

Problem A. Erase Nodes

Input file: standard input
Output file: standard output
Time limit: 6 seconds
Memory limit: 256 megabytes

小 Q 有一个包含 n 个点和 n 条无向边的连通图，但是现在他决定毁掉这张图。

在摧毁过程中，他需要不断地选择一个还存在的点 u ，然后访问每个与 u 连通的点 v （包括 u 本身），更新 v 点所存储的连通信息。他已经意识到，如果他以某种不好的顺序删除这些点，那么摧毁过程中总的访问次数会大大增加。为了避免这样的事情发生，每次选择时他都会从还存在的点里随机选出一个点 u 进行摧毁，这里每个当前还存在的点被选中的概率均等。

请你帮他计算一下他所需要的期望访问次数。这里为了避免可能的精度问题，若令答案为 $\frac{p}{q}$ ，则你只需要给出最小非负的整数 r 使得 $qr \equiv p \pmod{998244353}$ 。例如 $6 \times 166374072 \equiv 79, 3 \times 332748131 \equiv 40 \pmod{998244353}$ 。

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

第一行包含一个整数 n 。

接下来 n 行中，每行包含两个整数 u 和 v ，表示有一条连接第 u 个点和第 v 个点的边。

- $1 \leq T \leq 100$
- $3 \leq n \leq 10^5$
- $1 \leq u < v \leq n$
- 所有测试数据的 n 之和不超过 5×10^5 。
- 对于每组测试数据，保证边互不相同，也即不存在重边。

Output

对于每组测试数据，输出一行 “Case #x: y”（不含引号），其中 x 是测试数据的编号（从 1 开始编号）， y 是这组数据的答案。

Example

standard input	standard output
2	Case #1: 166374072
5	Case #2: 332748131
1 2	
1 3	
1 4	
2 4	
2 5	
5	
1 2	
1 3	
1 4	
1 5	
2 5	

Problem B. Erase Numbers III

Input file: standard input

Output file: standard output

Time limit: 3 seconds

Memory limit: 256 megabytes

给定 n 个正整数 a_1, a_2, \dots, a_n ，它们组成了序列 A 。

你的任务是，对于 $k = 1, 2, \dots, n-1$ ，删除其中 k 个数字，使得 A 中剩余的数字连起来的所表示的数字最大，并给出所有 k 对应的这个值之和。

严格来讲，如果剩余数字在 A 中的下标为 p_1, p_2, \dots, p_m ($1 \leq p_1 < p_2 < \dots < p_m \leq n$)，则它们连起来所表示的数字为 $\overline{a_{p_1}a_{p_2}\dots a_{p_m}}$ 。

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

第一行包含一个整数 n 。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ，保证每个数字不含前导零。

- $1 \leq T \leq 3000$
- $2 \leq n \leq 6000$
- $1 \leq a_i \leq 10^9$
- 所有测试数据的 n 之和不超过 6000。

Output

对于每组测试数据，输出一行 “Case #x: y”（不含引号），其中 x 是测试数据的编号（从 1 开始编号）， y 是这组数据的答案。

Example

standard input	standard output
3	Case #1: 72
3	Case #2: 213363
6 6 6	Case #3: 999353971308526
4	
21 12 12 21	
6	
998 244 353 985 661 441	

Problem C. Fibonacci Strikes Back

Input file: standard input
 Output file: standard output
 Time limit: 4 seconds
 Memory limit: 256 megabytes

在本题中，你需要解决的是关于 P -斐波那契序列的一个经典问题：

$$F_n = \begin{cases} 0 & \text{if } n = 0 \\ 1 & \text{if } n = 1 \\ PF_{n-1} + F_{n-2} & \text{otherwise} \end{cases}$$

现在，给定 P 和 m 以及 F_{F_n} 的十进制表示最低 k 位，你需要找出最小可能的 n 使得 $n \geq m$ 且 F_{F_n} 的十进制表示最低 k 位存在并如上所示，或者确定这样的解是不存在的。

