

北方工业大学

10、11 程序设计大赛试题



1. 求指定区间上的素数

Description

素数是上帝用来描写宇宙的文字（伽利略语）。

素数，又称为质数，是不能被 1 与它本身以外的其它整数整除的整数。如 2，3，5，7，11，13，17 是前几个质数，其中 2 为唯一的偶素数。

与此相对应，一个整数如果能被除 1 以外的整数整除，该整数称为合数或复合数。例如，15 能被除 1 与 15 以外的整数 3 和 5 整除，所以 15 是一个合数。

特别的，数 1 既不是素数，也不是合数。

作为一类特殊的整数，素数是数论中探讨最多也是难度最大的一类整数，其中有些问题是著名数学家提出并研究过的经典问题。

求素数的常用方法有试商判别法和筛选法两种。

现在要求你输出指定开区间 (a,b) 上的所有素数，并且统计这些素数的个数。

Input

输入包括两个数据：开区间 a 和 b 的值 $(a < 10000, b < 10000)$ 。

Output

输出包括两大部分，第一部分为该指定区间上所有的素数，相邻两个素数之间用一个空格分隔，然后换行；第二部分输出一个整数，表示该指定区间上素数的个数，输出之后不用换行。

如果该区间内没有素数，则输出只有一行，该行只有一个数字 0，输出之后不用换行。

Sample Input

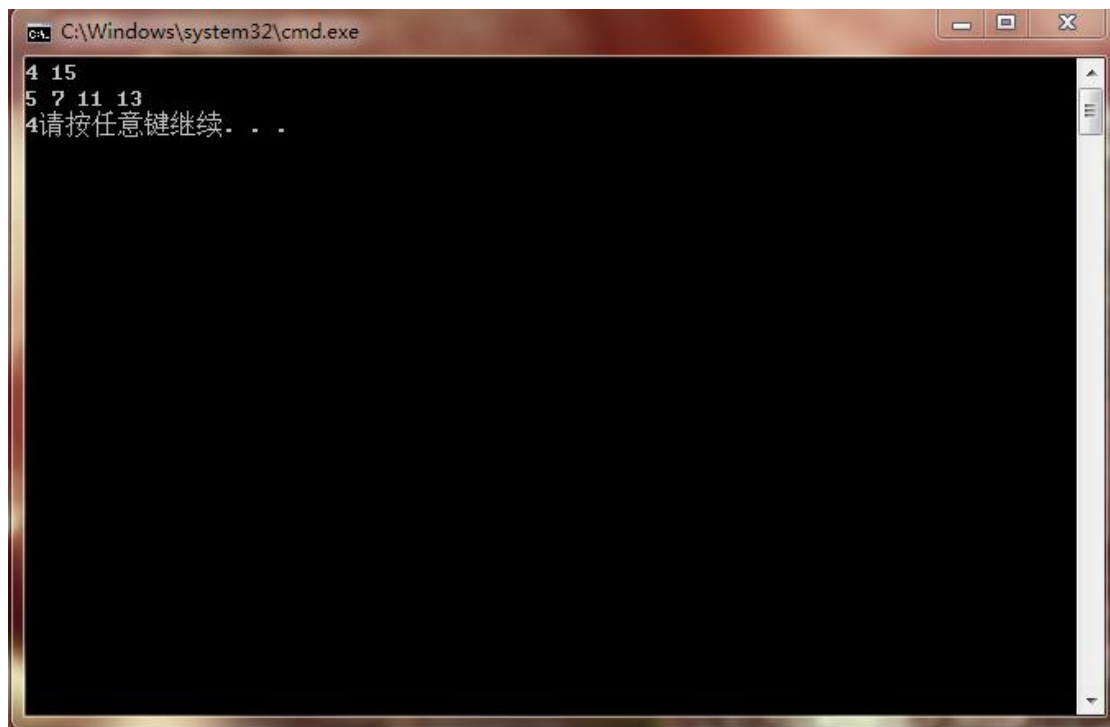
4 15

Sample Output

5 7 11 13

4

附实际测试截图：



2. 美丽的金字塔

Description

应用程序打印一些简单的图案是很有吸引力的课题之一，其中打印金字塔是基本的图案设计。金字塔作为古埃及文明的象征，具有非常独特的历史与文化意义。本题要求你打印 m 层金字塔图案。

