

# 1. 图书管理员

(librarian.cpp/c/pas)

## 【问题描述】

图书馆中每本书都有一个图书编码，可以用于快速检索图书，这个图书编码是一个正整数。

每位借书的读者手中有一个需求码，这个需求码也是一个正整数。如果一本书的图书编码恰好以读者的需求码结尾，那么这本书就是这位读者所需要的。

小D 刚刚当上图书馆的管理员，她知道图书馆里所有书的图书编码，她请你帮她写一个程序，对于每一位读者，求出他所需要的书中图书编码最小的那本书，如果没有他需要的书，请输出-1。

## 【输入格式】

输入文件名为 librarian.in。

输入文件的第一行，包含两个正整数 n 和 q，以一个空格分开，分别代表图书馆里书的数量和读者的数量。

接下来的n 行，每行包含一个正整数，代表图书馆里某本书的图书编码。

接下来的q 行，每行包含两个正整数，以一个空格分开，第一个正整数代表图书馆里读者的需求码的长度，第二个正整数代表读者的需求码。

## 【输出格式】

输出文件名为 librarian.out。

输出文件有 q 行，每行包含一个整数，如果存在第 i 个读者所需要的书，则在第 i 行输出第i 个读者所需要的书中图书编码最小的那本书的图书编码，否则输出-1。

## 【输入输出样例 1】

librarian.in	librarian.out
5 5	23
2123	1123
1123	-1
23	-1
24	-1
24	
2 23	
3 123	
3 124	
2 12	
2 12	

见选手目录下的 librarian/librarian1.in和librarian/librarian1.ans。

## 【输入输出样例 1 说明】

第一位读者需要的书有 2123、1123、23，其中 23 是最小的图书编码。第二位读者需要的书有 2123、1123，其中 1123 是最小的图书编码。对于第三位，第四位和第五位读者，没有书的图书编码以他们的需求码结尾，即没有他们需要的书，输出-1。

**【输入输出样例 2】**

见选手目录下的 `librarian/librarian2.in` 和 `librarian/librarian2.ans`。

**【数据规模与约定】**

对于 20% 的数据， $1 \leq n \leq 2$ 。

另有 20% 的数据， $q = 1$ 。

另有 20% 的数据，所有读者的需求码的长度均为 1。

另有 20% 的数据，所有的图书编码按从小到大的顺序给出。

对于 100% 的数据， $1 \leq n \leq 1,000$ ， $1 \leq q \leq 1,000$ ，所有的图书编码和需求码均不超过 10,000,000。

## 2、扫雷游戏

(mine.cpp/c/pas)

### 【问题描述】

扫雷游戏是一款十分经典的单机小游戏。在  $n$  行  $m$  列的雷区中有一些格子含有地雷（称之为地雷格），其他格子不含地雷（称之为非地雷格）。玩家翻开一个非地雷格时，该格将会出现一个数字——提示周围格子中有多少个是地雷格。游戏的目标是在不翻出任何地雷格的条件下，找出所有的非地雷格。

现在给出  $n$  行  $m$  列的雷区中的地雷分布，要求计算出每个非地雷格周围的地雷格数。  
注：一个格子的周围格子包括其上、下、左、右、左上、右上、左下、右下八个方向上与之直接相邻的格子。

### 【输入格式】

输入文件名为 mine.in。

输入文件第一行是用一个空格隔开的两个整数  $n$  和  $m$ ，分别表示雷区的行数和列数。

接下来  $n$  行，每行  $m$  个字符，描述了雷区中的地雷分布情况。字符 '\*' 表示相应格子是地雷格，字符 '?' 表示相应格子是非地雷格。相邻字符之间无分隔符。

### 【输出格式】

输出文件名为 mine.out。

输出文件包含  $n$  行，每行  $m$  个字符，描述整个雷区。用 '\*' 表示地雷格，用周围的数字表示非地雷格。相邻字符之间无分隔符。

### 【输入输出样例 1】

mine.in	mine.out
3 3	*10
*??	221
???	1*1
?*?	

见选手目录下的 mine/mine1.in 和 mine/mine1.ans。

### 【输入输出样例 2】

mine.in	mine.out
2 3	2*1
?*?	*21
*??	

见选手目录下的 mine/mine2.in 和 mine/mine2.ans。

### 【输入输出样例 3】

见选手目录下的 mine/mine3.in 和 mine/mine3.ans。

### 【数据说明】

对于 100% 的数据， $1 \leq n \leq 100$ ， $1 \leq m \leq 100$ 。

### 3、比例简化

(ratio.cpp/c/pas)

#### 【问题描述】

在社交媒体上，经常会看到针对某一个观点同意与否的民意调查以及结果。例如，对某一观点表示支持的有 1498 人，反对的有 902 人，那么赞同与反对的比例可以简单的记为 1498:902。

不过，如果把调查结果就以这种方式呈现出来，大多数人肯定不会满意。因为这个比例的数值太大，难以一眼看出它们的关系。对于上面这个例子，如果把比例记为 5:3，虽然与真实结果有一定的误差，但依然能够较为准确地反映调查结果，同时也显得比较直观。

现给出支持人数  $A$ ，反对人数  $B$ ，以及一个上限  $L$ ，请你将  $A$  比  $B$  化简为  $A'$  比  $B'$ ，要求在  $A'$  和  $B'$  均不大于  $L$  且  $A'$  和  $B'$  互质两个整数的最大公约数是 1 的前提下， $A'/B' \geq A/B$  且  $A'/B' - A/B$  的值尽可能小。

#### 【输入】

输入文件名为 ratio.in。

输入共一行，包含三个整数  $A$ ， $B$ ， $L$ ，每两个整数之间用一个空格隔开，分别表示支持人数、反对人数以及上限。

#### 【输出】

输出文件名为 ratio.out。

输出共一行，包含两个整数  $A'$ ， $B'$ ，中间用一个空格隔开，表示化简后的比例。

#### 【输入输出样例】

ratio.in	ratio.out
1498 902 10	5 3

#### 【数据说明】

对于 100% 的数据， $1 \leq A \leq 1,000,000$ ， $1 \leq B \leq 1,000,000$ ， $1 \leq L \leq 100$ ， $A/B \leq L$ 。

## 4、回文日期 (date)

### 【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。

### 【提示】

一个8位数字是回文的，当且仅当对于所有的  $i$  ( $1 < i < 8$ ) 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。

例如：

- 以下几个年份都是闰年：2000、2012、2016。
- 以下几个年份是平年：1900、2011、2014 ”

### 【输入格式】

从文件 **date.in** 中读入数据。

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  $date_1$  ”

第二行表示牛牛指定的终止日期  $date_2$  ”

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  $date_2$

**【输出格式】**

输出到文件 *date.out* 中。

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1输入】**

20110101

20111231

**【样例1输出】**

1

**【样例2输入】**

20000101

20101231

**【样例2输出】**

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  $date_1 = date$