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

仅一行，包含两个整数 P, m 和一个长度为 k 的数字串，这个串是 F_{F_n} 的十进制表示最低 k 位。

- $1 \leq T \leq 10^4$
- $1 \leq P, m \leq 10^{18}$
- $1 \leq k \leq 18$
- 所有测试数据的 k 之和不超过 10^4 。
- 保证 P 和 10^{18} 的最大公约数小于 5。

Output

对于每组测试数据，输出一行 “Case #x: y”（不含引号），其中 x 是测试数据的编号（从 1 开始编号），而在答案存在时 y 是最小可能的 n ，在答案不存在时 y 是 -1 。

Example

standard input	standard output
7	Case #1: 3
1 3 1	Case #2: 6
1 4 1	Case #3: 101
1 6 01	Case #4: 10
1 1 45	Case #5: 5
2 1 0482149	Case #6: -1
998 244 353	Case #7: 233
998244 1 353	

Note

当 $P = 1$ 时 $\{F_{F_n}\}_{n=0}^{\infty} = \{0, 1, 1, 1, 2, 5, 21, 233, 10946, 5702887, 139583862445, \dots\}$ 。

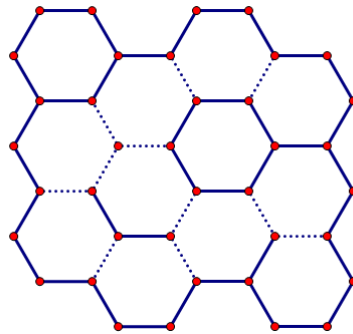
当 $P = 2$ 时 $\{F_{F_n}\}_{n=0}^{\infty} = \{0, 1, 2, 29, 13860, 44560482149, \dots\}$ 。

Problem D. Honeycomb

Input file: **standard input**
 Output file: **standard output**
 Time limit: 4 seconds
 Memory limit: 256 megabytes

蜂窝是由一系列六边形棱柱格子组成，其中每个格子有六个相邻的格子，而每两个相邻的格子共享一条边界。这里有些边界是可穿行的，但有些则不是。

小汉是一个勤劳的工蜂，他每天都在为打通边界或是切断两个相邻格子的联系辛苦工作着。几天后，他将有机会任意改动一片蜂窝的格局。这片蜂窝与外界不连通，包含如下图所示的 n 行 m 列格子。



为了切断两个格子的联系，他可能需要将某些边界设为不可穿行。他很好奇这至少要改动多少条边才能实现，你能帮他计算出，对于任意两个特别的格子，他最少需要改动的边数吗？为了避免输出过大，你只需要给出这些最小边数的之和即可。

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

第一行包含两个整数 n 和 m 。

接下来 $(4n + 3)$ 行描述了这片蜂窝，其中每一行至多有 $(6m + 3)$ 个字符。奇数行包含用加号（“+”）表示的顶点和水平方向的边界，而偶数行包含对角线方向的边界。

具体来说，每个格子由 6 个顶点和 6 条边界组成，每条边所示的字符会恰好位于对应端点之间，其中可穿行的边用空格（“ ”）表示，不可穿行的边遵循如下规则：

- 每个格子的上边界或下边界由三个连续的减号（“-”）或三个连续的空格表示，取决于它是否可以被穿行；
- 每个格子的每条不可穿行的对角线边界由一个正斜线（“/”）或一个反斜线（“\”）字符表示，取决于它的方向。

此外，在每个特别的格子中心处有一个星号（“*”）字符。其他字符将会是空白符，且输入每一行不会包含行末空格。

- $1 \leq T \leq 100$
- $2 \leq n, m \leq 100$
- 所有测试数据的特别格子的数量之和不超过 3000。
- 保证每个不特别的格子没有可以穿行的边界。

Output

对于每组测试数据，输出一行 “Case #x: y”（不含引号），其中 x 是测试数据的编号（从 1 开始编号），y 是这组数据的答案。

Example

standard input	standard output
<pre> 2 2 2 +---+ / \ + * +---+ \ / \ +---+ * + / / + * + + \ \ +---+ * + \ / +---+ 2 3 +---+ +---+ / \ / \ + * +---+ + \ \ / +---+ * +---+ / \ + * +---+ * + \ / +---+ * +---+ \ / +---+ </pre>	<pre> Case #1: 6 Case #2: 16 </pre>

