ICPC-LATAM Subregionals

2026

This problem set is used simultaneously in the following contests:
Maratona SBC de Programação 2026
Competencia Boliviana de Programación
The 2026 ICPC Gran Premio de Centroamerica
The 2026 ICPC Gran Premio de Mexico

August 29, 2026

Contest Session

This problem set contains 14 problems; pages are numbered from 1 to 19.

General information

Unless otherwise stated, the following conditions hold for all problems.

Program name

1. Your solution must be called *codename*`.c`, *codename*`.cpp`, *codename*`.java`, *codename*`.kt` or *codename*`.py` (C, C++20, Java, Kotlin and Python 3, respectively), where *codename* is the capital letter which identifies the problem.
2. The extension identifies the language your submission is judged as.

Input

1. The input must be read from standard input.
2. The input consists of a single test case, which is described using a number of lines that depends on the problem. No extra data appear in the input.
3. When a line of data contains several values, they are separated by *single* spaces. No other spaces appear in the input. There are no empty lines.
4. The English alphabet is used. There are no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, has the usual end-of-line mark.

Output

1. The output must be written to standard output.
2. The result of the test case must appear in the output using a number of lines that depends on the problem. No extra data should appear in the output.
3. When a line of results contains several values, they must be separated by *single* spaces. No other spaces should appear in the output. There should be no empty lines.
4. The English alphabet must be used. There should be no letters with tildes, accents, diaereses or other diacritical marks (ñ, Ã, é, Ì, ô, Ü, ç, etcetera).
5. Every line, including the last one, must have the usual end-of-line mark.

Problem A – High Frequency

You have been hired by a Brazilian HFT (*High Frequency Trading*) company, which uses robots to buy and sell stocks on the stock exchange in fractions of a second in a fully systematic and automated way. Your first task is to implement a *signal*: a number, computed from the current state of the market, that guides the robot's decision between buying, selling, or doing nothing. The signal you will compute determines whether, at that moment, there is more buying demand or more selling supply for a stock.

On the exchange, whoever wants to buy or sell stocks registers an order in the order book. The book consists of two queues: buy orders and sell orders. Orders are grouped into price levels, ordered from best to worst. For each level i, you are given c_i (total shares in buy orders) and v_i (total shares in sell orders).

The signal measures the book's imbalance for the first n levels:

$$I_n = \frac{C_n - V_n}{C_n + V_n}, \qquad \text{where } C_n = \sum_{i=1}^{n} c_i \text{ and } V_n = \sum_{i=1}^{n} v_i.$$

The value of I_n always lies between -1 and 1 and is interpreted as follows:

- if $I_n > 0$, the signal indicates `COMPRA`;
- if $I_n < 0$, the signal indicates `VENDA`;
- if $I_n = 0$, the signal indicates `NEUTRO`.

Given the order book and a series of queries, where the j-th query has depth n_j, report the signal corresponding to each query.

Input

The first line contains an integer N $(1 \le N \le 10^5)$, the number of levels in the order book.

Each of the following N lines contains two integers c_i and v_i $(0 \le c_i, v_i \le 10000)$, respectively the total shares in the buy orders and sell orders at level i. It is guaranteed that $c_1 + v_1 > 0$.

The next line contains an integer Q $(1 \le Q \le 10^5)$, the number of queries. Each of the following Q lines contains an integer n_j $(1 \le n_j \le N)$, the depth of the j-th query.

Output

Your program must produce Q lines. The j-th line must contain a single word: `COMPRA` if $I_{n_j} > 0$, `VENDA` if $I_{n_j} < 0$, or `NEUTRO` if $I_{n_j} = 0$.

Sample input 1	Sample output 1
4 10 2 1 9 0 7 8 1 4 1 2 3 4	COMPRA NEUTRO VENDA NEUTRO

Sample input 2	Sample output 2
1 5 3 2 1 1	COMPRA COMPRA

Problem B – Sequences

You are given N integer intervals $[L_i, R_i]$, in a fixed order. For each index i, exactly one integer x_i belonging to the i-th interval must be chosen, that is, $L_i \leq x_i \leq R_i$.

A choice is valid if the sequence $x_1, x_2, \ldots, x_N$ is strictly increasing, that is, if $x_1 < x_2 < \cdots < x_N$. Two choices are considered distinct if they differ in at least one value x_i.

Compute the number of valid choices, modulo $10^9 + 7$.

Input

The first line contains an integer N $(1 \leq N \leq 300)$.

Each of the following N lines contains two integers L_i and R_i $(1 \leq L_i \leq R_i \leq 10^9)$, describing the i-th interval.

Output

Print a single integer: the number of valid choices, modulo $10^9 + 7$.

Sample input 1	Sample output 1
3 2 4 1 4 1 5	4

Explanation of sample 1:

The four valid sequences are $(2, 3, 4)$, $(2, 3, 5)$, $(2, 4, 5)$, and $(3, 4, 5)$. Therefore, the answer is 4.

Problem C – Exchange Rate

Robertinho has been considering buying another keyboard, one of his favorite objects. Since he is going to travel abroad and knows that some products can cost much less outside the country, he wants help deciding whether it is worth buying the new keyboard there.

In Brazil, where he lives, the keyboard costs C reais, the local currency. In Nlogônia, the country he will visit, the same keyboard costs K nlogarian dollars. Given that the exchange rate is T reais per nlogarian dollar, after conversion, the cost of the purchase abroad is equivalent to $K \times T$ reais.

