单调试程序。

#ex0101

r=3.2

area=3.14\*r\*r

perimeter=2\*3.14\*r

print(“圆形的面积：{:.2f},周长：{:.2f}”.format(area.perimeter)

#ex0102

#ex0103 海伦公式计算三角形的面积

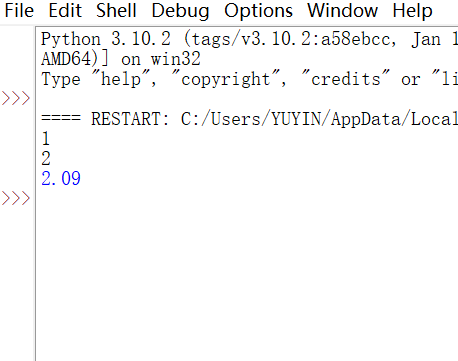
#ex0104

**实验 2：计算圆椎体体系积。**

要求: 交互式输入圆椎体的底面半径和高。

提示：（1）使用两个函数 input()和 eval()，其中 input()函数用于接收用户的输入，接收的值是字符串；eval()函数用来执行一个字符串表达式，并返回该表达式的值。

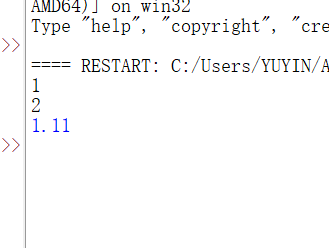
1. 圆锥体体积的公式 v=math.pi\*r\*r\*h/3



**实验 3：计算存款收益。**

要求：交互输入存款金额和存款年限（n），计算收益。其中，默认利率为 5.2%。

提示：收益=金额×(1+利率)n”，表达式为 total=num\*(1+rate)\*\*years



**实验 4：使用 Python 帮助文档中的函数实现计算。**

要求：查找其中的 Numeric Types 类型，试使用其中的代数函数、指数和对数函数、三角函数等完成计算功能。

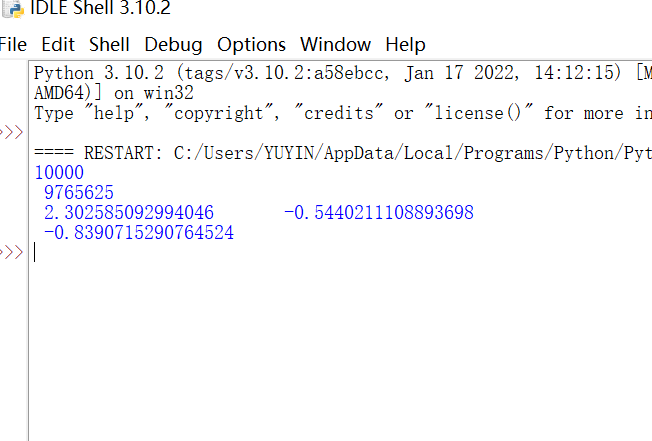
输入x = 10, 计算x4，5x，log(x)，sin(x)，cos(x)等值。

提示：（1）在 IDLE 环境下，选择[Help]/[Python Docs]命令或按 F1 键，就可以启动 Python 帮助文档。

（2）在帮助文档的“The Python Standard Library”选项下，选择“Numeric and Mathematical Modules”模块，其中包括数学运算的相关函数。

（3）可以选择 Python 的内置函数，其在“The Python Standard Library”选项下的“Built-in

Functions”。



**第 2 章 Python 基础知识**

**实验 1：调试完成教材例题。**

要求：掌握代码缩进、注释、数据类型、运算符等基础知识。

**实验 2：分析下面各语句的输出结果。**

01 >>>0.2+0.4==0.6

02 >>>round(0.2+0.4,1)==0.6

03 >>>0xAF

04 >>>9\*\*0.5

05 >>>-10%-3

06 >>>-3\*\*2 ; (-3)\*\*2

07 >>>4.0+3

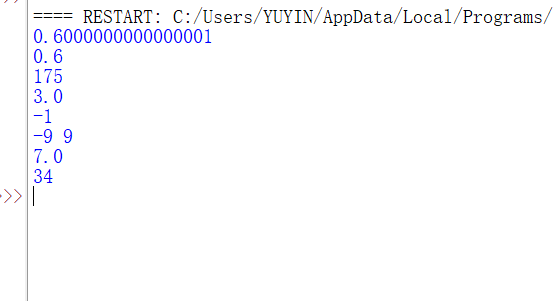
08 >>>12 and 34

提示：

（1）浮点数运算存在不确定的尾数，测试 0.2+0.4 的结果是 0.6000000000000001

（2）0x 开头表示这是一个十六进制整数。

（3）\*\*表示幂运算，指数如果是小数，表示根式运算，结果为浮点类型数据，幂运算优先级较高。



**实验 3：阅读并解析程序。**

要求：两个变量的值互换。若 x 值为 3，y 值为 4，则互换后 x 的值为 4，y 的值为 3。程序代码如下。（行号不能输入）

01 #demo0201.py

02 x=eval(input("请输入 x 的值："))

03 y=eval(input("请输入 y 的值："))

04 print("输入值：x=",x,"y=",y)

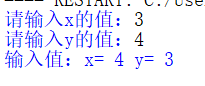
05 t=x

06 x=y

07 y=t

08 print("互换后：x=",x,"y=",y)

提示：两数互换算法，首先 t=x 语句将 x 的值存于临时变量 t 中，然后 x=y 语句将 y 的值赋予 x，最后 y=t 语句将 t 中保存的 x 的原始值赋予 y，从而实现 x 和 y 数值的互换。也可以使用同步赋值语句实现两数互换 x,y=y,x