Problem E. Power of Function

Input file: standard input
 Output file: standard output
 Time limit: 4 seconds
 Memory limit: 256 megabytes

小 Q 有一个函数

$$f(n) = \begin{cases} \frac{n}{k} & \text{if } n \bmod k = 0 \\ n - 1 & \text{otherwise} \end{cases},$$

其中这个函数的定义域是全部非负整数。

定义这个函数的 m 次幂为 $f^m(n)$, 满足

$$f^m(n) = \begin{cases} f^{m-1}(f(n)) & \text{if } m > 0 \\ n & \text{otherwise} \end{cases}.$$

他想知道最大可能的整数 m 使得存在至少一个整数 n 满足 $l \leq n \leq r$ 且 $f^m(n) = 1$ 。此外, 请在确定 m 后帮他找到 n 可能的最小值和最大值, 以便他验证你的结果是正确的。

Input

输入包含多组测试数据。第一行包含一个整数 T , 表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据:

仅一行, 包含三个整数 k, l, r 。

- $1 \leq T \leq 3 \times 10^5$
- $2 \leq k \leq 10^{18}$
- $1 \leq l \leq r \leq 10^{18}$
- 对于每组数据, 保证解一定存在。

Output

对于每组测试数据, 输出一行 “Case #x: m a b” (不含引号), 其中 x 是测试数据的编号 (从 1 开始编号), m 是最大可能的指数, 且对于这个 m 来说, a 是最小可能的自变量, 而 b 是最大可能的自变量。

Example

standard input	standard output
5	Case #1: 0 1 1
2 1 1	Case #2: 1 2 2
2 1 2	Case #3: 2 3 4
2 1 4	Case #4: 35 998244353 998244354
2 998244353 998244354	Case #5: 55 998244354 998244354
10 998244353 998244354	

Note

当 $k = 2$ 时 $\{f(n)\}_{n=0}^{\infty} = \{0, 0, 1, 2, 2, 4, 3, 6, 4, 8, \dots\}$, 而 $\{f^2(n)\}_{n=0}^{\infty} = \{0, 0, 0, 1, 1, 2, 2, 3, 2, 4, \dots\}$ 。

Problem F. Square Subsequences

Input file: standard input

Output file: standard output

Time limit: 1 second

Memory limit: 256 megabytes

小 Q 有一个字符串 $s_1s_2\cdots s_n$ ，他想找出其中最长的子序列使得这个子序列能够表示成平方串，你能帮他找出来吗？

$s_1s_2\cdots s_n$ 的每个子序列可以用字符串 $s_{p_1}s_{p_2}\cdots s_{p_m}$ 表示，其中 $1 \leq p_1 < p_2 < \cdots < p_m \leq n$ 。

字符串 $t_1t_2\cdots t_m$ 是平方串当且仅当：

- m 是偶数；
- 对于 $i = 1, 2, \cdots, \frac{m}{2}$ ，有 $t_i = t_{i+\frac{m}{2}}$ 。

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

仅一行，包含一个由小写字母组成、长度为 n 的字符串 $s_1s_2\cdots s_n$ 。

- $1 \leq T, n \leq 3000$
- 所有测试数据的 n 之和不超过 3000。

Output

对于每组测试数据，首先输出一行 “Case #x: m”（不含引号），其中 x 是测试数据的编号（从 1 开始编号）， m 是最长平方串子序列的长度。

如果 m 为正，再输出一行，包含一个字符串，表示这个最长平方串子序列。如果有多种最优解，请输出任意一种。

Example

standard input	standard output
5	Case #1: 2
abba	aa
abbab	Case #2: 4
abac	abab
abcd	Case #3: 2
bbabab	aa
	Case #4: 0
	Case #5: 4
	bbbb

