2018-2019学年第2学期素质教育公选课

《Python语言程序设计》课程平时作业

**姓名：王乐卿**

**学号：2018091602007**

**学院：信息与软件工程学院**

**专业：信息与软件工程专业**

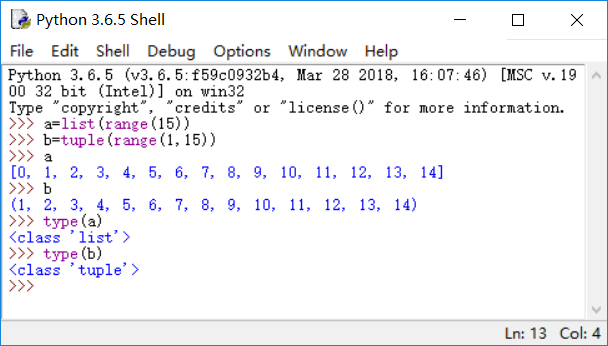
1. 先执行下列语句。

>>>a=list(range(15))

>>>b=tuple(range(1,15))

然后完成操作或回答问题：

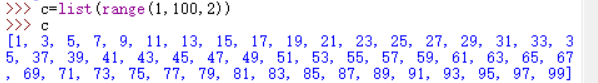
1. 显示变量a、b的值和变量a、b的数据类型



1. range()函数的作用是什么？range(15)与range(1,15)有何区别？

range（）的作用：调用格式为range（[start,]end[,step],默认start为0，默认step为1，产生相应的可迭代对象。range（15）等同于range（0，15），比range（1，15）多了一个0

1. 生成由100以内的奇数构成的列表c，请写出语句并验证。



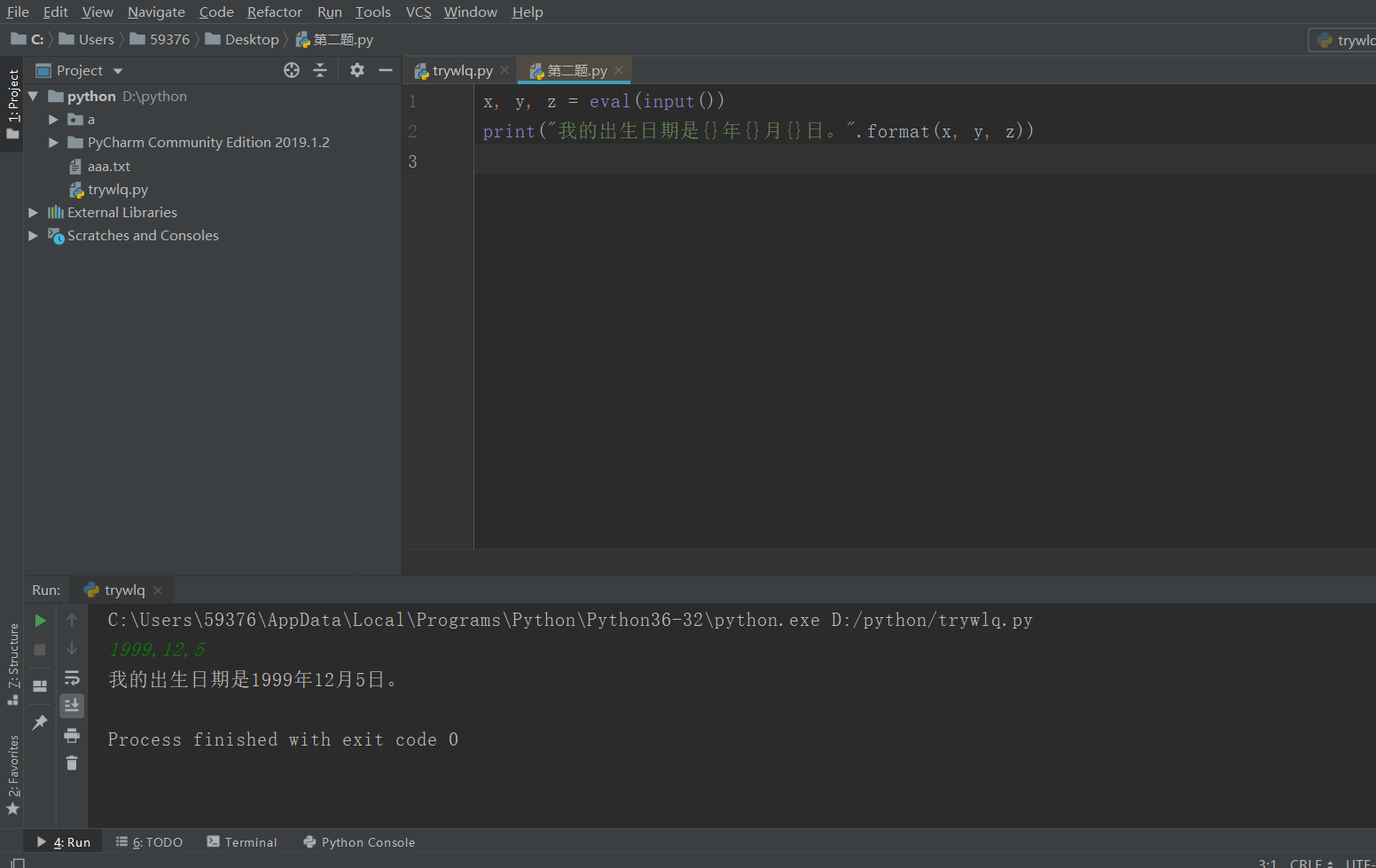
2. 编程实现：输入自己的出生年、月、日，按下列格式输出自己的出生日期信息。

1999,12,5↙

我的出生日期是1999年12月5日。

x, y, z = eval(input())

print("我的出生日期是{}年{}月{}日。".format(x, y, z))



3. 编程实现：输入一个整数，若为奇数则输出其平方根，否则输出其立方根。要求分别用单分支、双分支及条件运算实现。

import math

x = eval(input())

print("单分支实现:")

if x%2!=0:

print("{}".format(math.sqrt(x)))

if x%2==0:

print("{:f}".format(math.pow(x, 1.0/3)))

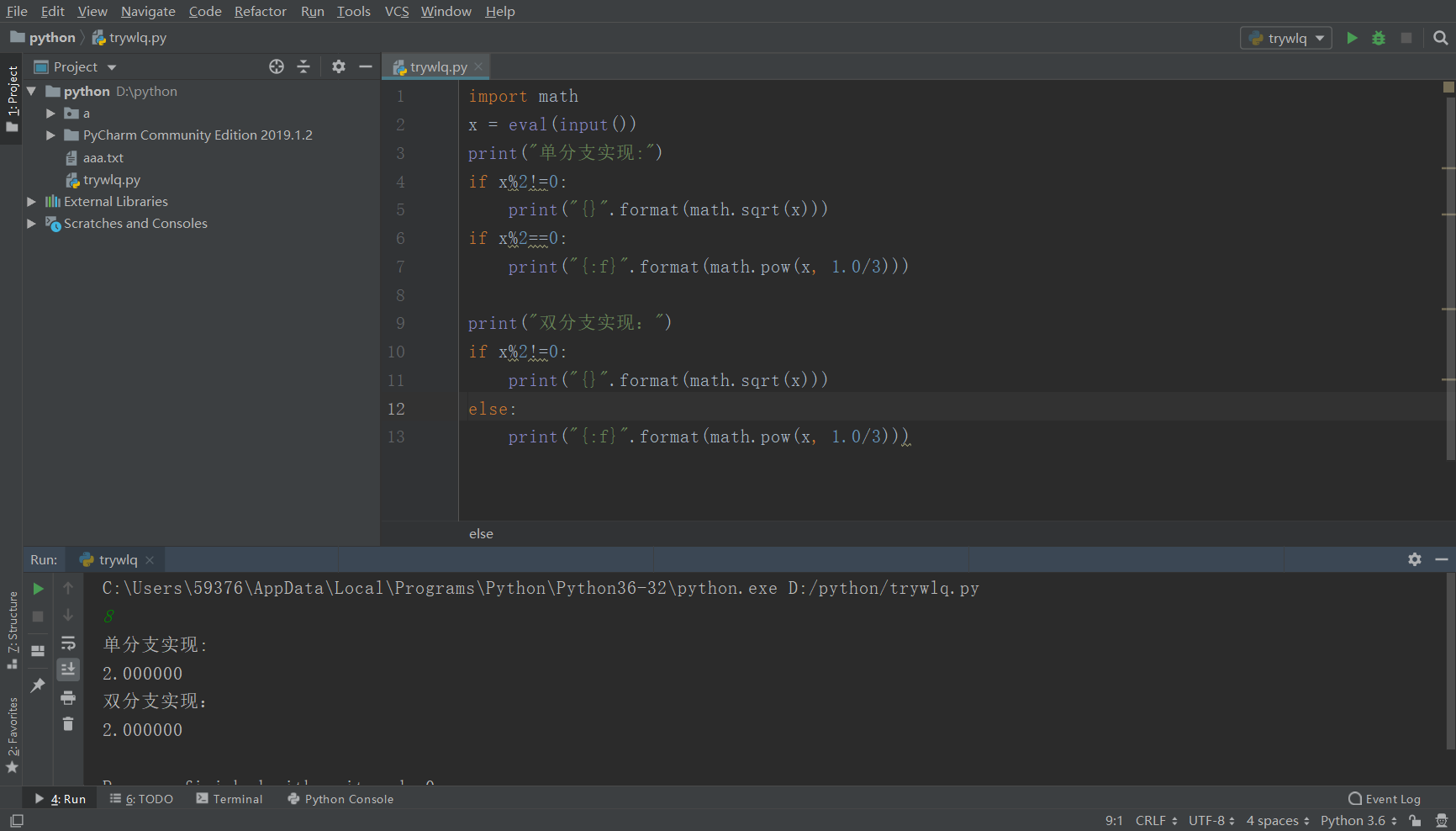
print("双分支实现：")

if x%2!=0:

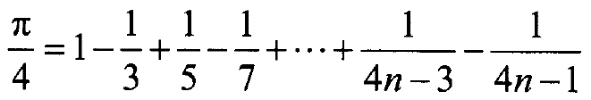
print("{}".format(math.sqrt(x)))

else:

print("{:f}".format(math.pow(x, 1.0/3)))