**实验 4：编写程序，计算并输出方程的实数根。**

要求：输入一元二次方程 ax 2 +bx+c=0 的参数 a、b、c，计算并输出方程的实数根（结果保留两位小数）。若方程没有实数根，输出“方程没有实数根”。

提示：

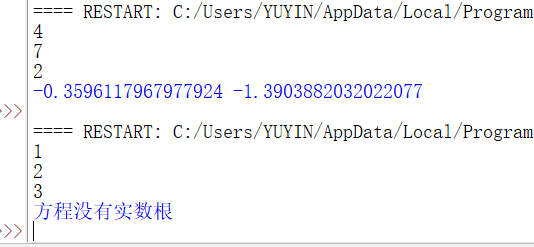
p=b\*b-4\*a\*c

if p>0:

x1=(-b+math.sqrt(p))/(2\*a)

x2=(-b-math.sqrt(p))/(2\*a)

23



**第 3 章 Python 中的字符串**

**实验 1：调试完成教材例题。**

要求：掌握字符串操作常用方法和输入输出语句。

**实验 2：分析代码运行结果。**

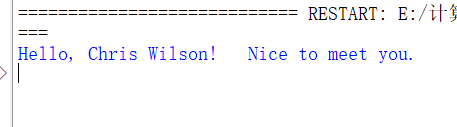
要求：阅读下面的语句，分析其功能。代码如下

01 >>> first\_name="chris"

02 >>> last\_name="Wilson"

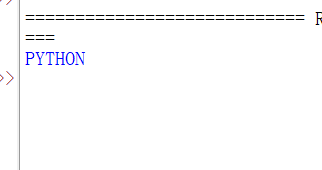
03 >>> full\_name=first\_name+" "+last\_name

04 >>> print("Hello, "+full\_name.title()+"!"+" "\*3+"Nice to meet you.")

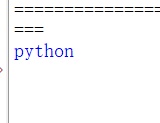
05 \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

06 >>> language="python"

07 >>> print(language.upper())

08 \_\_\_\_ \_\_\_\_\_

09 >>> print(language)

10 \_\_\_\_ \_\_\_\_\_

11 >>> product\_num="201906C15M"

12 >>> print(product\_num[6])

13 \_\_\_\_\_\_\_\_\_

14 >>> print(product\_num[-1])

15 \_\_\_\_\_\_\_\_\_

16 >>> print(product\_num[4:6])

17

18 >>> print(product\_num[:4])

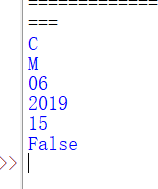
19 \_\_\_\_\_\_\_\_\_

20 >>> print(product\_num[-3:-1])

21 \_\_\_\_\_\_\_\_\_

22 >>> print('m' in product\_num)

23 \_\_\_\_\_\_\_\_\_



**实验 3：分析代码运行结果。**

要求：阅读下面的程序代码，分析其功能。

01 >>> name="Evan"

02 >>> money=45.7803

04 >>> number=10

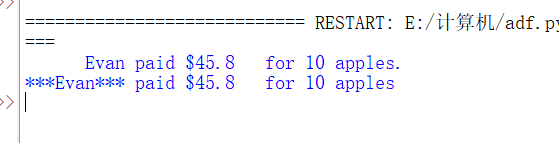
05 >>> print("%10s paid $%-6.1f for %d apples."%(name,money,number))

06 \_\_\_\_\_\_\_\_\_

07 >>> print("{0:\*^10} paid ${1:<6.1f} for {2:d} apples".format(name,

money,number))

08 \_\_\_\_\_\_\_\_\_



提示：

（1）第 5 行，采用格式化操作符%对输出字符串进行格式化，用法是将跟在字符串后面的%后的值插入对应的有格式控制符的字符串中。其中，格式控制符%s 用来格式化字符串；%f 用来格式化浮点数，可指定小数点后的精度；%d 用来格式化十进制整数。例如，%-6.1f 对变量 money 进行格式化，6 为宽度，.1 为小数位数，-表示左对齐。

（2）第 7 行，采用 str.format()方法对输出字符串进行格式化。模板字符串 str 中可包括4 多个{}表示的占位符，与 format()方法中的对应参数进行匹配。模板字符串的{}内部可以使用{<参数序号>：<格式控制标记>}的方式来对关联参数的输出格式进行控制。例如，{0:\*^10}中，0 为参数序号，10 为宽度，\*为填充字符，^表示居中对齐。

**实验 4：编程实现英文句子反转功能。**

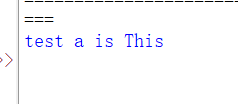
要求：给定一个只包括字符和空格的句子，将句子中的单词位置反转后输出。输入的句子占一行，各个单词之间以空格分隔。例如，输入 this is a test，输出 test a is this。

提示：

（1）将输入的字符串用 split()函数分隔，转换为列表，将列表反转后再生成字符串。

（2）使用 str[::-1]”将列表反转。

（3）使用“" ".join(a)”函数，功能是以空格为分隔符，将列表 a 中的所有元素合并为一个新的字符串。

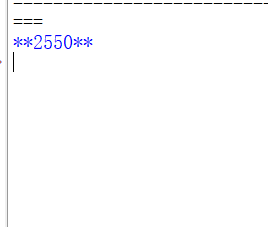


**第 4 章 Python 程序的流程**

**实验 1：编程计算 1～100 偶数的和。**

要求：输出结果为整数，宽度为 8，居中对齐，空白处填充符号“\*”。

提示： range(2,101,2)生成的整数列表从 2 开始到 100 结束，步长为 2，即 1～100 的所有偶数。



**实验 2：编程计算字符的个数。**

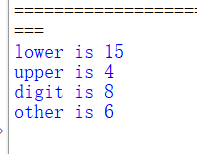
要求：输入一个字符串，计算其中小写字符、大写字符、数字字符、其他字符的个数。

提示：

（1）用 for n in s 对字符串遍历，执行时依次从字符串 s 中取出每个字符放入变量 n 中，并执行一次循环体。

（2）设置了 4 个计数器变量，num\_lower、num\_upper、num\_digit、other 分别存放小写字符、大写字符、数字字符、其他字符的个数。