Note

对于样例的最后一组测试数据，最长平方串子序列也可以是 `baba`。

Problem G. Cosmic Cleaner

Input file: standard input

Output file: standard output

Time limit: 3 seconds

Memory limit: 256 megabytes

在一片小行星带里有 n 颗小行星，它们在万有引力的作用下绕着一颗行星旋转。在这一刻时，它们之间不存在碰撞的情况。一位清洁工奉命前来清理这颗行星，Ta 会动用某种先进技术使这颗行星顷刻间从宇宙中消失，任何距离这颗行星的中心在一定范围内的事物都会在一瞬间被清除。假设这些天体都是完整的球体，你能计算出清除的区域里有多少体积的事物原本属于这些小行星吗？

注意，这些天体在此刻满足两两不存在交集的条件。

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

第一行包含一个整数 n 。

接下来的 n 行里，每行包含四个整数 x, y, z 和 r ，表示有一颗中心位于 (x, y, z) 、半径为 r 的小行星。

最后一行包含四个整数 x', y', z' 和 r' ，表示行星的中心位于 (x', y', z') ，而清洁工的清理半径为 r' （一个大于该行星半径的值）。

- $1 \leq T \leq 6000$
- $1 \leq n \leq 100$
- $-10^3 \leq x, y, z, x', y', z' \leq 10^3$
- $1 \leq r, r' \leq 10^3$

Output

对于每组测试数据，输出一行 “Case #x: y”（不含引号），其中 x 是测试数据的编号（从 1 开始编号）， y 是这组数据的答案，要求相对误差或绝对误差不超过 10^{-6} 。

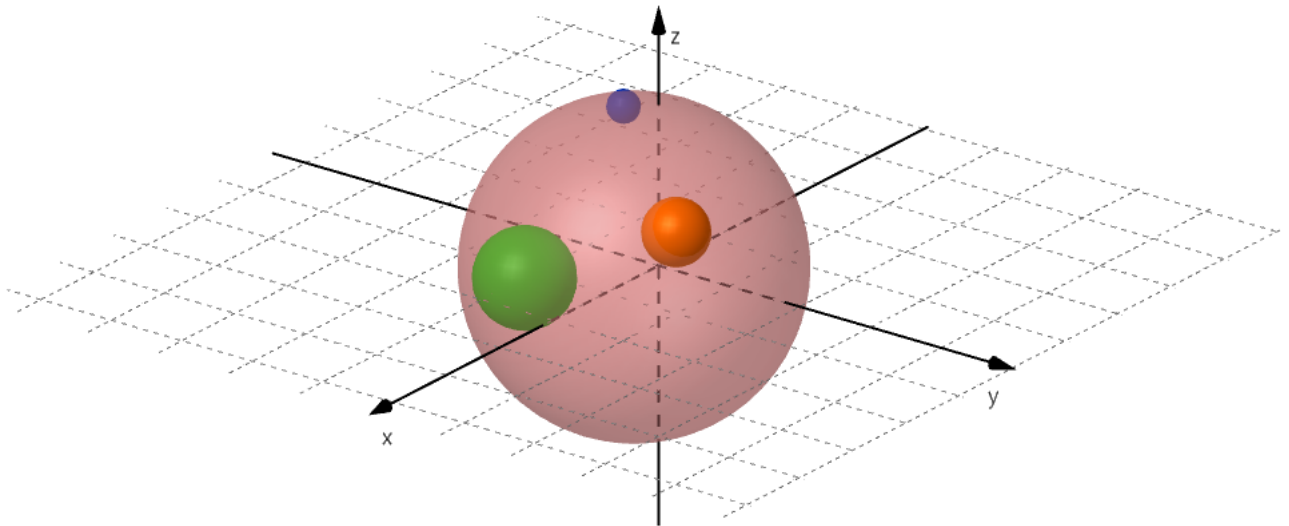
严格来讲，如果你的答案是 a ，而标准答案是 b ，那么当 $\frac{|a-b|}{\max\{1, |b|\}} \leq 10^{-6}$ 时你的答案会被认为是正确的。

Example

standard input	standard output
1 3 5 5 5 2 -6 -7 6 1 6 -5 0 3 1 -1 0 10	Case #1: 142.76246874761383764962

