实 验 报 告

**课程名称** Python **实验名称** 实验2

**课程代号** **实验编号**

**姓 名** 陈邦杰 **学 号** 1950300146 **专业班级** 19计算机科学与技术2班

**实验日期** 2021年11月2日  **成绩** **指导教师** 郭波

1. 实验目的

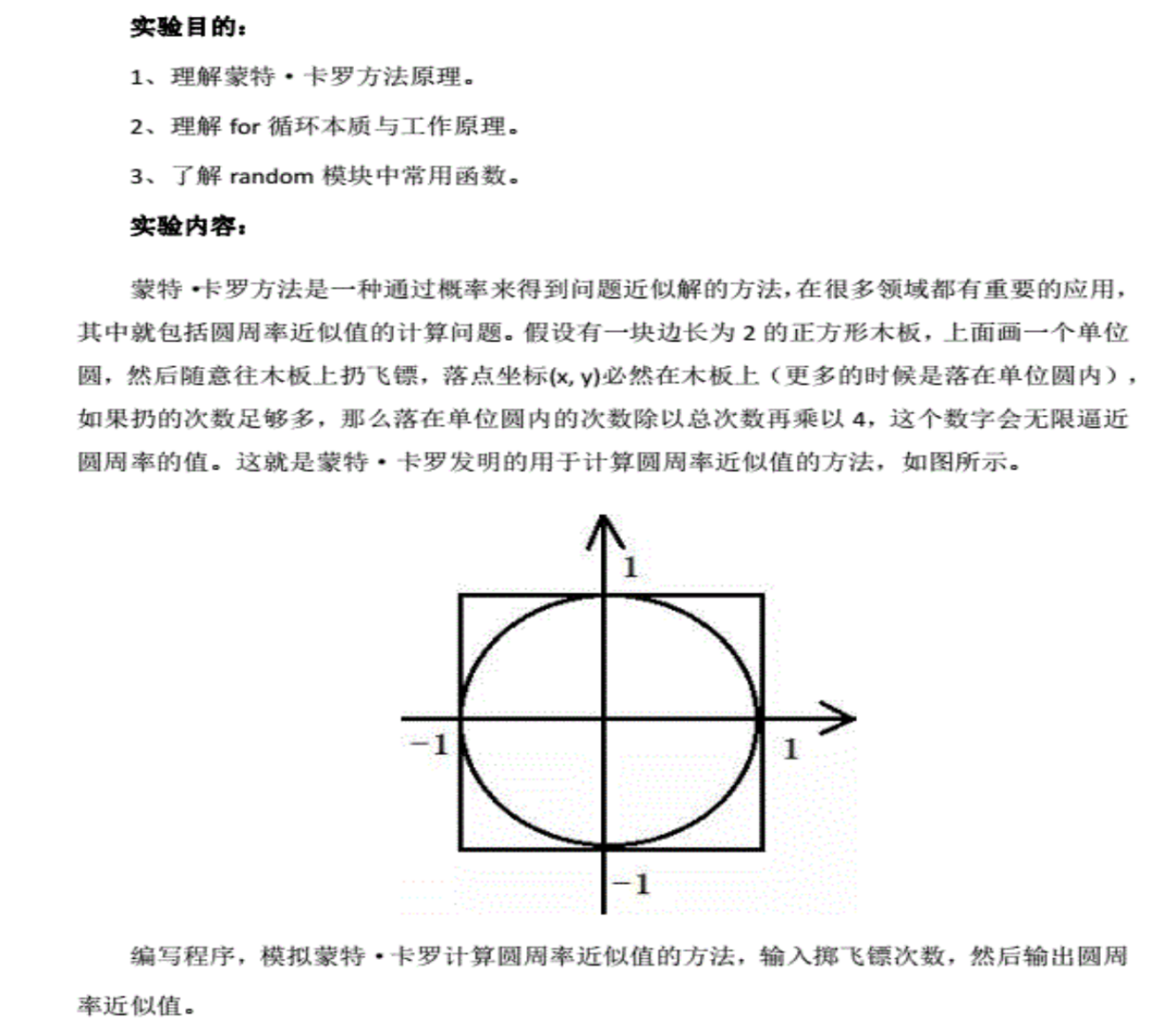
熟悉Python的环境，编写小程序并且运行

二、实验环境

Anaconda或者Python3.8

三、实验内容、代码、运行结果与分析

第一题：



实验代码：

# -\*- coding =utf-8 -\*-

import tqdm

import random

def get\_pi(n):