（3）判断字符是否为大写字符、小写字符或数字字符，可调用字符串函数 n.islower()、n.isupper()、n.isdigit()。



**实验 3：编程计算序列和。**

要求：有一分数序列：2/1，3/2，5/3，8/5，13/8，21/13，…，计算这个序列的前 20 项之和。

提示：

（1）采用遍历循环 for 语句，循环执行 20 次，每次循环进行一次分数的累加，最终 得到分数序列前 20 项的和。

（2）观察这个分数序列的规律，从第 2 个分数开始，分子=前一个分数的分子+前一个数的分母，分母=前一个分数的分子。

（3）代码如下：

01 #计算序列和。

02 a,b=2,1

03 sum=0

04 for i in range(20):

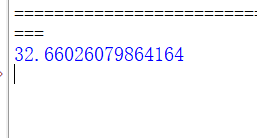
05 sum+=a/b

06 t=a

07 a=a+b

08 b=t

09 print(sum)



**实验 4：阅读程序，分析下面程序完成的功能。**

要求：程序代码如下，重点分析 break 和 continue 语句功能。

提示：（1）第 6 行，如果 x>100，则执行第 7 行 break 语句退出循环，故此循环执行 100 次，x 分别取 1～100 的每个整数。

（2）第 8 行判断如果 x 是偶数，则第 9 行的 continue 语句将结束本次循环，开始下次循环。因此，每个偶数将被忽略；而对于奇数，将执行其后的第 10 行语句累加和。

01 # while True 循环

02 sum=0

03 x=0

04 while True:

05 x=x+1

06 if x>100:

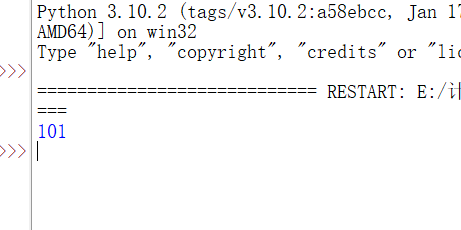
07 break

08 if x%2==0:

09 continue

10 sum+=x

11 print(sum)



**实验 5：编程输出一个 m 行 n 列的字符矩形。**

要求：输入行数 m、列数 n、组成矩形的字符，输出字符矩形。

提示：输入代码如下。

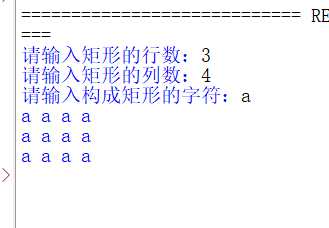
01 #输出字符矩形

02 m=int(input("请输入矩形的行数："))

03 n=int(input("请输入矩形的列数："))

04 ch=input("请输入构成矩形的字符：")

……



**第 5 章 Python 的组合数据类型**

**实验 1：完成教材例题。**

要求：（1）重点掌握列表、字典定义和常用操作。（2）掌握元组、集合的操作。

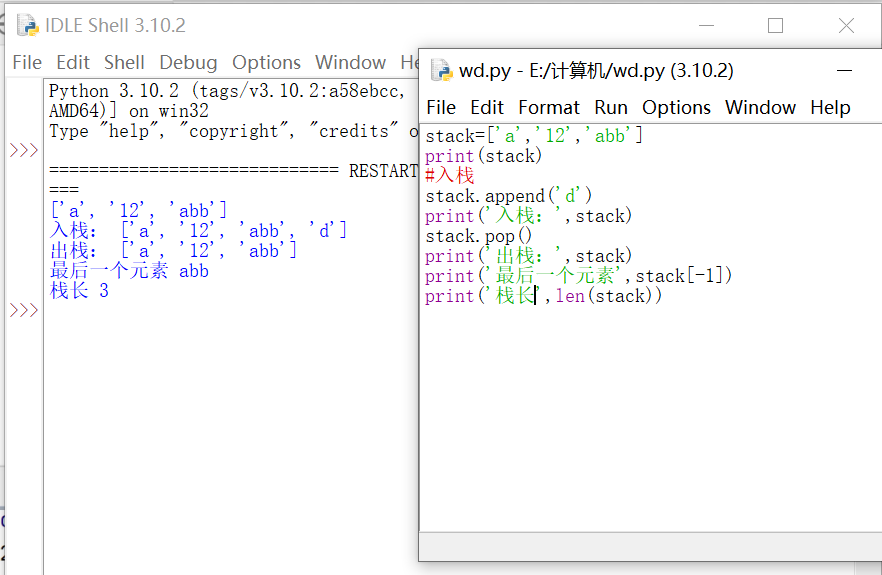
**实验 2：编写程序，模拟栈操作。 堆栈：先入后出/队列：先入先出**

要求：栈的操作包括入栈、出栈、查看栈顶元素、栈的长度、栈是否为空、退出。

提示：

（1）使用列表实现。列表是组合数据类型中非常重要的一种。列表的常用操作，包括添加列表元素 append()、移除并返回列表的最后一个元素 pop()、列表元素个数 len()，以及根据索引访问列表元素，并以此模仿栈的操作。

（2）列表的元素可以用索引表示，如 stack[i]表示 stack 列表中的第 i-1 个元素，索引可以为负数；stack[-1]表示列表中的最后一个元素。在不确定列表长度的情况下，用这种方法表示列表的末尾元素十分方便。