Input

输入包括两部分：第一个部分是一个字符符号 ‘*’ 这三个字符中的一个，表示你要打印的金字塔是由这个基本符号组成的；第二部分是一个整数 m ，用来表示要求你打印的金字塔层数。

特别地，我们要求打印的金字塔都是奇数层，如果输入的是一个偶数，则打印出 $m+1$ 层金字塔的图形。

Output

输出用该符号组成的 m 层或 $m-1$ 层金字塔图案。

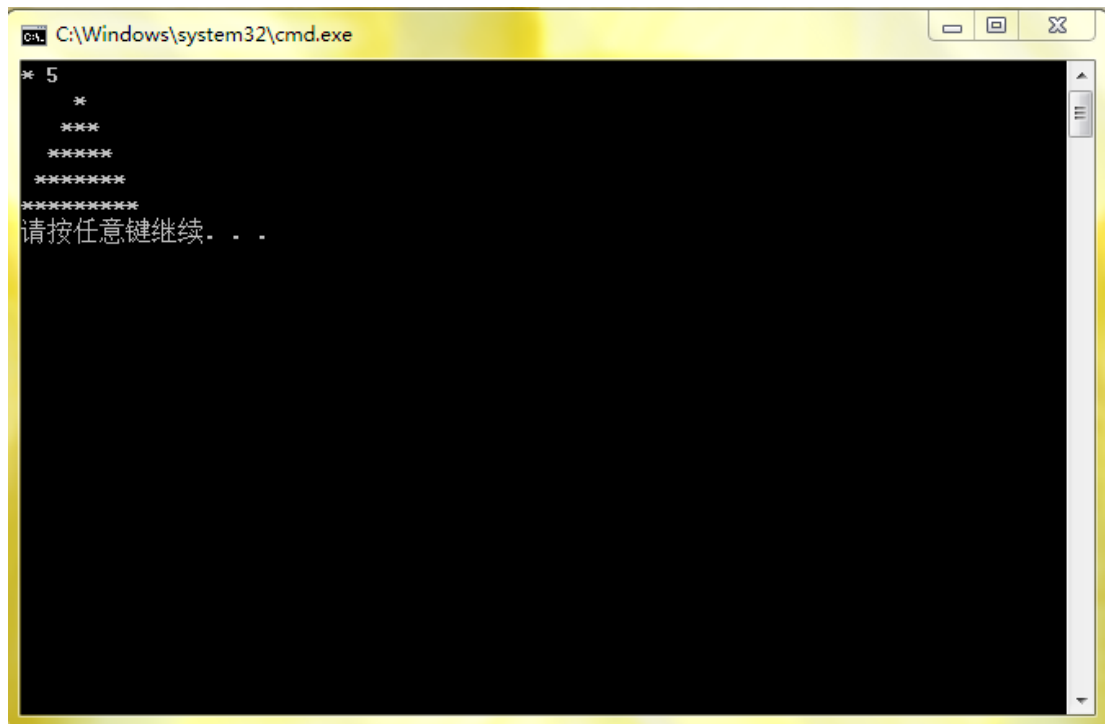
Sample Input

* 5

Sample Output

```
  *
 ***
*****
*****
*****
```

附实际测试截图：



3.可爱的甲虫

Description

有 n 只甲虫 ($0 < n < 10000$), 依次编号为 1,2,3, ..., n , 开始时头都朝东。第一秒钟, 编号为 1 的倍数的甲虫向右转 90 度; 第二秒钟, 编号为 2 的倍数的甲虫向右转 90 度; 第三秒钟, 编号为 3 的倍数的甲虫向右转 90 度, ..., 如此进行。我们的问题是, n 秒钟后, 第 n 号甲虫头朝哪个方向?

