Problem A. Array

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Chiaki has an array of integers $a_1, a_2, \ldots, a_n$. Chiaki can replace an element a_x to another integer y. Let the resulting array be $b_1, b_2, \ldots, b_n$. Chiaki would like to know the minimum value of $|a_x - y| + \sum\limits_{k=1}^{n} k \cdot c_k$, where c_k is the number of distinct integers in $b_1, b_2, \ldots, b_k$.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^6$) — the length of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

It is guaranteed that the sum of n in all test cases will not exceed 10^6.

Output

For each test case, output an integer in a single line, denoting the answer.

Example

standard input	standard output
1 4 1 2 3 4	22

Problem B. Chiaki Chain

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Chiaki has a graph consisting of n vertices and m edges. Each edge connects two vertices. After a short time of research, she has realized that the graph may represent a special graph — the k-th order Chiaki Chain.

An ordinary chain is a graph consisting of sequential (at least two) vertices. Every two adjacent vertices are connected by an edge. The k-th order Chiaki Chain looks slightly different from a chain. There are k sub-chains extended from k different vertices on the main chain. At the other side of each sub-chain, there is a simple cycle of length $3, 4, \ldots, k+2$ respectively. There is no useless vertices or edges on the k-th order Chiaki Chain.

Chiaki would like to know whether the graph represents the k-th order Chiaki Chain or not.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains three integers n, m and k $(1 \le n, m, k \le 2 \times 10^5)$ — the number of vertices and the number of edges in the graph and the order of Chiaki Chain.

Then followed by m lines. The i-th line contains two integers x_i and y_i $(1 \le x_i, y_i \le n)$ representing the vertices the i-th edge connects.

It is guaranteed that the sum of m in all test cases will not exceed 2×10^5.

Output

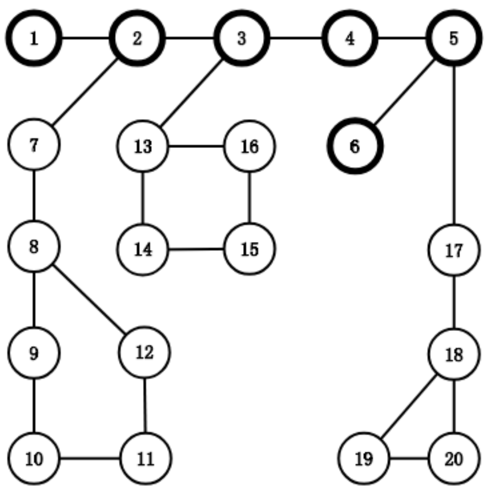
For each test case, output "*Yes*" if the graph represents the k-th order Chiaki Chain, or "*No*" otherwise.

Example

standard input	standard output
2 20 22 3 1 2 2 3 3 4 4 5 5 6 2 7 7 8 8 9 9 10 10 11 11 12 12 8 3 13 13 14 14 15 15 16 16 13 5 17 17 18 18 19 19 20 20 18 5 6 3 1 2 2 3 3 4 4 5 5 1 1 3	Yes No

Note

The following image corresponds to the first sample case.



Problem C. Cut the Plane

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

There are n distinct points on the plane, any three of which are not colinear.

You are asked to use $\lceil \frac{n}{2} \rceil$ distinct lines passing through no given points to cut the plane into pieces such that no two given points lie in the same piece.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n $(1 \le n \le 100)$ — the number of points.

Each of the following n lines contains two integers x and y $(-1000 \le x, y \le 1000)$ describing a point on the plane.

It is guaranteed that there always exists a solution for each test case and the sum of n in all test cases will not exceed 10^5.

Output

For each test case, output $\lceil \frac{n}{2} \rceil$ lines describing a solution.

Each line contains four integers x_1, y_1, x_2 and y_2 representing a line passing through (x_1, y_1) and (x_2, y_2), where $(x_1, y_1) \neq (x_2, y_2)$ and the absolute value of each coordinate should not exceed 10^9.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>2
3
0 0
2 1
4 0
4
0 1
1 0
2 1
1 2</pre></td><td><pre>1 0 1 1
3 0 3 1
0 0 2 2
2 0 0 2</pre></td></tr>
</table>

Problem D. Edges Counting

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

We call a simple graph good if each component of the graph has at most one cycle.

Your task is to count the number of edges belonging to one cycle for all the good graphs with n labeled vertices.

In order to avoid calculations of huge integers, please report the total number of these edges modulo p.

Input

There are multiple test cases. The first line of the input contains two integers T and p ($1 \le T \le 3000$, $1 \le p \le 2^{30}$), indicating the number of test cases and the modulus. For each test case:

The first line contains the only integer n ($1 \le n \le 3000$).

