

# 替罪羊

Aori是个很粗心的人，所以她总是会遇到麻烦。有一天，她遇到了  $N$  个大麻烦！但这一次，她似乎很放心，因为她找到了  $M$  个 Inkling 来承担所有的责任。每个麻烦都可以用数字  $a_i$  来表示它的严重程度。每一个麻烦都需要至少一个 Inkling 来承担，而且每个 Inkling 只能帮Aori承担一个麻烦。如果两个或两个以上的 Inkling 承担同一个麻烦，他们会将这个麻烦分解为一些小麻烦，再分配给每个Inkling 来减少每个 Inkling 的压力。分解方法只要满足所有 Inkling 承担的严重程度之和等于此麻烦的严重程度即可。

Inkling 都是很好的人，Aori 想寻找出一个最小化不同 Inkling 承担的严重程度的方差。你能替Aori将  $N$  个麻烦分给  $M$  个 Inkling 吗？

形式上，变量方差是变量与其均值的差的平方的期望：

$$\sigma^2 = \frac{1}{n} \sum \left( x_i - \frac{1}{n} \sum x_j \right)^2$$

## 输入

输入的第一行给出了测试用例数  $T$ 。

对于每个测试用例，第一行包含两个整数  $N$  和  $M$ ，其中  $N$  是麻烦的数量， $M$  是 Inkling 的数量。

下一行包含  $N$  整数  $a_1, a_2, \dots, a_N$ ，表示各麻烦的严重性。

- $1 \leq T \leq 100$ .
- $1 \leq N \leq M \leq 2 \times 10^5$ .
- $1 \leq a_i \leq 10000$ .
- $\sum M \leq 3 \times 10^6$ .

## 输出

对于每个测试用例，输出一行包含 `Case #x: y`，其中 `x` 是测试用例编号（从 1 开始），`y` 是最小方差。

如果  $y$  在正确答案的绝对或相对误差  $10^{-8}$  范围内，则它将被认为是正确的。

## 样例

输入

```
3
3 6
1 2 3
5 10
5 6 7 8 9
6 6
1 1 4 5 1 4
```

## 输出

```
Case #1: 0.000000000000
Case #2: 0.400000000000
Case #3: 2.888888888889
```

对于第一个样例，Aori 可以让一个 Inkling 承担第一个麻烦，让两个 Inkling 承担第二个麻烦，让三个 Inkling 承担第三个麻烦。每个 Inkling 承担的严重程度是1，因此它们的方差是0。

# 熊猫先生和生日歌

熊猫夫人的生日快到了。熊猫先生想创作一首歌作为生日礼物。

众所周知，熊猫夫人不喜欢一首歌的前提是它的歌词中连续包含  $X$  个元音，或连续包含  $Y$  个辅音（否则，熊猫夫人喜欢这首歌）。字母 'a'、'e'、'i'、'o'、'u' 是元音，其他字母都是辅音。

虽然潘达先生是个有天赋的歌手，但他不擅长作词。熊猫先生希望这首歌很特别。因此，他在谷歌上搜索了这首歌的歌词模板。

该模板由小写英文字母和一些问号组成。例如，“happybirthday”（不带引号，下同）、“???ybirthday”都是有效的模板，但“生日快乐”和“HappyBirthday”均不是。熊猫先生需要用小写英文字母替换所有问号，使其成为歌曲的实际歌词。

熊猫先生想给熊猫夫人一个惊喜。因此，熊猫先生用同一个模版，不仅创作出熊猫夫人喜欢一首歌，还希望创作出熊猫女士不喜欢的一首歌。因为熊猫先生真的不擅长作词，即使是用模板，这项任务也困扰了他好几天。幸运的是，熊猫先生知道你很有天赋，因此他想向你寻求帮助。

对于给定的模板，如果熊猫夫人不喜欢模板中生成的所有歌曲，则输出“DISLIKE”（不带引号，下同）（这意味着你不能用字母代替问号，这样熊猫夫人就喜欢这首歌）；如果熊猫女士喜欢模板中生成的所有歌曲则输出“LIKE”；如果这个模版既能创作出熊猫夫人喜欢的歌又能创作出熊猫夫人不喜欢的歌，则输出“SURPRISE”。

## 输入

输入的第一行给出了测试用例的数量  $T$ ，接下来是  $T$  组测试用例。

每个测试用例由一行组成，其中包含一个字符串  $S$ ，代表熊猫先生得到的模板，一个整数  $X$ ，表示熊猫夫人不喜欢的连续元音的最小数量，以及一个整数  $Y$ ，表示熊猫女士不喜欢的连续辅音的最小数量。在模板  $S$  中只包含小写英文字母（'a'到'z'）和问号（'?'）。

- $1 \leq T \leq 300$ .
- $2 \leq |S| \leq 10^6$ .
- $1 \leq X \leq |S|$ .
- $1 \leq Y \leq |S|$ .

