Problem A. Erase Nodes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	7 seconds
Memory limit:	256 mebibytes

Bob's job is to maintain a network, which can be represented as a connected graph having n nodes and n undirected edges, but now he has to close the entire network.

In the beginning, all nodes are active. During the processing of closing, he will repeatedly select one active node and inactivate it until there is no active node. However, each active node has a table that stores the connectivity information from this node to any other active node. When a node u is inactivated, every active node v (including u itself) that used to be able to reach u have to update its own table. Each update may change several records, as each inactivation can cut a connected component into several smaller components, but the cost of each update is almost the same — running a breadth-first search from v.

Now Bob is wondering in which order he should close these nodes because he knows if he operated these nodes in a bad order, the number of updates would be a bit large. He is not good at finding a good solution, so he chooses to randomly select one active node with equal probability at any time. Could you help him estimate the expected number of updates?

To avoid any precision issue, if the answer can be represented as an irreducible fraction $\frac{p}{q}$, then you are asked to report the minimum non-negative integer r such that $qr \equiv p \pmod{998244353}$. For example, $6 \times 166374072 \equiv 79, 3 \times 332748131 \equiv 40 \pmod{998244353}$.

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The first line contains an integer n.

Each of the following n lines contains two integers u and v, representing an edge between the u-th node and the v-th node.

- $1 \le T \le 100$
- $3 \le n \le 10^5$
- $1 \le u < v \le n$
- The sum of n in all test cases does not exceed 5×10^5.
- It is guaranteed that edges for each test case are distinct, which means there are no multiple edges.

Output

For each test case, output a line containing "`Case #x: y`" (without quotes), where `x` is the test case number starting from 1, and `y` is the answer to this test case.

Example

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

Problem B. Linear Congruential Generator

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

You are given a generator defined by the recurrence relation

$$X_{n+1} = ((aX_n + c) \bmod m),$$

where $X = \{X_n\}_{n=0}^{\infty}$ is the generated sequence of pseudorandom values, and m, a, c, X_0 are integer constants which specify the generator.

Additionally, two integer intervals $[l_1, r_1]$ and $[l_2, r_2]$ are given. Please calculate

$$\sum_{i=l_1}^{r_1} \sum_{j=l_2}^{r_2} (X_i \bmod (X_j + 1))$$

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The only line contains eight integers $m, a, c, X_0, l_1, r_1, l_2, r_2$.

- $1 \le T \le 10^5$
- $1 \le m \le 10^6$
- $0 \le a, c, X_0 < m$
- $0 \le l_1 \le r_1 \le 10^6$
- $0 \le l_2 \le r_2 \le 10^6$
- The sum of m in all test cases does not exceed 2×10^6.

Output

For each test case, output a line containing "`Case #x: y`" (without quotes), where `x` is the test case number starting from 1, and `y` is the answer to this test case.

Example

standard input	standard output
2 7 1 4 1 2 3 4 5 10 3 6 1 2 3 1 2	Case #1: 4 Case #2: 12

Note

In the first sample case, $\{X_n\}_{n=0}^{\infty} = \{1, 5, 2, 6, 3, 0, \ldots\}$.

In the second sample case, $\{X_n\}_{n=0}^{\infty} = \{1, 9, 3, 5, \ldots\}$.

Problem C. Fibonacci Strikes Back

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

In this problem, you need to solve a well-known problem about P-Fibonacci sequence:

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

Now given P, m and the lowest k decimal digits of F_{F_n}, you are asked to determine the minimum possible n such that $n \geq m$ and F_{F_n} has at least k lowest decimal digits as given above, or report it is impossible otherwise.

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The only line contains two integers P, m and a string of length k, consisting of only digits, which represents the lowest k decimal digits of F_{F_n}. Note the string may contain leading zeros.

- $1 \leq T \leq 10^4$
- $1 \leq P, m \leq 10^{18}$
- $1 \leq k \leq 18$
- The sum of k in all test cases does not exceed 10^4.
- It is guaranteed that the greatest common divisor of P and 10^{18} is less than 5 for each test case.

Output

For each test case, output a line containing "`Case #x: y`" (without quotes), where `x` is the test case number starting from 1, and `y` is the minimum possible n to this test case if it exists, or `y` is -1 otherwise.

Example

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

Note

When $P = 1$, $\{F_{F_n}\}_{n=0}^{\infty} = \{0, 1, 1, 1, 2, 5, 21, 233, 10946, 5702887, 139583862445, \ldots\}$.