Output

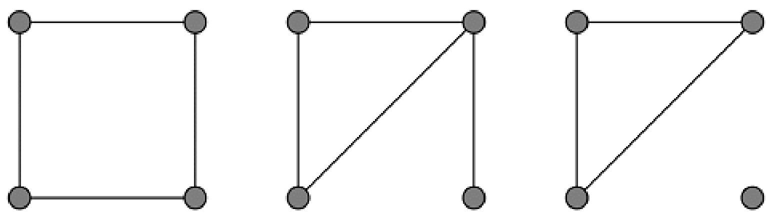
For each test case, output the total numbers of edges, each of which belongs to one cycle of a good graph with n labeled vertices, modulo p in a single line.

Example

standard input	standard output
7 998244353 1 2 3 4 5 6 7	0 0 3 60 1050 19380 393750

Note

There are three types of good graphs having four labeled vertices in which at least one cycle exists.



The numbers of these types of graphs are 3, 12 and 4 respectively. Consequently, the total number of required edges is $3 \times 4 + 12 \times 3 + 4 \times 3 = 60$.

Problem E. Equanimous

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

Alice, Bob and Eve are playing a game on craft papers. Every time Eve shows a natural number, Alice and Bob should write the number (in the decimal representation) on their own papers, add a plus sign or a minus sign before each digit, and then evaluate the arithmetic expression he or she has wrote. The one with the lower **absolute value** of his or her expression wins. If their absolute values are the same, it will cause a draw and they need to play one more time.

Actually, the game will never end if they are smart enough, so after a while, they turn to focus on the optimal solution of the puzzle game. Let $f(m)$ be the minimal absolute value that can be built from m. They are wondering if you can help them determine $f(m)$ for every integer m satisfying $l \le m \le r$.

Wait. After realizing your perfect programming skill, they decided to make a puzzle for you as well. They have set several questions (l, r) for you, and your task is to find the sum of all the integers m between l and r (inclusive) satisfying $f(m) = k$ for $k = 0, 1, 2, \ldots, 9$ and report the answer modulo $(10^9 + 7)$.

Input

The first line contains one integer n ($1 \le n \le 10^4$) indicating the number of questions.

Each of the next n lines contains two integers l and r ($1 \le l \le r \le 10^{100}$) representing a question.

Output

For each question, output ten space-separated integers in one line, where the i-th integer indicates the sum of all the integers m, satisfying that $l \le m \le r$ and $f(m) = i$, modulo $(10^9 + 7)$.

Example

standard input
7 1 10 11 50 51 100 101 500 501 1000 19260817 19260818 1234567890123456789 1234567890987654321

standard output
0 11 2 3 4 5 6 7 8 9 110 210 211 193 166 180 84 47 19 0 385 770 579 497 424 310 306 243 171 90 19080 34666 27312 19047 10615 5490 2594 1097 299 0 43695 81005 67134 55962 46289 35085 23872 13924 6385 1899 19260817 19260818 0 0 0 0 0 0 0 0 230833519 749351908 0 0 0 0 0 0 0 0

Note

The digits of 19260817 in the decimal representation are $\{1, 9, 2, 6, 0, 8, 1, 7\}$, which can build an arithmetic expression $(+1 - 9 - 2 - 6 + 0 + 8 + 1 + 7)$, whose value and abosolute value are 0.

The digits of 19260818 in the decimal representation are $\{1, 9, 2, 6, 0, 8, 1, 8\}$, which can build an arithmetic expression $(+1 - 9 + 2 + 6 - 0 - 8 - 1 + 8)$, whose value is -1 and aboslute value is 1.

Problem F. Fighting Against Monsters

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 mebibytes

One day, a hero and three monsters are fighting in the forest through turn-based battles. These monsters are a boss monster, which has extremely high health points, and two little monsters, which have fairly low health points. The health points of the three monsters are HP_A, HP_B and HP_C respectively, and their attack values are ATK_A, ATK_B and ATK_C respectively.

The turn-based battle occurs every second. During the i-th second, the hero will be attacked by monsters at first, and the damage is the sum of attack values of all alive monsters. Then he will select **exactly one** monster which is still alive and attack it. The selected monster will suffer damages of value i (i.e. its health points will be decreased by i). For instance: during the 1-st second, one of these three monsters will be under an attack of damage 1; during the 2-nd second, one of them, which is alive, will be under an attack of damage 2; during the 3-rd second, one of them, which is alive, will be under an attack of damage 3; and so on.

Once the value of a monster's health points is less than or equal to zero, it will die immediately. The hero will win if all the monsters have been killed.

The hero knows that health is very precious! He wants you to develop a strategy to minimize the total damages the hero should suffer before he wins the battle.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 50$), indicating the number of test cases. For each test case:

The first line contains six integers HP_A, HP_B, HP_C, ATK_A, ATK_B and ATK_C ($1 \le HP_A, HP_B \le 100$, $1 \le HP_C \le 10^{18}$, $1 \le ATK_A, ATK_B, ATK_C \le 10^9$).

