实验三　PYTHON 函数定义与面向对象程序设计

一、目的和要求

1. 熟悉 Python 的函数定义；

2. 熟悉 Python 的面向对象定义；

3. 掌握 Python 语言基本语法；

二、实验环境

1. 操作系统不限；

2. Python IDLE、PyCharm 等开发环境；

三、实验内容

（一）验证实验（每个同学完成，第 6-8 小题写入实验报告）

1．运行调试第五章各小节例示代码及课后练习的程序设计题，检查运行结果是否正确，记录实验结果。

2．运行调试第六章各小节例示代码及课后练习的程序设计题，检查运行结果是否正确，记录实验结果。

3．阅读和运行 birthday.py，理解代码的组织结构。

4．阅读和运行 reducible.py，比较该代码和你的小组在实验二中所完成的查找可缩减单词代码查找的效率。

5．函数的作用：编写一个函数，绘制如下的表格。

```
+———————+———————+
|           |           |
|           |           |
|           |           |
|           |           |
+———————+———————+
|           |           |
|           |           |
|           |           |
|           |           |
+———————+———————+
```

比较 Python2.x 和 Python3.x 中 print 的差异。

（提示，同一行打印多个值，可以使用逗号分隔不同的值：print ‘+’, ‘—’；如果值序列的结尾有一个逗号，则 Python 输出不会换行，如：

print ‘+’,

print ‘—’

两个语句的输出是 ‘+—’)

6．阅读和运行 grid.py (支持 Python2.x，请修改代码使得 Python3.x 下同样可以正常运行)，分析 grid 所实现功能中各个函数之间的调用关系，绘制这种调用关系的流程图（可用 Visio 等软件绘制，流程图放到实验报告中）。比较 grid.py 的实现方法和 3（1）中你所实现的绘制表格函数的差异，并且把学习体会写在实验报告中。

7．阅读和运行 myArray.py 和 myMatrix.py，分析其中类的功能，比较类的定义中同名函数实现上的差异，并写入实验报告。

8、阅读和运行 Kangaroo.py，调用和测试其种所定义的类 Kangaroo 的方法，分析方法实现中的 bug，修正，写入实验报告。

（二）设计实验（小组验收，代码提交，算法设计和测试写入实验报告）

1. 函数和数据结构复习

（1）编写 Ackermann 函数的递归实现 Ack（m,n）

$$A(m,n) = \begin{cases} n+1 & \text{if } m = 0 \\ A(m-1,1) & \text{if } m > 0 \text{ and } n = 0 \\ A(m-1, A(m,n-1)) & \text{if } m > 0 \text{ and } n > 0. \end{cases}$$

测试 Ack（3，4）的值，阅读 https://en.wikipedia.org/wiki/Ackermann_function，分析 m 和 n 取值对函数值计算的影响，深入理解递归。

（2）编写一个函数，实现从序列中移除重复项，且保持元素间顺序不变。生成随机的列表和字典，验证所实现函数的功能。

2．编写拥有 a、对象成员 hour，minute 和 second 的时间类 Time；b、重载__str__和__add__方法；c、方法 time2int：把时间对象转换为秒数；d、方法 printtime：输出时间；e、方法 isafter：判断两个时间对象的先后；f、方法 increment：计算对象经过 n〉0 秒后时间；g、方法 isvalid：判断时间对象合法性。在主函数设计代码验证 Time 各个方法的正确性。

3．马尔可夫文本分析和应用

（1）马尔可夫文本分析计算文本中单词组合和其后续单词（含标点符号）的映射，这个单词组合被称为马尔可夫分析的前缀，前缀中单词的个数被称为马尔可夫分析的“阶数”。编写 Python 代码实现某个文本的 n 阶马尔可夫文本分析，并且将分析结果记录在字典中。

（2）采用（1）所实现的马尔可夫分析模块，对“emma.txt”或“whitefang.txt”进行马尔可夫分析，运用 n 阶马尔可夫分析的结果生成由 m 个句子（注意首字母大写和结尾标点符号）组成的随机文本。分析所生成文本的语义自然性和阶数 n 的关系。

（3）尝试采用 Python 不同的序列数据结构表示前缀，比较运行效率的差异。

4．模拟快餐订餐场景

（1）定义 4 个类：Customer 顾客类，Employee 商户类，Food 食物类 以及 Lunch 订餐管理。

（2）Lunch 类包含 Customer 和 Employee 实例，具有下单 order 方法，该方法要求 Customer 实例调用自身的 placeOrder 向 Employee 对象要求下单，并且获得 Employee 对象调用 takeOrder 生成和返回一个 Food 对象，Food 对象应当包含了食物名字符串。调用关系如下：

Lunch.order—〉Customer.placeOrder—〉Employee.takeOrder—〉Food

（3）Lunch 类包含 result 方法，要求 Customer 打印所收到的食物订单。

（4）编写交互式界面验证所设计的订餐系统。

5．编制系列单词处理函数，分别实现下述功能，并设计测试用例验证程序的正确性，请在实验报告中说明所使用的正则表达式。

（1）编写函数 rotateword，接收一个字符串 strsrc 以及一个整数 n 作为参数，返回新字符串 strdes，其各个字母是 strsrc 中对应位置各个字母在字母表中“轮转”n 字符后得到的编码。

（2）编写函数 avoids，接收一个单词和一个含有禁止字母的字符串，判断该单词是否含有禁止字母。

（3）编写函数 useonly，接收一个单词和一个含有允许字母的字符串，判断该单词是否仅仅由允许字母组成。

（4）编写函数 useall，接收一个单词和一个含有需要字母的字符串，判断该

单词是否包含了所有需要字母至少一个，并输出 words.txt 中使用了所有元音字母 aeiou 的单词。

（5）编写函数 hasnoe，判断一个英语单词是否包含字母 e，并计算 words.txt 中不含 e 的单词在整个字母表中的百分比。

（6）编写函数 isabecedarian，判断一个英语单词中的字母是否符合字母表序，并且输出 words.txt 中所有这样的单词。