

实验二： Python 函数和面向对象的实践应用

(1) 编写程序，输出 n 行由大写字母 A 开始构成的三角形字符阵列。例如输入为 4，输出如下：

```
A B C D
E F G
H I
J
```

(2) 定义一个函数来判断单词 m 是否可以由字符串 n 中出现的字母来组成。
本题保证字符串中出现的字母均为小写字母，且不考虑 n 中的字母使用次数。在两行中分别输入两个字符串 m,n。如果 m,n 满足条件，则输出'FOUND'，否则输出'NOT FOUND'。如果输入的 m 包含有除字母外的其他字符，输出'ERROR'结束

输入输出示例

输入		输出
示例 1	word world	FOUND
示例 2	1a3e	ERROR
示例 3	at bcda	NOT FOUND

(3) 身份证号校验

描述

中国目前采用的是18位身份证号，其第7-10位数字是出生年，11-12位是出生月份，13-14是出生日期，第17位是性别，奇数为男性，偶数为女性，第18位是校验位。如果身份证号码的其中一位填错了（包括最后一个校验位），则校验算法可以检测出来。如果身份证号的相邻2位填反了，则校验算法可以检测出来。校验规则如下：

1. 将前面的身份证号码17位数分别乘以不同的系数。从第一位到第十七位的系数分别为：7 - 9 - 10 - 5 - 8 - 4 - 2 - 1 - 6 - 3 - 7 - 9 - 10 - 5 - 8 - 4 - 2。
2. 将这17位数字和系数相乘的结果相加。
3. 用加出来和除以11，看余数只可能是：0 - 1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10

分别对应的最后一位身份证的号码为：1 - 0 - X - 9 - 8 - 7 - 6 - 5 - 4 - 3 - 2

4. 通过上面得知如果余数是2，就会在身份证的第18位数字上出现罗马数字的X（大写英文字母X）。如果余数是10，身份证的最后一位号码就是2。

用户输入一个身份证号，校验其是否是合法的身份证号码。

输入格式

一个18位身份证号，末位为数字或大写字母X

输出格式

'身份证号码校验为合法号码!' 或 '身份证校验位错误!'

输入输出示例

	输入	输出
示例 1	220221197302286534	身份证校验位错误!
示例 2	220221197302296536	身份证号码校验为合法号码!

(4) 个人信息提取（字符串）

描述

用户输入自己的个人信息，格式如下：

```
0122923450321 王昊 法学1801 河北 2001年
```

数据分别表示：学号 姓名 专业班级 籍贯 出生年份，各数据间空格间隔

有些用户没有按照规则输入数据，输入自己出生年份的时候写成了类似 ‘1900年生’ 的数据，题目保证这些用户输入年份时一定是以四位数字开始
请注意程序仍然需要正确读取该项数据

请按照输出样式输出姓名，班级，年龄

提示：

列表中的数据 and 字符串当中的字符一样，都具有有序的索引，且引用数据和切片方式一致。

输入示例

```
0122923450321 王昊 法学1801 河北 2001年
```

输出示例

```
姓名：王昊  
班级：法学1801  
年龄：19岁
```

	输入	输出
示例 1	0122923450321 王昊 法学1801 河北 2001年	姓名：王昊 班级：法学1801 年龄：19岁

(5) 个人数据脱敏

描述

随着计算机与互联网技术快速发展，电话号码，家庭住址，姓名等个人隐私信息被泄露的风险也越来越高。

数据脱敏是指对敏感信息进行变形处理，比如将电话号码 ‘13000000000’ 中的四位用 ‘*’ 来代替，变为 ‘130****0000’ 。

本题要求从输入的学生信息中将手机号码，姓名，学号数据进行脱敏处理。

处理规则如下：

学号（13 位或 14 位数串）：第 5-11 位修改为 ‘*’ 如 ‘0121134567801’ 修改为 ‘0121*****01’

姓名：第2位修改为‘*’ 如 ‘贾诩’ 修改为‘贾*’

电话号码（11位数串）：第4-7位 修改为‘*’ 如 ‘13000000000’ 中修改为 ‘130****0000’

输入

第一行输入 n ，代表 n 个同学的信息

依次输入 n 行学生信息数据，每行保存一名学生的学号、姓名、电话号码以空格间隔

输出

将学生数据脱敏后的信息保存到二维列表中，并输出这个二维表。

如果 n 不是正整数，输出 'ERROR'。（本题保证 n 是整数，且信息完整）

输入输出示例

输入	输出
<div>示例 1</div> <div>2 01211345678011 张辽 14457673508 01211345678012 徐晃 18514863218</div>	<div>[['0121*****011', '张*', '144****3508'], ['0121*****012', '徐*', '185****3218']]</div>

（6）用字典来统计词频

描述

使用 `freqDict = eval(input())` 读入单词词频字典，再读入一段英文，默认按照英文输入的顺序，统计更新单词词频字典，并输出。

输入格式：

输入为两行，第一行是一个字典，形如 `{'hello': 12, 'world': 10}`，其中存储初始的词频数据。第二行是一段英文文本。

输出格式：

输出一行，直接打印输出更新后的字典。

输入输出示例

输入	输出
<div>示例 1</div> <div>{ hello world</div>	<div>{'hello': 1, 'world': 1}</div>
<div>示例 2</div> <div>{'and':20,'in':10} Anna and Elsa</div>	<div>{'and': 21, 'in': 10, 'Anna': 1, 'Elsa': 1}</div>