count = 0

for i in tqdm.tqdm(range(n)):

x = random.random()

y = random.random()

if x\*\*2 + y\*\*2 <= 1:

count += 1

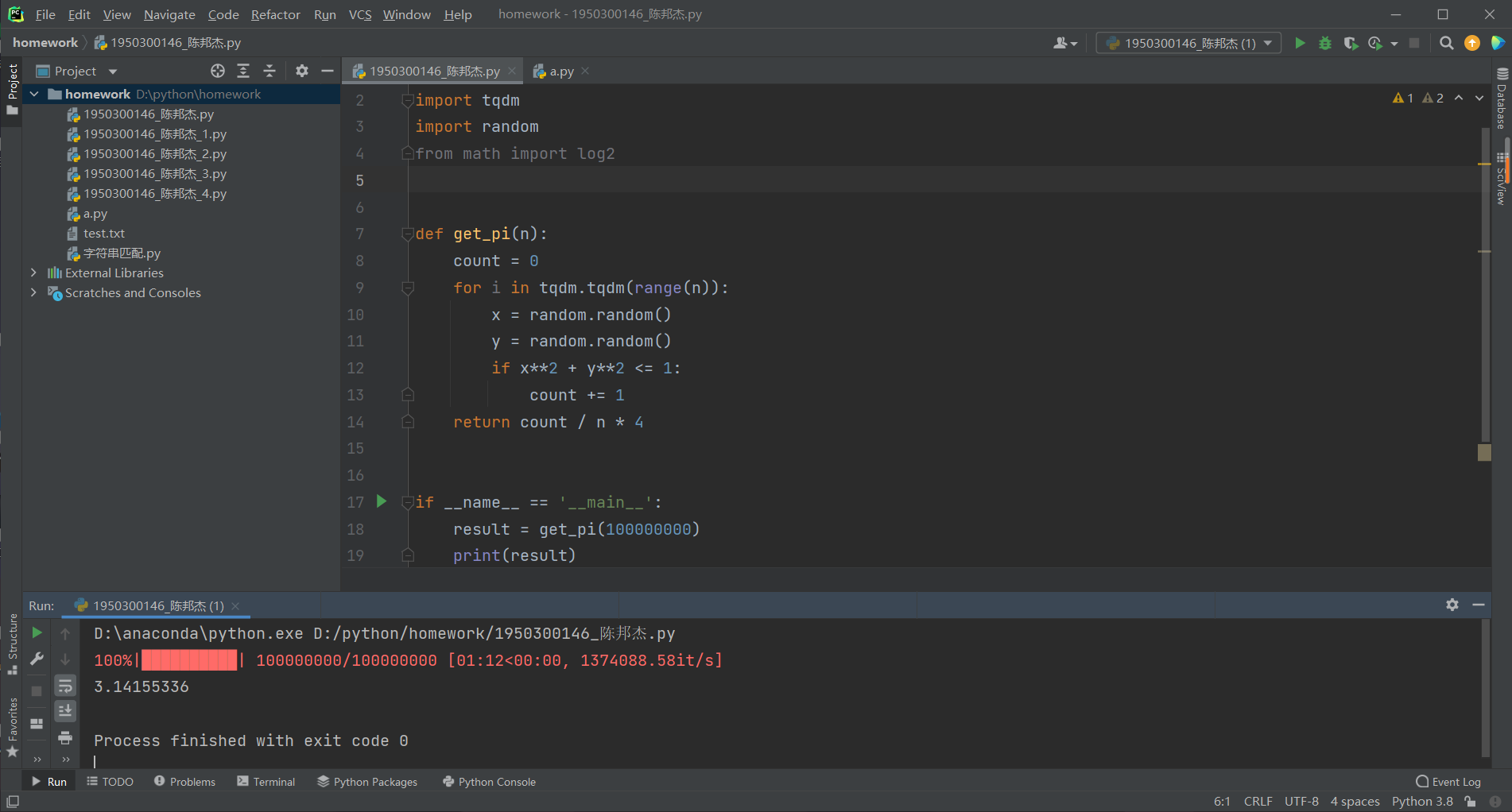
return count / n \* 4

if \_\_name\_\_ == '\_\_main\_\_':