Output

For each test case, output an integer in a single line, denoting the minimal total damages the hero should suffer.

Example

standard input	standard output
2 1 10 100 3 2 1 3 2 1 1 10 100	28 123

Problem G. Mysterious Triple Sequence

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

Jeffery found an amazing sequence of triples $\{(a_k, b_k, c_k)\}_{k=0}^{\infty}$:

- $(a_0, b_0, c_0) = (2, 1, 0)$; and
- for each non-negative integer k, $(a_{k+1}, b_{k+1}, c_{k+1}) = (a_k^2 + b_k^2, a_k b_k + b_k c_k, b_k^2 + c_k^2)$.

For example, $(a_1, b_1, c_1) = (5, 2, 1)$ and $(a_2, b_2, c_2) = (29, 12, 5)$.

If we consider the sequence in modulo an integer p, some triples would never appear in this sequence, some triples would appear periodically and other triples would appear only once.

Jeffery is wondering if you could help him find out the first appearance of some triples starting from given positions. Could you help him, please?

Input

The first line contains two integers n and p ($1 \le n \le 5000$, $1 \le p \le 2^{30}$) where n indicates the number of questions and p indicates all the following questions are considered in modulo p.

Each of the next n lines contains four integers x, y, z and m ($0 \le x, y, z < p$, $0 \le m \le 10^{18}$) representing a question that queries you to find the minimum integer k such that $k \ge m$ and $(a_k, b_k, c_k) \equiv (x, y, z) \pmod{p}$.

Output

For each question, output an integer in a single line, indicating the answer to the question. If there is no such integer k, output -1 instead.

Examples

standard input	standard output
5 11 6 1 4 10 4 10 6 3 7 1 5 0 2 1 0 1 2 1 0 0	11 4 2 -1 0
5 10 5 8 9 5 0 2 6 0 9 2 5 6 5 5 5 7 5 2 1 2	5 -1 6 -1 -1

Note

In the first sample, $(a_0, b_0, c_0) \equiv (2, 1, 0)$, $(a_1, b_1, c_1) \equiv (5, 2, 1)$, $(a_2, b_2, c_2) \equiv (7, 1, 5)$, $(a_{2T+3}, b_{2T+3}, c_{2T+3}) \equiv (6, 1, 4)$, $(a_{2T+4}, b_{2T+4}, c_{2T+4}) \equiv (4, 10, 6) \pmod{11}$ where $T = 0, 1, 2, \ldots$

In the second sample, $(a_0, b_0, c_0) \equiv (2, 1, 0)$, $(a_1, b_1, c_1) \equiv (5, 2, 1)$, $(a_{2T+2}, b_{2T+2}, c_{2T+2}) \equiv (9, 2, 5)$, $(a_{2T+3}, b_{2T+3}, c_{2T+3}) \equiv (5, 8, 9) \pmod{10}$ where $T = 0, 1, 2, \ldots$

Problem H. Inner Product

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

Chiaki has two trees and each tree has n vertices, labeled by $1, 2, \ldots, n$. Consider the following two arrays $A = [d_1(1,1), d_1(1,2), \ldots, d_1(1,n), d_1(2,1), d_1(2,2), \ldots, d_1(2,n), \ldots, d_1(n,1), d_1(n,2), \ldots, d_1(n,n)]$, $B = [d_2(1,1), d_2(1,2), \ldots, d_2(1,n), d_2(2,1), d_2(2,2), \ldots, d_2(2,n), \ldots, d_2(n,1), d_2(n,2), \ldots, d_2(n,n)]$, where $d_1(i,j)$ is the distance between i and j on the first tree, and $d_2(i,j)$ is the distance between i and j on the second tree.

Chiaki would like to know the inner product of A and B. By the way, the inner product of two arrays $a = [a_1, a_2, \ldots, a_m]$ and $b = [b_1, b_2, \ldots, b_m]$ is defined as $\sum\limits_{k=1}^{m} a_k b_k$.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^5$) — the number of vertices in each tree.

Each of the next $(n-1)$ lines contains three integers u_i, v_i and w_i ($1 \le u_i, v_i \le n$, $1 \le w_i \le 10^9$) — an edge of length w_i between vertices u_i and v_i on the first tree.

Each of the next $(n-1)$ lines contains three integers u_i, v_i and w_i ($1 \le u_i, v_i \le n$, $1 \le w_i \le 10^9$) — an edge of length w_i between vertices u_i and v_i on the second tree.

It is guaranteed that the sum of n in all test cases will not exceed 10^5.

Output

For each test case, output an integer in a single line, denoting the inner product of A and B modulo $(10^9 + 7)$.