4. 编程实现：利用下列公式



1. n取1000时π计算的近似值。

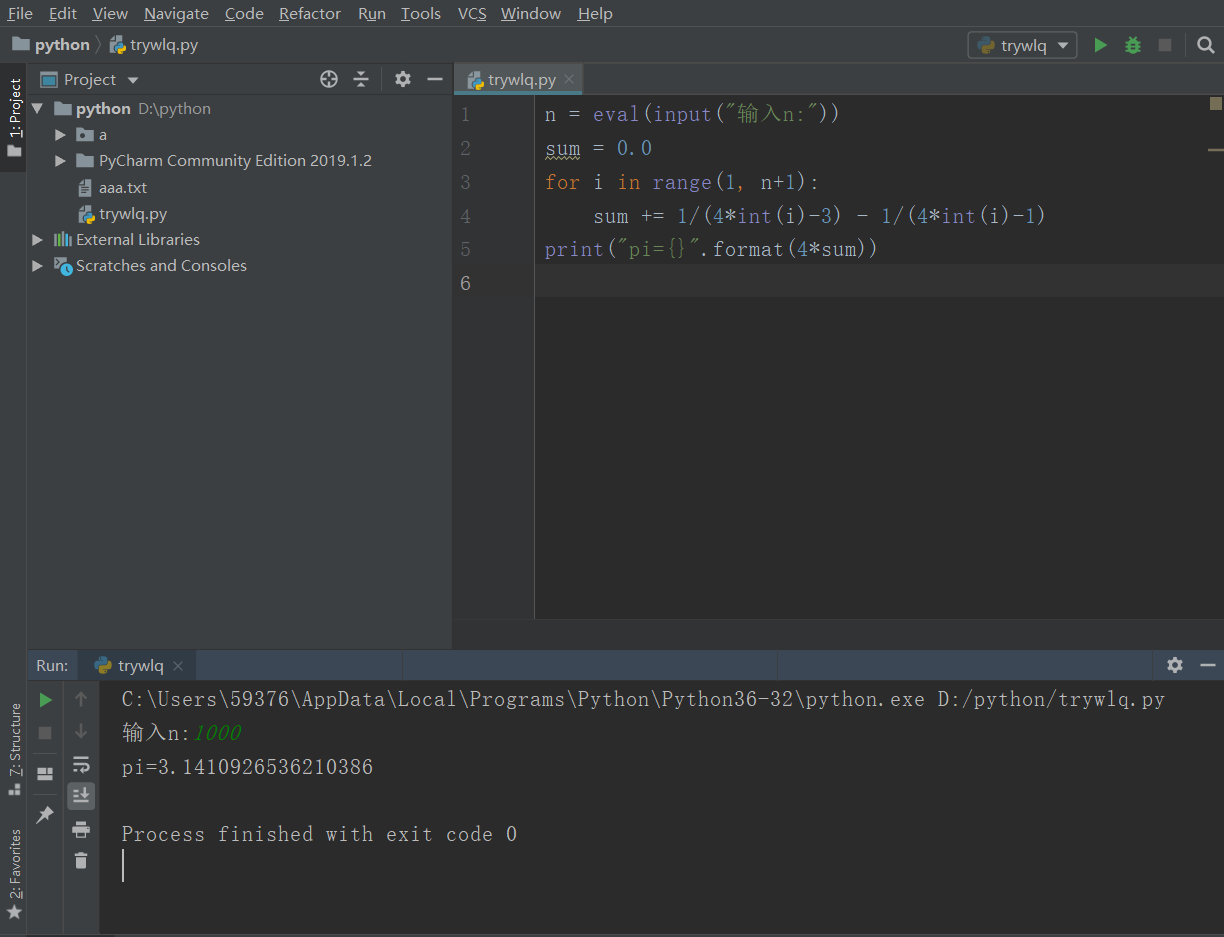
n = eval(input("输入n:"))

sum = 0.0

for i in range(1, n+1):

sum += 1/(4\*int(i)-3) - 1/(4\*int(i)-1)

print("pi={}".format(4\*sum))



1. 求π的近似值，直到最后一项的绝对值小于为止。

import math

i = 1

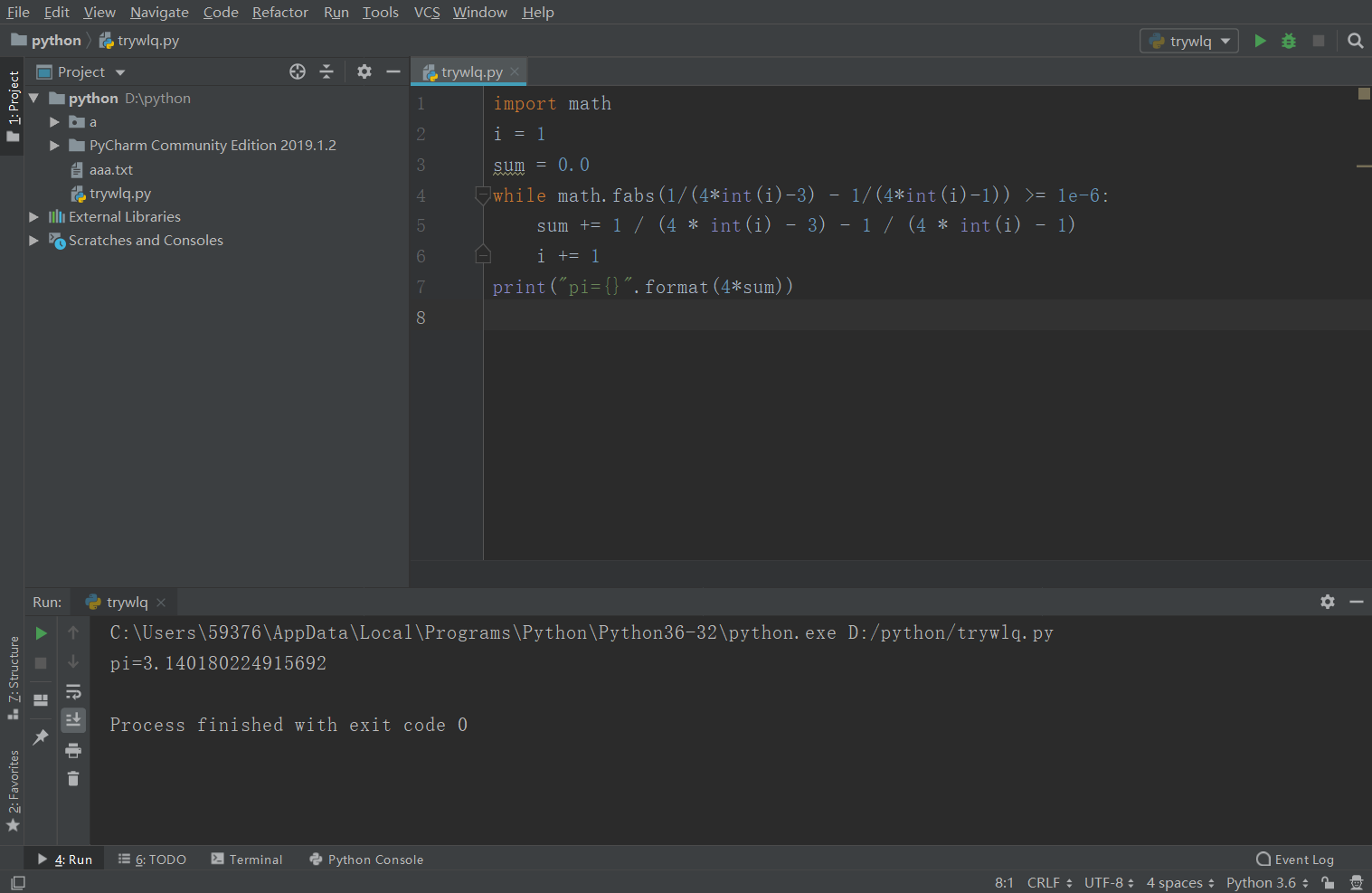
sum = 0.0

while math.fabs(1/(4\*int(i)-3) - 1/(4\*int(i)-1)) >= 1e-6:

sum += 1 / (4 \* int(i) - 3) - 1 / (4 \* int(i) - 1)

i += 1

print("pi={}".format(4\*sum))



5. 编程实现：输入一个字母标识符，计算标识符中各个字母的数值之和，其中A=a=1，B=b=2，…，Z=z=26。例如“uestc”的数值为21+5+19+20+3=68。

string = input('输入字母标识符：')