## 输出

对于每个测试用例，输出一行包含 `Case #x: y`，其中 `x` 是测试用例编号（从 1 开始），`y` 可以是问题描述中提到的“LIKE”、“DISLIKE”或“SURPRISE”。

## 样例

输入

```
5
happybirthda? 3 4
happybirth?ay 3 5
happybirthd?y 3 5
hellow?rld 3 5
hellllllowoorld 3 5
```

输出

```
Case #1: DISLIKE
Case #2: LIKE
Case #3: SURPRISE
Case #4: SURPRISE
Case #5: DISLIKE
```

## 图像识别

有  $N$  个二进制图像，编号从 0 到  $N - 1$ 。每个图像包含  $M$  个像素，编号从 0 到  $M - 1$ 。每个像素要么是黑色，要么是白色。图像之间两两不同（至少有一个像素不同）。

现在有一些查询。每个查询包含这些图像的一个子集。对于每个查询，希望找到一组可以区分其中每对图像的像素集。如果两个图像在至少一个给定像素集中的像素上中具有不同的像素值，则可以通过给定的像素集来区分它们。选择的像素数应小于每个查询中的图像数。

## 输入

输入的第一行给出了测试用例的数量  $T$ ，接下来是  $T$  组测试用例。

每个测试用例的第一行由三个整数  $N$ 、 $M$  和  $Q$  组成，表示图像的数量、每个图像中的像素数量和查询数量。下面的  $N$  行描述了每个图像的内容。每一行包含一个长度为  $M$  的 01 字符串，表示图像的内容。下面的  $Q$  行描述了这些查询。每个查询以指示该查询的图像数量的整数  $K_i$  开始，然后是  $K_i$  个图像的编号列表。

- $1 \leq T \leq 100$ .
- $1 \leq N \leq 2000$ .
- $1 \leq M \leq 4000$ .
- $1 \leq Q \leq 2000$ .

- $1 \leq K_i \leq N$ .
- 保证所有测试用例中  $N$  的和不超过  $2 \times 10^4$ .
- 保证所有测试用例中  $K_i$  的和不超过  $5 \times 10^6$ .

## 输出

对于每个测试用例，输出一行包含 `Case #x: y`，其中 `x` 是测试用例编号（从 1 开始），然后跟随  $Q$  行。每一行都以一个整数开始，该整数表示区分查询中的图像所需的像素数，然后是一个以空格分隔的像素的编号列表。每个查询可能有多个答案。只要选择的像素数小于  $K_i$ ，输出其中任何一个即可。

## 样例

输入

```
1
3 3 4
101
110
001
2 0 1
2 0 2
2 1 2
3 0 1 2
```

输出

```
Case #1:
1 1
1 0
1 0
2 0 1
```

## 熊猫先生与几何序列

熊猫先生喜欢玩数字游戏。一天，他拿起号码 3612，发现它很有趣。3612 可以分为 3 个数字 —— 3，6 和 12，这形成了一个整数几何序列。整数几何序列由至少 **3** 个正整数  $a_0, a_1, \dots, a_{n-1}$  构成的序列，还有一个正数  $D$  ( $D > 1$ )，对于每个  $i(0 \leq i < n - 1), a_i \times D = a_{i+1}$ 。

熊猫先生把这种数字命名为“快乐数字”。他还宣布，前导零是被禁止的，这意味着数字之前不应该有多余的零。现在熊猫先生想知道有多少快乐数字在  $L$  和  $R$  之间（含  $L$  和  $R$ ）。

## 输入

输入的第一行给出了测试用例的数量  $T$ ，接下来是  $T$  组测试用例。每个测试用例包含一行，由两个数字  $L$  和  $R$  组成。

- $1 \leq T \leq 2 \times 10^5$ .  
 $0 \leq L \leq R \leq 10^{15}$ .

## 输出

对于每个测试用例，输出一行包含 `Case #x: y`，其中 `x` 是测试用例编号（从 1 开始），`y` 是介于  $L$  和  $R$  之间的快乐数（包括  $L$  和  $R$ ）。

## 样例

输入

```
4
123 124
468 470
248 248
124816 124832
```

输出

```
Case #1: 1
Case #2: 1
Case #3: 1
Case #4: 1
```

在第一个测试用例中，123 和 124 之间唯一的快乐数是 124，其中  $1 \times 2 = 2$  和  $2 \times 2 = 4$ 。  
在第二个测试用例中，唯一的快乐数字是 248。  
在第三个测试案例中，唯一的快乐数字是 469，正数 D 是  $\frac{3}{2}$ 。  
在第四个测试案例中，124816 和 124832 之间唯一的快乐数是 124816，其中  $1 \times 2 = 2, 2 \times 2 = 4, 4 \times 2 = 8, 8 \times 2 = 16$ 。