# 广州理工学院实验报告

**学院：计算机科学与工程学院 专业： 数据科学与大数据技术 年级： 2022**

**姓名： 吴恩威 学号： 20220409430251 组 实验时间：2023.11.27**

**指导教师签字： 成绩：**

|  |
| --- |
| **实验项目名称：**  实验8函数与模块应用   * **实验目的和要求：**   1、熟练掌握函数的定义与调用；  2、掌握局部变量与全局变量的作用域；  3、学习使用函数的参数和返回值；  4、学习使用Python的内置函数；  5、掌握模块的应用。 |
| * **主要仪器设备：**   PC机一台；Windows操作系统；Python运行环境；开发工具（PyCharm)。 |
| * **实验内容和原理：**   1、练习函数的声明与调用；  2、函数的参数与返回值设置；  3、函数的主体功能设置；  4、Python内置函数的使用；  5、全局变量、局部变量、模块等知识内容的灵活运用。  6、函数的实际工程应用。 |
| * **操作方法和实验步骤（绘图）：**   1、 小球从给定高度的地方自由落下，每次落地后反弹回原高度的52%，然后再落下，定义一函数ball\_fun，实现：计算小球在第n次落地时，所经过的路共多少米以及第n次反弹高度位置。  操作方法与步骤：  （1）请定义全局变量Cn和Hn分别存储小球经过的路程和第n次反弹的高度。  （2）定义ball\_fun函数：实现计算小球在第n次落地时，所经过的路共多少米以及第n次反弹高度位置。  （2）主函数main：输入小球原始落地高度与求解n（n为小球第n次落地）。  （3）主函数main：调用ball\_fun函数并传递相关数据进去，最后得出Cn与Hn的值并输出。  过程代码截图如下：    2、编写程序：编写函数guess\_fun模拟猜数游戏。  系统随机产生一个1-n（n为最大值，比如100）范围内整数，玩家进行猜测，并设置猜测机会time（比如5次），系统在玩家猜测的过程中，根据玩家测试的数据与系统生成的数进行对比，并给予提示（比正确数字偏大或偏小了），玩家在机会范围内，可以根据系统的提示对下一次的猜测进行适当的调整数值范围，以提高猜中的机率。  操作方法与步骤：定义guess\_fun函数，并设置接收参数n,time，函数内实现调用random模块生成一个1-n的范围内的整数num，使用循环语句for控制猜测次数，每次玩家输入一个整数x，并与num做对比，根据对比结果进行不同的操作。最后调用函数，模拟实现猜字游戏。  过程代码截图如下：    3、汉诺塔问题基于递归算法的实现。据说古代有一座梵塔，塔内有3个底座A、B、C，A座上有64个盘子，盘子大小不等，大的在下，小的在上。有一个和尚想把这64个盘子从A座移到C座，但每次只能允许移动一个盘子，在移动盘子的过程中可以利用B座，但任何时刻3个座上的盘子都必须始终保持大盘在下、小盘在上的顺序。如果只有一个盘子，则不需要利用B座，直接将盘子从A移动到C即可。和尚想知道这项任务的详细移动步骤和顺序。    操作方法与步骤：定义honoi函数，接收盘子的个数，以及三座塔的命名字符，函数内使用递归调用的基本思路实现求解具体的步骤，第一调用honoi函数自身上一次的结果，第二使用if语句设置临界条件。  过程代码截图如下：    4、编写函数计算任意位数的黑洞数。黑洞数是指：有这一样一个整数，由这个整数每位数字出来重组成一个最大数，再减去每位数字重组成的最小数，结果仍然是得到这个数本身。比如，3位黑洞数495=954-459，4位黑洞数6174=7641-1467。  操作方法与步骤：定义black\_num函数，并设置接收参数n(n为位数)，函数内实现调用：对n位数范围内的所有数进行判断，是否满足为黑洞数（即取每位数字出来重组成一个最大数，再减去每位数字重组成的最小数，结果仍然是得到这个数本身）。  过程代码截图如下：    5、编写函数，实现将数据经MD5（需要导入hashlib模块）加密返回给调用端。主函数实现读取输入的用户名称和密码，调用加密函数，实现对用户名、密码加密输出结果。  输入示例：    输出示例：    操作方法与步骤：1.用户通过输入用户名和密码。2.encrypt 函数接收用户名和密码作为参数。3.使用 hashlib 库的 md5 方法创建 MD5 对象，然后分别对用户名和密码进行编码和加密。4.打印出未加密和已加密的用户名和密码。  过程代码截图如下： |
| * **实验结果与分析**   1、实验结果如下：    结果分析：(1)定义变量Cn和Hn记录小球经过的路程和第n次反弹的高度（2）定义函数ball\_fun来是实现题目要求，由于函数内要使用和更新Cn、Hn的值，所以在函数内要对Cn、Hn进行全局变量的定义 （3）先将初始高度赋给init\_high，然后通过for循环控制落地的次数，Cn += init\_high然后对init\_high的0.52赋值给Hn反弹高度，然后更新init\_high为Hn，最后循环结束，加上最后一次反弹的高度，返回即可 (4)调用main，按照要求输入数据，最后打印数据   1. 实验结果如下：     结果分析：(1)定义函数guess\_fun(n,times),传入一个数n，使用random模块下的randint生成从1-n范围的一个随机数 num,然后使用for循环控制猜的次数times (2)开始猜数字，用户输入一个数x，使用if-elif判断，如果x<num,输出偏小和剩下猜数字的次数times-i-1，然后continue,x>num时也同理，如果猜中了，按照要求输出打印，break (3)根据main中用户输入的随机数范围和次数，调用函数  3、实验结果如下：    结果分析：如下图所示,假设总共需要移动n个盘子 (1)将A柱上的n-1个盘子借助C柱移向B柱 (2)将A柱上仅剩的最后一个盘子移向C柱 (3)将B柱上的n-1个盘子借助A柱移向C柱  IMG_256  4、实验结果如下：    结果分析：（1）通过用户输入的n，来查看n位数的黑洞数，传入black\_num(n) (2)通过一个for循环(10\*\*(n-1),10\*\*n)拿到n位数的数，然后直接对num强转成str后sorted，然后使用字符串的join将排序后的num[::-1]拿到最大的max\_num,min\_num为排序后的num (3)最后根据题目要求进行判断输出即可   1. 实验结果如下：     结果分析：hexdigest() 是 Python 中 hashlib 库中 hash 对象的一个方法，用于获取摘要算法的十六进制表示形式。当你使用 hashlib 库中的哈希对象（比如 MD5、SHA-1、SHA-256 等）进行加密时，这些对象会产生一个哈希值或摘要。hexdigest() 方法可以将这个摘要以十六进制的形式返回，方便在程序中处理和展示。它将二进制摘要转换为易于阅读和处理的十六进制字符串表示形式。user\_md.hexdigest() 和 passwd\_md.hexdigest() 被用来获取对用户名和密码进行 MD5 加密后的十六进制表示形式的摘要。 |

**批阅评语：**