Input

输入 m 个数 ($0 < m < 10$), 表示有 m 个测试数据, 以下输入 m 个甲虫的编号 n_1 、 n_2 、...、 n_m 。

Output

输出共 m 行, 分别表示编号为 n_1 的甲虫在第 n_1 秒之后头的朝向, 编号为 n_2 的甲虫在第 n_2 秒之后头的朝向, ..., 编号为 n_m 的甲虫在第 n_m 秒之后头的朝向, 用 East、West、South、North 来表示。

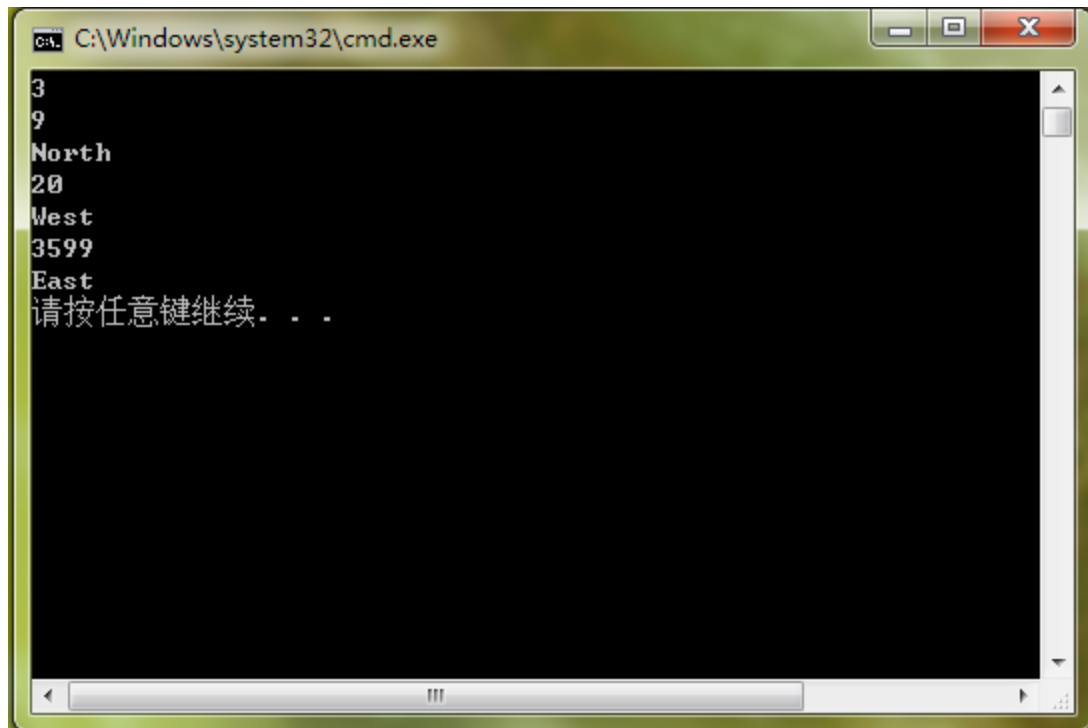
Sample Input

```
3
9 20 3599
```

Sample Output

```
North
West
East
```

附实际测试截图:



4、打印自方幂数

Description

一个 n 位正整数如果等于它的 n 个数字的 n 次方和，该数称为 n 位自方幂数，也称为 n 位 n 次幂回归数。

一位自方幂数又叫花数。

两位自方幂数又叫花朵数。

三位自方幂数又叫水仙花数。

四位自方幂数又叫玫瑰花数。

五位自方幂数又叫五角星数。

六位自方幂数又叫六合数。

叫法很多，实在不知道怎么称呼。此外还有广义的定义。

维基百科上是统称为水仙花数 Narcissistic number, PPDI, 阿姆斯特朗数 Armstrong_number。 (sequence A005188 in OEIS)

现在要求你打印出 1 到 $m-1$ 位的自方幂数。

Input

输入 m ($1 < m < 7$)，表示要打印的从 1 到 $m-1$ 个自方幂数。

Output

输出从 1 到 $m-1$ 个的所有自方幂数，如果不存在则显示如下引号中的信息：“does not exist!”，相邻的两个自方幂数用空格隔开，相邻两个不同位数的运行结果用换行符隔开（见附图）。

Sample Input

5

Sample Output

Armstrong number of 1:1

Armstrong number of 2:does not exist!

