

# NOIP 普及组复赛 B 类题解思路 (C++)

-----2016 T2

## 日期回文

### 【问题描述】

在日常生活中，通过年、月、日这三个要素可以表示出一个唯一确定的日期。牛牛习惯用 8 位数字表示一个日期，其中，前 4 位代表年份，接下来 2 位代表月份，最后 2 位代表日期。显然：一个日期只有一种表示方法，而两个不同的日期的表示方法不会相同。

牛牛认为，一个日期是回文的，当且仅当表示这个日期的 8 位数字是回文的。现在，牛牛想知道：在他指定的两个日期之间包含这两个日期本身），有多少个真实存在的日期是回文的。一个 8 位数字是回文的，当且仅当对于所有的  $i$  ( $1 \leq i \leq 8$ ) 从左向右数的第  $i$  个数字和第  $9-i$  个数字（即从右向左数的第  $i$  个数字）是相同的。

例如：

- 对于 2016 年 11 月 19 日，用 8 位数字 20161119 表示，它不是回文的。
- 对于 2010 年 1 月 2 日，用 8 位数字 20100102 表示，它是回文的。
- 对于 2010 年 10 月 2 日，用 8 位数字 20101002 表示，它不是回文的。

每一年中都有 12 个月份：其中，1、3、5、7、8、10、12 月每个月有 31 天；4、6、9、11 月每个月有 30 天；而对于 2 月，闰年时有 29 天，平年时有 28 天。一个年份是闰年当且仅当它满足下列两种情况其中的一种：

- 1.这个年份是 4 的整数倍，但不是 100 的整数倍；
- 2.这个年份是 400 的整数倍。

### 【输入格式】

输入包括两行，每行包括一个 8 位数字。第一行表示牛牛指定的起始日期。第二行表示牛牛指定的终止日期。保证  $date_1$  和  $date_2$  都是真实存在的日期，且年份部分一定为 4 位数字，且首位数字不为 0。保证  $date_1$  一定不晚于  $date_2$ 。

### 【输入输出样例】

输出一行，包含一个整数，表示在  $date_1$  和  $date_2$  之间，有多少个日期是回文的。

样例输入 1 Sample Input

20110101

20111231

样例输出 1 Sample Output

1

样例输入 2 Sample Input

20000101

20101231

样例输出 2 Sample Output

2

### 【样例说明】

对于样例 1，符合条件的日期是 20111102。

对于样例 2，符合条件的日期是 20011002 和 20100102。

### 【子任务】

对于 60%的数据，满足  $\text{date1} = \text{date2}$ 。

## 日期回文解析

- 1、按年反转解法，首先看年份表达 20011002，注意到前四位和后四位成反，对于给定年份，只需要判断前四位和后四位是否相反，即可得到 60 分。
- 2、第二步判断前四位翻转后的月份和日期是否是合法日期，如果不是合法日跳过。
- 3、合法日期，月份数字在 1-12 闭区间，日期在 1-31 闭区间，不用判断大小月，因为小月反过来的年份不合法，所以不用判断（比如：0321 1230），需要判断**不是闰年**的 2 月是否是 29，**是 29**也不合法。
- 3、要冲刺满分的同学在第一步完成后再写验证合法的分步函数，先保证 60 分。
- 4、第二解法，按照天数，计算每一天是否合法，这样的方法判断复杂，效率低下，容易出错。相比按年解法，按年运算次数每年一次运算，第二解法是按照日。比如间隔 5 年，按年解法计算 5 次，而第二解大于是  $5 \times 365$  次，还要外加判断大小月，计算量大。找到最优解是编程乐趣之一。
- 5、将闰年判断程序背下来，考试时候盲打。

```
bool isLeap(int date_year)
```

```
{  
  if(date_year%100!=0&&date_year%4==0)  
  {  
    return true;  
  }  
  if(date_year%400 == 0)  
  {  
    return true;  
  }  
  return false;  
}
```