Note

下图是对样例的解释，其中清理区域是标记为红色的球体内部，而小行星则被依次标记为橙色、蓝色和绿色。



Problem H. Quicksort

Input file: standard input
 Output file: standard output
 Time limit: 3 seconds
 Memory limit: 256 megabytes

小 Q 同学刚刚又写了一个假的快速排序，如下图所示。如果他随机地选择一个 1 到 n 的排列 p_1, p_2, \dots, p_n 去执行 $\text{QuickSort}(p, 1, n, k)$ 这个过程，你知道这个排列在执行完之后期望会有多少个逆序对吗？

Functions — a fake QuickSort implementation

```

1: function QUICKSORT( $A, l, r, h$ )                                ▷ Elements in  $A$  would be modified
2:   if  $h > 1$  and  $l < r$  then
3:      $m \leftarrow \text{PARTITION}(A, l, r)$ 
4:     QUICKSORT( $A, l, m - 1, h - 1$ )
5:     QUICKSORT( $A, m + 1, r, h - 1$ )
6: function PARTITION( $A, l, r$ )                                ▷ Elements in  $A$  would be modified
7:    $m \leftarrow \lfloor \frac{l+r}{2} \rfloor$ 
8:    $\text{pivot} \leftarrow A[m]$ 
9:    $A[m] \leftarrow A[l]$ 
10:   $i \leftarrow l$ 
11:   $j \leftarrow r$ 
12:  while  $i < j$  do
13:    while  $i < j$  and  $A[j] \geq \text{pivot}$  do
14:       $j \leftarrow j - 1$ 
15:    if  $i < j$  then
16:       $A[i] \leftarrow A[j]$ 
17:    while  $i < j$  and  $A[i] < \text{pivot}$  do
18:       $i \leftarrow i + 1$ 
19:    if  $i < j$  then
20:       $A[j] \leftarrow A[i]$ 
21:   $A[i] \leftarrow \text{pivot}$ 
22:  return  $i$ 

```

提示：一个排列 p_1, p_2, \dots, p_n 的逆序对个数等于满足 $1 \leq u < v \leq n$ 且 $p_u > p_v$ 的整数二元组 (u, v) 数量。

为了避免可能的精度误差，你只需要给出期望的逆序对数乘以 n 的阶乘后对 998244353 取模的值。显然，这个期望值乘以 n 的阶乘是一个整数。

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

仅一行，包含两个整数 n 和 k 。

- $1 \leq T \leq 3 \times 10^5$
- $1 \leq n, k \leq 6000$

Output

对于每组测试数据，输出一行 “Case #x: y”（不含引号），其中 x 是测试数据的编号（从 1 开始编号），y 是这组数据的答案。

Example

standard input	standard output
5	Case #1: 600
5 1	Case #2: 240
5 2	Case #3: 64
5 3	Case #4: 8
5 4	Case #5: 0
5 5	

Problem I. Routes

Input file: standard input
Output file: standard output
Time limit: 5 seconds
Memory limit: 512 megabytes

很久很久以前，有一个王国，它有 n 个城市和 m 条铁路，那时每个城市恰好位于一条铁路线上。人们为了在不同铁路线上的两个城市之间往来，通常需要跋山涉水，方能到达。

交通不便极大地限制了经济的发展，为了民众的利益着想，国王不久后便开放了空中路线。此后，王国划分为 k 个限飞区域，在每个区域里，人们可以选择搭乘热气球前往同一区域里的城市，也可以继续通过铁路到达远方的城市。幸运的是，人们可以从任何一个城市出发，只借助这两种交通方式，便可以到达其他任意一个城市，王国也开始繁荣昌盛起来。

国王想确定是否他的决策能够帮到所有人，但是由于城市数量太过庞大，在他临死前，他还是没能知道这个问题的答案。不过现在，人们的计算能力改善了许多，这个问题又被再次提起。假设用热气球从一个城市移动到一个可达城市需要一个小时的时间，而通过铁路在两个相邻的站点之间通行也只需要一个小时的时间，你能否确定从一个城市到达另一个城市在只使用这些交通方式的情况下期望需要多少时间呢？