Armstrong number of 3:153 370 371 407

Armstrong number of 4:1634 8208 9474

附实际测试截图:

```
5
Armstrong number of 1:1
Armstrong number of 2:does not exist!
Armstrong number of 3:153 370 371 407
Armstrong number of 4:1634 8208 9474
Press any key to continue
```

5.简单的语音计算器

Description

Suzumiya 最近开始无端刁难她的同学 ViVo，总是莫名其妙的问他一些简单而又离谱、没有实际意义的数学问题，比如她问 ViVo: “What will get if we add 300 to 500?”. 回答一次两次还可以，但不断这样的纠缠致使 ViVo 已经无法忍受了。所以 ViVo 决定制作一个语音装置来自动回答 Suzumiya 提出的无聊问题。

ViVo 知道你是个伟大的算法艺术家，所以希望该装置核心的数学计算模块你能来帮忙。装置接收到的语音会自动为你转化为对应的英文字符串，经过你的模块处理完成后输出结果的数字信息，Suzumiya 虽然懒得计算，但她至少还是会读数的。

为了给你带来方便，ViVo 已经提前收集好了 Suzumiya 可能会问到的问题，发现这些问题更加地无聊，尤其令人厌烦的是 Suzumiya 每次提问都会以“what will we get if ...”的形式开头，这些问题都是数学中最基本的加减乘除问题，并且题目中可能包含的数字都是从 1 到 9 的一位整数，而且答案都可以用数字来表示。考虑到你可能还不会用英文来表达数学计算，ViVo 耐心细致地把问题分成了以下四类，分别对应加、减、乘、除四种运算，这四种运算的问题形式分别是：

“+”: What will we get if 1 add 2?

“-”: What will we get if 6 minus 3?

“*”: What will we get if 3 times12?

“/”: What will we get if 8 divided by 4?

Input

输入的第一行是一个数字 n ，表示接下来有 n 个问题，每个问题占一行。如果遇到一行运算非法（如除数为 0）则显示信息：输入非法！如果运算结果为负数，不用计算出结果，只输出引号中的信息：“The result is minus!”。

Output

每行包含一个没有语病的中文表示最终的结果。

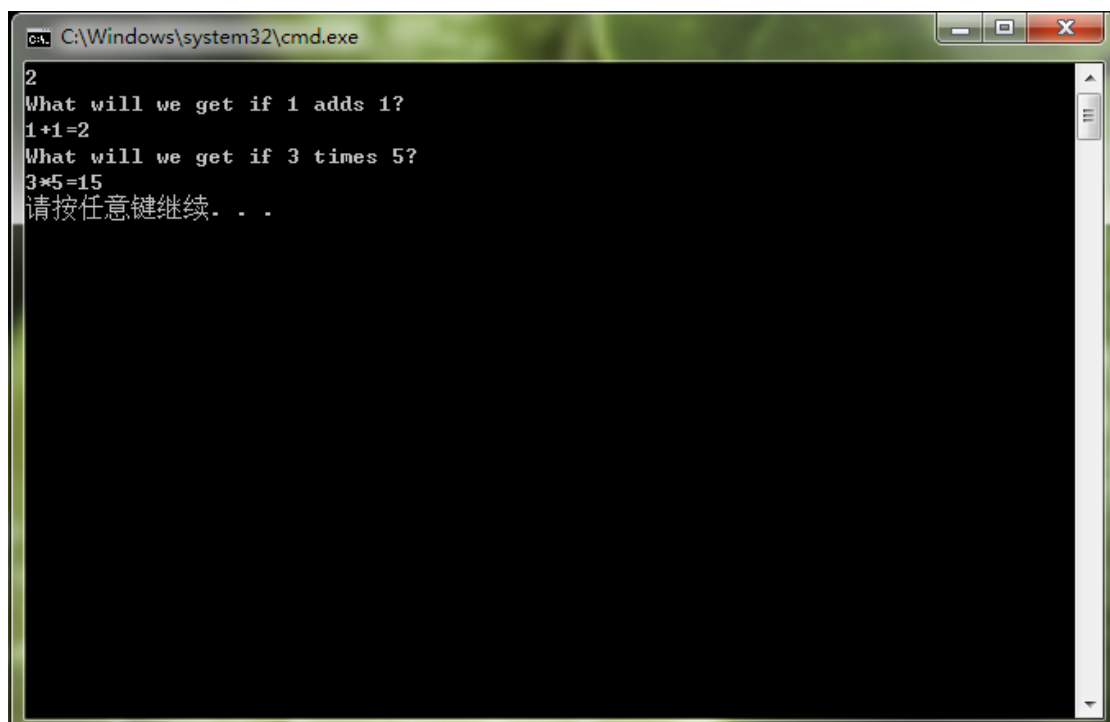
Sample Input

```
2
What will we get if 1 adds 1?
What will we get if 3 times 5?
```

