Problem A. RPG

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

To play the new role-playing game "Crazy Elves" more effectively, you have to understand two notions, a *skill point* and a *special command*. A character can boost up by accumulating his experience points. When a character boosts up, he can gain skill points.

You can arbitrarily allocate the skill points to the character's skills to enhance the character's abilities. If skill points of each skill meets some conditions simultaneously (e.g., the skill points of some skills are are greater than or equal to the threshold values and those of others are less than or equal to the values), the character learns a special command.

One important thing is that once a character learned the command, he will never forget it. And once skill points are allocated to the character, you cannot revoke the allocation. In addition, the initial values of each skill is 0.

The system is so complicated that it is difficult for ordinary players to know whether a character can learn all the special commands or not. Luckily, in the «real» world, you are a great programmer, so you decided to write a program to tell whether a character can learn all the special commnads or not. If it turns out to be feasible, you will be more absorbed in the game and become happy.

Input

The first line of the input contains two integers (M, N), where M is the number of special commands ($1 \le M \le 100$), N is the number of skills ($1 \le N \le 100$). All special commands and skills are numbered from 1.

Then M set of conditions follows. The first line of a condition set contains a single integer K_i ($0 \le K_i \le 100$), where K_i is the number of conditions to learn the i-th command. The following K_i lines describe the conditions on the skill values. $s_{i,j}$ is an integer to identify the skill required to learn the command. $cond_{i,j}$ is given by string "`<=`" or "`>=`". If $cond_{i,j}$ is "`<=`", the skill point of $s_{i,j}$-th skill must be less than or equal to the threshold value $t_{i,j}$ ($0 \le t_{i,j} \le 100$).

Otherwise, i.e. if $cond_{i,j}$ is "`>=`", the skill point of $s_{i,j}$ must be greater than or equal to $t_{i,j}$.

Output

Output "`Yes`" if a character can learn all the special commands in given conditions, otherwise print "`No`".

Examples

Standard input	Standard output
2 2 2 1 >= 3 2 <= 5 2 1 >= 4 2 >= 3	Yes
2 2 2 1 >= 5 2 >= 5 2 1 <= 4 2 <= 3	Yes
2 2 2 1 >= 3 2 <= 3 2 1 <= 2 2 >= 5	No
1 2 2 1 <= 10 1 >= 15	No
5 5 3 2 <= 1 3 <= 1 4 <= 1 4 2 >= 2 3 <= 1 4 <= 1 5 <= 1 3 3 >= 2 4 <= 1 5 <= 1 2 4 >= 2 5 <= 1 1 5 >= 2	Yes

Problem B. Integer in integer

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

Given an integer interval $[A, B]$ and an integer C, your job is to calculate the number of occurrences of C as string in the interval.

For example, 33 appears in 333 twice, and appears in 334 once. Thus the number of occurrences of 33 in $[333, 334]$ is 3.

Input

The input file is given by a line with the three integers, A, B, C ($0 \le A \le B \le 10^{10000}$, $0 \le C \le 10^{500}$).

Output

Print the number of occurrences of $C \bmod 10^9 + 7$.

Examples

Standard input	Standard output
1 3 2	1
333 334 33	3
0 10 0	2

Problem C. Card Game

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

Three animals Frog, Kappa (Water Imp) and Weasel are playing a card game.

In this game, players make use of three kinds of items: a guillotine, twelve noble cards and six action cards. Some integer value is written on the face of each noble card and each action card. Before stating the game, players put twelve noble cards in a line on the table and put the guillotine on the right of the noble cards. And then each player are dealt two action cards. Here, all the noble cards and action cards are opened, so the players share all the information in this game.

Next, the game begins. Each player takes turns in turn. Frog takes the first turn, Kappa takes the second turn, Weasel takes the third turn, and Frog takes the fourth turn, etc. Each turn consists of two phases: action phase and noble phase in this order.

In action phase, if a player has action cards, he may use one of them. When he uses an action card with value on the face y, y-th (1-based) noble card from the front of the guillotine on the table is moved to in front of the guillotine, if there are at least y noble cards on the table. If there are less than y cards, nothing happens. After the player uses the action card, he must discard it from his hand.

In noble phase, a player just remove a noble card in front of the guillotine. And then the player scores x points, where x is the value of the removed noble card.

The game ends when all the noble cards are removed.

Each player follows the following strategy:

Each player assume that other players would follow a strategy such that their final score is maximized. Actually, Frog and Weasel follows such strategy. However, Kappa does not. Kappa plays so that Frog's final score is minimized. Kappa knows that Frog and Weasel would play under "wrong" assumption: They think that Kappa would maximize his score. As a tie-breaker of multiple optimum choises for each strategy, assume that players prefer to keep as many action cards as possible at the end of the turn.