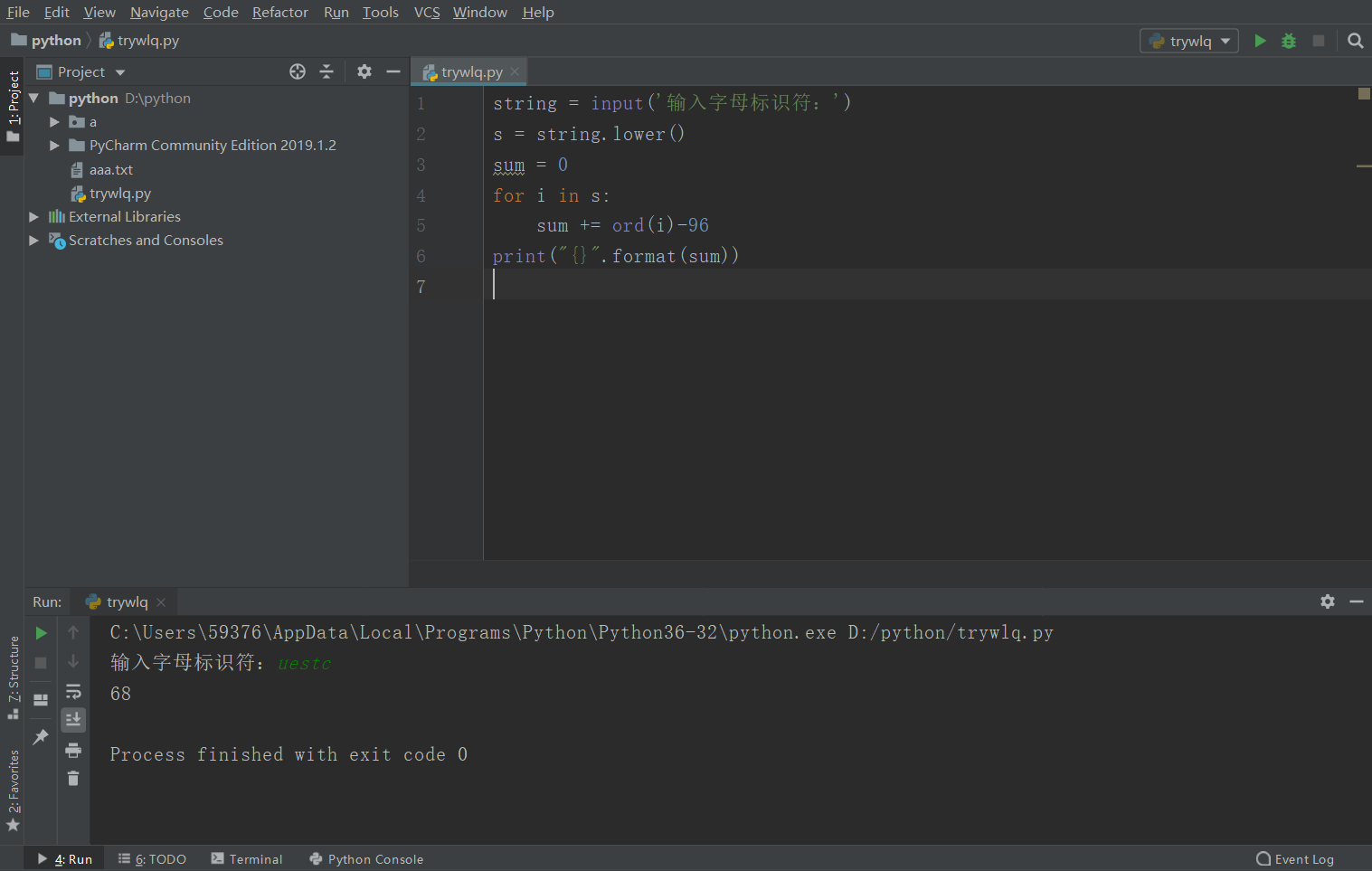
s = string.lower()

sum = 0

for i in s:

sum += ord(i)-96

print("{}".format(sum))



6. 编程输出如下6×6的矩阵：所有对角元素和靠边的元素全为1，其余元素全为0。并统计其0元素的个数。

x = [[1, 1, 1, 1, 1, 1],

[1, 1, 0, 0, 0, 1],

[1, 0, 1, 0, 0, 1],

[1, 0, 0, 1, 0, 1],

[1, 0, 0, 0, 1, 1],

[1, 1, 1, 1, 1, 1]]

sum0 = 0

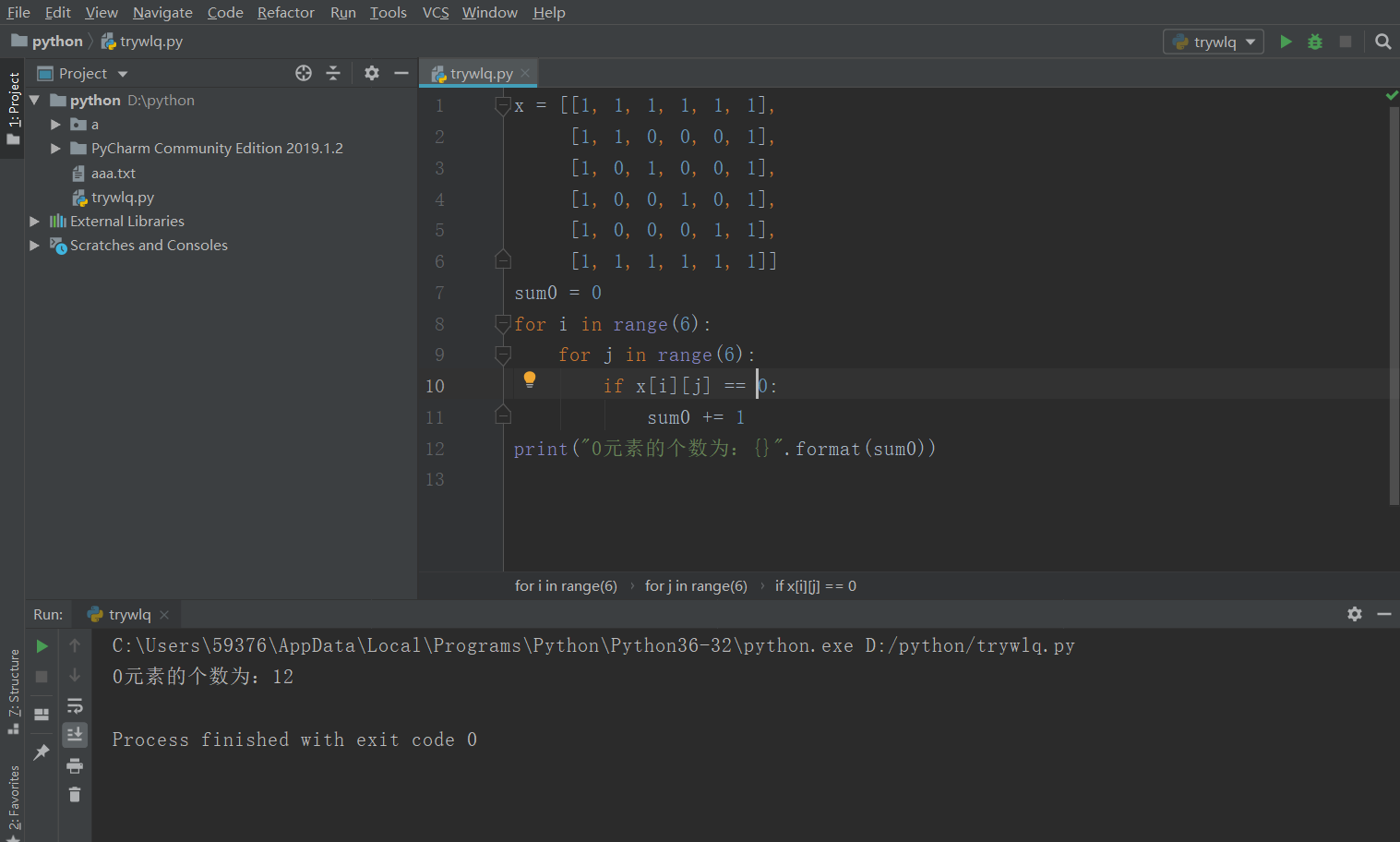
for i in range(6):

for j in range(6):

if x[i][j] == 0:

sum0 += 1

print("0元素的个数为：{}".format(sum0))



7. 编程实现：输入全班30名学生的姓名和成绩，输出其中最高分和最低分，并求全班同学的平均分。要求利用字典实现。

d = dict()

for i in range(1,31):

print('输入第{}个名字：'.format(int(i)))

name = input()

print('输入第{}个成绩：'.format(int(i)))

score = eval(input())

d.setdefault(name, score)

flag = 0

sum = 0.0

for v in d.values():

sum += v

if flag == 0:

max=min=v

flag = 1

else :

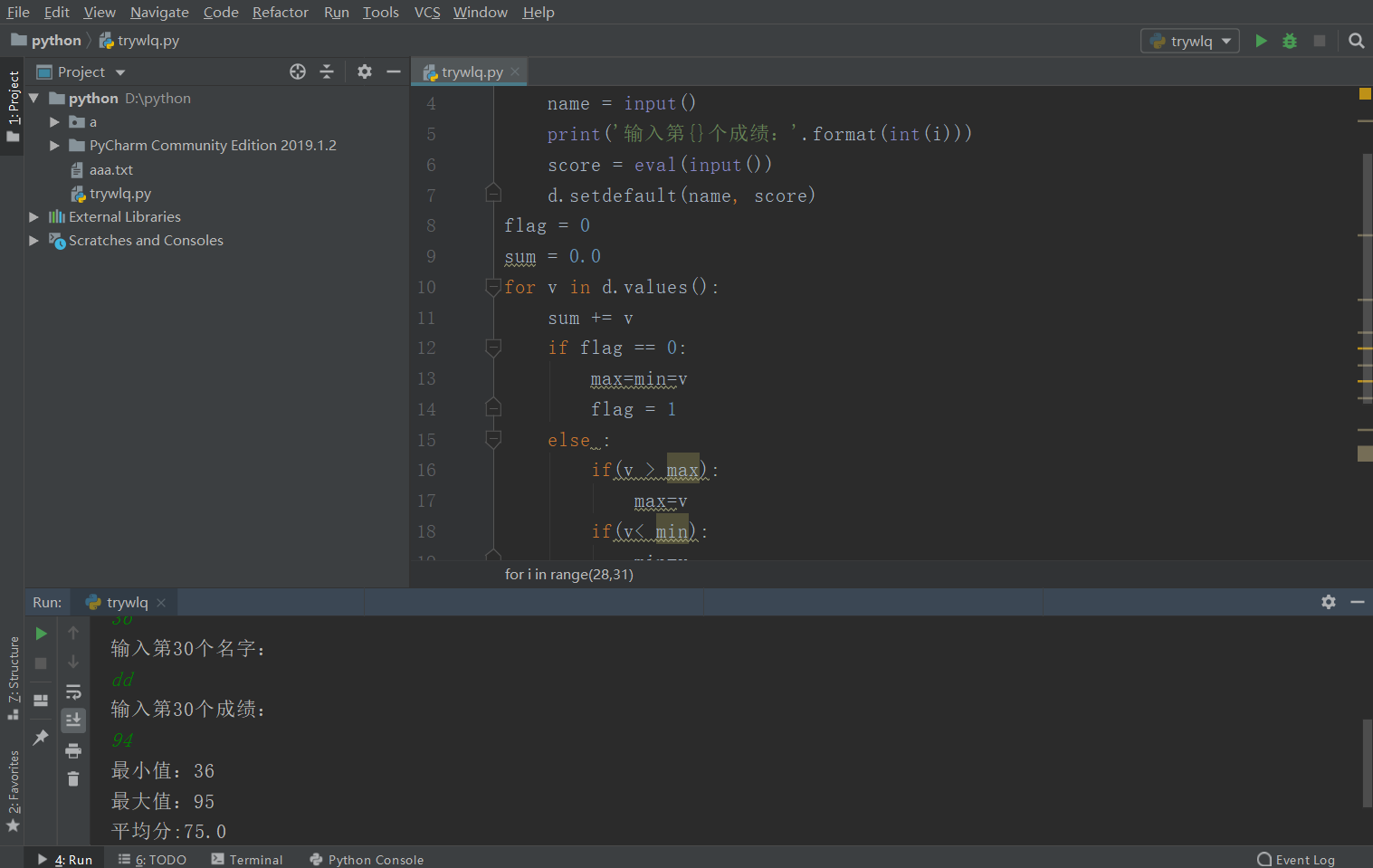
if(v > max):

max=v

if(v< min):

min=v

print('最小值：{}\n最大值：{}\n平均分:{}'.format(min,max,sum/len(d)))



