

2022 年 3 月 c 二级真题

1、温度统计

现有一段时间的温度数据，请统计指定温度出现的次数。

时间限制：1000

内存限制：65536

输入

第一行一个整数 n , 表示温度数据的个数。 ($0 < n \leq 200$) 第二行 n 个整数，以空格分隔，每个整数表示一个温度，温度的范围大于等于 0，小于等于 40。 第三行一个整数，表示需要查询的温度 t 。 ($0 \leq t \leq 40$)

输出

输出一个整数，表示温度 t 出现的次数。

样例输入

```
10
18 19 21 17 20 18 21 21 22 21
21
```

样例输出

```
4
```

2、序列排序

对于给定的正整数序列，按照每个数的各位数和从大到小排序，各位数和相同的按照本身大小排序，大的在前，小的在后。

时间限制：1000

内存限制：65536

输入

第一行 1 个整数 n , 表示序列的大小。 ($0 < n \leq 1000$) 第二行 n 个正整数，表示序列的每个数，每个数不大于 100000000。

输出

输出按照题目要求排序后的序列

样例输入

```
6
17 26 9 13 88 10
```

样例输出

```
88 9 26 17 13 10
```

3、单词的长度

输入一行单词序列，相邻单词之间由 1 个或多个空格间隔，请对应地计算各个单词的长度。

注意，如果有标点符号（如连字符，逗号），标点符号算作与之相连的词的一部分。没有被空格间开的符号串，都算作单词。

时间限制：1000

内存限制：65536

输入

一行单词序列，最少 1 个单词，最多 300 个单词，单词之间用至少 1 个空格间隔。单词序列总长度不超过 1000。

输出

依次输出对应单词的长度，之间以逗号间隔。

样例输入

```
She was born in 1990-01-02 and from Beijing city.
```

样例输出

```
3,3,4,2,10,3,4,7,5
```

4、扫雷游戏地雷数计算

扫雷游戏是一款十分经典的单机小游戏。它的精髓在于，通过已翻开格子所提示的周围格地雷数，来判断未翻开格子里是否是地雷。

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的周围格地雷数。

注：每个格子周围格有八个：上、下、左、右、左上、右上、左下、右下。

时间限制：1000

内存限制：65536

输入

第一行包含两个整数 n 和 m ，分别表示雷区的行数和列数。 $1 \leq n \leq 100$, $1 \leq m \leq 100$ 。接下来 n 行，每行 m 个字符，‘*’表示相应格子中是地雷，‘?’表示相应格子中无地雷。字符之间无任何分隔符。

输出

n 行，每行 m 个字符，描述整个雷区。若相应格中是地雷，则用‘*’表示，否则用相应的周围格地雷数表示。字符之间无任何分隔符。

样例输入

```
3 3
*??
???
?*?
```

样例输出

```
*10
221
1*1
```

5、古代密码

古罗马帝国有一个拥有各种部门的强大政府组织。其中一个部门就是保密服务部门。为了保险起见，在省与省之间传递的重要文件中的大写字母是加密的。当时最流行的加密方法是替换和重新排列。

替换方法是将所有出现的字符替换成其它的字符。有些字符会替换成它自己。例如：替换规则可以是将'A'到'Y'替换成它的下一个字符，将'Z'替换成'A'，如果原词是"VICTORIOUS"则它变成"WJDUPSJPVT"。

排列方法改变原来单词中字母的顺序。例如：将顺序例如将顺序 < 2 1 5 4 3 7 6 10 9 8 > 应用到"VICTORIOUS"上，则得到"IVOTCIRSUO"。

人们很快意识到单独应用替换方法或排列方法加密，都是很不保险的。但是如果结合这两种方法，在当时就可以得到非常可靠的加密方法。所以，很多重要信息先使用替换方法加密，再将加密的结果用排列的方法加密。用两种方法结合就可以将"VICTORIOUS"加密成"JWPUDJSTVP"。

考古学家最近在一个石台上发现了一些信息。初看起来它们毫无意义，所以有人设想它们可能是用替换和排列的方法被加密了。人们试着解读了石台上的密码，现在他们想检查解读的是否正确。他们需要一个计算机程序来验证，你的任务就是写这个验证程序。

时间限制：1000

内存限制：65536

输入

输入有两行。第一行是石台上的文字。文字中没有空格，并且只有大写英文字母。第二行是被解读出来的加密前的文字。第二行也是由大写英文字母构成的。两行字符数目的长度都不超过100。

输出

如果第二行经过某种加密方法后可以产生第一行的信息，输出"YES"，否则输出"NO"。

样例输入

```
JWPUDJSTVP
VICTORIOUS
```

样例输出

YES