Python 程序设计 作业 1

注意事项：

（1） 作业提交**截止日期：2021.04.28，23:59pm**，迟交扣 20%，缺交 0 分。
（2） 提交方法：**Blackboard。**
（3） 提交要求：作业回答（word 文件）+源代码，打包上传，命名：学号_姓名_作业_1。
（4） 作业回答包括问答题的解释，代码题的解题思路、运行结果等。
（5） 如同一题目需要多个 Python 文件运行，请写清楚程序运行方法。
（6） **禁止代码抄袭；一经发现，抄袭者和提供代码者统一 0 分处理！**

1. 矩阵：在 Python 中，我们可以使用列表的列表 (a list of lists) 来存储矩阵，每个内部列表代表矩阵一行。例如，我们可以用

```
M = [ [5, 6, 7],
      [0, -3, 5] ]
```

来存储矩阵

$$\begin{pmatrix} 5 & 6 & 7 \\ 0 & -3 & 5 \end{pmatrix}$$

我们可以使用 M[1]来访问矩阵的第二行（即[0, -3, 4]），也可以使用 M[1][2]来访问矩阵的第二行的第三项（即 5）。

(a) 编写函数 *matrix_dim(M)*，该函数输入上面格式的矩阵 M，返回矩阵 M 的维度。例如, matrix_dim([[1,2],[3,4],[5,6]]) 返回 [3, 2]。

(b) 编写函数 *mult_M_v(M, v)*, 返回 $n \times m$ 矩阵 M 和 $m \times 1$ 向量 v 的乘积。

(c) 编写函数 *transpose(M)*, 返回矩阵的转置。

(d) 编写函数 *largest_col_sum(M)*, 寻找矩阵 M 中元素总和最大的列，如上面矩阵中第三列元素的总和最大，则返回 12（7+5=12）。

(e) 编写函数 *switch_columns(M, i, j)*, 交换矩阵 M 的第 i 列和第 j 列，返回新的矩阵 M。

(f) 把以上函数（a-e）写进模块 matrix.py。编写 test_matrix.py 调用模块 matrix 并测试函数（a-e）。

2. 信用卡号码

我们知道信用卡的号码非常长，但是仔细研究会其中也有规律，例如

（1） American Express 使用 15 位数字，MasterCard 使用 16 位数字，Visa 使用 13 位或 16 位数字。

（2） American Express 号码以 34 或 37 开头；MasterCard 号码以 51、52、53、54 或 55 开头；Visa 号码以 4 开头。

（3） 信用卡号码有一个内置的“数学关系”。

对于这个“数学关系”，大多数信用卡都使用 IBM 的 Hans-Peter Luhn 发明算法。根据 Luhn 的算法，我们可以确定信用卡号是否有效。方法如下

（1） 从倒数第二个数字开始，每隔一个数字乘以 2；然后将得到的所有乘积的各个数位相加（注意不是乘积本身相加），得到结果 sum1；

（2） 对于在（1）中没有乘上 2 的数字，直接相加得到结果 sum2；

（3） 把（1）的结果和（2）的结果相加，即 sum3=sum1+sum2。若 sum3 的最后一位数字是 0，则信用卡号码有效。

例如有一个号码是 4003600000000014，在第一步中，我们把标红色的数字乘以 2 然后相加，得到

$$1*2+0*2+0*2+0*2+0*2+6*2+0*2+4*2 = 2+12+8$$

将所得的所有乘积的各个数位相加，得到

$$sum1 = 2+1+2+8 = 13$$

在第二步中，我们把标黑色的数字直接相加，得到

$$sum2 = 4+0+0+0+0+0+3+0 = 7$$

最后第三步 sum3 = sum1+sum2 = 20，因为 20 的最后一位是 0，那么这个信用卡卡号是有效的。

任务：

编写函数 *validCredit(cardNum)*，检查输入信用卡号码 cardNum 是否有效；若有效，则输出该卡的发行公司（American Express，Mastercard，Visa）。

例子：

```
validCredit(4003600000000014)    # 输出：Valid，Visa
validCredit(6177292929)    # 输出：Invalid
```

参考的有效信用卡号码：

American Express	378282246310005
American Express	371449635398431
MasterCard	5555555555554444
Visa	4111111111111111
Visa	4012888888881881
Visa	4222222222222