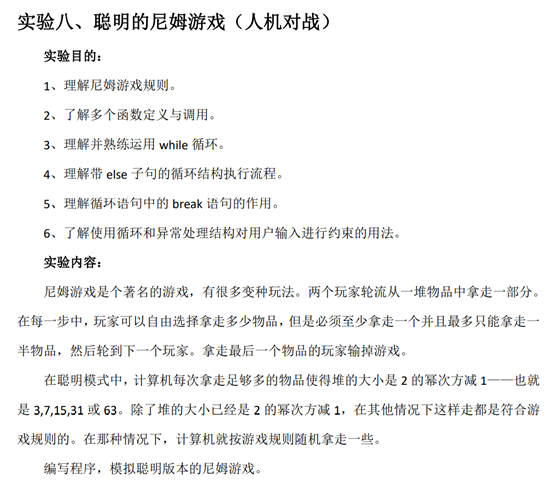
result = get\_pi(100000000)

print(result)

运行截图：



第二题：



实验代码：

import random

from math import log2

def rot\_get(total\_count):

num = total\_count - (2\*\*int(log2(total\_count))-1)

half = total\_count // 2

if num <= half:

print(f'电脑拿走:{num}')

return num

else:

num = random.randint(1, half)

print(f'电脑拿走:{num}')

return num

def nim\_game():

total\_count = random.randint(100,200)

while total\_count > 1:

print(f'剩余物品数为:{total\_count}')

while True:

num = int(input(f'输入拿走的值(1-{total\_count // 2}):'))

if num > total\_count // 2:

print('超过总数一半，请重新输入')

continue

if num < 1:

print('拿走物品不能小于1，请重新输入')

continue

break

total\_count -= num

if total\_count == 1:

print('玩家获胜!')