为了避免输入过大，每个城市将由所在铁路线上的一个小写字母表示，使得在同一个限飞区域里的城市拥有相同的字母。为了避免输出过大，你只需要给出，对于任意两个城市，通行所需要的时间之和。

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

第一行包含三个整数 n , m 和 k 。

接下来 m 行，每行包含一个由前 k 种小写字母组成的非空字符串，表示一条铁路线上的站点。

- $1 \leq T \leq 1000$
- $1 \leq m \leq n \leq 10^6$
- $1 \leq k \leq 16$
- 对于每组测试数据，保证前 k 种小写字母分别出现至少一次。
- 所有测试数据的 n 之和不超过 5×10^6 。
- 保证不超过 5 组测试数据满足 $k > 8$ 。

Output

对于每组测试数据，输出一行 “Case #x: y”（不含引号），其中 x 是测试数据的编号（从 1 开始编号）， y 是这组数据的答案（单位：小时）。

Example

standard input	standard output
3	Case #1: 1
2 1 2	Case #2: 1
ab	Case #3: 20
2 2 1	
a	
a	
5 2 3	
abb	
ac	

Problem J. Square Substrings

Input file: standard input

Output file: standard output

Time limit: 6 seconds

Memory limit: 512 megabytes

小 Q 给出了一个字符串 $s_1s_2\cdots s_n$ 和 q 个询问 (l_i, r_i) 。对于每个询问 (L, R) ，他想知道字符串 $s_Ls_{L+1}\cdots s_R$ 有多少个子串是平方串，你能帮他解决这个问题吗？

对于字符串 $t_1t_2\cdots t_m$ 来说，它的子串 $t_xt_{x+1}\cdots t_y$ ($1 \leq x \leq y \leq m$) 是平方串当且仅当：

- $(y - x + 1)$ 是偶数；
- 对于 $i = x, x + 1, \dots, \frac{x+y-1}{2}$ ，有 $t_i = t_{i+\frac{y-x+1}{2}}$ 。

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

第一行包含两个整数 n 和 q 。

第二行包含一个由小写字母组成的长度为 n 的字符串 $s_1s_2\cdots s_n$ 。

在接下来的 q 行中，第 i 行包含两个整数 l_i 和 r_i ，表示关于 $s_{l_i}s_{l_i+1}\cdots s_{r_i}$ 的一个询问。

- $1 \leq T \leq 100$
- $1 \leq n, q \leq 10^6$
- $1 \leq l_i \leq r_i \leq n$
- 所有测试数据的 n 之和不超过 10^6 。
- 所有测试数据的 q 之和不超过 10^6 。

Output

对于每组测试数据，首先输出一行 “Case #x:” (不含引号)，其中 x 是测试数据的编号 (从 1 开始编号)。

接下来，对于每个询问，输出一行，包含一个整数，表示这组询问的答案。

Example

standard input	standard output
1	Case #1:
7 5	1
ababbab	1
1 4	1
2 5	1
3 6	3
4 7	
1 7	

Note

bb 和 abab 以及 babbab 是平方串，但 abba 不是平方串。

Problem K. Sticks

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

小 Q 手上有 12 根木棒，它们的长度依次为 l_1, l_2, \dots, l_{12} 。他想用一些木棒来拼出三角形，这里每个三角形要用三根合适的木棒 l_a, l_b, l_c 组成，它们满足 $l_a + l_b > l_c$ 和 $l_a + l_c > l_b$ ，以及 $l_b + l_c > l_a$ 。此外，每根木棒只能用在至多一个三角形中。你能帮他尽可能多的拼出三角形吗？