3. 分析名著

编写程序，通过统计名著中使用的 20 个最常用单词来分析两位作家的写作风格。我们分析的名著和作家如下：

- *The Strange Case of Dr. Jekyl and Mr. Hyde*, Robert Louis Stevenson (hyde.txt)
- *Treasure Island*, Robert Louis Stevenson (treasure.txt)
- *War and Peace*, Leo Tolstoy (war.txt)

我们定义单词是连续的字母序列（小写或大写）。一个单词紧靠前的字符，以及紧随其后的字符是一个非字母。比如，缩写 we’re，我们把它看成两个单词 we 和 re。本次作业无需考虑诸如 we’re，other’的单词。

任务和步骤：

（1）编写函数 *parse(string)*，接受字符串参数 string，并返回一个列表。列表包含 string 中所有长度至少为 4 个字母的单词，且所有单词都为小写。例如 string 为

“Shenzhen’s big, beautiful and rich place”

则返回

[“Shenzhen”, “beautiful”, “rich”, “place”]

（2）编写函数 *mostFrequentWords(filenames)*，接受参数 filenames，表示从文件 filenames 中读取数据，函数返回类型不限制。

（3）统计两位作家最常用的 20 个单词。注意三部名著中有两部是 Robert Louis Stevenson 的，需要综合考虑此作家的两部名著，给出他最常用的 20 个单词。

4. 生活中很多例子都涉及到字符串的“接近”问题。比如我们搜索 Pytho，搜索引擎会回答“您是指 Python 吗？”又如，科学家检查一些核苷酸序列，想知道基因序列 AGTCGTC 和 TAGTCGT 有多匹配，或者说有多接近。

本题要探讨的一个大问题是：什么时候我们可以认为一个字符串与另一个字符串接近？或者说，我们什么时候可以将一个字符串视为另一个字符串的“邻居”？这里的“邻居”有三种可能的定义：

- 如果两个字符串除了在一个位置上不一样，其他位置都一样，如“abc”和“abe”；
- 如果可以通过交换一个字符串中的两个相邻字符来获得另外一个字符串，如“abc”和“acb”；
- 如果从一个字符串中删除一个字符可以生成另一个字符串，如 “abc”和“abxc”。

任务和步骤：

（1） 编写函数 offByOne(str1,str2),输入参数为两个非空的字符串 str1 和 str2。仅当 str1 和 str2 具有相同的长度并且只在一个位置上不同时，返回 True。例如：

str1	str2	return
"read"	"rexd"	True
"read"	"xexd"	False
"read"	"readx"	False
"read"	"eadx"	False
"a"	"x"	True
"a"	"a"	False
"a"	"A"	True

（2） 编写函数 offBySwap(str1,str2),输入参数为两个非空的字符串 str1 和 str2。仅当 str1 不等于 str2，且可以将 str1 中任意两个相邻字符交换可以得到 str2 时，返回 True。 例如：

str1	str2	return
"read"	"raed"	True
"read"	"erad"	True
"reaxd"	"read"	False
"read"	"erda"	False
"read"	"erbx"	False
"x"	"Y"	False
"aaa"	"aaa"	False

（3） 编写函数 offByExtra(str1,str2),输入参数为两个非空的字符串 str1 和 str2。仅当从 str1 中删除一个字符可以得到 str2，或者从 str2 中删除一个字符可以得到 str1 时，返回 True。 例如

str1	str2	return
"abcd"	"abxcd"	True
"abxcd"	"abcd"	True
"abcda"	"abcd"	True
"abcd"	"bcda"	False
"abcd"	"abcdef"	False
"abcd"	"abcd"	False

（4） 我们定义两个字符串 str1 和 str2 是邻居，仅当 str1 和 str2 不是同一个字符串，并且三个函数 offByOne(str1,str2)， offBySwap(str1,str2)，offByExtra(str1,str2)中任意一个为 True。

按照上述定义，编写函数 ListOfNeighbors(str,L)，输入参数 str 为字符串（要求全小写），L 为字符串列表（见英文单词表 EnglishWords.txt），返回一个列表，为 L 中所有的 str 的邻居。