If there are still multiple optimum choises, players prefer to maximuze the total value of remaining action cards.

Your task in this problem is to calculate the final score of each player.

Input

The first line of the input contains twelve integers x_{12}, x_{11}, ..., x_1 ($-2 \le x_i \le 5$). x_i is integer written on i-th noble card from the guillotine. Following three lines contains six integers y_1, ..., y_6 ($1 \le y_j \le 4$). y_1, y_2 are values on Frog's action cards, y_3, y_4 are those on Kappa's action cards, and y_5, y_6 are those on Weasel's action cards.

Output

Print three space-separated integers: the final scores of Frog, Kappa and Weasel.

Example

Standard input	Standard output
3 2 1 3 2 1 3 2 1 3 2 1 1 1 1 1 1 1	4 8 12
4 4 4 3 3 3 2 2 2 1 1 1 1 2 2 3 3 4	8 10 12
0 0 0 0 0 0 0 -2 0 -2 5 0 1 1 4 1 1 1	-2 -2 5

Problem D. Epidemy (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	3 seconds
Memory limit:	256 mebibytes

The government has declared a state of emergency due to the MOFU syndrome epidemic in your country. Persons in the country suffer from MOFU syndrome and cannot get out of bed in the morning. You are a programmer working for the Department of Health. You have to take prompt measures.

The country consists of n islands numbered from 1 to n and there are ocean liners between some pair of islands. The Department of Health decided to establish the quarantine stations in some islands and restrict an infected person's moves to prevent the expansion of the epidemic. To carry out this plan, there must not be any liner such that there is no quarantine station in both the source and the destination of the liner. The problem is the department can build at most K quarantine stations due to the lack of budget.

Your task is to calculate whether this objective is possible or not. And if it is possible, you must calculate the minimum required number of quarantine stations.

Input

The input file starts with a line containing three integers N ($2 \le N \le 3{,}000$), M ($1 \le M \le 30{,}000$) and K ($1 \le K \le 32$). Each line in the next M lines contains two integers a_i ($1 \le a_i \le N$) and b_i ($1 \le b_i \le N$). This represents i-th ocean liner connects island a_i and b_i. You may assume $a_i \neq b_i$ for all i, and there are at most one ocean liner between all the pairs of islands.

Output

If there is no way to build quarantine stations that satisfies the objective, print "`Impossible`" (without quotes). Otherwise, print the minimum required number of quarantine stations.

Examples

Standard input	Standard output
3 3 2 1 2 2 3 3 1	2
3 3 1 1 2 2 3 3 1	Impossible
7 6 5 1 3 2 4 3 5 4 6 5 7 6 2	4
10 10 10 1 2 1 3 1 4 1 5 2 3 2 4 2 5 3 4 3 5 4 5	4

Problem E. MST (Division 1 Only!)

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

You are given an undirected weighted graph G with n nodes and m edges. Each edge is numbered from 1 to m.

Let G_i be an graph that is made by erasing i-th edge from G. Your task is to compute the cost of minimum spanning tree in G_i for each i.

Input

The first line of the input contains two integers n ($2 \leq n \leq 10^5$) and m ($1 \leq m \leq 2 \cdot 10^5$). n is the number of nodes and m is the number of edges in the graph. Then m lines follow, each of which contains a_i ($1 \leq a_i \leq n$), b_i ($1 \leq b_i \leq n$) and w_i ($0 \leq w_i \leq 10^6$). This means that there is an edge between node a_i and node b_i and its cost is w_i. It is guaranteed that the given graph is simple: That is, for any pair of nodes, there is at most one edge that connects them, and $a_i \neq b_i$ for all i.

Output

Print the cost of minimum spanning tree in G_i for each i, in m line. If there is no spanning trees in G_i, print -1 instead.

Examples

Standard input	Standard output
4 6 1 2 2 1 3 6 1 4 3 2 3 1 2 4 4 3 4 5	8 6 7 10 6 6
4 4 1 2 1 1 3 10 2 3 100 3 4 1000	1110 1101 1011 -1
7 10 1 2 1 1 3 2 2 3 3 2 4 4 2 5 5 3 6 6 3 7 7 4 5 8 5 6 9 6 7 10	27 26 25 29 28 28 28 25 25 25
3 1 1 3 999	-1

Problem F. Molecules (Division 1 Only!)