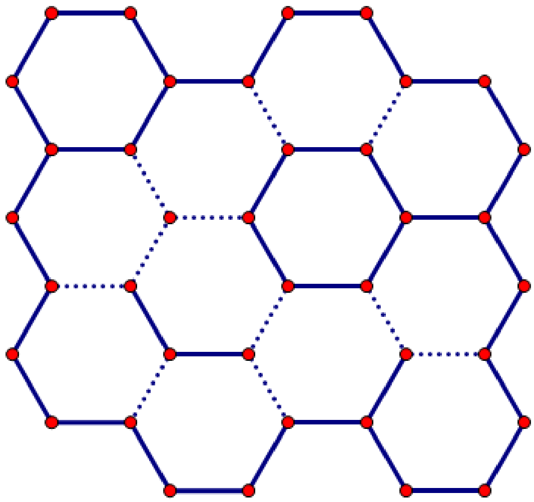
When $P = 2$, $\{F_{F_n}\}_{n=0}^{\infty} = \{0, 1, 2, 29, 13860, 44560482149, \ldots\}$.

Problem D. Honeycomb

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

A honeycomb is a mass of hexagonal prismatic cells, where each cell has six adjacent cells, and every two adjacent cells share an edge. In addition, some edges are traversable but others are not.

Hamilton is an industrious worker bee. Every day he works for connecting or disconnecting some pairs of adjacent cells in a honeycomb. In a few days, he will have a great chance to design a tiny block of cells by himself. This block is disconnected from its outside and it could be represented as cells in n rows and m columns, like the figure shown below.



To cut off the connection of two cells, he may have to change some edges into untraversable. He is wondering the minimum number of edges he has to change so that two specified cells in the block would be disconnected. Could you please help him find out the minimum number of changed edges for every two special cells in this block? To avoid huge output data, you are asked to report the sum of these minimum numbers of edges.

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The first line contains two integers n and m.

The following $(4n+3)$ lines describe the block, where each line contains at most $(6m+3)$ characters. Odd lines contain grid vertices represented as plus signs ("+") and horizontal edges, while even lines contain diagonal edges.

Specifically, a cell is described as 6 vertices and 6 edges. All edge characters will be placed exactly between the corresponding vertices, such that edges are described as following:

- Its upper boundary or lower boundary is represented as three consecutive minus signs ("-") if the edge is untraversable, or three consecutive spaces ("   ") otherwise;
- Each one of its diagonal edges is represented as a single space if the edge is traversable, or otherwise as a single forward slash ("/") or a single backslash ("\") character, depending on the direction of the edge.

Besides, there is an asterisk ("`*`") character at the center of each special cell. All other characters in the input will be spaces, and no input line will contain trailing spaces.

- $1 \le T \le 100$
- $2 \le n, m \le 100$
- It is guaranteed that every non-special cell has no traversable edge.
- The sum of the numbers of special cells in all test cases does not exceed 3000.

Output

For each test case, output a line containing "`Case #x: y`" (without quotes), where `x` is the test case number starting from 1, and `y` is the answer to this test case.

Example

standard input

```
2
2 2
  +---+
 /     \
+   *   +---+
 \     /     \
  +---+   *   +
 /           /
+   *   +   +
 \           \
  +---+   *   +
       \     /
        +---+
2 3
  +---+       +---+
 /     \     /     \
+   *   +---+       +
 \           \     /
  +---+   *   +---+
 /                 \
+   *   +---+   *   +
 \                 /
  +---+   *   +---+
       \     /
        +---+
```

standard output

```
Case #1: 6
Case #2: 16
```

Problem E. Power of Function

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

Bob has a function

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

which is defined for all **non-negative** integers.

Denote the m-th power of this function as $f^m(n)$ such that

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

He would like to know the maximum possible integer m meeting the condition that there exists at least one integer n such that $l \le n \le r$ and $f^m(n) = 1$. Besides, please help him find out the minimum and the maximum n for the maximum possible m so that he could easily validate your answer is correct.

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The only line contains three integers k, l, r.

- $1 \le T \le 3 \times 10^5$
- $2 \le k \le 10^{18}$
- $1 \le l \le r \le 10^{18}$
- It is guaranteed that solution exists for each test case.

Output

For each test case, output a line containing "`Case #x: m a b`" (without quotes), where `x` is the test case number starting from 1, `m` is the maximum possible exponent, `a` is the minimum possible argument, and `b` is the maximum possible argument with respect to `m`.