**实验 3：编写程序，完成字典操作。**

要求：（1）建立字典 dict，包含以下内容："数学":"L04","语文":"W01","英语":"W02","物理

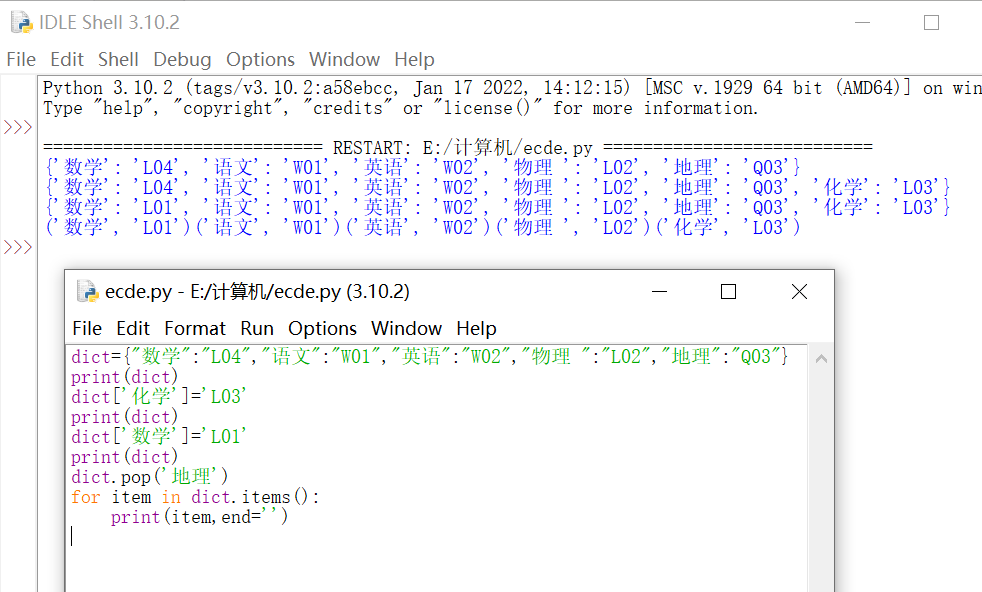
":"L02","地理":"Q03"。

（2）向字典中添加键值对"化学":"L03"。

（3）修改"数学"对应的值为"L01"。

（4）删除"地理"对应的键值对。

（5）输出字典 dict 的全部信息。



**实验 4：列表中的数据统计的实现。**

要求：定义一个列表，存储一个班级所有学生某科目考试的成绩等级，编写程序，统计输出各级别的人数，并按各级别人数的倒序输出。

输出结果如下。

良好:9

及格:7

优秀:5

不及格:3

提示：（1）程序数据代码如下。

01 demo0404.py

02 ls\_score=["及格","优秀","良好","优秀","及格","不及格","良好","及格",\

"良好","不及格","良好","良好","良好","及格","及格","优秀",\

"优秀","良好","不及格","良好","及格","良好","及格","优秀"]

03 d={}

……

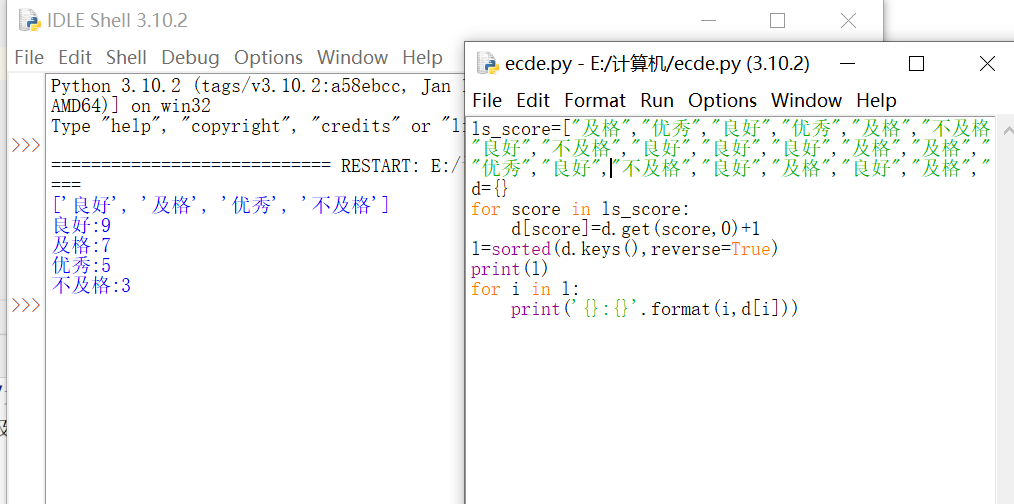
（2）首先定义一个空字典 d，然后采用 for 循环从 ls\_score 列表中逐一取出每个元素，即成绩等级，并判断这个元素是否在 d 中，这里使用了 d.get()函数。字典的 get()函数用于从字典中获取键对应的值，如果键不在字典中，给出默认值。第 5 行，d.get(score,0)将成绩等级 score 作为键检索字典，如果该键在字典中，则返回该键的值，即该等级的次数，再加 1，表示该等级又出现了一次；如果该键不在字典中，则返回 0，再加 1，并将 1 赋值给 [score]，相当于在字典中增加一个新元素，其键为 score，值为 1。字典添加元素与修改元素的方法相同，都是 dicts[key]=value 的形式。

参考代码如下。

04 for score in ls\_score:

05 d[score]=d.get(score,0)+1

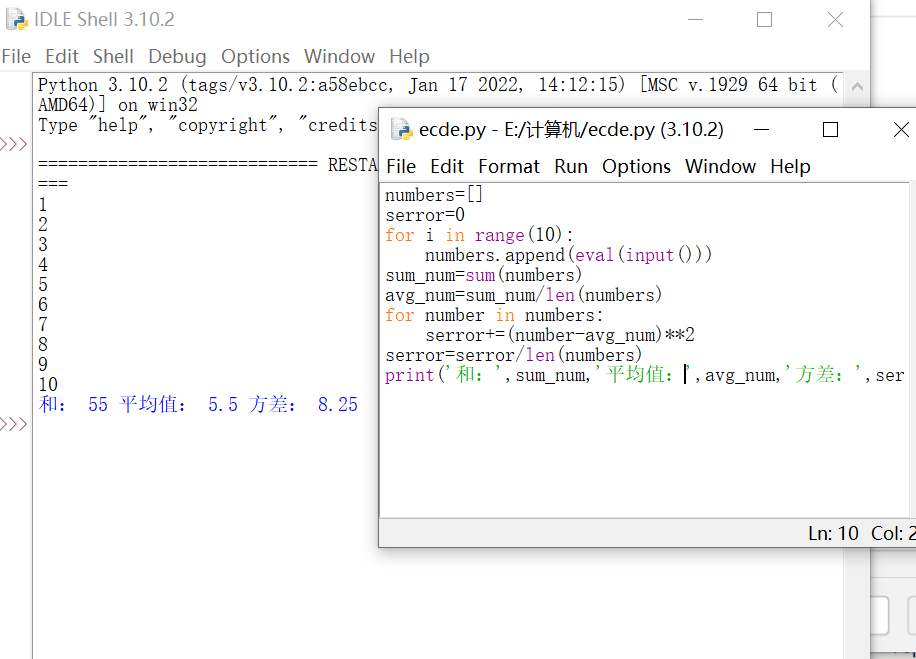
……



**实验 5：编写程序，计算和、平均值、方差**

要求：输入一组 10 个整数，存放在列表 numbers 中，计算这组数的和、平均值、方差。

提示：请自行搜索计算方差的公式。



**第 6 章 用函数实现代码复用**

**实验 1：编程求两点之间的曼哈顿距离。**

要求：主程序代码如下，实现函数 abs()。

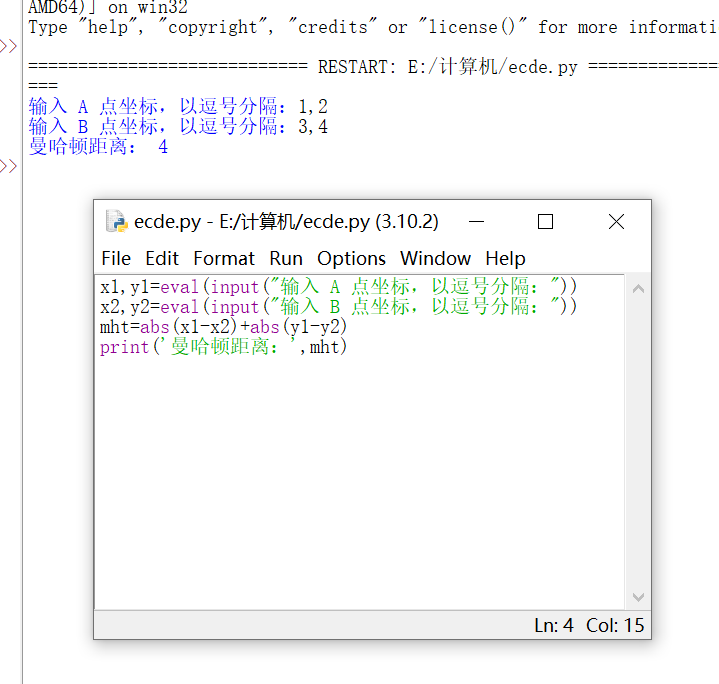
01 #主程序

02 x1,y1=eval(input("输入 A 点坐标，以逗号分隔："))

03 x2,y2=eval(input("输入 B 点坐标，以逗号分隔："))

04 mht=abs(x1-x2)+abs(y1-y2)

提示：给出平面上两个点的坐标，平面上点 A(x1,y1)与点 B(x2,y2)的曼哈顿距离为：|x1-x2|+|y1-y2|。



**实验 2：编写函数，计算某班级学生考试的平均分。**

要求：（1）班级共 10 人，计算平均分时可以根据全部人数或者实际参加考试人数计算。（2）完成 avgScore()函数。

01 #demo0503.py

02 def avgScore(scores,n=10):

03 ……

……

07 #主程序

08 scores=[90,88,76,45,77,95,66,88,91]

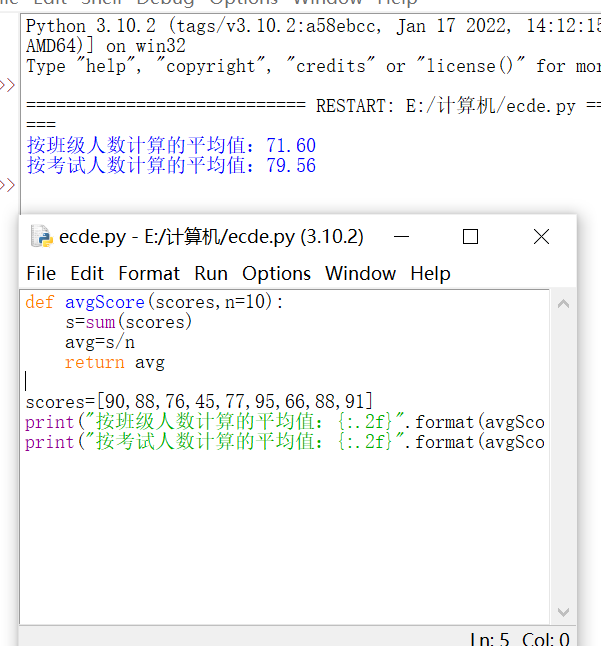
09 print("按班级人数计算的平均值：{:.2f}".format(avgScore(scores)))

10 print("按考试人数计算的平均值：{:.2f}".format(avgScore(scores,len

(scores))))

提示：（1）定义函数 avgScore()时，参数 n 为默认参数，其默认值为 10。在调用函数 avgScore() 时，如果没有传入 n 的实参，则 n 取默认值；如果传入 n 的实参，则函数会使用传递给 n 的新值。

1. 函数 avgScore()用于计算考试成绩的平均分，接收列表类型的参数 scores，遍历列表 scores，计算元素的累加和 s，返回 s/n 的值。



**实验 3：编写函数，利用可变参数计算一组数的最大值。**

提示：部分代码如下，请补全代码。

01 #demo0504.py

02 def maxnum(\*nums):