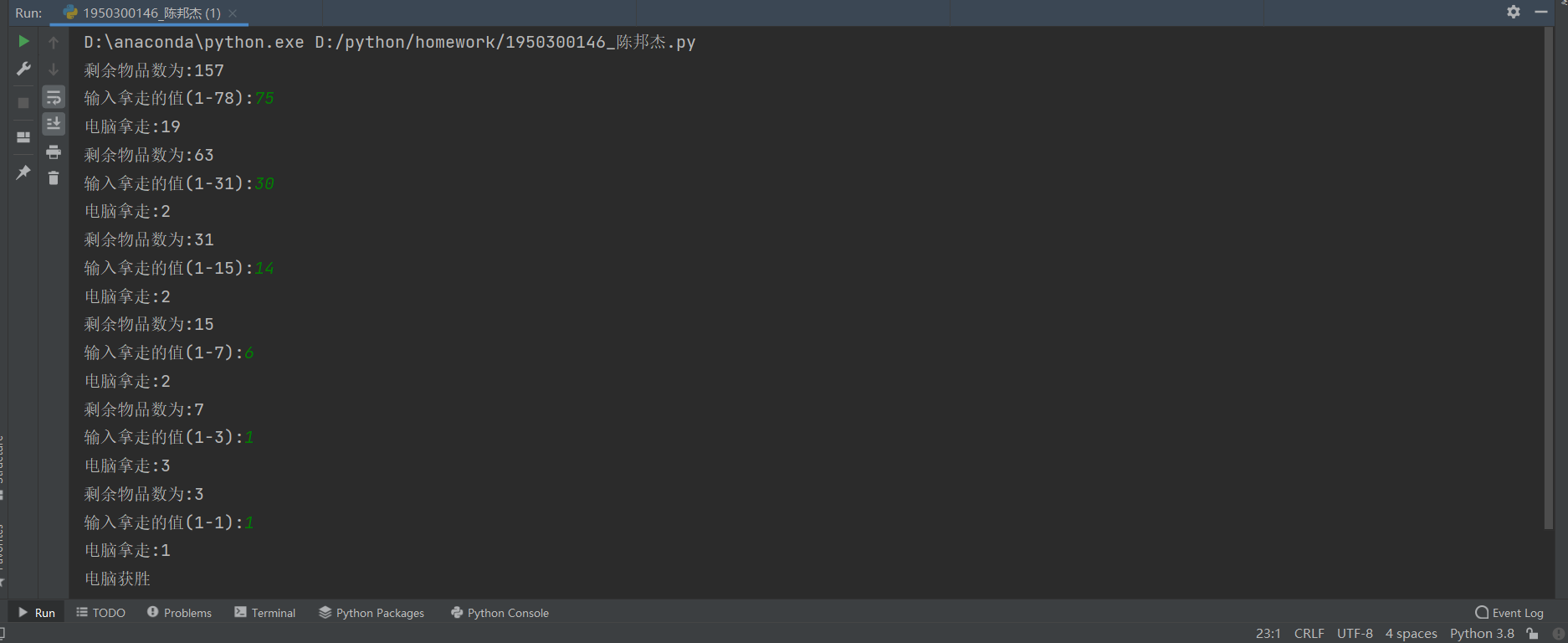
break

total\_count -= rot\_get(total\_count)

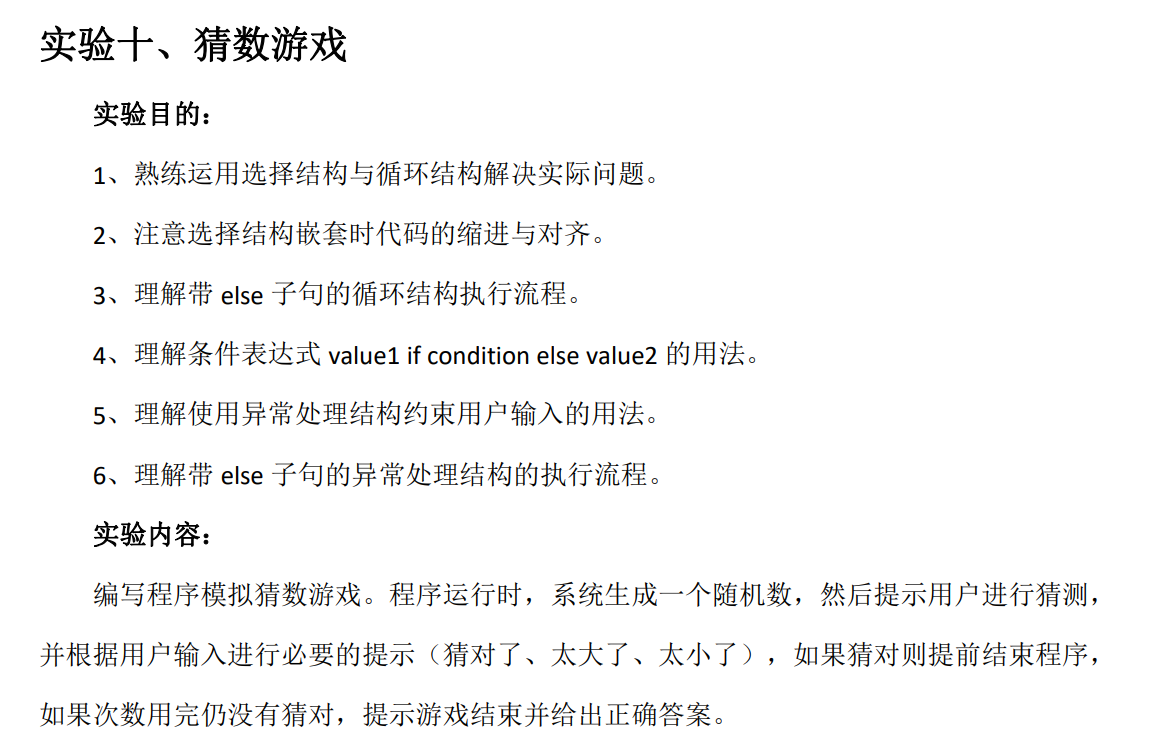
else:

print('电脑获胜')