Sample Output

```
1+1=2
3*5=15
```

附实际测试截图：



```
C:\Windows\system32\cmd.exe
2
What will we get if 1 adds 1?
1+1=2
What will we get if 3 times 5?
3*5=15
请按任意键继续. . .
```

6、角谷猜想

Description

考拉兹猜想，又称为 $3n+1$ 猜想、角谷猜想、哈塞猜想、乌拉姆猜想或叙拉古猜想，是由日本数学家角谷静夫发现，是指对于每一个正整数，如果它是奇数，则对它乘 3 再加 1，如果它是偶数，则对它除以 2，如此循环，最终都能够得到 1。取一个数字。

如 $n = 6$ ，根据上述公式，得出 $6 \rightarrow 3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 。(步骤中最大的数是 16，共有 8 个步骤)。

如 $n = 11$ ，根据上述公式，得出

$11 \rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 。(步骤中最大的数是 52，共有 14 个步骤)

如 $n = 27$ ，根据上述公式，得出：

$27 \rightarrow 82 \rightarrow 41 \rightarrow 124 \rightarrow 62 \rightarrow 31 \rightarrow 94 \rightarrow 47 \rightarrow 142 \rightarrow 71 \rightarrow 214 \rightarrow 107 \rightarrow 322 \rightarrow 161 \rightarrow 484 \rightarrow 242 \rightarrow 121 \rightarrow 364 \rightarrow 182 \rightarrow 91 \rightarrow 274 \rightarrow 137 \rightarrow 412 \rightarrow 206 \rightarrow 103 \rightarrow 310 \rightarrow 155 \rightarrow 466 \rightarrow 233 \rightarrow 700 \rightarrow 350 \rightarrow 175 \rightarrow 526 \rightarrow 263 \rightarrow 790 \rightarrow 395 \rightarrow 1186 \rightarrow 593 \rightarrow 1780 \rightarrow 890 \rightarrow 445 \rightarrow 1336 \rightarrow 668 \rightarrow 334 \rightarrow 167 \rightarrow 502 \rightarrow 251 \rightarrow 754 \rightarrow 377 \rightarrow 1132 \rightarrow 566 \rightarrow 283 \rightarrow 850 \rightarrow 425 \rightarrow 1276 \rightarrow 638 \rightarrow 319 \rightarrow 958 \rightarrow 479 \rightarrow 1438 \rightarrow 719 \rightarrow 2158 \rightarrow 1079 \rightarrow 3238 \rightarrow 1619 \rightarrow 4858 \rightarrow 2429 \rightarrow 7288 \rightarrow 3644 \rightarrow 1822 \rightarrow 911 \rightarrow 2734 \rightarrow 1367 \rightarrow 4102 \rightarrow 2051 \rightarrow 6154 \rightarrow 3077 \rightarrow 9232 \rightarrow 4616 \rightarrow 2308 \rightarrow 1154 \rightarrow 577 \rightarrow 1732 \rightarrow 866 \rightarrow 433 \rightarrow 1300 \rightarrow 650 \rightarrow 325 \rightarrow 976 \rightarrow 488 \rightarrow 244 \rightarrow 122 \rightarrow 61 \rightarrow 184 \rightarrow 92 \rightarrow 46 \rightarrow 23 \rightarrow 70 \rightarrow 35 \rightarrow 106 \rightarrow 53 \rightarrow 160 \rightarrow 80 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 。(步骤中最大的数是 9232，共有 111 个步骤)