03 ……

04

05

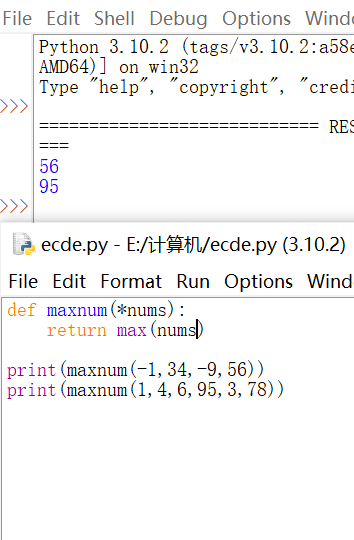
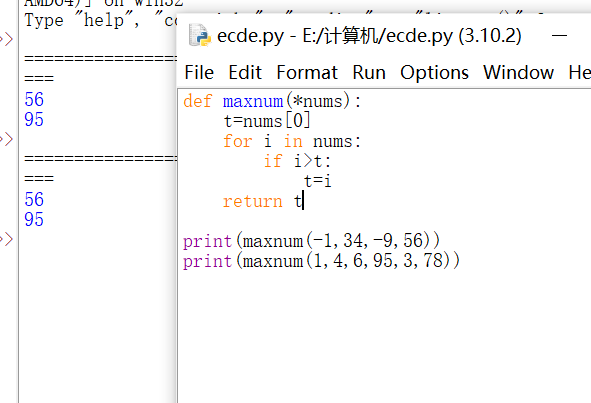
06

07

08 #主程序

09 print(maxnum(-1,34,-9,56))

10 print(maxnum(1,4,6,95,3,78))

**实验 4：分析程序的功能及程序中变量的作用域。**

要求：程序代码如下，分析程序中变量 s、ls 的作用范围。

01 #demo0507.py

02 def addLetter(a):

03 global s

04 s+=1

05 ls.append(a)

06 return

07 #主程序

08 ls=['G','g']

09 s=0

10 while True:

11 x=input("请输入要加入列表的字符：")

12 addLetter(x)

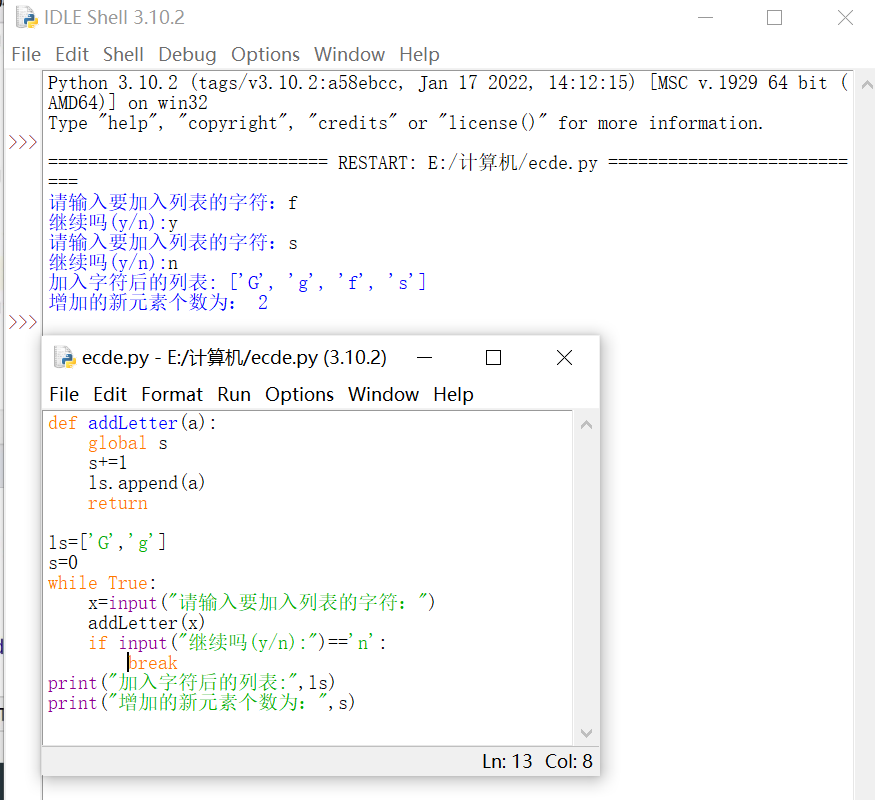
13 if input("继续吗(y/n):")=='n':

14 break

15 print("加入字符后的列表:",ls)

16 print("增加的新元素个数为：",s)

提示：函数中的变量默认为局部变量，作用域为函数内部，即使与函数外部变量同名，也不会影响函数外部变量的值。而用 global 声明的变量为全局变量，作用域为整个程序。列表等组合数据类型作为一种特殊的参数，如果在函数内部没有重新定义，则默认为全局变量。



**第 7 章 用类实现抽象和封装**

**实验 1：运行下面的程序，回答问题。**

要求：程序代码如下，（1）说明程序的执行过程；（2）程序运行结果是什么？

01 #demo1101.py

02 class FatherClass:

03 value=100 #类属性

04 def function1(self):

05 print("self.value=",self.value);

06 print("FatherClass.value=", FatherClass.value)

07

08 class ChildClass(FatherClass):

09 value=200 #类属性

10 def function1(self):

11 super().function1()

12 print("super().value=",super().value)

13 print("self.value=",self.value)

14 print("ChildClass.value=", ChildClass.value)

15

16 #主控程序

17 cc=ChildClass()

18 cc.function1()

19 print("用对象访问，cc.value=",cc.value)

20 print("用类访问，ChildClass.value=",ChildClass.value)

提示：

（1）使用 super()函数调用父类中的属性和方法的程序。

（2）使用 super().function1()语句，调用父类的 function1()方法，此时父类尚未初始化，第 5 行 self.value 的值为子类的 value 值 200。父类的 function1()方法执行后返回，在子类ChildClass 中继承执行第 12 行，输出父类的 value 值和子类的 value 值。

第 14 行用类名 ChildClass 访问子类自己的 value 值。

最后两行分别用对象名和类名访问子类的 value 值。

**实验 2：编写程序，重点理解构造方法在继承中的运用。**

要求：编写 Person 类，其具有 name、age、sex 等属性。继承于 Person 类的 Teacher 类，具有 title、quality、salary、prize 等属性。显示这些属性，并计算 salary、prize 之和。

