

Python 程序设计 作业 1

注意事项:

- (1) 作业提交截止日期: 2022.04.30, 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) 作业回答写解题思路、运行结果等(不需要使用实验报告模板)。
- (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. 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 $\text{sum3} = \text{sum1} + \text{sum2}$ 。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 400360000000014, 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

$$\text{sum1} = 2+1+2+8 = 13$$

在第二步中, 我们把标黑色的数字直接相加, 得到

$$\text{sum2} = 4+0+0+0+0+0+3+0 = 7$$

最后第三步 $\text{sum3} = \text{sum1} + \text{sum2} = 20$, 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	42222222222222

2. 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. Jekyll 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：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。

3. 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。

- 3. MasterMind:** 我们将实现一个简单的数字猜测游戏，通常被称为 **MasterMind**。游戏中，游戏管理员创建一个秘密数字，游戏玩家猜数字。管理员会给玩家一些关于他猜测的反馈。根据这一反馈，玩家做出另一个猜测。猜测一直持续到猜出秘密数字或达到最大尝试次数为止。理想情况下，管理员和玩家应该是不同的人，但出于测试的目的，我们这里不区分管理员和玩家。

要求：

- (1) 游戏开始时提示输入数字。数字长度应为 5 位，不得出现任何重复数字。例如，12345 满足要求，55432 和 444 不满足。如果输入了不满足的数字，请打印错误诊断信息并要求重新输入格式正确的数字。
- (2) 让玩家提供一个 5 位数的数字。如果玩家输入的不是 5 位数字（包括输入不全是数字、长度不为 5 等），请玩家提供另一个猜测。
- (3) 当玩家给出 5 位数的猜测，报告两个结果：
 - a) 首先，报告真实秘密数字中有多少位出现在了猜测数字中（正确位数），与是否在正确位置无关；
 - b) 其次，报告猜测数字中有多少位处于正确的位置（正确位置位数）。

例如，假设秘密数字是 12345，玩家猜测是 12567。程序应该将正确位数报告为 3，将正确位置位数报告为 2。

- (4) 每次猜测后，报告玩家的进度（猜测次数、猜测数字、正确位数、正确位置位数）。
- (5) 最后，如果玩家成功猜出了数字，打印出玩家使用了多少次猜数。
- (6) 为游戏设定最大猜测次数（在游戏开始时报告）。如果玩家超过此上限，则报告玩家失败。
- (7) 在此过程中，允许玩家退出游戏。如果他们退出，报告他们输了，猜测的次数是多少。

- 4. 抓狐狸:** 编写程序，模拟抓狐狸小游戏。假设一共有一排 `N` 个洞口，小狐狸最开始的时候出现在其中一个洞口，然后玩家随机打开一个洞口，如果小狐狸在里面就被抓到了。如果洞口里面没有狐狸就需要到第二天再来抓，但是小狐狸第二天会在玩家来抓之前随机跳到隔壁的洞口里。如果在规定的次数

内抓到了小狐狸就提前结束游戏并提示成功；如果规定次数用完之后还没有抓到小狐狸，就结束游戏并提示失败。

自行设计游戏，如游戏输出提示、实现细节、使用的数据结构等等。满足上述游戏要求即可。