Input file:	`Standard input`
Output file:	`Standard output`
Time limit:	8 seconds
Memory limit:	256 mebibytes

You work for invention center as a part time programmer. This center researches movement of protein molecules. It needs how molecules make clusters, so it will calculate distance of all pair molecules in grid map and will make a histogram.

You are given each molecule positions, please calculate histogram of all pair distance.

Input

First line contains the grid map size N ($1 \le N \le 1024$). Each next N line contains N numbers. Each numbers C_{xy} ($0 \le C_{xy} \le 9$) means the number of molecule in position (x, y). There are at least 2 molecules.

Output

Print an average distance of all pair molecules on first line. Next, print histogram of all pair distance. Each line contains a distance and the number of pair molecules of the distance. The distance should be calculated by squared Euclidean distance and sorted as increasing order. If the number of different distance is more than 10^4, please show the first 10^4 lines. You should not print a distance of pair molecules of the distance which don't exist. The answer may be printed with an arbitrary number of decimal digits, but may not contain an absolute or relative error greater than or equal to 10^{-8}.

Example

Standard input	Standard output
2 1 0 0 1	1.4142135624 2 1
3 1 1 1 1 1 1 1 1 1	1.6349751825 1 12 2 8 4 6 5 8 8 2
5 0 1 2 3 4 5 6 7 8 9 1 2 3 2 1 0 0 0 0 0 0 0 0 0 1	1.8589994382 0 125 1 379 2 232 4 186 5 200 8 27 9 111 10 98 13 21 16 50 17 37 18 6 20 7 25 6

Problem G. Carrier Pigeon (Division 1 Only!)

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

Flora is a freelance carrier pigeon. Since she is an excellent pigeon, there are too much task requests to her. It is impossible to do all tasks, so she decided to outsource some tasks to Industrial Carrier Pigeon Company.

There are N cities numbered from 0 to $N-1$. The task she wants to outsource is carrying f freight units from city s to city t. There are M pigeons in the company. The i-th pigeon carries freight from city s_i to t_i, and carrying cost of u units is ua_i if u is smaller than or equals to d_i, otherwise $d_i a_i + (u - d_i)b_i$. Note that i-th pigeon cannot carries from city t_i to s_i. Each pigeon can carry any amount of freights. If the pigeon carried freight multiple times, the cost is calculated from total amount of freight units he/she carried.

Flora wants to minimize the total costs. Please calculate minimum cost for her.

Input

The input file starts with a line containing five integers N ($2 \le N \le 100$), M ($1 \le M \le 1{,}000$), s ($0 \le s \le N-1$), t ($0 \le t \le N-1$) and f ($1 \le f \le 200$). You may assume $s \ne t$. Each of the next M lines contains five integers s_i ($0 \le s_i \le N-1$), t_i ($0 \le t_i \le N-1$), a_i ($0 \le a_i \le 1{,}000$), b_i ($0 \le b_i \le 1{,}000$) and d_i ($1 \le d_i \le 200$). Each denotes i-th pigeon's information. You may assume at most one pair of a_i and b_i satisfies $a_i < b_i$, all others satisfies $a_i > b_i$.

Output

Print the minimum cost to carry f freight units from city s to city t in a line. If it is impossible to carry, print "`Impossible`" (quotes for clarity).

Examples

Standard input	Standard output
2 2 0 1 5 0 1 3 0 3 0 1 2 1 6	9
4 4 0 3 5 0 1 3 0 3 1 3 3 0 3 0 2 2 1 6 2 3 2 1 6	18
2 1 0 1 1 1 0 1 0 1	Impossible
2 2 0 1 2 0 1 5 1 2 0 1 6 3 1	9
3 3 0 2 4 0 2 3 4 2 0 1 4 1 3 1 2 3 1 1	14

Problem H. Circles

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

There are two circles with radius 1 in 3D space. Please check two circles are connected as chained rings.

Input

First line contains three real numbers($-3 \leq x_i,\ y_i,\ z_i \leq 3$). It shows a circle's center position. Second line contains six real numbers($-1 \leq X_{i,j},\ Y_{i,j},\ Z_{i,j} \leq 1$). A unit vector ($X_{1,1}$, $Y_{1,1}$, $Z_{1,1}$) is directed to the circumference of the circle from center of the circle. The other unit vector ($X_{1,2}$, $Y_{1,2}$, $Z_{1,2}$) is also directed to the circumference of the circle from center of the circle. These two vectors are orthogonalized. Third and fourth lines show the other circle information in the same way of first and second lines. There are no cases that two circles touch.

