**深 圳 大 学 实 验 报 告**

**课程名称：­ python语言程序设计**

**实验项目名称： 实验2- 控制语句**

**学院： 医学部生物医学工程学院**

**专业： 生物医学工程**

**指导教师： 黄淦**

**报告人： 李昕萌**

**学号： 2021222001**

**实验时间： 2023年 02 月 27 日**

**教务处制**

|  |  |  |
| --- | --- | --- |
| **Exercise 1: Maximum**  Write a function my\_max(x,y) that returns the maximum of x and y. Do not use the max function, but use if instead in following two ways:  (a) Use both if and else.  (b) Use if but not else (nor elif).  编写代码如下    分别三次测试结果输出如下    **Exercise 2: For loops**  Use a for loop to:  (a) Print the numbers 0 to 100  (b) Print the numbers 0 to 100 that are divisible by 7  (c) Print the numbers 1 to 100 that are divisible by 5 but not by 3  (d) Print for each of the numbers x = 2…. 20, all numbers that divide x, excluding 1 and x. Hence, for 18, it should print 2 3 6 9.  （a）编写代码如图所示    输出部分结果如图所示    （b）编写代码如图所示    输出结果如图所示    （c）编写代码如图所示    输出结果如图所示    （d）编写代码如图所示    输出结果如图所示    **Exercise 3: While loops**  Use a while loop to find the first 20 numbers that are divisible by 5, 7 and 11, and print them Hint: store the number found so far in a variable.  编写程序如图所示    输出结果如下    **Exercise 4: Collatz sequence**  A Collatz sequence is formed as follows: We start with some number x0, and we find the next number in the sequence by  If xi = 1, we stop iterating and have found the full sequence.  For example, if we start with x0 = 5, we obtain the sequence:  5 16 8 4 2 1  It is conjectured, though not proven, that every chain eventually ends at 1.  Print the Collatz sequence starting at x0 = 103.  编写代码如图所示    输出部分结果如下    **Exercise 5:**  Print the numbers that are composed of 4 digits (1,2,3 and 4), and each of the digits is different  编写代码如图所示，本实验我采用了递归的方法实现，详细的算法介绍已在注释中说明，我基于已布置实验拓展了全排列的功能，实现了对任意长度的互不相同元素进行全排列并输出。在这个实验过程中，我注意到了python与以往常用的C++、C语言不同之处，传递进python函数的实参会随着形参的改变而改变，而通过.copy()传递实参的拷贝可以避免这一情况，在递归之中比较重要    四位互不相同元素全排列的输出结果和五位互不相同元素全排列的部分输出结果如下图 | | |
| 指导教师批阅意见：   1. 实验报告的完整性：   □实验报告完整  □实验报告不够完整  □实验报告缺少内容   1. 实验结果的正确性   □实验结果正确无误，  □有小的逻辑错误  □代码有很大错误   1. 代码的简洁和效率   □代码简洁优雅，可读性好，能高效运行  □代码具有较好的可读性，但执行效率可进一步优化  □代码可读性不佳，执行效率较好  □代码混乱，效率不佳  成绩评定：  指导教师签字：  2022年 0x 月 xx 日 |
| 备注： |