

Python 程序设计 实验 3: 列表、元组的应用

注意事项:

- (1) 实验报告提交截止日期: 2021.04.01, 23:59pm, 迟交扣 20%, 缺交 0 分。
- (2) 实验报告内容包括: 解决问题的思路与方法 (如代码的解释)、遇到的问题以及收获 (简单描述即可)、代码运行结果的展示。
- (3) 实验报告提交方法: **blackboard**。
- (4) 提交要求: 实验报告+源代码, 打包上传, 命名: 学号_姓名_实验报告_3。
- (5) **禁止抄袭**, 一经发现 0 分处理 (包括抄袭者和提供代码或实验报告者)!

1. 查找最大元素的索引: 编写函数 *indicesOfLargestElement(list)*, 输入整数列表 *list*, 返回 *list* 中最大元素所有的索引, 索引以列表形式返回。

例子:

```
indexOfLargestElement([2, 1, 3, 1, 3]) # return [2,4]
```

2. 合并两个排序的列表: 编写函数 *merge(list1, list2)*, 将两个排序好的整数列表合并到一个新的排序列表中, 返回这个新的列表。

使用两种方法实现 *merge* 函数:

- (1) 不使用 *sort()* 或 *sorted()*;
- (2) 使用 *sort()* 或 *sorted()*。

例子:

```
merge([2, 4, 7], [1,5,6]) # return [1, 2, 4, 5, 6, 7]
```

3. 子列表: 编写函数 *match_pattern(list1, list2)*, 仅当 *list2* 是 *list1* 的子列表时返回 *True*。

例子:

```
list1 = [4, 10, 2, 3, 50, 100]
list2 = [3, 2, 50]
list3 = [2, 3, 50]
list4 = [2, 3, 40]
match_pattern(list1, list2) # return False
match_pattern(list1, list3) # return True
match_pattern(list1, list4) # return False
```

4. 模式识别-四个连续相等的数字: 编写函数 *isconsecutivefour(list1)*, 用于测试列表 *list1* 中是否有四个连续相等的数字。编写测试程序, 提示用户输入一系列整数, 返回 *True/False*, 表示是否包含四个连续相等的数字。

5. “几乎对称”列表：如果在一个非对称列表中，我们交换任意两个元素之后，列表是对称的，则把这个列表称为“几乎对称”列表。例如，列表 `lst=[1,2,1,2]` 是几乎对称的，因为交换 `lst[2]`和 `lst[3]`之后，得到对称的列表`[1,2,2,1]`。编写函数 `is_almost_symmetric(lst)`，仅当列表 `lst` 是几乎对称列表时，返回 `True`。

6. 元组的理解：测试并回答以下问题

(1) 下面代码是否正确？解释原因。

```
t = (1, 2, 3)
t.append(4)
t.remove(0)
t[0] = 1
```

(2) 下面代码是否正确？解释原因。

```
t1 = (1, 2, 3, 7, 9, 0, 5)
t2 = (1, 2, 5)
t1 = t2
```

(3) 切片：测试下面代码，解释输出的结果。

```
t = (1, 2, 3, 7, 9, 0, 5)
print(t[3])
print(t[1: 3])
print(t[-1:-3])
print(t[-1:-3:-1])
print(t[-1: -3])
print(t[ : -1: 3])
print(t[3 : -1: 3])
print(t[3 : -1: -3])
```