Output

If two circles are connected as chained rings, you should print "YES". The other case, you should print "NO". (quotes for clarity)

Example

Standard input	Standard output
0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0 1.0 0.0 0.5 1.0 0.0 0.0 0.0 0.0 1.0	YES
0.0 0.0 0.0 1.0 0.0 0.0 0.0 1.0 0.0 0.0 3.0 0.0 0.0 1.0 0.0 -1.0 0.0 0.0	NO
1.2 2.3 -0.5 1.0 0.0 0.0 0.0 1.0 0.0 1.1 2.3 -0.4 1.0 0.0 0.0 0.0 0.70710678 0.70710678	YES
1.2 2.3 -0.5 1.0 0.0 0.0 0.0 1.0 0.0 1.1 2.7 -0.1 1.0 0.0 0.0 0.0 0.70710678 0.70710678	NO

Problem I. Queries

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

You are given N empty arrays, $t_1, \ldots, t_n$. At first, you execute M queries as follows.

- add a value v to array t_i $(a \le i \le b)$;

Next, you process Q following output queries.

- output the j-th number of the sequence sorted all values in t_i $(x \le i \le y)$.

Input

The first line contains three integers N $(1 \le N \le 10^9)$, M $(1 \le M \le 10^5)$ and Q $(1 \le Q \le 10^5)$. Each of the following M lines consists of three integers a_i, b_i and v_i $(1 \le a_i \le b_i \le N,\ 1 \le v_i \le 10^9)$. Finally the following Q lines give the list of output queries, each of these lines consists of three integers x_i, y_i and j_i $(1 \le x_i \le y_i \le N,\ 1 \le j_i \le \sum_{x_i \le k \le y_i} |t_k|)$.

Output

For each output query, print in a line the j-th number.

Examples

Standard input	Standard output
5 4 1 1 5 1 1 1 3 4 5 1 3 4 2 1 3 4	2
10 4 4 1 4 11 2 3 22 6 9 33 8 9 44 1 1 1 4 5 1 4 6 2 1 10 12	11 11 33 44

Note

In the first sample, after the M-th query is executed, each t_i is as follows:

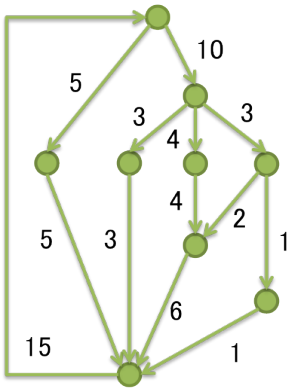
$[1, 3]$, $[1]$, $[1, 2]$, $[1, 1, 2]$, $[1, 1]$

The sequence of sorted values in t_1, t_2 and t_3 is $[1, 1, 1, 2, 3]$. In the sequence, the 4-th number is 2.

Problem J. Restoring the Flows

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

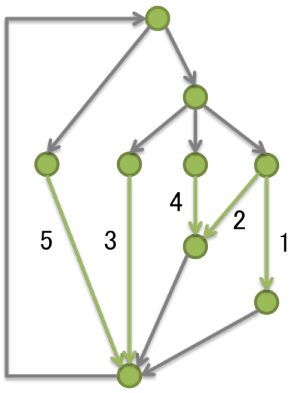
Some directed graphs are given. Each edge $e(u, v)$ has the value of flow from u to v. An example of the graph with the value of flow is shown in the following figure.



This graph is the same as the first case of the sample input.

The value of flow for all edges can be restored, even if you don't store the flow value of some edges. You want to restore the value of flow for all edges.

The first case can be restored from five edges (See the following figure).



Your task is to compute the minumum number of edges to restore all edges in the graph. You can assume that the following conditions are satisfied.

- For each node, the amount of incoming flow is equal to the amount of outgoing flow.
- The flow value is non-negative.

Input

The first line contains two integers N ($1 \leq N \leq 500$) and M ($0 \leq M \leq 3000$) that indicates the number of nodes and edges. The following M lines consists of two integers s_i and t_i, which means that there is a direct edge e (s_i, t_i).

You can assume that the following conditions are satisfied.

- $s_i \neq t_i$.
- If $i \neq j$, then $(s_i \neq s_j)$ or $(t_i \neq t_j)$.
- Input file consists of of some graphs and each graph is strongly connected.

Output

Print answer in one line.

Examples

Standard input	Standard output
9 13 1 2 1 3 2 9 3 4 3 5 3 6 4 9 5 7 6 7 6 8 7 9 8 9 9 1	5
7 9 1 2 1 3 2 4 3 4 4 5 4 6 5 7 6 7 7 1	3
4 4 1 2 2 1 3 4 4 3	2

Problem K. Genome (Division 2 Only!)