Example

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

Note

When $k = 2$, $\{f(n)\}_{n=0}^{\infty} = \{0, 0, 1, 2, 2, 4, 3, 6, 4, 8, \ldots\}$, and $\{f^2(n)\}_{n=0}^{\infty} = \{0, 0, 0, 1, 1, 2, 2, 3, 2, 4, \ldots\}$.

Problem F. Square Subsequences

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

Bob has a string $s_1 s_2 \cdots s_n$. He would like to find its longest subsequence that is a square. Could you please help him?

A subsequence of $s_1 s_2 \cdots s_n$ can be represented as a string $s_{p_1} s_{p_2} \cdots s_{p_m}$ meeting the condition that $1 \le p_1 < p_2 < \ldots < p_m \le n$.

A string $t_1 t_2 \cdots t_m$ is a square if and only if:

- m is even;
- $t_i = t_{i+\frac{m}{2}}$ for $i = 1, 2, \ldots, \frac{m}{2}$.

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The only line contains a string of n lowercase letters, $s_1 s_2 \cdots s_n$.

- $1 \le T, n \le 3000$
- The sum of n in all test cases does not exceed 3000.

Output

For each test case, firstly output a line containing "`Case #x: m`" (without quotes), where `x` is the test case number starting from 1, and `m` is the length of the longest square subsequence.

If `m` is positive, then output a string in the next line, representing the longest square subsequence. If there are many optimal solutions, please output any of them.

Example

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

Note

For the last sample test case, the longest square subsequence can be `baba` as well.

Problem G. Cosmic Cleaner

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

There are n asteroids forming an asteroid belt. Due to gravity, these asteroids revolve around a planet and they, asteroids and the planet, don't collide with each other at the moment. A cleaner is ordered to wipe out the planet, which means they will take advantage of their advanced technology to obliterate the planet, and anything close to its center will be erased immediately. If these celestial bodies are regarded as perfect balls, can you evaluate the total volume of areas that belong to asteroids but will be cleaned up after executing the mission?

Note that these celestial bodies don't intersect with each other, which means there is no point shared by any two bodies.

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The first line contains an integer n.

Each of the following n lines contains four integers x, y, z and r, representing an asteroid with radius r, whose center is located at (x, y, z).

The last line contains four integers x', y', z' and r', representing that the center of the planet, which is also the center of the cleaning area, is located at (x', y', z'), and the radius of the area affected by the cleaner is r', which is greater than the planet's radius.

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

Output

For each test case, output a line containing "`Case #x: y`" (without quotes), where `x` is the test case number starting from 1, and `y` is the answer to this test case, with an absolute or relative error of at most 10^{-6}.

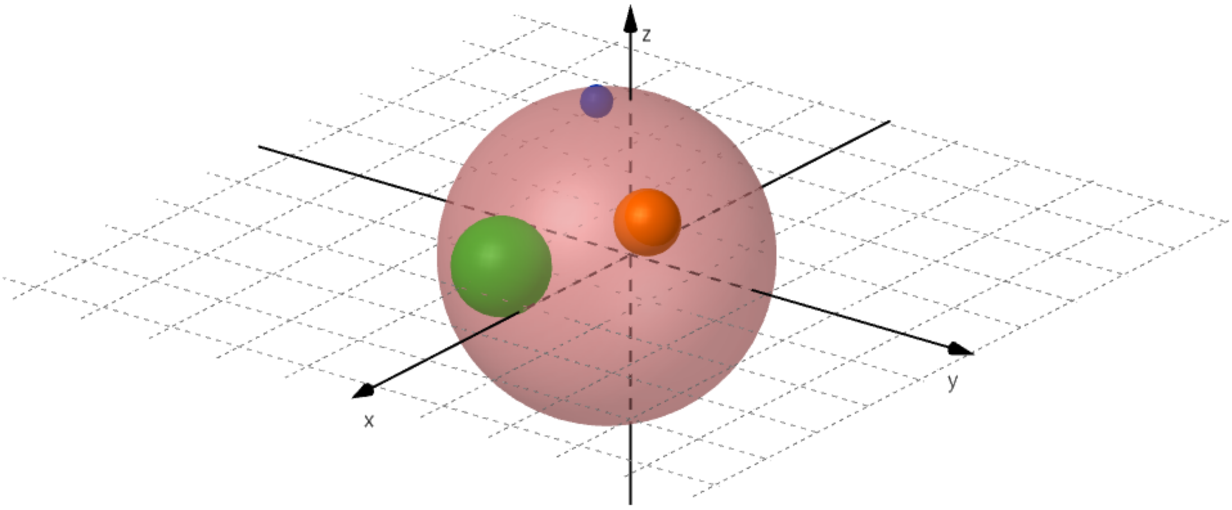
Formally, assuming that your answer is a and the jury's answer is b, your answer will be considered correct if $\frac{|a-b|}{\max\{1,|b|\}} \le 10^{-6}$.

