

# 1001 小 Q 系列故事——世界上最遥远的距离

Time Limit: 0.2 Seconds    Memory Limit: 65536K

世界上最遥远的距离  
不是生与死  
而是我就站在你面前  
你却不知道我爱你

世界上最遥远的距离  
不是我就站在你面前你却不知道我爱你  
而是明明知道彼此相爱  
却不能在一起

世界上最遥远的距离  
不是明明知道彼此相爱却不能在一起  
而是相约好了私奔的时间  
我穿越到了未来 你却回去了古代

——摘自《小 Q 失恋日记》第 117 卷 513 页

当小 Q 使出浑身解数，终于赢得 HR 女神芳心的时候，却出现了一个意外情况，那就是白富美 HR 的妈妈并不同意他们交往，当听说小 Q 只是一个码农，特别是听说小 Q 曾经参加过资本主义国家发起的 SM/ICPC 比赛的时候，更是坚决反对！

爱情是伟大的，但是得不到亲人祝福的爱情却备受折磨，小 Q 和 HR 相约在腾讯第二届编程马拉松大赛进行到第 5 天的时候（即 2013 年 3 月 24 日），一起“向前穿越 D 天，然后开启幸福新生活”。

其勇气可谓令人赞叹，但可怜的小 Q 却总是备受折磨——小 Q 理解的“向前穿越”是朝着未来的方向，而女友 HR 理解的“向前穿越”却是朝着古代的方向！

假设已知现在的日期和穿越的天数 D，你能计算出小 Q 和女友各自到达的年代吗？

## Input

输入首先包含一个整数 N，表示有 N 组测试用例；

接下来 N 行是 N 组数据，每一行包含一个正整数 D ( $D \leq 10,000$ )，D 表示向前穿越的天数。

## Output

请计算并输出小 Q 和女友分别到达的日期，日期格式为 YYYY/MM/DD，两个日期中间用一个空格隔开，每组数据占一行，具体输出格式请参见样例。

### **Sample Input**

2  
6  
30

### **Sample Output**

2013/03/30 2013/03/18  
2013/04/23 2013/02/22

## 1002 威威猫系列故事——因式分解

Time Limit: 0.2 Seconds    Memory Limit: 65536K

“春天来了，万物复苏，大地一片生机盎然，又到了动物们求偶的季节...”

周末的威威猫虽然眼睛盯着电视屏幕，但是思绪却停留在自己喜欢的数学问题上。

有人说孤独是可耻的，但是单身一人的威威猫并不孤独，随着对数学的深入学习，威威猫甚至很庆幸自己没有陷入儿女情长，因为他有喜爱的数学相伴，最近，他就在潜心研究因式分解问题。

在我们学习数学的过程中，经常需要把一个多项式进行因式分解，也就是把它写成乘积的形式，比如多项式  $x^2+3x+2$  分解因式的结果就是  $(x+1)(x+2)$ 。这个因式一眼就能看出来，但是当  $x$  的指数更高时，就不太容易分解了。

现在，威威猫就是在研究如何编写程序来实现对多项式的因式分解。

### Input

输入第一行是一个整数  $T$  ( $T \leq 50$ )，表示测试数据的组数；

接下来是  $T$  行字符串表示  $T$  个测试用例，每行 1 个数学多项式，多项式长度不会超过 100 个字符，每个多项式表示形式如下：

$$A[1]x^{P[1]} + A[2]x^{P[2]} + \dots + A[m]x^{P[m]}$$

其中  $0 \leq P[i] \leq 5$ ， $A[i]$  表示  $x^{P[i]}$  的系数， $A[i]=0$  时直接简写为 0， $A[i]=1$  和  $-1$  时分别简写为  $x^{P[i]}$  与  $-x^{P[i]}$ ， $P[i]=0$  和  $1$  时分别简写为  $A[i]$  与  $A[i]x$ ，且同一指数  $r$  的对应项系数之和的绝对值不超过 1000，每行中没有多余空格，具体格式可参考 Sample Input。