**实验 3：设计并测试一个表示一个点的 MyPoint 类。**

要求：该类包括以下属性：

x：点的横坐标。

y：点的纵坐标。

包括如下方法：

\_\_init\_\_() (self, x, y)：构造方法，创建对象的同时为属性 x、y 赋初值。

getX()：获得点的横坐标。

getY()：获得点的纵坐标。

getDdistance (self,p)：返回当前点与点 p 之间的距离。

**第 8 章 使用模块和库编程**

**实验 1：调试完成教材例题。**

要求：掌握模块的概念、导入和执行模块的过程；重点掌握 Python 标准库中函数的使用方

法。

**实验 2：运行并解析 turtle 程序。**

要求：（1）分析程序的运行结果。（2）说明 turtle 库各函数的功能。

01 #turtle1.py

02 import turtle as t

03 def move(x,y):

04 t.penup()

05 t.goto(x,y)

06 t.pendown()

07

08 t.setup(800,300)

09 move(-300,0)

10 for x in range(1,5):

11 t.forward(100)

12 t.left(90)

13

14 move(-100,0)

15 for x in range(8):

16 t.forward(50)

17 t.left(45)

18

19 move(50,40)

20 for x in range(8):

21 t.forward(120)

22 t.right(225)

23

24 move(300,60)

25 for x in range(18):

26 t.forward(60)

27 if x%2==0:

28 t.left(175)

29 else:

30 t.left(225)

31 t.done()

提示：第 2 行导入 turtle 库，并命名为 t。第 3～6 行的 move()函数的功能是移动画笔的位置，后面的代码用于分别绘制四边形、八边形和不同的星形。

**实验 3：编写程序，使用 turtle 库绘制图形。**

要求：（1）使用 turtle 库绘制红色五角星图形，效果如图 8-1 所示。

（2）使用 turtle 库绘制花形图形，效果如图 8-2 所示。



图 8-1 程序运行效果 1



图 8-2 程序运行效果 2

**实验 4：中文文本分析与统计。**

要求：给出一篇文档“wind.txt”，统计其中词频前 3 位的词及出现次数。

提示：中文文本分析与统计通常是将文件操作与第三方库 jieba 库、字典、列表等知识点结合在一起，重点在于知识综合运用能力。

（1）导入第三方库 jieba 库，该库用于中文分词。

（2）使用当前操作系统默认的编码方式，打开文本文件“wind.txt”，并将文件内容读取到变量 txt 中。

（3）使用 jieba 库的 lcut()方法将 txt 变量分词，分词结果保存到列表 ls 中。

（4）完成词频统计功能，并保存在字典中；将字典中的统计结果保存到列表 freq\_word中，然后根据出现频次排序，最后输出。

**第 9 章 Python 的文件操作**

**实验 1：完成教材例题 ex0902.py~ex0909.py。**

要求：（1）掌握文件打开和关闭方法。（2）掌握文件的读写操作。

**实验 2：逐一统计文本文件中的字符数。**

要求：统计某一文本文件中的大写字母、小写字母和数字出现的次数。

提示：

（1）读取文件内容后，使用字符串对象的内置方法 isupper()、islower()、isdigit()判断字符的类别。（2 打开文件时，未指定打开模式，默认为 r 模式，编码方式为 UTF-8，可以写为 file=open("wind.py",'r',encoding= 'utf-8')。

（3）通过一个 while 循环逐一判断文件中的每一个字符是否是大写字母、小写字母或数字。

**实验 3：计算文件中数值数据的算术平均数和中位数。**

要求：（1）文本文件 numbers.txt 中给出了若干数据，部分数据如下：

647

862

889

1436

……

（2）参照如下格式输出：

算术平均数：3428.96

中位数：3966.5

提示：（1）算术平均数。算术平均数（arithmetic mean）是统计学中最常用的一种平均指标，又称均值，主要用于未分组的原始数据。设一组数据为 X1，X2，…，Xn，算术平均数 m 的计算公式为 m=(X1+X2+…+Xn)/n

（2）中位数。中位数（median）是统计学中的专有名词，又称中点数或中值。中位数是按顺序排列的一组数据中居于中间位置的数。

设有一组数据，按从小到大的顺序排序为 X1，X2，…，Xn，中位数用 m0.5 表示，当 n 为奇数时，中位数的计算公式为 m0.5=X（n-1）/2

当 n 为偶数时，中位数的计算公式为 m0.5=(X（n/2）+ X（n/2-1）)/2

**实验 4：编程实现文本文件的加密和解密功能。**

要求：程序使用 key 作为参数，对给定的文本文件执行加密运算，加密后的文件输出到另一文本文件中。

提示：加密算法是对于文件中的每个字母，用字母表中其后第 n 个字母来替代，其中 n 为密钥。加密后的文件可以用密钥-n 来解密。

例如，文件内容如果是“abc123<(”，密钥是 4，则加密后的文件是“efg567@,”。

**第 10 章 异常处理**

**实验 1：完成教材例题 ex1003.py~ex1007.py。**

要求：（1）掌握异常处理的基本方法。（2）掌握异常处理各关键字的使用。

**实验 2：阅读程序 except01.py，回答问题。**

要求：（1）描述程序的功能。

11（2）程序的第 11 行和第 13 行，为什么要进行 if source!=None 和 if target!=None 的判断？

（3）查阅文献，说明 IOError 异常的功能。

01 # except01.py

02 source=target=None

03 try :

04 source=open("temp.py",encoding="utf8")

05 print(source.read(9))

06 target=open("801.txt","w+")

07 target.writelines(source.readline())

08 except (FileNotFoundError,IOError):

09 print("没有找到文件或读写失败")

10 finally:

11 if source!=None:

12 source.close()

13 if target!=None:

14 target.close()

提示：finally 语句块是一定会被执行的语句块，其中的代码也可能抛出异常。在程序中，第4 行的 temp.py 文件如果不存在，第 12 行执行 source.close()语句时，将会因对象不存在而抛出异常。为避免因 source 对象不存在而导致程序因异常退出，增加一个判断语句，使程序更为健壮。