8. 编程实现：随机生成10个[0，10]范围的整数，分别组成集合A和集合B，输出A和B的内容、长度、最大值、最小值以及它们的并集、交集和差集。

import random

A=set()

for i in range(10):

A.add(random.randint(0, 10))

B=set()

for j in range(10):

B.add(random.randint(0, 10))

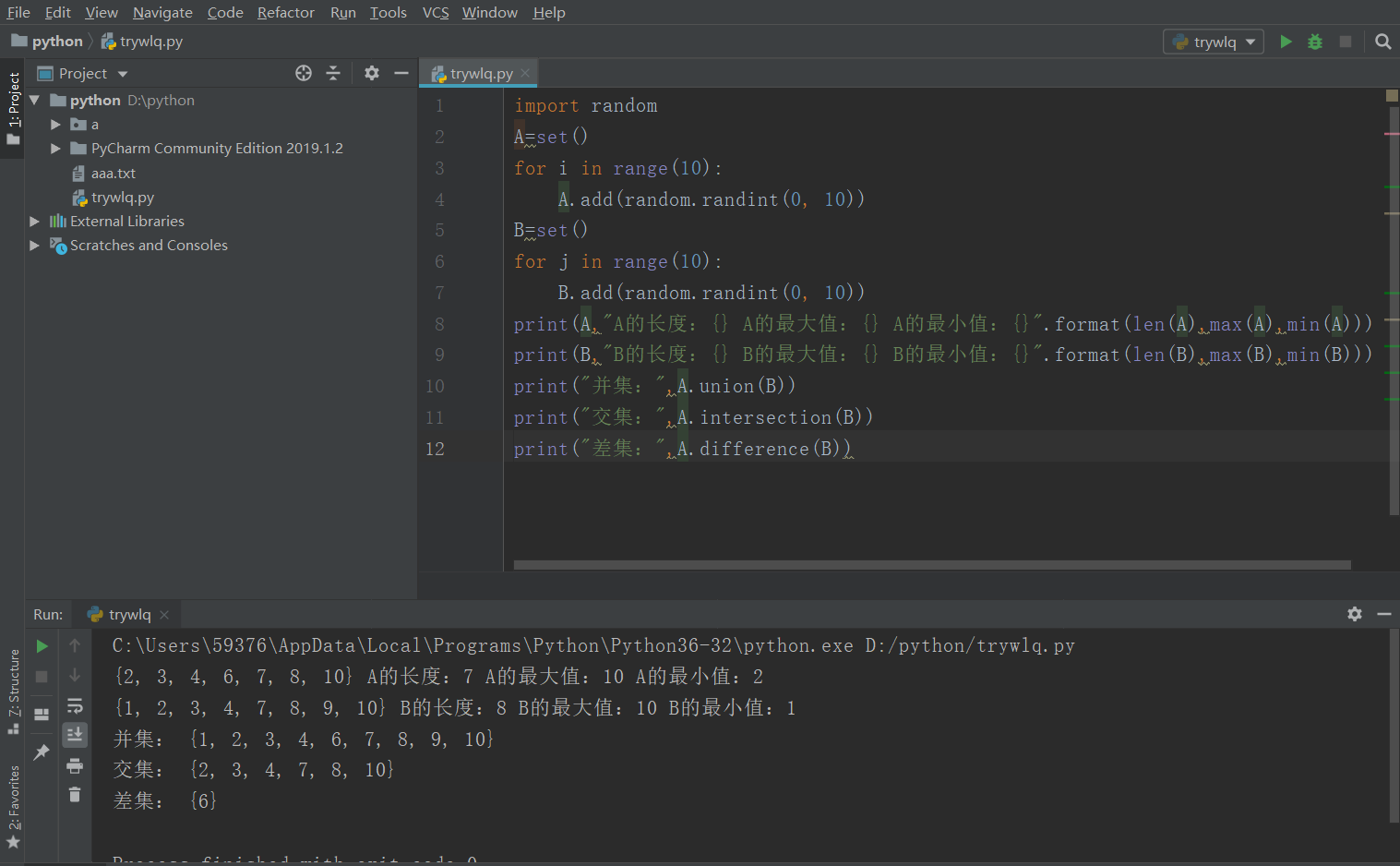
print(A,"A的长度：{} A的最大值：{} A的最小值：{}".format(len(A),max(A),min(A)))

print(B,"B的长度：{} B的最大值：{} B的最小值：{}".format(len(B),max(B),min(B)))

print("并集：",A.union(B))

print("交集：",A.intersection(B))

print("差集：",A.difference(B))



9. 编程实现：定义一个点类，为这个点类设置一个方法来计算其方向余弦。并为这个点类设置另一个方法来计算两个点之间的距离。

from math import sqrt

class POINT:

def \_\_init\_\_(self, m1, n1, z1, m2, n2, z2):

self.x1 = m1

self.y1 = n1

self.z1 = z1

self.x2 = m2

self.y2 = n2

self.z2 = z2

def direction\_cosine(self):

cosx = (self.x1 - self.x2) / sqrt((self.x1 - self.x2) \*\* 2 + (self.y1 - self.y2) \*\* 2 + (self.z1 - self.z2) \*\* 2)

cosy = (self.y1 - self.y2) / sqrt((self.x1 - self.x2) \*\* 2 + (self.y1 - self.y2) \*\* 2 + (self.z1 - self.z2) \*\* 2)

cosz = (self.z1 - self.z2) / sqrt((self.x1 - self.x2) \*\* 2 + (self.y1 - self.y2) \*\* 2 + (self.z1 - self.z2) \*\* 2)

print("({},{},{})".format(cosx, cosy, cosz))

def distance(self):

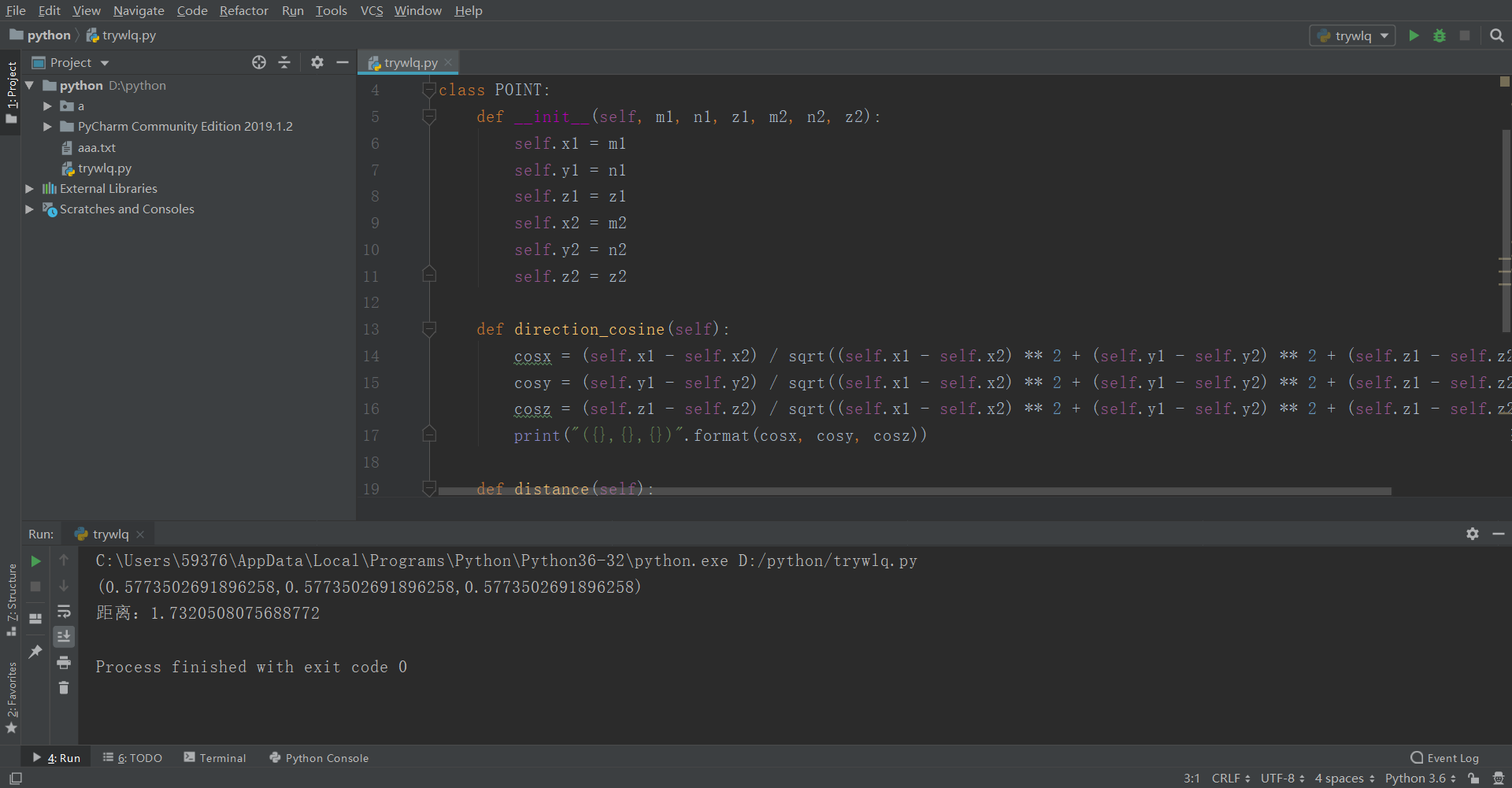
dis = sqrt((self.x1 - self.x2) \*\* 2 + (self.y1 - self.y2) \*\* 2 + (self.z1 - self.z2) \*\* 2)

print("距离：{}".format(dis))

a = POINT(1, 1, 1, 0, 0, 0)

a.direction\_cosine()

a.distance()



