

Problem A. Boomerang Decoration

Input file: **standard input**
Output file: **standard output**
Time limit: 30 seconds
Memory limit: 512 megabytes

As an experienced boomerangist, you take great pride in your trusty boomerang, and you'd like to be able to show it off! Unfortunately, it doesn't look quite right yet...

Like all boomerangs, yours is symmetrical in shape, with two N millimetre-long halves. However, what's not necessarily symmetrical is its paint job — each millimetre-long section of each half has been painted some color. There are 26 possible colors, represented by uppercase letters ("A" to "Z"), and the i th section of the left half (counting from the left end) has color A_i , while the i th section of the right half (counting from the right end) has color B_i . You'd like your boomerang to be completely symmetrical (which is the case when the i th section in the left half has the same color as the i th section in the right half, for all $1 \leq i \leq N$) as soon as possible!

To that end, you're enlisting the help of your two best friends, Jack and Jill, to repaint the left half of your boomerang until it matches the right half. Every second, Jack may paint over a "prefix" of the left half with a solid color, while Jill may simultaneously paint over a disjoint "suffix". In other words, every second, Jack and Jill will each select a color (from "A" to "Z"), Jack will select a value x ($0 \leq x \leq N$), and Jill will select a value y ($0 \leq y \leq N - x$). Then, Jack will repaint the first x sections of the left half of the boomerang with his color, while Jill will repaint the last y sections of the left half of the boomerang with her color. Painting over a section completely overwrites its previous color.

What's the minimum amount of time (in seconds) it can take Jack and Jill to make your boomerang completely symmetrical?

Input

Input begins with an integer T ($1 \leq T \leq 75$), the number of boomerangs you want to decorate. For each boomerang, there is first a line containing the integer N ($1 \leq N \leq 10^5$), then two lines that each contain a string of length N . The i -th character of the second line is A_i . The i -th character of the third line is B_i .

Output

For the i th boomerang, print a line containing "Case #i: " followed by the minimum number of seconds it will take Jack and Jill to make your boomerang's paint job symmetrical.

Examples

standard input	standard output
5	Case #1: 0
3	Case #2: 1
ABC	Case #3: 2
ABC	Case #4: 3
3	Case #5: 4
ABC	
CBA	
3	
ABC	
BAC	
9	
FOXENRULE	
NOREAALLY	
19	
DOYOUlikeBOOMERANGS	
00B0000VVB00EAAAGGS	

Note

The first boomerang is already symmetrical, so no work needs to be done. The second boomerang can be made symmetrical in 1 second. Jack will cover the first millimetre with 'C', and Jill will cover the third millimetre with 'A'. The third boomerang takes 2 seconds to paint. One possible solution is for Jack to paint the first two millimetres with 'A' and then the first millimetre with 'B'.

Problem B. Carnival Coins

Input file: **standard input**
Output file: **standard output**
Time limit: 30 seconds
Memory limit: 512 megabytes

At the local carnival you find a midway game advertising "TONS OF FABULOUS PRIZES". Certainly a fabulous prize or two would make your time worthwhile. It turns out that "TONS" is actually a bit of an understatement. There are in fact infinitely many prizes available. Consequently, the game operator is willing to give you a chance to score multiple prizes in a single game (for a nominal fee).

After you hand over your money, the game operator gives you N coins. Each coin has the same probability p of landing on heads when flipped (and consequently probability $1 - p$ of landing on tails). She also gives you a goal, an integer K .

As long as you still have some coins remaining, you can select any number of them and flip them all simultaneously. These coins are then taken away. If at least K of them land on heads, you win a prize. If you play optimally, what is the expected number of prizes you'll manage to win?

Input

Input begins with an integer T ($1 \leq T \leq 100$), the number of times you play the game. For each attempt, there is a line containing the space-separated values N , K ($1 \leq N, K \leq 3000$) and p ($0 \leq p \leq 1$). N and K are integers, and p is given with at most 16 decimal places.

Output

For the i th attempt, print a line containing "Case #i: " followed by the expected number of prizes you'll win. Your output should have at most 10^{-6} absolute or relative error.

Examples

standard input	standard output
5	Case #1: 1.000000000
2 1 0.5	Case #2: 1.180980000
10 5 0.9	Case #3: 0.001634937
10 5 0.1	Case #4: 5.712907306
3000 50 0.123	Case #5: 56.182751225
3000 50 0.987	

Note

In the first case, flipping the coins together gives you a 75 percent chance of winning a prize. If you flip them separately, you get a 50 percent chance on each flip. The latter approach is better, giving you 1 expected prize rather than 0.75 expected prizes. In the second case, it is optimal to partition the ten coins into two sets of five. In the third case, it is optimal to flip all ten coins at once.

Problem C. Snakes and Ladders

Input file: standard input
Output file: standard output
Time limit: 30 seconds
Memory limit: 512 megabytes

As the owner of a vast collection of fine ladders, you would like to display them to potential onlookers. You've decided to stand your N ladders up vertically, with their bases at distinct points on a horizontal number line. The i -th ladder's base is at position X_i , and it has height H_i .

