## A: 判断回文数

**描述**

输入一个正整数， 判断它是不是回文数？（回文数是顺着看和倒着看都一样的数，例如1221，12321都是回文数）。

**输入**

第一行输入一个正整数n，表示有n组数据。

每组数据为一个正整数。

**输出**

对于每组数据，若是回文数，输出“yes”，否则输出“no”

**样例**

|  |  |
| --- | --- |
| 输入 | 输出 |
| 2  121  1231 | yes  no |

## B: 分解质因数

**描述**

请编写程序，将整数分解为若干个质数的乘积。

**输入**

输入的第一行含一个正整数k，表示测试例的个数，后面紧接着k行，每行对应一个测试例，包含一个正整数x(2<=x<=10000)。

**输出**

每个测试例对应一行输出，输出x的质数乘积表示式，式中的质数从小到大排列，两个质数之间用“\*”表示相乘。中间不要添加任何空格。

**样例**

|  |  |
| --- | --- |
| 输入 | 输出 |
| 3  13  4950  262 | 13  2\*3\*3\*5\*5\*11  2\*131 |

## C: 数字显示

**描述**

用点阵的形式在字符界面上显示数字。数字按如下形式显示。每个数字占5列7行，数字和数字之间用2个空格隔开

\*\*\*\*\*      \*  \*\*\*\*\*  \*\*\*\*\*  \*   \*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\*

\*   \*      \*      \*      \*  \*   \*  \*      \*          \*  \*   \*  \*   \*

\*   \*      \*      \*      \*  \*   \*  \*      \*          \*  \*   \*  \*   \*

\*   \*      \*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\*      \*  \*\*\*\*\*  \*\*\*\*\*

\*   \*      \*  \*          \*      \*      \*  \*   \*      \*  \*   \*      \*

\*   \*      \*  \*          \*      \*      \*  \*   \*      \*  \*   \*      \*

\*\*\*\*\*      \*  \*\*\*\*\*  \*\*\*\*\*      \*  \*\*\*\*\*  \*\*\*\*\*      \*  \*\*\*\*\*  \*\*\*\*\*

**输入**

输入的第一行含一个正整数k，表示测试例的个数，后面紧接着k行，每行对应一个测试例，包含一个正整数x(0<=x<=9999)。

**输出**

每个测试例，先输出x和冒号(英文字符，半角)，换行后输出x的点阵显示形式，每位数字之间用2个空格隔开。如果行末有空格，请保留。

**样例**

|  |  |
| --- | --- |
| 输入 | 输出 |
| 2  13  4950 | 13:      \*  \*\*\*\*\*      \*      \*      \*      \*      \*  \*\*\*\*\*      \*      \*      \*      \*      \*  \*\*\*\*\*  4950:  \*   \*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\*  \*   \*  \*   \*  \*      \*   \*  \*   \*  \*   \*  \*      \*   \*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\*  \*   \*      \*      \*      \*  \*   \*      \*      \*      \*  \*   \*      \*  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\* |

## D: 最多的单词

**描述**

统计一段文本中出现次数最多的英文单词。单词不区分大小写，即“Hello”和“hello”表示同一个单词。单词和单词之间由一个或多个**非字母字符**隔开。

**输入**

输入数据为一段文本，可能包含若干行，直到文件末尾(EOF)为止。即只有一组测试用例。

**输出**

输出该段文本中出现最多的单词(全变换成小写字母)，及该单词出现的次数，中间用空格隔开。如果出现次数最多的单词有多个，输出字典序最大(全转换成小写字母比较)的那个。

**样例1**

|  |  |
| --- | --- |
| 输入 | 输出 |
| Is this a   very hard problem?  No, it is not! | is 2 |

**样例2**

|  |  |
| --- | --- |
| 输入 | 输出 |
| I like   computer.   I like Computer Science.  I like School   of Computer Science.  I like   Sichuan University. | like 4 |

**提示**

如果用getchar()或者scanf()函数读入数据，那么当返回 EOF 时表示已经读取到文件末尾。

如果用gets()函数读入数据，那么返回NULL时表示已经读取到文件末尾。

## E: 合并有序数组

**描述**

合并两个有序数组a,b。

**输入**

第一行输入两个数据m,n，分别代表数组a,b的长度，第二行第三行分别输入两个数组a,b,保证数组有序。

**输出**

将两个数组合并后输出，保证数组仍然有序。输出数据之间用空格隔开。

**样例**

|  |  |
| --- | --- |
| 输入 | 输出 |
| 3 3  1 3 5  2 4 6 | 1 2 3 4 5 6 |