### Output

对于每组测试数据，首先输出 Case #X:， $X$  代表多项式编号，从 1 开始计数，然后输出因式分解的结果，分解结果的表示形式规定如下：

$$(x+B[1])(x+B[2])\dots(x+B[m])$$

其中， $B[1] \leq B[2] \leq \dots \leq B[m]$ ，若  $B[i]=0$  则不加括号直接简写为  $x$ ，如果无法表现成上述格式，则输出  $-1$ 。

具体可参考 Sample Output。

### Sample Input

```
4
x
x+1
-2x^2+x^2+x^3
2x+2
```

### Sample Output

```
Case #1: x
Case #2: (x+1)
Case #3: (x-1)xx
Case #4: -1
```

## 1003 小小明系列故事——游戏的烦恼

Time Limit: 1.0 Seconds    Memory Limit: 32768K

小小明最近在玩一款游戏，它由  $n*m$  大小的矩阵构成，矩阵上会随机产生一些黑色的点，这些点它们可能会连在一起也可能会分开，这些点的个数没有限制，但是每个  $1*1$  方格中最多只可能有一个黑点产生。游戏要求玩家以最短的时间用  $x*y$  的小矩阵覆盖这个大矩阵，覆盖的要求有以下 2 点：

1.  $x*y$  大小的小矩阵内必须有  $x*y$  个黑点。
2. 多个小矩阵可以重叠，但是每个小矩阵放置的位置必须是独一无二的，即不同的小矩阵内的黑点不能完全相同。例如  $1*2$  的矩阵可以横着放，也可以竖着放，这两种方法是不同的，即使它们可能共用黑点。

小小明是个粗心的孩子，他尝试了很多遍都无法将所有的符合要求的小矩阵找到，聪明的你，能不能告诉烦恼中的小小明这个大矩阵里有多少个满足要求的小矩阵呢？

### Input

题目有多组测试数据（不多于 100 个）；

每组测试数据的第一行包含 2 个正整数  $n$  和  $m$ ，然后第二行是  $x$  和  $y$ （ $n, m, x, y$  的意思如题），接下来  $n$  行，每行  $m$  个字符，其中 '\*' 表示黑点，'.' 表示空白。

$n$  和  $m$  为 0 则结束输入。

### [Technical Specification]

$0 < n, m \leq 2000$

$0 < x, y \leq 1000$

### Output

请计算并输出一共有多少个满足要求的小矩阵，每组输出占一行。

### Sample Input

```
2 3
1 2
**.
.**
0 0
```

### Sample output

```
3
```

## 1004 吉哥系列故事——最终数

---

Time Limit: 0.2 Seconds    Memory Limit: 65536K

---

在 2012 年腾讯编程马拉松比赛中，吉哥解决了一道关于斐波那契的题目，这让他非常高兴，也更加燃起了它对数学特别是斐波那契数的热爱。现在，它又在思考一个关于斐波那契的问题：

假如我们现在已知斐波那契数是 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89...

由于吉哥特别喜欢斐波那契数，它希望每个数中都包含斐波那契数，比如说 130，其中包含了 13，或者 5534 包含了 55 和 34，只要数位中含有至少一个斐波那契数就是吉哥想要的数。但是这种数实在太多了，于是它去掉那些仅仅含有小于 10 的斐波那契数的数，比如说 1，仅仅含有 1，所以被去掉；或者 335 只含有 3 和 5，都是小于 10 的斐波那契数，所以也去掉；但是 131 是留下的，因为它含有 13，我们暂且称这类数为 F 数，不难得到前几个 F 数是 13, 21, 34, 55, 89, 113, 121, 130, 131...