10. 读入Python源程序文件a.py，删除程序中的注释，再显示该文件的内容。

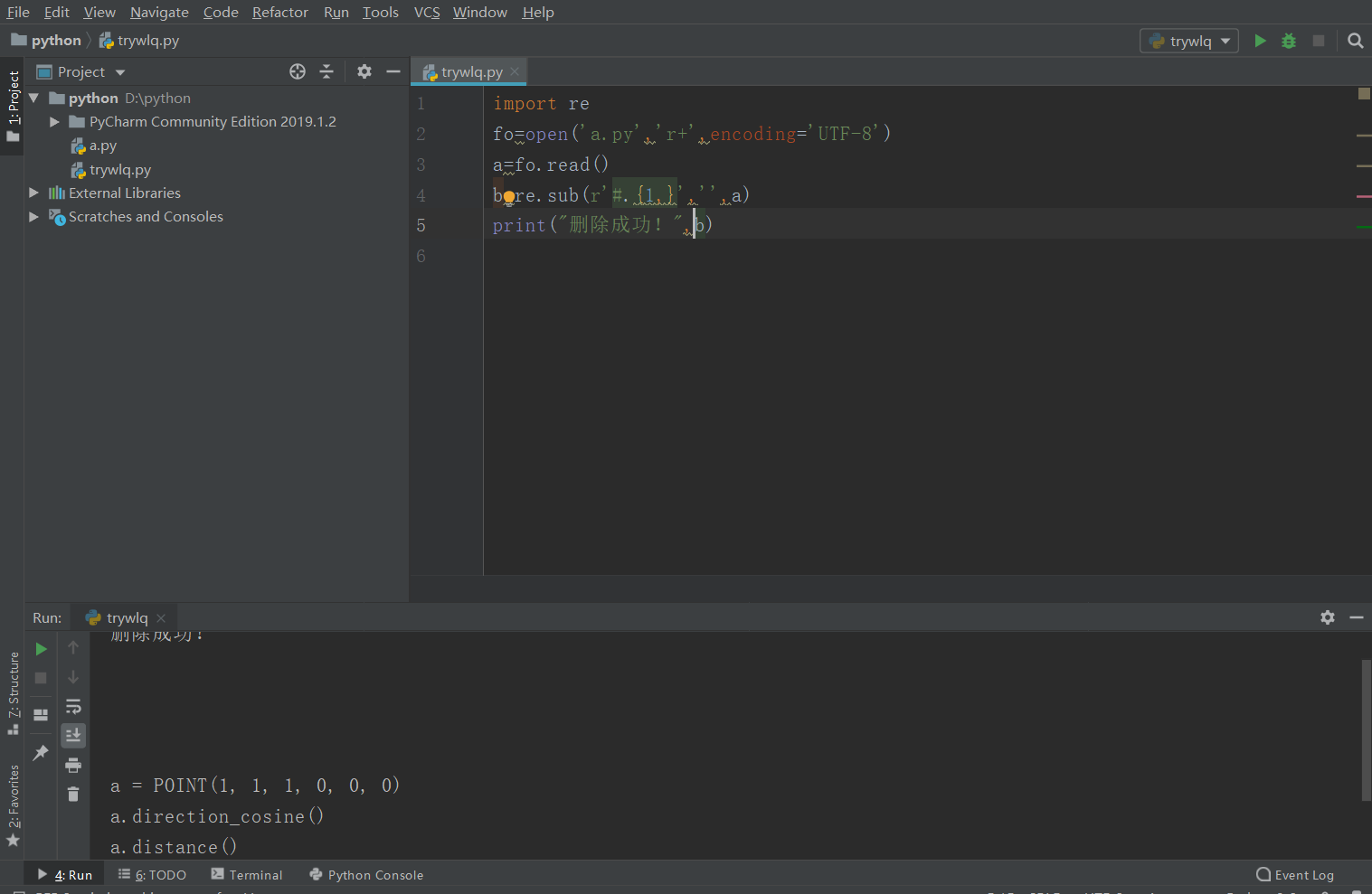
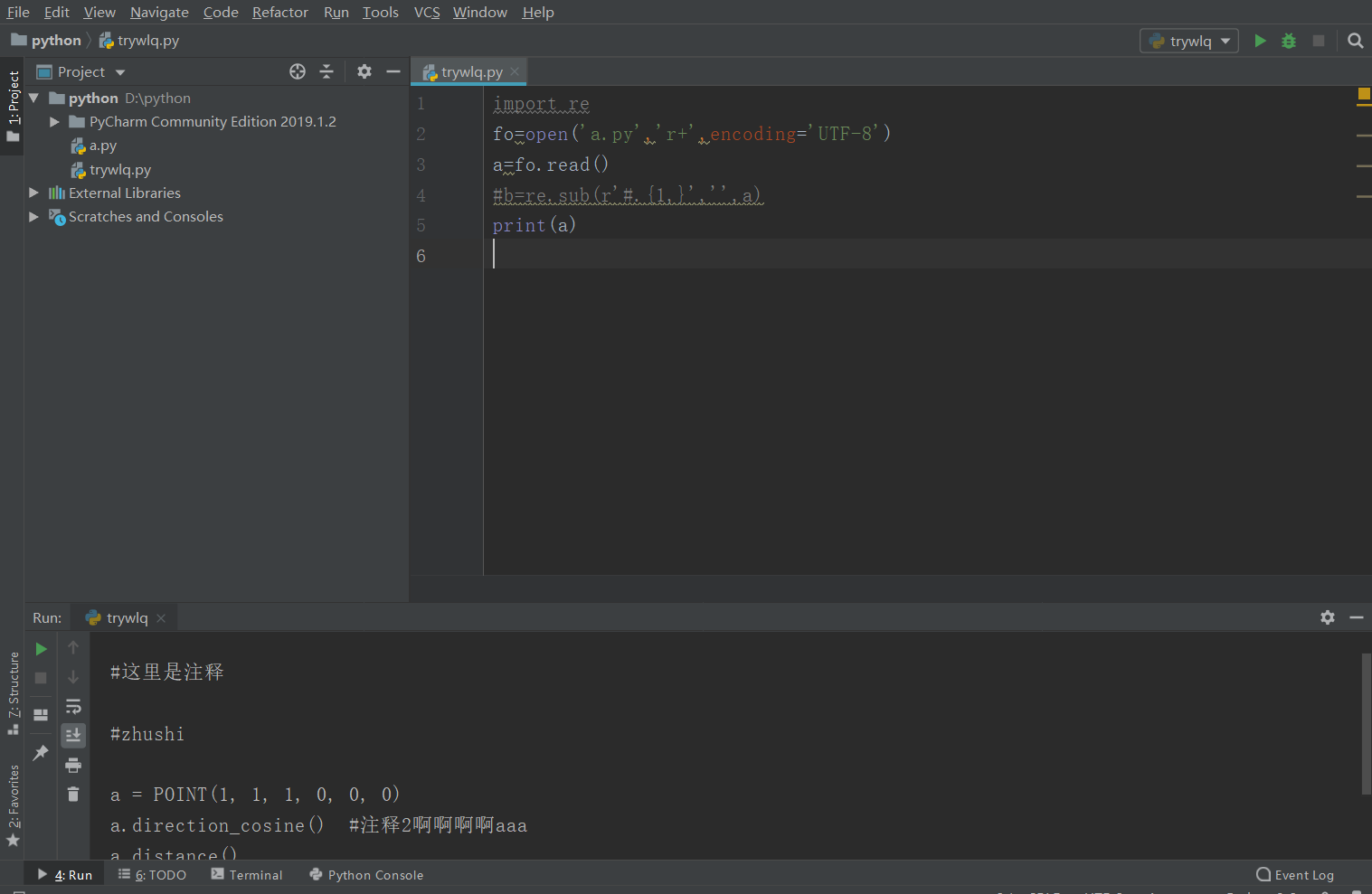
import re

fo=open('a.py','r+',encoding='UTF-8')

a=fo.read()

b=re.sub(r'#.{1,}','',a)

print("删除成功！",b)



11. 编程实现：从键盘输入x和y的值，计算y=ln(3x-y+1)的值。考虑用户输入数据的多种可能性，进行异常处理。

import math

def main():

x,y=eval(input())

try:

z=math.log(3\*x-y+1)

print(z)

except TypeError:

print("数据类型错误")

except ValueError:

print("3\*x-y+1不可以小于等于0")