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

仅一行，包含十二个整数 l_1, l_2, \dots, l_{12} 。

- $1 \leq T \leq 6000$
- $1 \leq l_i \leq 10^9$

Output

对于每组测试数据，首先输出一行 “Case #x: m”（不含引号），其中 x 是测试数据的编号（从 1 开始编号）， m 是三角形数量的最大值。

接下来输出 m 行，每行包含三个整数，表示三角形三条边的长度。

如果有多种最优解，请输出任意一种。注意，每根木棒只能使用至多一次，而且对于输出中位于同一行的相邻整数，你需要输出一个空格将它们隔开。

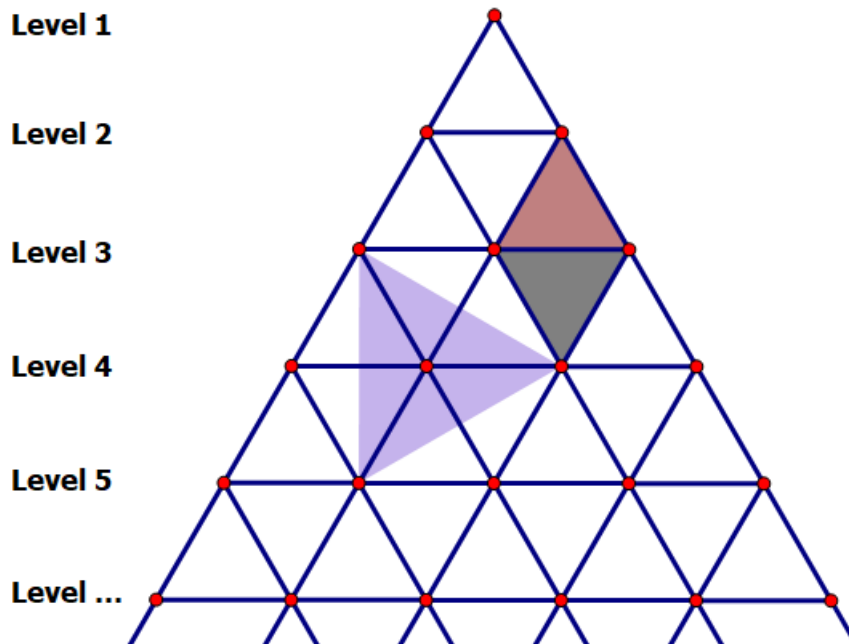
Example

standard input	standard output
5	Case #1: 4
1 2 1 3 1 4 1 5 1 6 1 7	1 1 1
1 2 3 4 5 6 7 8 9 10 11 12	4 3 2
1 2 3 5 8 13 21 34 55 89 144 233	1 1 1
2 3 6 15 27 59 72 83 121 159 201 234	6 7 5
2 2 4 8 16 32 64 128 256 512 1024 1281	Case #2: 3
	6 5 4
	10 12 11
	9 8 7
	Case #3: 0
	Case #4: 2
	83 121 72
	234 159 201
	Case #5: 1
	1024 1281 512

Problem L. Pyramid

Input file: standard input
 Output file: standard output
 Time limit: 2 seconds
 Memory limit: 256 megabytes

金字塔的侧表面可以表示成许多个等边三角形，并且可以将它们的顶点划分层次，使得第一层有一个顶点，第二层有两个，以此类推。



如上图所示，第 k 层每两个相邻的顶点和某个位于第 $(k - 1)$ 层的顶点组成了一个正常放置的等边三角形，而这两个来自第 k 层的顶点也能和某个位于第 $(k + 1)$ 层的顶点一起组成一个倒立放置的等边三角形。此外，来自三个不同层次的顶点也可能组成一个倾斜放置的等边三角形。

如果我们只考虑第 l 层到第 r 层的顶点，那么 we 有多少种选择三个顶点组成等边三角形的方案呢？

Input

输入包含多组测试数据。第一行包含一个整数 T ，表示测试数据的组数。随后的内容是各组测试数据。对于每组测试数据：

仅一行，包含两个整数 l 和 r 。

- $1 \leq T \leq 3 \times 10^5$
- $1 \leq l \leq r \leq 10^5$

Output

对于每组测试数据，输出一行 “Case #x: y”（不含引号），其中 x 是测试数据的编号（从 1 开始编号）， y 是这组数据的答案。

Example

standard input	standard output
3	Case #1: 5
1 3	Case #2: 12
2 4	Case #3: 20
3 5	