《Python程序设计》课程实验报告

信息 学院 智能科学与技术 专业 2020 级

实验时间 2021 年 10 月 11 日

姓名 ***Steven*** 学号

实验名称 作业三

实验成绩

一、实验目的

练习使用Python进行程序开发，解决问题的方法。

二、实验仪器设备及软件

电脑，Pycharm ，Anaconda Python 3.8

三、实验方案

使用构造函数，列表、集合以及一些内置函数解决实验问题。

1. 可以使用循环完成。循环接受玩家输入的编号，如果与实际编号一致，则输出正确信息，退出程序；如果输入编号与实际编号不一致，则使用随机数的方式选择狐狸去小1的洞里还是去大1的洞里，然后进入下次循环。
2. Python列表的insert的方法是将目标对象插入列表的某个位置中，原本位置及其后的数据会依次向后移1。故遍历原列表，将对象pop出来，然后根据原index的位置作为下限，从头正序遍历，最后在符合条件的位置insert进去即可。由于排序过程中列表的元素数不变，所以遍历全列表并不会漏掉元素。
3. 本题可以用排列组合和递归两种思路解决。排列组合的方式是讨论所有的1个台阶和2个台阶的步数，然后进行排列组合。递归方式就是根据总台阶为15，f(15)=f(13)+f(14)，f(14)=f(13)+f(12)……

四、实验步骤

1. 编写程序，实现抓狐狸游戏。假设墙上有5个洞（编号分别为0、1、2、3、4），其中一个洞里有狐狸，人类玩家输入洞口编号，如果洞里有狐狸就抓到了；如果洞里没有狐狸就第二天再来抓。但在第二天人类玩家来抓之前，狐狸会跳到隔壁的洞里。（课本5题）：

|  |
| --- |
| def question5():  foxHole = random.randint(0, 4) *# 先随机选一个洞* while True:  *# 决定狐狸是去前一个洞还是后一个洞* if random.random() > 0.5:  foxHole = foxHole + 1 if foxHole + 1 <= 4 else foxHole - 1  else:  foxHole = foxHole - 1 if foxHole - 1 >= 0 else foxHole + 1  choiceHole = eval(input("请在0~4内选择一个洞"))  if choiceHole == foxHole:  print("您已经抓到狐狸，即将退出")  break  else:  print("狐狸不在这个洞里，明天再来吧") |

2.编写程序，生成一个包含20个介于1到50之间随机整数的列表，然后使用插入法对给定整数列表中的所有元素升序排序。（课本6题）：

|  |
| --- |
| def question6():  list = [random.randint(1, 50) for i in range(20)]  for i in range(1, len(list)):  temp = list.pop(i) *# 取出数组中第i个值* for j in range(i): *# temp和已排序的部分作比较* if temp >= list[i - 1]: *# 和倒数第一个比* list.insert(i, temp)  break  elif temp < list[j]: *# 然后从前往后比，直到倒数第一个和倒数第二个* list.insert(j, temp)  break   print("排序结果为：", list) |

3.6岁的小明上楼梯，一步只能迈1个或3个台阶，现在有段15个台阶的楼梯，要想上到第15个台阶，编写程序计算小明有多少种上法。（课本9题）：

|  |
| --- |
| def que9\_recu(remainStep):  if remainStep <= 0:  return 0  if remainStep == 1:  return 1  elif remainStep == 3:  return 2  else:  return que9\_recu(remainStep - 1) + que9\_recu(remainStep - 3) |
| def ques9\_loop(totalStep):  sum = 0  for threeNum in range(0, int(totalStep/3 + 1)): *# 遍历到全部情况* oneNum = totalStep - threeNum \* 3  *# 算组合数并累加* sum = sum + math.factorial(oneNum + threeNum) / (math.factorial(oneNum) \* math.factorial(threeNum))  return sum |

五、实验结果及分析

1.

图形用户界面, 文本

描述已自动生成 图形用户界面, 应用程序

描述已自动生成 图片包含 文本

描述已自动生成

（一开始随机选择了狐狸在0号洞里，选择了1号洞）

图形用户界面, 文本, 应用程序

描述已自动生成 图形用户界面, 文本

中度可信度描述已自动生成

（之后狐狸在1号洞里，选择了4号洞）

图形用户界面, 文本, 应用程序

描述已自动生成 文本

描述已自动生成

（之后狐狸在2号洞里，选择了2号洞）

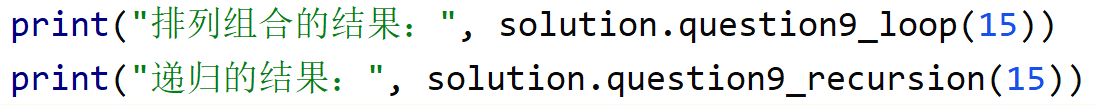
2.

图形用户界面, 应用程序

描述已自动生成



3.



文本

低可信度描述已自动生成

六、实验总结及体会

使用Python利用常见的循环、分支等结构解决问题，相比于之前曾学过的Java、C语言解决又有新的感受。由于Python内置了丰富的库函数，使得在完成功能的时候减少了很多“造轮子”的步骤，更多时候是调用现成的库函数完成具体的小功能，并且也因此可读性较好。

但不管怎么说，循环和分支都是构成程序、算法的基础结构，所以尽管在Python中简化了书写过程，但仍需多加练习，同时需要更加充分地理解各个函数的作用，才能更好地利用Python的高效率语法。

七、教师评语