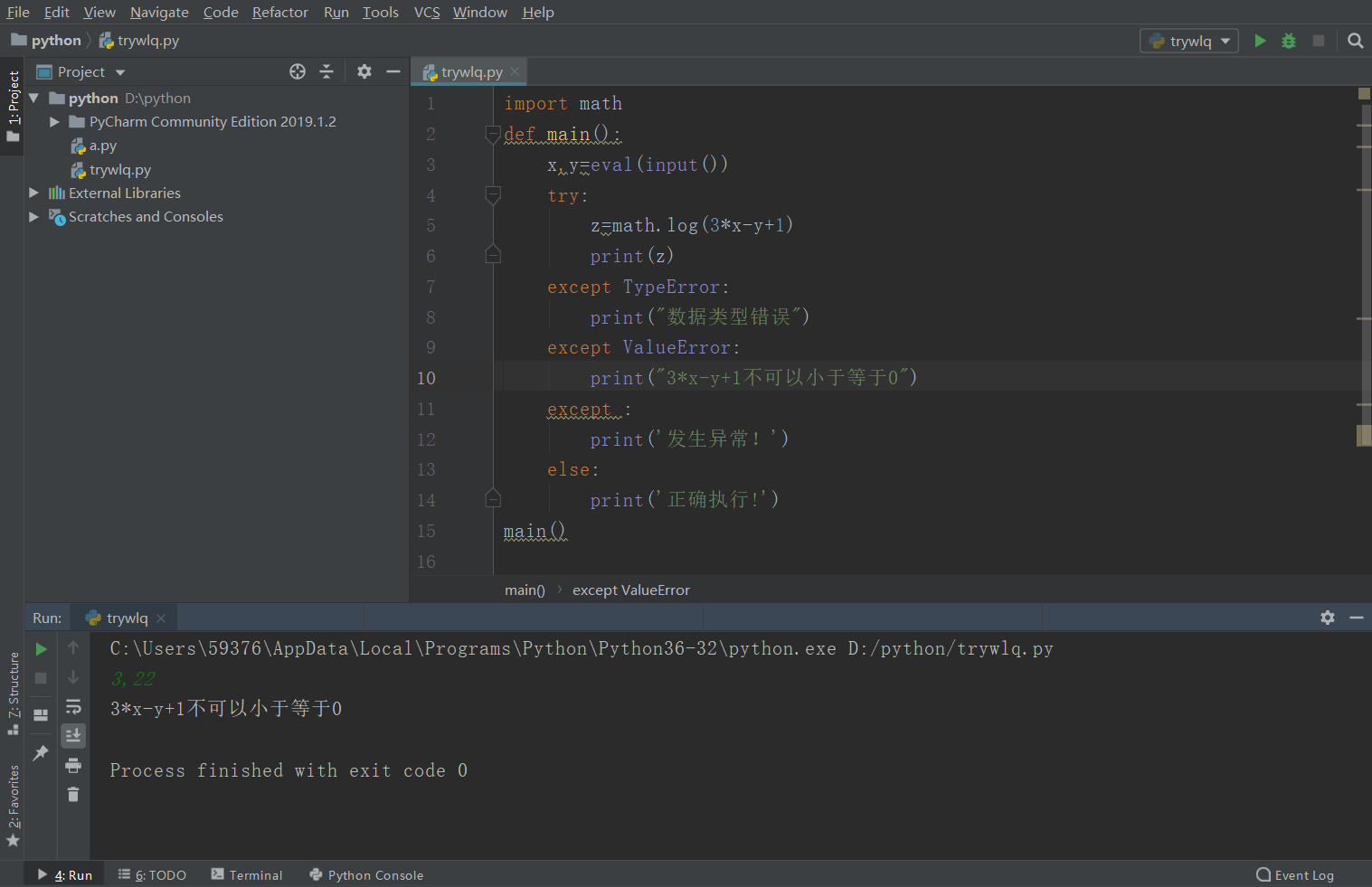
except :

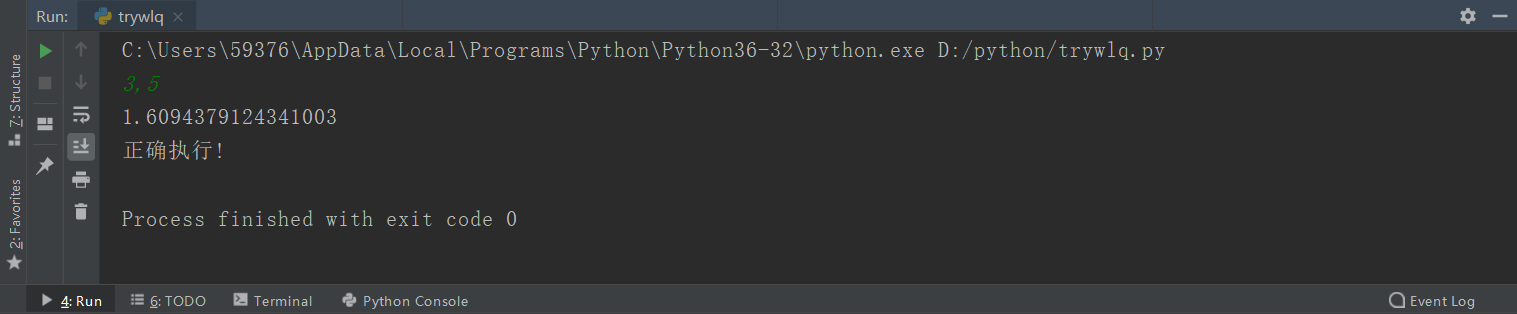
print('发生异常！')

else:

print('正确执行!')

main()





12. 编程实现：绘制曲线。

from graphics import \*

import math

win=GraphWin("绘制曲线",300,200)

i=0

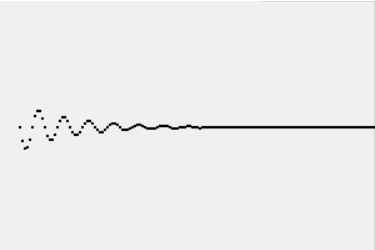
while i<=10\*math.pi:

p=Point(i\*20+15,(2\*math.exp(-0.5\*i)\*math.sin(2\*math.pi\*i))\*10+100)

p.draw(win)

i+=0.1

win.mainloop()



13. 编程实现：分别用tkinter模块、turtle模块和graphics模块绘制一个正方形及其内接圆。

from tkinter import \*

w=Tk()

w.title('tkinter绘制正方形及其内置圆')

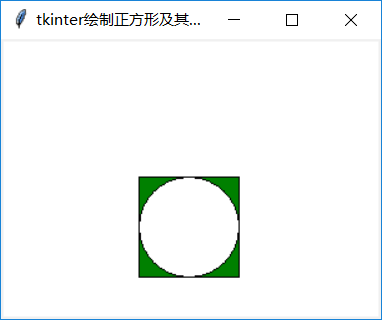
c=Canvas(w,width=300,height=220,bg='white')

c.pack()

c.create\_rectangle(110,110,190,190,fill='green')

c.create\_oval(110,110,190,190,fill='white')

w.mainloop()



import turtle

turtle.color('black')

turtle.pensize(5)

turtle.speed(10)

for i in range(4):

turtle.forward(100)

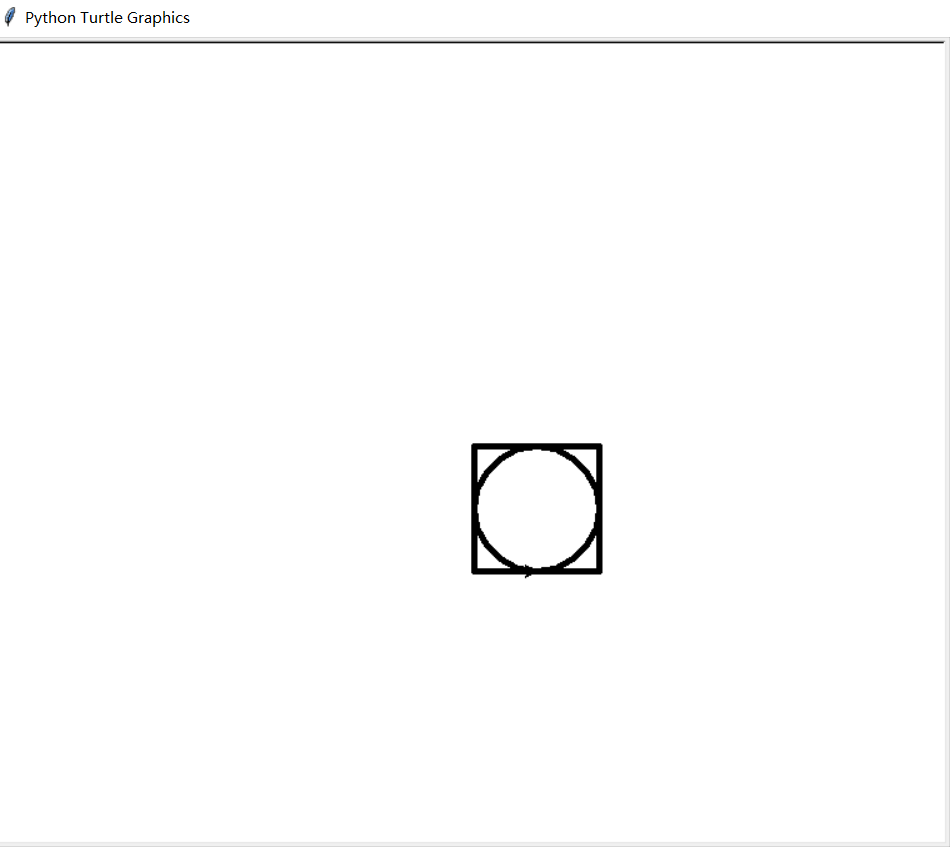
turtle.right(90)

turtle.up()

turtle.goto(50,-100)

turtle.down()

turtle.circle(50,extent=360)



from graphics import \*

win=GraphWin('graphics绘制正方形极其内接圆',300,200)

p1=Point(50,50)

p2=Point(100,100)

p3=Point(75,75)

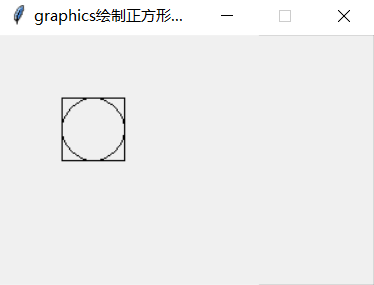
r=Rectangle(p1,p2)

c=Circle(p3,25)

r.draw(win)

c.draw(win)

win.mainloop()



14. 编程实现：在窗口中添加菜单栏，在菜单栏中添加菜单项，并添加下拉菜单，通过选择不同的菜单项可以为窗口设置不同的背景颜色。

from tkinter import \*

def call\_red():

w['bg']='red'

def call\_yellow():

w['bg']='yellow'

def call\_blue():

w['bg']='blue'

w=Tk()

w.title('颜色菜单')

m=Menu(w)

w.config(menu=m,bg='white')

colormenu=Menu(m,tearoff=0)

m.add\_cascade(label='背景颜色',menu=colormenu)

colormenu.add\_command(label='红',command=call\_red)

colormenu.add\_command(label='黄',command=call\_yellow)

colormenu.add\_command(label='蓝',command=call\_blue)

w.mainloop()



15. 编写一个跟踪鼠标位置的图形界面程序，单击鼠标时在所处位置绘制一个十字，同时在窗口上方显示鼠标所在位置的坐标，双击鼠标时擦除十字。

from tkinter import \*

def callback1(event):

print('({},{})'.format(event.x,event.y))

c.create\_line(event.x-5,event.y,event.x+5,event.y)

c.create\_line(event.x, event.y-5, event.x, event.y+5)

def callback2(event):

c.create\_line(event.x - 5, event.y, event.x + 5, event.y,fill='white')

c.create\_line(event.x, event.y - 5, event.x, event.y + 5,fill='white')

w=Tk()