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

An alien species has a strange genome with a DNA sequence of only two types of nucleotides: 'A' and 'B'. A valid DNA sequence must also obey the constraint that no two A's can be adjacent in the sequence. For example, "ABBAB" is a valid sequence and "BAABA" is invalid. Such a DNA sequence has a natural orientation, so the reverse of a sequence is not necessarily the same as the original. For example, "ABBAB" and "BABBA" are considered two distinct sequences.

You are to write a program to compute the number of valid alien DNA sequences of a given length n modulo m.

Input

There will be several (20 or less) sets of input. Each set will consist of two integers on a single line. The first number, n ($1 \le n \le 8 \cdot 10^{18}$), represents the length of a DNA sequence. The second number, m ($1 \le m \le 2 \cdot 10^9$), is the modulus for the operation. The input will be terminated by a line with two 0's, which shouldn't be processed.

Output

For each input set, print a single line containing one integer, specifying the number of valid DNA sequences modulo m. There should be no blank lines between outputs.

Example

Standard input	Standard output
1 10 2 10 5 10 6 8 0 0	2 3 3 5

Problem L. Encryption (Division 2 Only!)

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

Alice and Bob like to encrypt data. They have a simple encryption scheme. Given a number n (which represents an 8-bit character), they encode it by replacing the number with the n-th prime sum. The n-th prime sum is the sum of all the prime numbers less or equal to nth (0-indexed) prime number.

Your task is to encode input numbers.

Input

The first line contains an integer T ($1 \le T \le 100$), T being the number of test cases. In the next T lines, each line contains an integer n ($0 \le n \le 255$).

Output

For each integer n, output the special code.

Example

Standard input	Standard output
3 0 5 7	2 41 77

Problem M. Points in triangle (Division 2 Only!)

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

A lattice point is an ordered pair (x, y) where x and y are both integers. Given the coordinates of the vertices of a triangle (which happen to be lattice points), you are to count the number of lattice points which lie completely inside of the triangle (points on the edges or vertices of the triangle do not count).

Input

The input will contain multiple test cases (maximum of 200 test cases). Each input test case consists of six integers x_1, y_1, x_2, y_2, x_3, y_3, where (x_1, y_1), (x_2, y_2), (x_3, y_3) are the coordinates of vertices of the triangle. All triangles in the input will be non-degenerate (will have positive area), and be in the range $-15000 \le x_1, y_1, x_2, y_2, x_3, y_3 \le 15000$. The end of input is marked by a test case with $x_1 = y_1 = x_2 = y_2 = x_3 = y_3 = 0$ and should not be processed.

Output

For each test case, the program should print the number of internal lattice points on a single line.

Example

Standard input	Standard output
0 0 1 0 0 1 0 0 5 0 0 5 0 0 0 0 0 0	0 6

Problem N. Word Stats (Division 2 Only!)

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

We all know how to crunch statistics with numbers. But, let's crunch some statistics with words! That's your job with this program. You want to input all the words from a text file, convert them to lowercase, sort them lexicographically, then calculate the median of your word distribution, and the mode (the word with the largest frequency). The median is defined as the word in the middle of your sorted distribution. If you have an odd number of words, there is only one median. But, if you have an even number of words, you will have two words that define your median.

The mode is the word that occurs more often than any others. In the event of a tie (multiple words with the same largest frequency), you will print all words with the largest frequency.

Input

Your program will take one or more words as input. The number of words will be not greater, than 500. A word is defined as one or more contiguous characters, containing atleast one upper- or lowercase letter, followed by white space (space, tab, or newline). The length of each word is not greater, than 100. All words should be converted to lowercase upon input, and any non-alphabetic characters that start, end, or are contained within the word should be removed.

Output

Your median should be output first on a single line with a prompt as shown below. If you have an odd number of words in your input, a single median word will be output. If you have an even number of words, you will output two words with a single comma between them, and brackets surrounding them. For example, "`My median=[easter,egg]`" . Your mode should be output on a single line following the median with a prompt as shown below. Include the number of occurrences of the mode in parentheses following. Enclose your answer with brackets like you did for median. In the event of a tie, use a single comma to separate your words and their occurrences. For example, "`My mode=[an(23),the(23)]`".

With ties, all words should be listed in alphabetical order for both the median and mode.

Example

Standard input	Standard output
When April with his sweet showers has pierced the drought of March to the root, and bathed every vein in such moisture as has power to bring forth the flower	My median=[moisture,of] My mode=[the(3)]
a test test1 2test2 3te3st3 4tests 123456	My median=[test,test] My mode=[test(4)]
a test test1 2test2 3te3st3 123456	My median=[test] My mode=[test(4)]