The local snake population has taken an interest in your ladders. As everyone knows, snakes love nothing more than to suspend themselves horizontally above the ground! In particular, a snake of length L is able to suspend itself between the tops of two ladders a and b if and only if they meet the following conditions:

- the ladders are exactly L units apart ($|X_a - X_b| = L$)
- the ladders are of equal height ($H_a = H_b$)
- there are no taller ladders in between them (there exists no ladder c such that $\min\{X_a, X_b\} < X_c < \max\{X_a, X_b\}$ and $H_c > H_a$)

A number of snakes are planning to take up residence amongst your ladders. In particular, for every position in which a snake could suspend itself (in other words, for every distinct, unordered pair of ladders a and b), one snake will move in of the appropriate length.

You'll have no choice but to take care of these snakes, of course. To feed a snake of length L , it'll cost you L^2 dollars daily. To prepare your budget, you'd like to calculate how many dollars you'll be spending each day on your new pets!

Input

Input begins with an integer T ($1 \leq T \leq 50$), the number of sets of ladders you own. For each set, there is first a line containing the integer N ($1 \leq N \leq 2 \cdot 10^5$). Then, N lines follow, the i -th of which contains two space-separated integers, X_i and H_i ($0 \leq X_i, H_i \leq 10^9$).

Output

For the i th set of ladders, print a line containing "Case #i: " followed by the daily feeding cost of all the snakes that move in, modulo $10^9 + 7$.

Examples

standard input	standard output
5	Case #1: 400
2	Case #2: 1400
10 100	Case #3: 10
30 100	Case #4: 238
3	Case #5: 14991178
10 100	
30 100	
40 100	
5	
1 3	
4 3	
3 2	
5 2	
2 2	
8	
5 6	
8 1	
9 5	
13 5	
15 6	
16 6	
19 7	
100 6	
4	
1 1000000000	
1000 1000000000	
1000000 1000000000	
1000000000 1000000000	

Note

In the first case, one snake will move in between your two ladders. It will have length 20, so it will cost $20^2 = 400$ dollars a day to feed. In the third case, one snake will move in between the ladders of height 3, and another will move in between the ladders at $X = 2$ and $X = 3$. The first snake has length 3, and the second has length 1. The total cost is therefore $3^2 + 1^2 = 10$.

Problem D. Costly Labels

Input file: standard input
Output file: standard output
Time limit: 30 seconds
Memory limit: 512 megabytes

The original test have been splitted to block with $T \leq 5$.

You've got yourself an unrooted tree with N nodes — that is, a connected, undirected graph with N nodes numbered from 1 to N , and $N - 1$ edges. The i th edge connects nodes A_i and B_i .

You'd like to spend as little money as possible to label each node with a number from 1 to K , inclusive. It costs $C_{i,j}$ dollars to label the i -th node with the number j .

Additionally, after the whole tree has been labelled, you must pay P more dollars for each node which has at least one pair of neighbours that share the same label as each other. In other words, for each node u , you must pay P dollars if there exist two other nodes v and w which are both adjacent to node u , such that the labels on nodes v and w are equal (note that node u 's label is irrelevant). You only pay the penalty of P dollars once for a given central node u , even if it has multiple pairs of neighbours which satisfy the above condition.

What's the minimum cost (in dollars) to label all N nodes?

Input

Input begins with an integer T ($1 \leq T \leq 5$), the number of trees. For each tree, there is first a line containing the space-separated integers N ($1 \leq N \leq 1000$), K ($1 \leq K \leq 30$), and P ($0 \leq P \leq 10^6$). Then, N lines follow, the i -th of which contains the space-separated integers $C_{i,1}$ through $C_{i,K}$ in order ($0 \leq C_{i,j} \leq 10^6$). Then, $N-1$ lines follow, the i -th of which contains the space-separated integers A_i and B_i ($1 \leq A_i, B_i \leq N$).

Output

For the i -th tree, print a line containing "Case #i: " followed by the minimum cost to label all of the tree's nodes.

Examples

standard input	standard output
5	Case #1: 111
1 1 1	Case #2: 15
111	Case #3: 15
3 1 8	Case #4: 99
1	Case #5: 1
2	
4	
1 2	
2 3	
3 2 10	
4 7	
8 9	
2 3	
1 2	
2 3	
4 2 99	
0 1	
0 1	
0 1	
0 0	
4 1	
2 4	
4 3	
4 3 99	
0 1 0	
0 1 0	
0 1 0	
0 0 0	
4 1	
2 4	
4 3	

Note

In the first case, there is only one node which must be painted the only possible color for 111 dollars. In the second case, there is only one color, so a penalty of 8 dollars must be paid since node 2 has two neighbors with the same color. In total we pay $1 + 2 + 4 + 8 = 15$ dollars. In the third case, it's optimal to paint nodes 1 and 2 with color 1, and node 3 with color 2. The total cost is $4 + 8 + 3 = 15$ dollars.