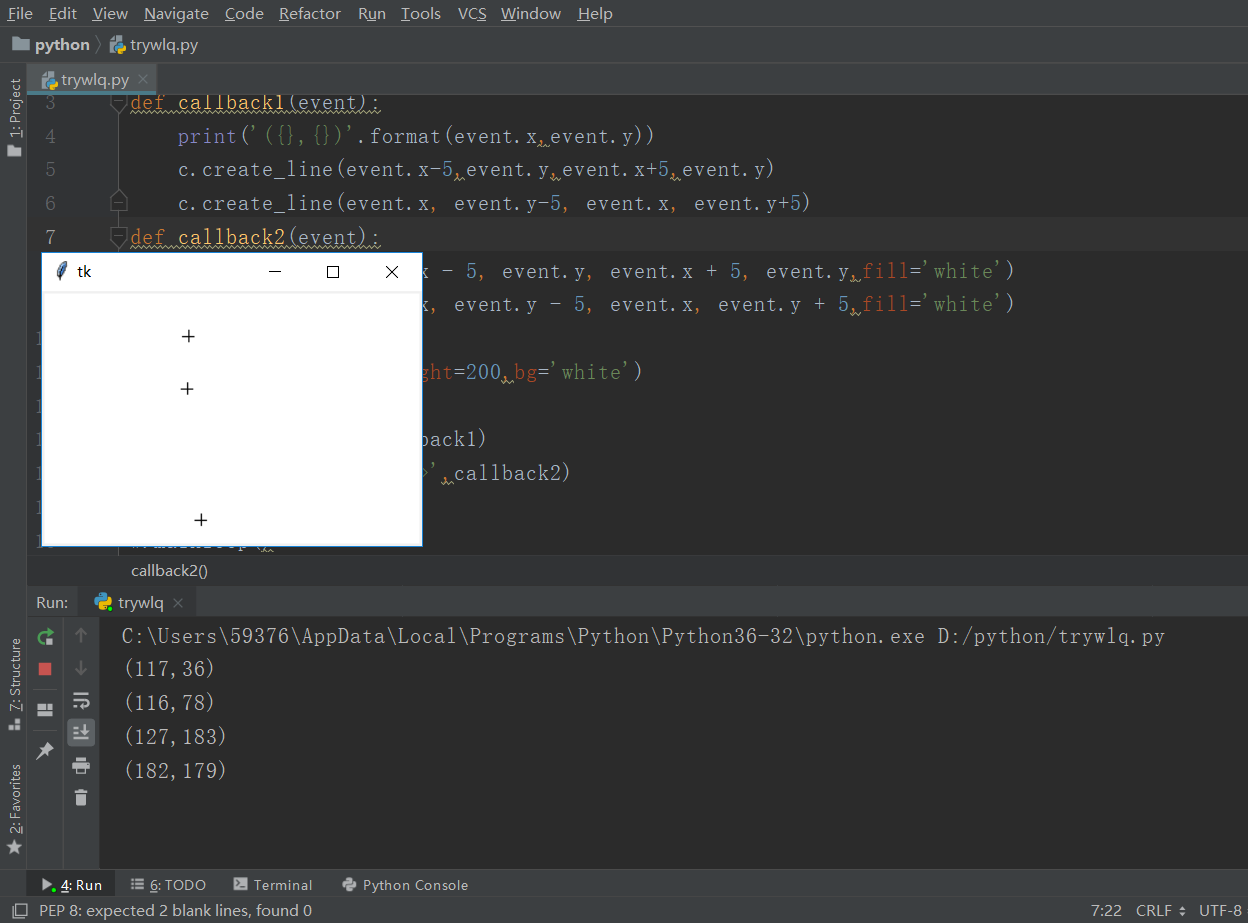
c=Canvas(w,width=300,height=200,bg='white')

c.pack()

w.bind('<Button-1>',callback1)

w.bind('<Double-Button-1>',callback2)

w.mainloop()



16. 编程创建图形用户界面，实现求一个数平方的运算。

from tkinter import \*

def calc():

value=eval(v.get())\*eval(v.get())

Label(w,text=str(value)).pack()

w=Tk()

w.title('平方计算')

v=StringVar()

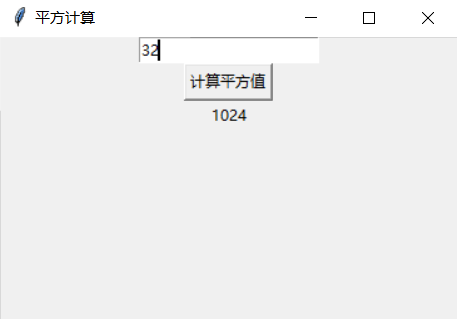
ety=Entry(w,textvariable=v)

ety.pack(side=TOP)

btn=Button(w,text='计算平方值',command=calc)

btn.pack()

w.mainloop()



17. 请用Python语言编写一个系统登录程序。

lock\_file=open('file\_lock.txt','r+')

user\_pass=open('username\_file.txt','r+')

count=0

cmd=input('''

1：登录系统

2：退出系统

请输入您的操作：''')

if cmd.isdigit() and int(cmd)==2:

exit()

elif cmd.isdigit() and int(cmd)==1:

while count <3:#只要重试不超过3次就不断循环

username=input('请输入您的用户名:')

for i in lock\_file.readlines(): #判断用户名是否在锁文件中

i=i.split()

if username in i[0]:

print('对不起 %s 已锁定'%username)

exit()

match = False

for j in user\_pass.readlines():

user,password = j.strip('\n').split() #去掉每行多余的\n并把这一行按空格分成两列，分别赋值为user,passwd两个变量

if username == user: #判断输入的用户是否存在

passwd=input ('请输入密码:')

if password == passwd:

print ('用户名和密码正确')

match = True

break

elif password != passwd:#在用户名正确的前提下，判断输入的密码是否正确

for i in range(2):

passwd=input ('密码错误，请重新输入密码:')

if password == passwd:

print ('用户名和密码正确')

match = True

break

if username != user :#判断用户不存在

print('您输入用户名不存,程序已退出')

exit()

elif match == False :#如果match还为False,代表上面的循环中跟本就没有match上用户名和密码

print('密码和用户名不匹配，尝试超过三次，用户被锁定')

lock\_file.write('%s \n'%username)

lock\_file.close()

user\_pass.close()

exit()

elif match==True:

print('登录成功')

break

else:

print('无效选项，程序已退出'

18. 请用Python语言编写一个计算器程序。

from tkinter import \*

from tkinter.ttk import \*

def my\_frame(master):

w=Frame(master)

w.pack(side=TOP,expand=YES,fill=BOTH)

return w

def my\_button(master,text,command):

w=Button(master,text=text,command=command,width=6)

w.pack(side=LEFT,expand=YES,fill=BOTH,padx=2,pady=2)

return w

def back(text):

if len(text)>0:

return text[:-1]

else:

return text

def calc(text):

try:

if sep\_flag.get()==0:

return eval(del\_sep(text))

else:

return add\_sep(str(eval(del\_sep(text))))

except(SyntaxError,ZeroDivisionError,NameError):

return 'Error'

def add\_sep(text):

dot\_index=text.find('.')

if dot\_index>0:

text\_head=text[:dot\_index]

text\_tail=text[dot\_index:]

elif dot\_index<0:

text\_head=text

text\_tail=''

else:

text\_head=''

text\_tail=text

list\_=[char for char in text\_head]

length=len(list\_)

tmp\_index=3

while length-tmp\_index>0:

list\_.insert(length-tmp\_index,',')

tmp\_index+=3

list\_.extend(text\_tail)

new\_text=''

for char in list\_:

new\_text+=char

return new\_text

def del\_sep(text):

return text.replace(',','')

wind=Tk()

wind.title('简易计算器')

main\_menu=Menu(wind)

calc\_menu=Menu(main\_menu,tearoff=0)

calc\_menu.add\_command(label='退出',command=lambda:exit())

main\_menu.add\_cascade(label='计算',menu=calc\_menu)

text=StringVar()

sep\_flag=IntVar()

sep\_flag.set(0)

view\_menu=Menu(main\_menu,tearoff=0)

view\_menu.add\_checkbutton(label='显示千位分隔符',variable=sep\_flag,\

command=lambda t=text:t.set(add\_sep(t.get())))

main\_menu.add\_cascade(label='视图',menu=view\_menu)

wind['menu']=main\_menu

Entry(wind,textvariable=text).pack(expand=YES,fill=BOTH,\

padx=2,pady=4)

style=Style()

style.configure('TButton',padding=3)

fedit=my\_frame(wind)

my\_button(fedit,'Backspace',lambda t=text:t.set(back(t.get())))

my\_button(fedit,'Clear',lambda t=text:t.set(''))

my\_button(fedit,'±',lambda t=text:t.set('-('+t.get()+')'))

for key in('789/','456\*','123-','0.=+'):

fsymb=my\_frame(wind)

for char in key:

if char=='=':