霸气的吉哥觉得这样还不够，它想将斐波那契进行到底——在前面 F 数的基础上，吉哥要得到那些是第斐波那契数个的 F 数！就是说，我们假设 F 数从 1 开始标号，那么 13 是第 1 个 F 数，吉哥想要那些在 F 数中的排列或者说下标也要是斐波那契数的数，吉哥称之为最终数，如 13, 21, 34, 89, 130...

现在给你一个数 n，吉哥想知道离 n 最近的最终数与 n 的差的绝对值是多少。

### Input

输入包含多组测试数据。

每组测试数据包含一个整数 n ( $1 \leq n \leq 10^{11}$ )；

若 n = -1，表示输入结束。

### Output

对于每个 n，请输出离 n 最近的最终数与 n 的差的绝对值。

### Sample Input

```
1
100
-1
```

### Sample Output

```
12
11
```

# 1005 郑厂长系列故事——体检

Time Limit: 1.0 Seconds    Memory Limit: 32768K

郑厂长不是正厂长  
也不是副厂长  
他根本就不是厂长  
只是公司的一个码农

郑厂长所在的腾讯公司每一年都要组织员工体检，比如量身高体重、测血压之类的，今年也不例外。

这次总共有  $N$  位员工接受体检，并且每个员工都需要做  $K$  个项目的检查才算完成整个体检的流程。现在来了  $M$  个医生为员工做身体检查，并且每一位医生都带齐了检查这  $K$  个项目的器材来(也就是说每个医生都能进行这  $K$  个项目中的任意一项检查)。

体检的详细流程是这样的：

公司事先制定好了  $M$  份体检单，每个医生手上都各自拿到一份体检单，上面已经安排好了检查的次序，以及每一次检查所对应的员工和项目。每个医生按照体检单上的次序为相应的员工做相应的项目检查。医生拿到的体检单上的名单也可以是空的，就是这个医生不需要检查任何员工的任何项目。

当然，制定出的这  $M$  份体检单不能有问题存在，否则就会有混乱的情况发生。按照常理来说，同一个医生在同一时间只能为一个员工做一个项目的检查。另外，同一个员工在同一时间也只能进行一个项目的检查，当然，不同的医生或不同的员工可以在同一时间进行项目检查。现在假设每个员工的每个项目的检查时间都是一分钟（其它时间花费忽略不计，只考虑项目检查工作所花费的一分钟）。

公司希望体检的工作越快完成越好，由于郑厂长大学期间曾经是一个 ACMer，所以公司就将体检的安排工作交给了他，他需要计算出最快需要多少分钟能完成所有员工的体检工作。

## Input

输入的第一行为一个正整数  $T$ ，表示有  $T$  组测试数据；

接下去有  $T$  组测试数据，每组测试数据占一行，包含三个整数  $N, K, M$ ， $N$  表示员工的人数， $K$  表示体检的项目数， $M$  表示医生的人数。

### [Technical Specification]

$T \leq 1000$

$1 \leq N \leq 100$

$1 \leq K \leq 10$

$1 \leq M \leq 100$

## Output

对于每组数据，输出一个整数，表示最快需要多少分钟才能完成所有员工的体检工作。

### Sample Input

```
2
2 1 1
3 2 2
```

### Sample Output

```
2
3
```

### Hint:

对于第二组数据体检单的安排可以是如下情况：

第 1 个医生的体检单：员工 A 的项目 1、员工 A 的项目 2、员工 B 的项目 2；

第 2 个医生的体检单：员工 B 的项目 1、员工 C 的项目 1、员工 C 的项目 2。

第一分钟：第 1 个医生检查员工 A 的项目 1，而第 2 个医生检查员工 B 的项目 1；

第二分钟：第 1 个医生检查员工 A 的项目 2，而第 2 个医生检查员工 C 的项目 1；

第三分钟：第 1 个医生检查员工 B 的项目 2，而第 2 个医生检查员工 C 的项目 2；

这样就只需要 3 分钟即可完成体检工作。