Python 程序设计 实验 6：函数式编程

注意事项：

（1） 实验报告提交**截止日期：2021.04.29， 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. 函数的参数传递:定义一个简单的函数 sum 如下，

```
def sum(a, b, c):
    print("a=%d, b=%d, c=%d"%(a,b,c))
    print(a+b+c)
```

以下哪些语句是合法的，哪些是不合法的？分别输出什么？解释原因。

a) sum(*(1, 2, 3))
b) sum(1, *(2, 3))
c) sum(*(1,),b=2, 3)
d) sum(*(1,),b=2, c=3)
e) sum(*(1, 2),c=3)
f) sum(a=1, *(2, 3))
g) sum(b=1, *(2, 3))
h) sum(c=1, *(2, 3))

2. Lambda: 思考以下句子的输出，解释每句话意思，并通过实际测试验证自己想法。

3.1. (lambda val: val ** 2)(5)
3.2. (lambda x, y: x * y)(3, 8)
3.3. (lambda s: s.strip().lower()[:2])(' PyTHon')

3. Map: 使用 map 语句将以下输入，分别转化为指定的输出。

3.1. ['12', '-2', '0'] --> [12, -2, 0]
3.2. ['hello', 'world'] --> [5, 5]
3.3. ['hello', 'world']` --> ['olleh', 'dlrow']
3.4. range(2, 6) --> [(2, 4, 8), (3, 9, 27), (4, 16, 64), (5, 25, 125)]
3.5. zip(range(2, 5), range(3, 9, 2)) --> [6, 15, 28]

4. Filter: 使用 filter 语句将以下输入，分别转化为指定的输出。

4.1. ['12', '-2', '0'] --> ['12', '0']

4.2. ['hello', 'world'] --> ['world']

4.3. ['technology', 'method', 'technique'] --> ['technology', 'technique']

4.4. range(20) --> [0, 3, 5, 6, 9, 10, 12, 15, 18]

5. Reduce: 使用 reduce 语句编写函数 lcm(*nums)，计算任意数量个正整数的最小公倍数，**要求只写一句 python 语句** （提示：可使用 math 模块的 gcd 函数先求出最大公约数）。

例子：

```
lcm(3, 5)                    # 15
lcm(41, 106, 12)             # 26076
lcm(1, 2, 6, 24, 120, 720)   # 720
lcm(3)                       # 3
lcm()                        # 如果没有向函数提供数字，可以返回值 1。
```

6. Iterator: 运行以下代码，观察输出并解释输出的原因。

```
it = iter(range(100))
67 in it   # => True

print(next(it))   # => ??
print(37 in it)   # => ??
print(next(it))   # => ??
```

7. **Generator:**

7.1. 编写一个生成器 generate_triangles()，连续地产生三角数 1，3，6，10，… 三角数通过连续的正整数相加来生成（如 1=1，3=1+2，6=1+2+3，10=1+2+3+4，…）。

例子：

```
g=generate_triangles()
for _ in range(5):
    print(next(g))      #输出 1，3，6，10，15
```

7.2. 使用生成器 generate_triangles()，编写函数 generate_triangles_under(n)，返回小于 n 的所有三角数。