考拉兹猜想称，任何正整数，经过上述计算步骤后，最终都会得到 1。

Input

输入仅包含一个需要验证角谷猜想的数 n ， $0 < n < 100$ 。

Output

从数字 n 开始，依次输出经过变换后得到的每一个数字，数与数之间用“->”连接，直到输出 1 为止步。统计步骤中最大的数字 m 以及所需的步骤数 x ，然后在下一行输出如下引号中的信息：“Max is m and we need x steps.”。

对于步骤数的规定：我们约定，经过一次运算为 1 步，比如，在下面的输入样例中，从 11 到 34 就是第 1 步，如此循环，直到最后数字变成 1 为止。

Sample Input

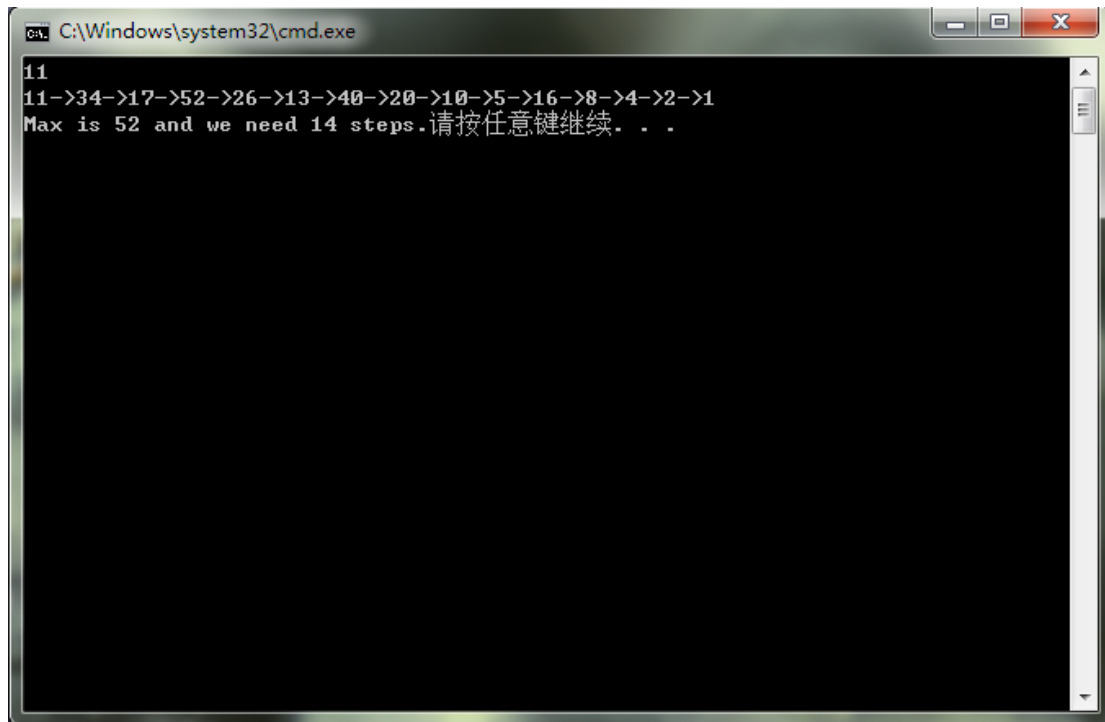
11

Sample Output

$11 \rightarrow 34 \rightarrow 17 \rightarrow 52 \rightarrow 26 \rightarrow 13 \rightarrow 40 \rightarrow 20 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

Max is 52 and we need 14 steps.

附实际测试截图：



```
C:\Windows\system32\cmd.exe
11
11->34->17->52->26->13->40->20->10->5->16->8->4->2->1
Max is 52 and we need 14 steps.请按任意键继续. . .
```

7、DNA Sorting

Description

One measure of "unsortedness" in a sequence is the number of pairs of entries that are out of order with respect to each other. For instance, in the letter sequence "DAABEC", this measure is 5, since D is greater than four letters to its right and E is greater than one letter to its right. This measure is called the number of inversions in the sequence. The sequence "AACEDGG" has only one inversion (E and D)---it is nearly sorted---while the sequence "ZWQM" has 6 inversions (it is as unsorted as can be---exactly the reverse of sorted).