**实验 3：**编写异常处理程序。

要求：输入姓名和月工资计算年薪，如果输入格式不正确则抛出异常**。**

提示：程序中的异常可能来自两个方面，一是输入工资信息时输入了非数字格式，会抛出ValueError 异常；二是输入工资范围不正确，如输入了负值或数值太大，这是一个用户自定义的异常。

**实验 4：**编写异常处理程序。

要求：定义一个 Circle 类，其中有求面积的方法，当半径小于 0 时，抛出一个用户自定义异常。

**第 11 章 tkinter GUI 编程**

**实验 1：完成教材例题 ex1101.py~ex1108.py。**

要求：（1）掌握 tkinter GUI 编程的基本方法。（2）掌握 tkinter GUI 编程布局的方法。

**实验 2：编写图形用户界面的应用程序。**

要求：编制求两个正整数的最小公倍数的程序。（1）两个文本框 txt1、txt2，用来输入整型数据；（2）一个按钮；一个不可编辑的文本组件 txt3。当单击按钮时，在 txt3 中显示两个整数的最小公倍数。

提示：根据题目要求，设计出与图 11-1 所示类似的界面。



图 11-1 程序运行结果

**实验 3：编写图形用户界面的应用程序。**

要求：计算用户输入的若干整数之和，整数之间用英文逗号分隔。运行结果如图 11-3 所示，在文本框中输入数据后，单击“计算”按钮，输出结果。

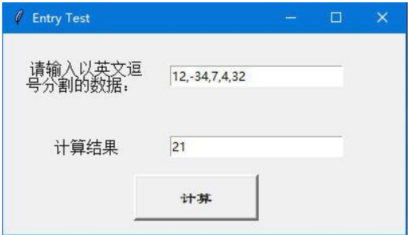


图 11-2 程序运行结果

**实验 4：编写图形用户界面的应用程序。**

要求：设计GUI界面，模拟QQ登录界面，输入用户名和密码，如果输入正确，提示登陆成功；否则提示登陆失败。

**第 12 章 Python 的数据库编程**

**实验 1：下载和安装 SQLite 数据库**

要求：（1）从网址 http://www.sqlite.org/download.html 下载并安装数据库。

（ 2 ）测试 SQLite3 交 互 模 式 常 用 的 命 令 ， 包 括 sqlite3.exe [dbname] 、 .open dbname、.databases、.tables、.schema [tbname]、.help 等。

**实验 2：在 sqlite 数据库中的操作**

要求：在 sqlite 数据库中，创建数据库，并完成数据的插入、更新和删除操作（参考 12.4 节）。

提示：插入、更新、删除记录的一般步骤如下。

（1）建立数据库连接。

（2）创建游标对象 cur，使用 cur.execute(sql)方法执行 SQL 的 insert、update、delete 等 语句，完成数据库记录的插入、更新、删除操作，并根据返回值判断操作结果。

（3）提交操作。

（4）关闭数据库。

已有test.db数据库和employee表，表中初始数据如下。

[1132, 李四，男，经理，7548.6，11；

1443，王五，男，职员，6656，14；

1036，高七，女，经理，7600，10]

**实验 3：Python实现QQ登陆数据库编程**

设计GUI界面，模拟用户登录功能，用户输入用户名和密码，如果正确提示登陆成功；否则提示登陆失败，

用户的密码信息保存在SQLite数据库中。

注意：有sqlite数据库userData.db, 其中有表users，其结构为（username，pwd），

并在该表中保存了用户登陆信息。

username,pwd

David，123456

John，654321

Lily，345678

Lucy，876543

Mary，567890

**实验 4：Python的数据库编程练习**

创建sqlite数据库nameList.db并建立连接，在数据库中创建user\_id表，并存入如下数据，建立自动id。提取id>2的数据元组打印输出，输出时将表头信息一并打印输出。

存入：('zhang3','123456','male'),

('zhao4','123456','male'),

('wang5','123456','female'),

('li6','123456','female')

**第 13 章 科学计算与图表绘制**

**实验 1：创建 numpy 数组。**

要求：（1）使用 array()函数、empty()函数、zeros()函数、linspace()函数等创建 numpy 数组。

1. 使用 numpy 数组的索引和切片方法访问数组元素。

**实验 2：阅读程序并分析 numpy 库的运算功能。**

要求：（1）写出运行结果。

（2）分析运算功能。

01 >>> import numpy as np

02 >>> a1=np.array([1,2,3])

03 >>> b1=np.array(([1,2,3],[4,5,6],[7,8,9]))

04 >>> a1+100

05 \_\_\_\_\_\_\_\_\_\_\_

05 >>> b1\*2

07 \_\_\_\_\_\_\_\_\_\_\_

06 >>> a1

11 \_\_\_\_\_\_\_\_\_\_\_

07 >>> b1

13 \_\_\_\_\_\_\_\_\_\_\_

08

09 >>> a1+np.array([100,200,300])

19

10 >>> a1+b1

21 \_\_\_\_\_\_\_\_\_\_\_

11 >>> np.sum(b1)

25 \_\_\_\_\_\_\_\_\_\_\_

12 >>> np.sum(b1,axis=0)

27

13 >>> np.sum(b1,axis=1)

29 \_\_\_\_\_\_\_\_\_\_\_

**实验 3：绘制散点图。**

要求：绘制如图 13-1 所示的余弦三角函数 y=cos(2x)散点图。

提示：使用 import matplotlib.pyplot as plt 和 help(plt.scatter)命令查看绘制散点图的帮助信息。

**第 14 章 爬取和分析网页中的数据**

**实验 1：爬取网页中的数据。**

要求：使用 urllib 库和 requests 库分别爬取 http://www.sohu.com 首页的前 360 个字节的数据。

**实验 2：测试 BeautifulSoup 对象的方法。**

要求：（1）完成教材例题 ex1404.py，创建 BeautifulSoup 对象。

（2）测试 Tag 对象的 contents、children、string、strings、parent 等属性。

（3）测试搜索文档树的 find\_all()方法和 find()方法。

**实验 3：爬取并分析网页页面数据。**

要求：完成教材例题 ex1405.py，查看爬取数据的结果，注意比较与例题结果的不同。