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

The following figure illustrates the sample test case, where the affected area is inside the red sphere, and asteroids are colored by orange, blue and green in the order they appear in the sample.


z
x
y

Problem H. Quicksort

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Bob has written a fake QuickSort implementation as below. Can you figure out, if he randomly chooses a permutation $p = [p_1, p_2, \ldots, p_n]$ obtained from $1, 2, \ldots, n$ with equal probability, the expected number of inversions of p after calling `QuickSort(p, 1, n, k)`?

Functions — a fake QuickSort implementation

```
function QuickSort(A, l, r, h)                ▷ Elements in A would be modified
    if h > 1 and l < r then
        m ← Partition(A, l, r)
        QuickSort(A, l, m − 1, h − 1)
        QuickSort(A, m + 1, r, h − 1)
function Partition(A, l, r)                   ▷ Elements in A would be modified
    i ← l
    j ← r
    m ← ⌊(l+r)/2⌋
    pivot ← A_m
    A_m ← A_i
    while i < j do
        while i < j and A_j ≥ pivot do
            j ← j − 1
        if i < j then
            A_i ← A_j
        while i < j and A_i < pivot do
            i ← i + 1
        if i < j then
            A_j ← A_i
    A_i ← pivot
    return i
```

The number of inversions of a permutation $[p_1, p_2, \ldots, p_n]$ is the number of integer pairs (u, v) such that $1 \le u < v \le n$ and $p_u > p_v$.

To avoid any precision issue, you are asked to report the product of $n!$, the factorial of n, and the expected number in modulo 998244353, which ought to be an integer.

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The only line contains two integers n and k.

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

Output

For each test case, output a line containing "`Case #x: y`" (without quotes), where `x` is the test case number starting from 1, and `y` is the answer to this test case.

Example

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

Problem I. Routes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

A long long time ago, there were n cities and m railways in a kingdom. Each railway contained many cites, and each city was located on **exactly one** railway. Since some cities were not connected by a railway, people often made an arduous journey when moving from one city to another. The inconvenient traffic had greatly blocked the development of the kingdom's economy, and therefore its king soon made a decision to launch several airline routes.

Under his directive, the cities were divided into k districts, and thus people were able to travel between every two cities belonging to the same district by hot air balloons. Furthermore, when planning to travel to any other city, people in the kingdom could choose to go by the combination of trains and hot air balloons without any other kinds of transportation, which was also a reason why the economy continued to prosper.

The king wanted to know if the policy he made had benefited everyone. However, he didn't find it out before he died, as a huge amount of calculation was required. Now that the computing ability is better, the problem is called back to solve again. Assuming that moving from a city to another city directly by hot air balloon will spend one hour, and traveling from a city to an adjacent city on a railway will spend one hour as well, you are asked to determine the average time cost needed to travel from a city to another city by using only these two kinds of transportation, and then report the product of $\frac{n(n-1)}{2}$ and the average value, which ought to be an integer.

To avoid huge input data, each city is represented as one of the first k lowercase letters, representing a station on a railway, such that cities represented by the same letter are in the same district.

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The first line contains three integers n, m and k.

Each of the following m lines contains a non-empty string, consisting of the first k lowercase letters, describing a railway's stations in order.

- $1 \leq T \leq 1000$
- $1 \leq m \leq n \leq 10^6$
- $1 \leq k \leq 16$
- Each district contains at least one city.
- The sum of n in all test cases does not exceed 5×10^6.
- There are at most 5 test cases with $k > 8$.

Output

For each test case, output a line containing "`Case #x: y`" (without quotes), where `x` is the test case number starting from 1, and `y` is the answer (in hours) to this test case.