Example

standard input	standard output
1 2 1 2 3 1 2 4	24

Problem I. Counting Polygons

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

Mr. Panda loves counting polygons. Today, he wants to count the number of non-isomorphic convex polygons with m sides such that the length of each side is a positive integer, the perimeter is exactly n and no two sides are collinear.

Mr. Panda thinks such a polygon could be described by the sequence $[l_1, l_2, \ldots, l_m]$ — lengths of its sides in some order. A polygon may have several describing sequences. Each describing sequence may start at any side of the polygon and go through all the sides in clockwise or counterclockwise such that each side appears exactly once.

Mr. Panda calls two convex polygons P and Q isomorphic if and only if there exists a describing sequence $A = [a_1, a_2, \ldots, a_x]$ of P and a describing sequence $B = [b_1, b_2, \ldots, b_y]$ of Q such that $x = y$ and $a_i = b_i$ for $i = 1, 2, \ldots, x$.

Could you please help Mr. Panda count the number of these polygons? To avoid huge output data, you are only asked the answer modulo $(10^9 + 7)$.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^4$), indicating the number of test cases. For each test case:

The first line contains two integers n and m ($3 \le m \le n \le 10^7$).

Output

For each test case, output the number of different polygons modulo $(10^9 + 7)$ in a single line.

Example

standard input	standard output
4 3 3 4 3 5 3 7 4	1 0 1 3

Note

For the third sample case, there is only one type of convex polygon, whose describing sequence can be $[1, 2, 2]$, or $[2, 1, 2]$, or $[2, 2, 1]$.

For the last sample case, there are three types of convex polygons, whose describing sequences can be $[1, 2, 2, 2]$, $[1, 1, 2, 3]$ and $[1, 2, 1, 3]$ respectively.

Note that a polygon with describing sequence $[1, 2, 1, 3]$ and a polygon with describing sequence $[1, 3, 1, 2]$ are isomorphic.

Problem J. Square Graph

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Prof. Elephant has a sequence $a_1, a_2, \ldots, a_n$. He has used the sequence to generate an undirected graph G with n vertices labeled by $1, 2, \ldots, n$.

For each even-length contiguous subsequence $a_l, a_{l+1}, \ldots, a_{l+2k-1}$, if $a_{l+i-1} = a_{l+k+i-1}$ always holds for $i = 1, 2, \ldots, k$, Prof. Elephant would add k edges to G, where the endpoints of the i-th edge are vertices labeled by $(l + i - 1)$ and $(l + k + i - 1)$, and its weight is w_k.

Prof. Elephant would like to know the total weight of the minimum spanning forest of G.

Input

There are multiple test cases. The first line of the input contains an integer T $(1 \le T \le 10^4)$, indicating the number of test cases. For each test case:

The first line contains an integer n $(2 \le n \le 3 \times 10^5)$.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le n)$.

The third line contains $\lfloor \frac{n}{2} \rfloor$ integers $w_1, w_2, \ldots, w_{\lfloor \frac{n}{2} \rfloor}$ $(1 \le w_i \le 10^9)$.

It is guaranteed that the sum of n in all test cases will not exceed 3×10^5.

Output

For each test case, output an integer in a single line, denoting the total weight of the minimum spanning forest of G.

Example

standard input	standard output
1 8 2 2 5 6 2 5 6 2 5 1 4 4	21

Problem K. Three Dimensions

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Let's define a strange "distance" between two lattice points $a = (x_a, y_a, z_a)$ and $b = (x_b, y_b, z_b)$ in three-dimensional space:

$$d(a, b) = \max\{|x_a - x_b|, |y_a - y_b|, |z_a - z_b|\} \oplus x_a \oplus y_a \oplus z_a \oplus x_b \oplus y_b \oplus z_b,$$

where $\max\{S\}$, $|x|$ and $\oplus$ correspond to the maximum value in S, the absolute value of x and the bitwise exclusive-or operator respectively.

Given six non-negative integers mx_a, my_a, mz_a, mx_b, my_b, mz_b, please calculate the sum of $d(a, b)$ for all lattice points a and b meeting the conditions that $x_a \in [0, mx_a]$, $y_a \in [0, my_a]$, $z_a \in [0, mz_a]$ and $x_b \in [0, mx_b]$, $y_b \in [0, my_b]$, $z_b \in [0, mz_b]$. Since the sum may be very large, please output it modulo 2^{30}.

Note that x_a, y_a, z_a, x_b, y_b, z_b should all be integers.

Input

The input only contains six non-negative integers mx_a, my_a, mz_a, mx_b, my_b, mz_b, each of which is not larger than 10^9.

Output

Output an integer denoting the sum modulo 2^{30}.

Example

standard input	standard output
`3 2 1 2 1 3`	`778`