运行截图：



第三题：



实验代码：

def get\_number(m):

n = random.randint(1, m)

print(f'right answer: {n}')

r = -1

while True:

o = yield r

if o == n:

r = '对'

elif o > n:

r = '大'

else:

r = '小'

print(r)

def guess\_big\_small(m):

g = get\_number(m)

r = g.send(None)

up = m

down = 0

count = 0

while r != '对' and count < 100:

n = int(round((up + down) / 2, 0))

print(f'guess {n}')

r = g.send(n)

if r == '大':

up = n

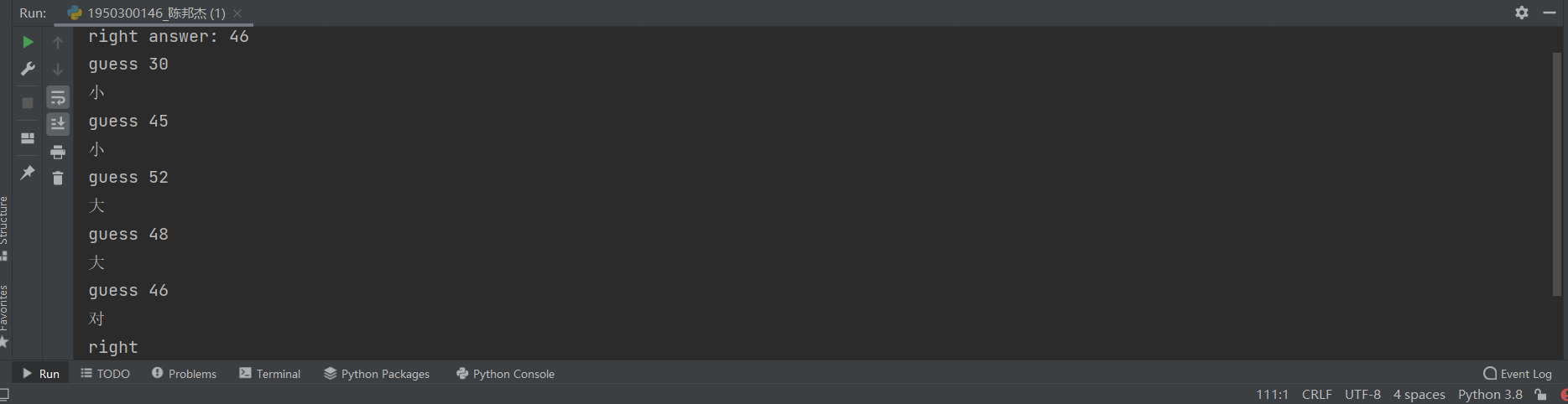
else:

