程序员的算法趣题之Q02：数列的四则运算

有一种游戏叫做“组合数字”，组合的方法是在给定的数字之间插入四则运算的运算符组成算式，然后计算算式的结果（某些数位之间可以没有运算符，但最少要插入1个运算符）。

这样说很抽象，举个栗子：

1234 → 1+2×3-4 = 3

9876 → 9×87+6 = 789

假设这里的条件是，计算结果为“将原数字逆序排列得到的数”，并且算式的运算遵循四则运算法则（先乘除，后加减）。

比如351 → 3×51 = 153

**那么问题来了……**

求位于1000-9999，满足上述条件的数。

好好思考一下吧！

==================分割线===================

思路

解决这个问题时，“计算算式的方法”会影响实现方法。如果要实现的是计算器，那么通常会用到逆波兰表示法（所有操作符置于操作符的后面，也称为后缀表示法）。而本题则是使用编程语言内置的功能来实现更为简单。

Python代码如下

op = [**"+"**, **"-"**, **"\*"**, **"/"**, **""**]  
*# 可以将上述运算符改为  
# op = ["\*", ""]  
# 因为很容易发现，一旦用了'\*'以外的任意运算符，最终的结果就凑不够4位数了***for** num **in** range(1000, 10000):  
 *# 排除末尾数字为0的情况* **if** num % 10 == 0:  
 **continue** numlist = list(str(num))  
 **for** op1 **in** op:  
 **for** op2 **in** op:  
 **for** op3 **in** op:  
 *# 排除对0开头的数字进行运算的情况，比如1\*01等* **if** (op1 != **'' and** numlist[1] == **'0'**) **or** (op2 != **'' and** numlist[2] == **'0'**):  
 **continue** val = numlist[0] + op1 + numlist[1] + op2 + numlist[2] + op3 + numlist[3]  
 **if** len(val) > 4: *# 至少插入一个运算符* **if** int(str(num)[::-1]) == eval(val): *# 如果逆序排序的数字与运算后的结果相等* print(str(num)+**" ---> "**+val+**"="**+str(num)[::-1])

5931 ---> 5\*9\*31=1395

参考《程序员的算法趣题》