You are responsible for cataloguing a sequence of DNA strings (sequences containing only the four letters A, C, G, and T). However, you want to catalog them, not in alphabetical order, but rather in order of "sortedness", from "most sorted" to "least sorted". All the strings are of the same length.

Input

The first line contains two integers: a positive integer n ($0 < n \leq 50$) giving the length of the strings; and a positive integer m ($0 < m \leq 100$) giving the number of strings. These are followed by m lines, each containing a string of length n .

Output

Output the list of input strings, arranged from "most sorted" to "least sorted". Since two strings can be equally sorted, then output them according to the original order.

Sample Input

```
10 6
AACATGAAGG
```

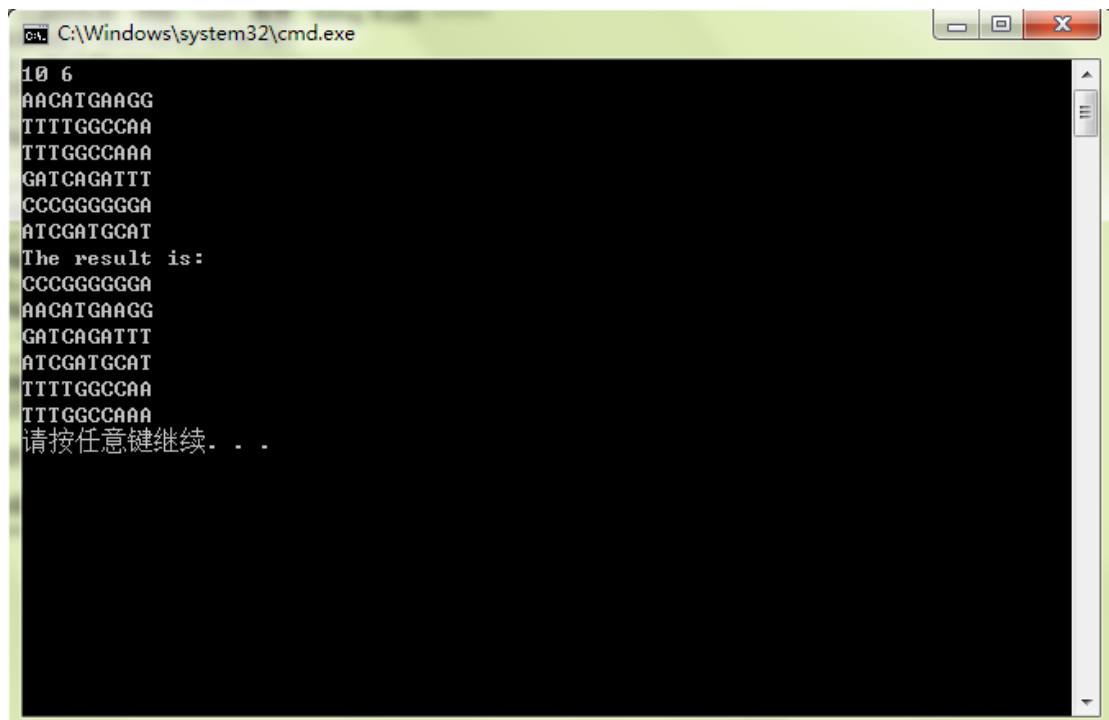

TTTTGGCCAA
TTTGGCCAAA
GATCAGATTT
CCCGGGGGGA
ATCGATGCAT

Sample Output

The result is:

CCCGGGGGGA
AACATGAAGG
GATCAGATTT
ATCGATGCAT
TTTTGGCCAA
TTTGGCCAAA

附实际测试截图:



```
C:\Windows\system32\cmd.exe
10 6
AACATGAAGG
TTTTGGCCAA
TTTGGCCAAA
GATCAGATTT
CCCGGGGGGA
ATCGATGCAT
The result is:
CCCGGGGGGA
AACATGAAGG
GATCAGATTT
ATCGATGCAT
TTTTGGCCAA
TTTGGCCAAA
请按任意键继续. . .
```