**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3 - 列表**

**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 03 月 06 日**

**教务处制**

|  |  |  |
| --- | --- | --- |
| **Exercise 1: Short questions**  (a) Write a function that prints the elements of a list  (b) Write a function that prints the elements of a list in reverse  (c) Write your own implementation of the len function that returns the number of elements in a list.  编写代码如下：      运行结果如下    **Exercise 2: Copying lists**  (a) Create a list a with some entries.  (b) Now set b = a  (c) Change b[1]  (d) What happened to a?  (e) Now set c = a[:]  (f) Change c[2]  (g) What happened to a?  Now create a function set\_first\_elem\_to\_zero (l) that takes a list, sets its first entry to zero, and  returns the list. What happens to the original list?    通过输出结果我们可以得知，执行b=a时，b和a指向同一对象，属于引用的相互赋值，二者任意一个列表中的元素改变都会引起对方列表元素改变，属于浅层拷贝，这也是当实参作为参数传递的时候会随形参改变的原因。而当使c=a[:]时，实现的是元素间的赋值，c与a在单层列表的情况下实现深拷贝，c的元素改变不会引起a中元素改变，经过查询资料得知，若a中存在列表元素，则对a列表中的列表元素，这种方法实现的仍然是浅拷贝。若想实现任意层数列表的深拷贝，正确的做法是导入copy包，使用deepcopy()方法来实现。  编写函数set\_first\_elem\_to\_zero如下，正如先前所说，实参到形参的传递过程实际上可以简化为形参=实参，是一个赋值过程，是显然的浅拷贝，因此在函数中更改形参中的数据时，实参的数据也被一同更改了    输出结果如下 | | |
| 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： |