down = n

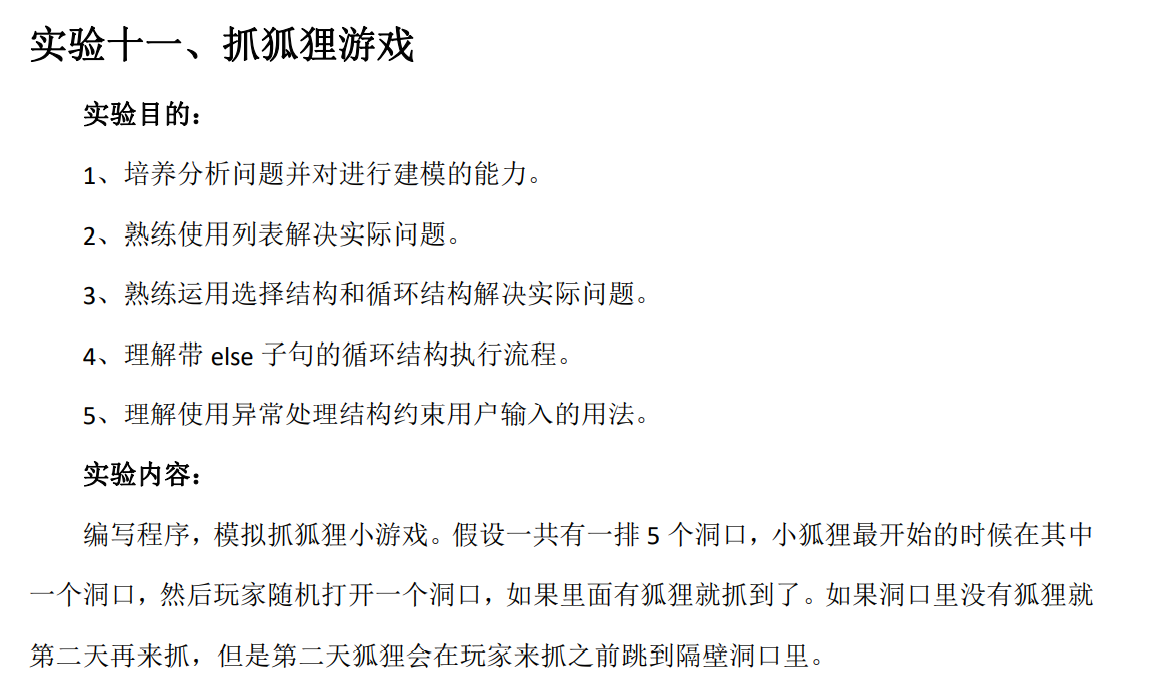
count += 1

print('right')

运行截图：



第四题：



实验代码：

def catch\_fox():

hole = [1, 2, 3, 4, 5]

fox = random.randint(0, 4)

while True:

choice = int(input('这里有五个洞口1~5，请输入小狐狸所在洞口:'))

if choice == hole[fox]:

print('恭喜你抓到小狐狸了！')

break

else:

print('很遗憾没有抓到!')

if 0 <= fox < 4:

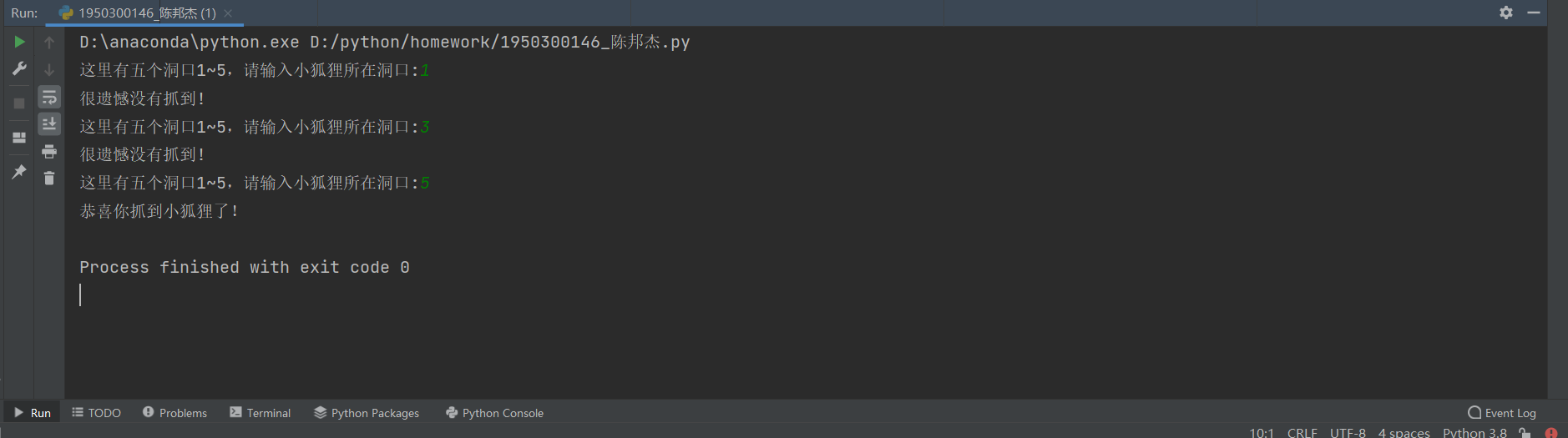
fox += 1

else:

fox -= 1

continue

运行截图：



**四、实验小结**

通过本次实验我熟悉了列表推导的使用方法和for、while循环语句的使用，强化了分析问题的能力。Python for循环可以遍历任何可迭代的对象，如一个列表或者一个字符串或者生成器；在 python 中，for … else 表示这样的意思，for 中的语句和普通的没有区别，else 中的语句会在循环正常执行完（即 for 不是通过 break 跳出而中断的）的情况下执行，while … else 也是一样。Python 编程中 while 语句用于循环执行程序，即在某条件下，循环执行某段程序，以处理需要重复处理的相同任务。