my\_button(fsymb,char,\

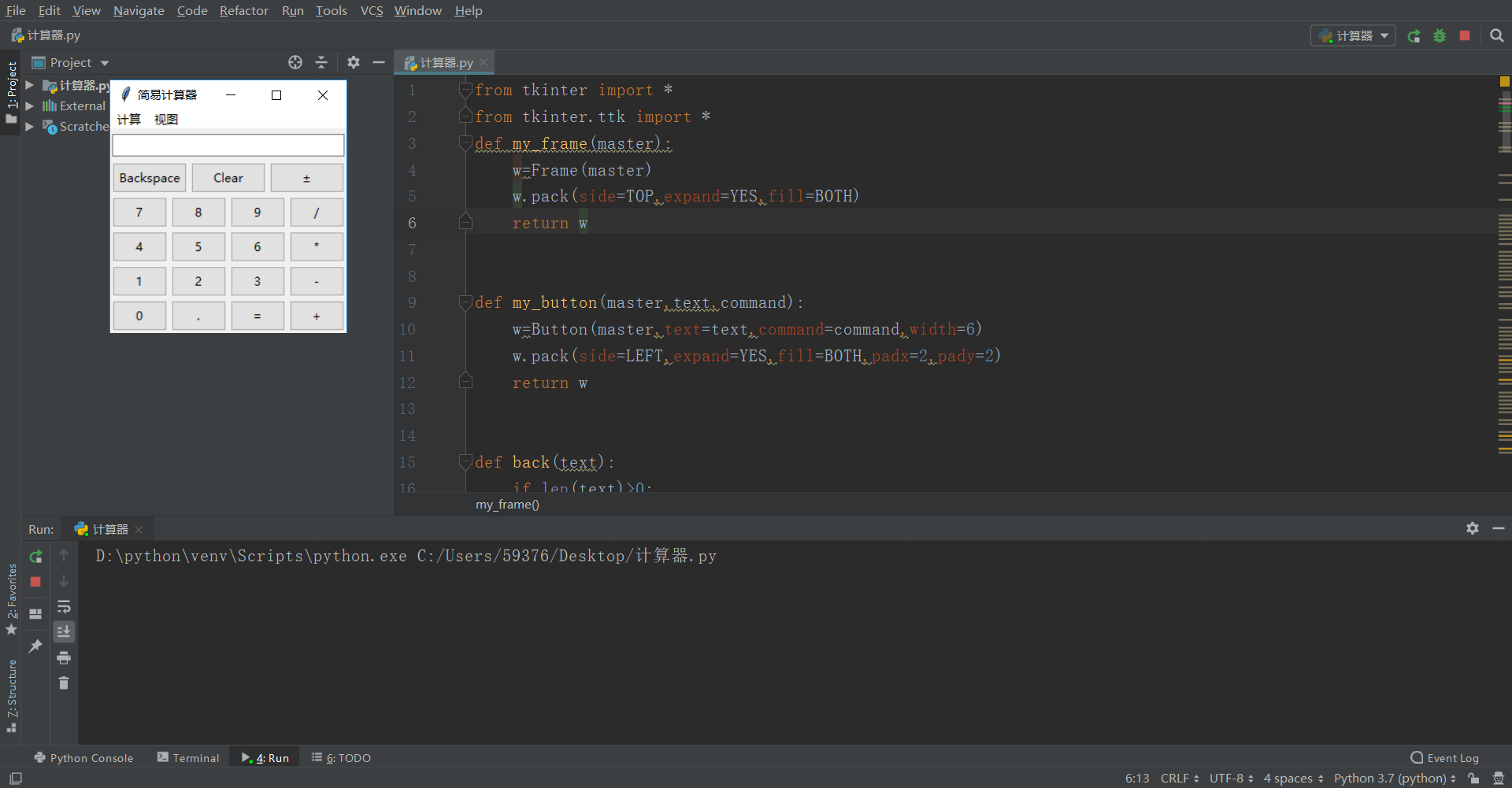
lambda t=text:t.set(calc(t.get())))

else:

my\_button(fsymb,char,\

lambda t=text,c=char:set(t.get()+c))

wind.mainloop()



19. 请用Python语言编写一个猜数字游戏程序。

from tkinter import \*

from tkinter.messagebox import \*

import random

def call\_begingame():

num=int(random.uniform(0,10))

showinfo(message='猜一个1-10的数字')

top = Toplevel()

v = IntVar()

guess\_ety = Entry(top, textvariable=v)

guess\_ety.pack()

print(num)

print(v.get())

v.set(1)

while True:

if v.get()>10:

continue

if v.get() == num:

if askyesno(title='YOU WIN!', message='再玩一次？'):

showinfo(title='you choose again!', message='游戏愉快！')

call\_begingame()

else:

showinfo(title='you choose give up!', message='再见了您嘞！')

exit()

elif v.get() < num:

showwarning(title='wrong!', message='猜大一点试试呢！')

guess\_ety.pack()

else:

showwarning(title='wrong!', message='猜小一点试试呢！')

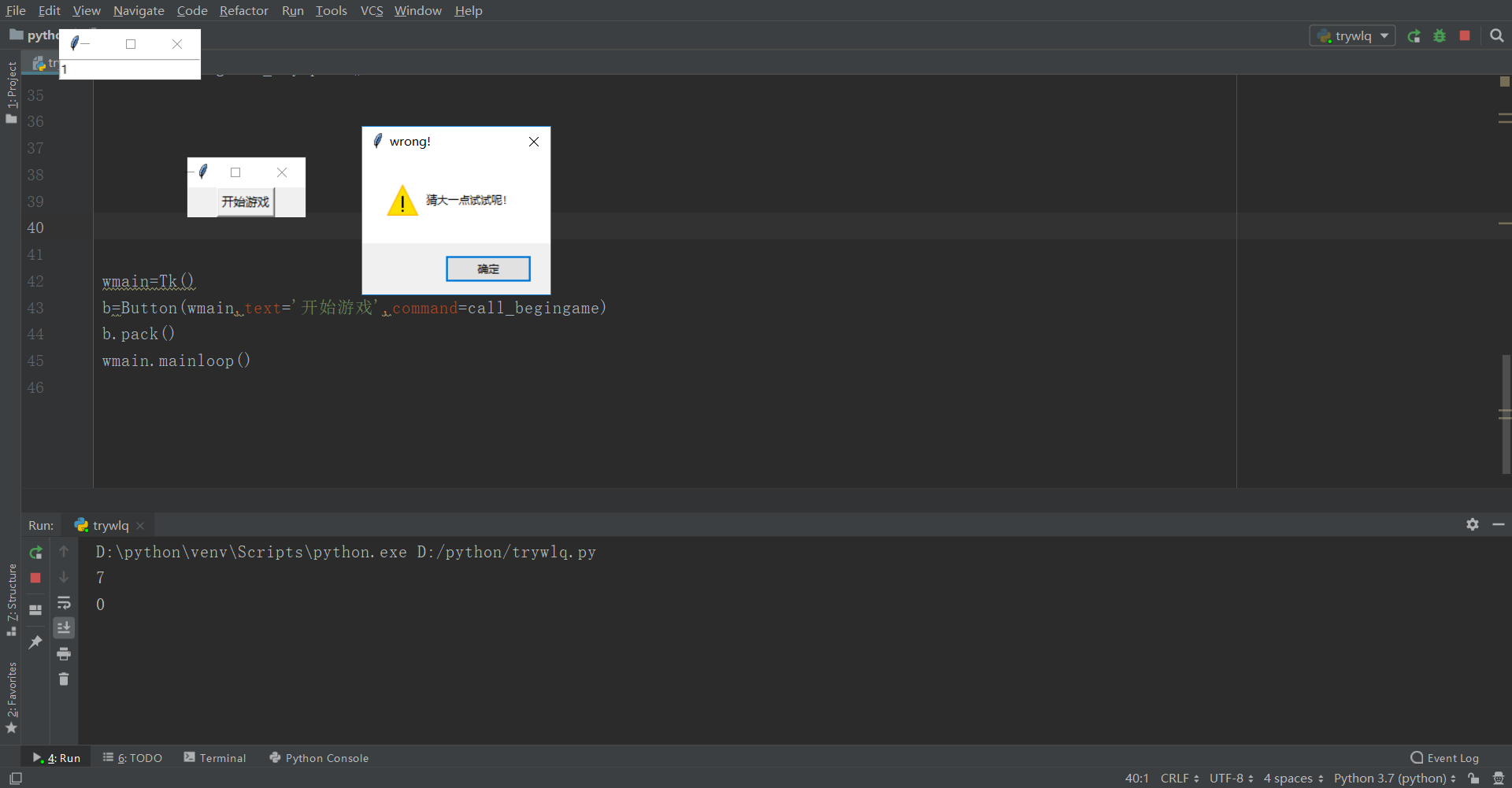
guess\_ety.pack()

wmain=Tk()

b=Button(wmain,text='开始游戏',command=call\_begingame)

b.pack()

wmain.mainloop()



20. 请用Python语言编写一个通讯录管理程序。

# 1.增加姓名和手机

# 2.删除姓名

# 3.修改手机

# 4.查询所有用户

# 5.根据v姓名查找手机号

# 6.退出

# 遍历列表

# list1=[1,2,4,5]

# for i in range(len(list1)):

#     print(list1[i])

#通讯录

telbook={}

print("欢迎使用XXX的通讯录，请输入您想要进行的操作")

flag="A"

while flag=="A" or "B" or "C" or "D" or "E":

    m=input("1.输入A为通讯录添加联系人和相应手机号. \n2.输入B删除通讯录里的联系人. \n3.输入C修改通讯录里联系人手机号。\n4.输入D查询通讯录里的所有用户。 \n5.输入E根据姓名查找响应联系人手机号  \n")

    if m=="A":

        username=input("请输入联系人姓名：")

        telnumber=input("请输入联系人电话：")

        telbook[str(username)]=str(telnumber)

        print("添加成功,通讯录中现有联系人为：",end=" ")

        print(telbook)

        n=input("您是否想继续操作彭赫妍通讯录？任意键为继续，N为退出通讯录")

        if n=="N":

            print("您已经退出通讯录")

            break

        else:

            continue

    elif m=="B":

        print("您是否想删除通讯录里联系人")

        deusername=input("请输入要删除的联系人名字")

        del telbook[deusername]

        print("删除成功，删除后的通讯录内容为",telbook,end=" ")

        n=input("您是否想继续操作彭赫妍通讯录？任意键为继续，N为退出通讯录")

        if n=="N":

            print("您已经退出通讯录")

            break

        else:

            continue

    elif m=="C":

        mousername=input("请输入要修改的联系人姓名")

        for key in sorted(telbook.keys()):

            if str(mousername)==key:

                motelnumber=input("联系人存在，请输入修改后的新号码")

                telbook[str(key)]=str(motelnumber)

                print("修改成功,修改后的通讯录为",telbook)

                break

            n=input("您是否想继续操作彭赫妍通讯录？任意键为继续，N为退出通讯录")

            if n=="N":

                print("您已经退出通讯录")

                break

            else:

                continue

    elif m=="D":

        print(telbook.keys)

        n=input("您是否想继续操作彭赫妍通讯录？任意键为继续，N为退出通讯录")

        if n=="N":

            print("您已经退出通讯录")

            break

        else:

            continue

    elif m=="E":

        searchname=input("请输入您想查询响应手机号的联系人姓名")

        for key2 in sorted(telbook.keys()):

            if str(searchname)==key2:

                print(key2,telbook[key2])

            else:

                print("没有您输入的联系人信息，请重新输入或添加后输入")

        n=input("您是否想继续操作彭赫妍通讯录？任意键为继续，N为退出通讯录")

        if n=="N":

            print("您已经退出通讯录")

            break

        else:

            continue

    else:

        print("您的输入有误")