Example

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

Problem J. Square Substrings

Input file:	*standard input*
Output file:	*standard output*
Time limit:	8 seconds
Memory limit:	512 mebibytes

Bob has a string $s_1 s_2 \cdots s_n$ and q queries (l_i, r_i) $(i = 1, 2, \ldots, q)$. For each query (l_i, r_i), he would like to know the number of intervals (L, R) such that $l_i \le L \le R \le r_i$ and $s_L s_{L+1} \cdots s_R$ is a square. Could you please help him?

A string $t_1 t_2 \cdots t_m$ is a square if and only if:

- m is even;
- $t_i = t_{i+\frac{m}{2}}$ for $i = 1, 2, \ldots, \frac{m}{2}$.

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The first line contains two integers n and q.

The second line contains a string of n lowercase letters, $s_1 s_2 \cdots s_n$.

The i-th one of the following q lines contains two integers l_i and r_i, representing a query.

- $1 \le T \le 100$
- $1 \le n, q \le 10^6$
- $1 \le l_i \le r_i \le n$
- The sum of n in all test cases does not exceed 10^6.
- The sum of q in all test cases does not exceed 10^6.

Output

For each test case, firstly output a line containing "`Case #x:`" (without quotes), where `x` is the test case number starting from 1.

Then, for each query, output a line containing an integer, denoting the answer to this query.

Example

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

Note

`bb`, `abab` and `babbab` are squares, while `abba` is not a square.

Problem K. Sticks

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

Bob has 12 sticks of lengths $l_1, l_2, \ldots, l_{12}$. He wants to use some sticks to form triangles as many as possible. Each triangle can be built by three different sticks l_a, l_b, l_c such that $l_a + l_b > l_c$, $l_a + l_c > l_b$ and $l_b + l_c > l_a$. If each stick can be used for at most one triangle, how many triangles can he build at most? Also, could you please find a way to build them all?

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The only line contains twelve integers $l_1, l_2, \ldots, l_{12}$.

- $1 \le T \le 6000$
- $1 \le l_1, l_2, \ldots, l_{12} \le 10^9$

Output

For each test case, firstly output a line containing "`Case #x: m`" (without quotes), where `x` is the test case number starting from 1, and `m` is the maximum number of triangles that can be built.

Then, output `m` lines, each line of which contains three integers, representing three side lengths of a triangle.

If there are many optimal solutions, please output any of them. Note that every stick for each test case can be used at most once, and every two adjacent integers in a line of the output should be separated by one space.

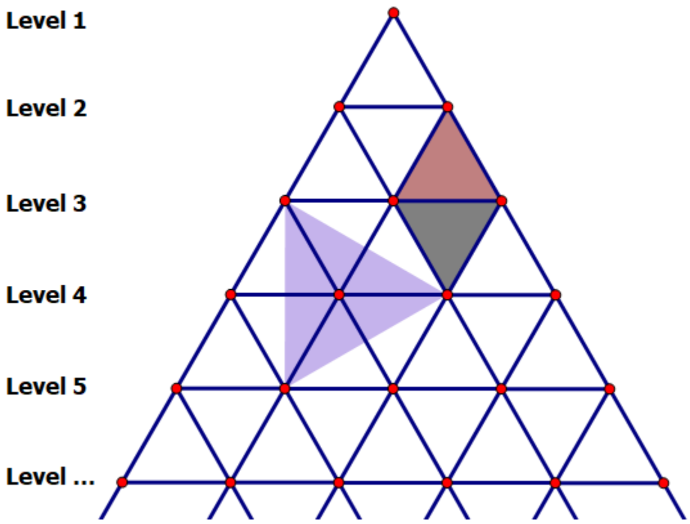
Example

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

A side surface of a pyramid can be cut into many equilateral triangles, whose vertices can be grouped into different levels from top to bottom, such that the first level contains one vertex, the second level contains two, and so on.



As the image shows, for example, two adjacent level-k vertices with a level-$(k-1)$ vertex can form an upright equilateral triangle, and two adjacent level-k vertices with a level-$(k+1)$ vertex can form an inverted equilateral triangle as well. Also, three vertices at three different levels can form an equilateral triangle, which may be oblique.

If we only consider vertices between level l and level r (inclusive), in how many ways can we choose three equidistant vertices so that they can form an equilateral triangle?

Input

The input contains several test cases. The first line contains an integer T indicating the number of test cases. The following describes all test cases. For each test case:

The only line contains two integers l and r.

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

Output

For each test case, output a line containing "`Case #x: y`" (without quotes), where `x` is the test case number starting from 1, and `y` is the answer to this test case.

Example

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