## **一.实验目的**

1. 熟悉并掌握Python中函数的定义方式、调用方法、参数传递以及结果的返回
2. 建立使用函数来处理问题的逻辑，能够定义合适的函数来解决问题
3. 体会模块化编程的便利，思考使用函数对于代码可读性、复用性、处理能力的影响

## **二.实验内容**

### **背景知识**

国际标准书号（International Standard Book Number），简称ISBN，是专门为识别图书等文献而设计的国际编号。采用ISBN编码系统的出版物有：图书、小册子、缩微出版物、盲文印刷品等。2007年1月1日前，ISBN由10位数字组成，分四个部分：组号(国家、地区、语言的代号)，出版者号，书序号和检验码。2007年1月1日起，实行新版ISBN，新版ISBN由13位数字组成，分为5段，即在原来的10位数字前加上3位EAN（欧洲商品编号[1]  ）图书产品代码“978”。（来自 百度百科）

我们考虑2007年前的ISBN编码。

### **ISBN实例**

《算法导论》第二版中文译本的ISBN为7-111-18777-6

其他可能的ISBN例子：0-13-030657-6，0-32-108599-X和0-471-58719-2

### **题目描述**

ISBN前9个字符是数字，而最后一个字符是一个数字或字母X（代表10）。连字符将这些字符分为四个区块。第一个区块表示国家或地区编号（如0代表美国所出版的内容，7为中国大陆出版物使用的号码等）。第二个区块标识出版商。第三区块是出版商为一本书选择的数字。第四区块总是包含一个字符，叫做验证位，用于测试错误。

让我们将ISBN的10个字符从左往右分别记为d1，d2，d3，d4，d5，d6，d7，d8，d9，d10。选择验证位使得

的和是11的倍数。如果ISBN的最后一位是一个X，那么求和的时候，d10则用10代替。

以ISBN号0-32-108599-X为例，其和应该是

由于165/11是15，其和是11的倍数，其是一个合法的ISBN。

上述验证机制可以检测每一位数字和相邻数字移位错误。也就是说，如果在复制一个ISBN号码时，你错抄了一位或调换了相邻的两个字符，其和也就不会是11的倍数。

编写程序接受连续的多行输入，每行输入表示一个ISBN号码（包括连字符）。对每个ISBN号码，计算其和(\*)，并给出它是否是一个合法的ISBN。在计算和之前，程序应该检查前9位的每一位都是数字，并且最后一位是一个数字或字母X(大写或小写)。输入单独一行"$$$"表示输入结束，程序退出。

如果用户输入的ISBN不满足格式要求，如数字位数不足、分隔符个数不合规范、包含非法字符等，程序输出"Invalid ISBN Format"

如果用户输入的ISBN合乎格式要求，但不满足ISBN校验性质（按规则求和，所得值是11倍数），程序输出"Invalid ISBN"

如果输入ISBN正确，程序输出"Valid ISBN"

*注：以上要求，需要按顺序满足*

*提示：一些可能用到的Python内建（built-in）字符串操作函数搜索提示：字符串按某个字符(串)分割的函数*

## **三.测试用例**

****sample 1****

****input****

 0-32-108599-X

 0-32-10859-32

 0-32108599-X

 0-3X-108599-X

 0-32-108589-X

$$$

****output****

Valid ISBN

Invalid ISBN Format

Invalid ISBN Format

Invalid ISBN Format

Invalid ISBN

## **四.实验要求**

1. 编写合适的函数，完成实验功能。
2. 使用规范的变量名定义，在合适的地方添加真实代码逻辑的描述——注释

## **\*五.兴趣扩展**

现在我们通常看到的书籍的ISBN为13位，有兴趣的同学可以网上查询一下新版ISBN编码的校验规则，再编写一个程序自己测试一下。

更好地，可以把这两个程序合起来，能够自动根据输入的ISBN分辨出是新版或旧版的ISBN并作出是否合法的判断。相信这样做能让你更加体会到使用函数的方便和灵活。