火柴等式

**时间限制: 2000/1000 MS (Java/Others)**

**内存限制: 524288/524288 K (Java/Others)**

**问题描述**

给出一个火柴棒表示的三位数加法等式。移动、删除或添加一根火柴算作一次操作。请问，最少需要多少次操作才能变成合法的三位数加法等式。

注意：

1. 只能对数字进行操作，不可以操作加法符号和等号。
2. 等式只能由三个可以包含前导零的三位数、一个加号、一个等号组成。
3. 10种数字的形状如下（其中‘1’视为靠右的两根火柴组成）：



**输入描述**

第一行一个正整数 表示数据组数。

接下来 行每行一个等式，具体格式请参考输入样例。

**输出描述**

行，每行一个整数表示最小操作次数。

**输入样例**

4

123+321=444

088+111=999

808+111=991

009+004=009

**输出样例**

0

1

1

1

**样例解释**

第一组，等式正确，不需要操作。

第二组，将0添加一根火柴，变为8。

第三组，将第二个8的火柴移动到0的中间。

第三组，将第一个9的一根火柴去掉，变成5。