Given the information about the current exchange rate T, the cost in Brazil C, and the cost abroad K, determine the smallest amount (in reais) that Robertinho can pay for the product.

Input

The input consists of a single line containing three integers T, C, and K ($1 \leq T, C, K \leq 1000$), representing, respectively, the exchange rate, the price of the product in Brazil, and the price of the product abroad in the foreign currency.

Output

Print a single line containing an integer: the smallest amount in reais that Robertinho can pay for the product.

Sample input 1	**Sample output 1**
`5 30 5`	`25`

Explanation of sample 1:

In this case, with a conversion rate of $T = 5$ reais per nlogarian dollar, the purchase options are:

- Buy in Brazil: $C = 30$ reais;
- Buy abroad: $K \times T = 5 \times 5 = 25$ reais.

Therefore, the smallest amount Robertinho can pay is 25 reais.

Sample input 2	**Sample output 2**
`10 10 2`	`10`

Problem D – Dragons in Harmony

The dragon keepers of Dragonland are reorganizing a large rectangular plot of land, divided into square cells of the same size, arranged in N rows and M columns. Some cells contain obstacles, such as mountains and lava pits, while the remaining cells are free spaces, each inhabited by a dragon.

To give the environment a fresh look without disrupting the group's dynamics, the keepers drew the current position of all the dragons on a transparent magic scroll, so that each free cell of the land corresponds to a mark on the scroll. They plan to relocate the dragons using this same scroll as a template. To do this, the keepers may freely manipulate the scroll on the table: they can rotate it in any direction or even flip it over (looking through the back of the transparent sheet).

A new arrangement is considered valid if the scroll aligns perfectly with the original boundaries of the land and if all the dragon marks fall exactly on free cells. Obviously, no dragon can be placed on an obstacle.

Your task is to compute the total number of valid ways to align the scroll to the land. Each distinct physical orientation of the scroll is counted as a different way, even if two orientations make the marks fall on exactly the same cells. The initial orientation, with the scroll neither rotated nor flipped, already counts as one valid way.

Input

The first line of input contains two integers N and M ($1 \le N, M \le 1000$), representing the dimensions of the land.

Each of the following N lines contains M characters, describing the land:

- The character '`.`' indicates free land, where a dragon lives.
- The character '`#`' indicates an obstacle.

Output

Print a single line containing an integer: the number of valid ways to align the scroll to the land.

Sample input 1	Sample output 1

`3 3`
`..#`
`...`
`###`
	`1`

Sample input 2	Sample output 2

`3 3`
`.#.`
`...`
`#.#`
	`2`

Sample input 3	Sample output 3

`2 4`
`....`
`....`
	`4`

Problem E – Network Editing

The company Sistemas Brasileiros de Conectividade (SBC) manages a network formed by N stations, numbered from 1 to N. Each connection in the network is a dedicated link formed by two terminals, one at each connected station. Connections are bidirectional, and there is at most one connection between each pair of distinct stations.

To modernize its infrastructure, SBC has prepared a desired configuration for the network. For cost reasons, during the reconfiguration the company neither buys nor discards terminals, but reuses the existing links. Furthermore, its field team is small and can only relocate one terminal per intervention.

In each intervention, the team chooses a connection between two stations u and v, keeps one of the terminals at station v, and moves the other terminal from station u to a station w. As a result, the connection between u and v is removed and a connection between v and w is installed. Stations u, v, and w must be distinct, and the connection between v and w must not exist before the intervention.

Given the current and desired configurations of the network, determine the minimum number of interventions needed to transform the first into the second, or report that this transformation is impossible.

Input

The first line contains three integers N, M_1, and M_2 ($2 \leq N \leq 500$, $0 \leq M_1, M_2 \leq 500$), representing the number of stations, the number of connections in the current configuration, and the number of connections in the desired configuration, respectively.

Each of the following M_1 lines contains two integers a and b ($1 \leq a, b \leq N$, $a \neq b$), indicating that there is a connection between stations a and b in the current configuration.

Each of the following M_2 lines contains two integers a and b ($1 \leq a, b \leq N$, $a \neq b$), indicating that there is a connection between stations a and b in the desired configuration.

In each configuration, each pair of stations appears in at most one line.

Output

Print a single line containing an integer corresponding to the minimum number of interventions needed to obtain the desired configuration, or -1 if this is impossible.

Sample input 1	**Sample output 1**
5 2 2 1 2 2 3 1 3 4 5	3

Explanation of sample 1:

An optimal sequence consists of replacing, in this order, connection 1–2 with 1–3, connection 2–3 with 2–4, and connection 2–4 with 4–5.

It is not possible to use only two interventions, since the connection between 4 and 5 does not share a station with any of the connections in the initial configuration.

Sample input 2	Sample output 2

`4 1 2`
`1 2`
`1 3`
`2 4`
	`-1`

Sample input 3	Sample output 3

`4 5 5`
`1 2`
`1 3`
`1 4`
`2 3`
`2 4`
`1 3`
`1 4`
`2 3`
`2 4`
`3 4`
	`2`

Explanation of sample 3:

First, the connection between 1 and 3 can be removed and the connection between 3 and 4 installed. Then, the connection between 1 and 2 is removed and the connection between 1 and 3 is reinstalled.

A single intervention is not enough, since the only connections that differ between the configurations, 1–2 and 3–4, do not share any station.

Problem F – Flower

In the serene waters of the Amazon, riverside legends tell of a magical Victoria water lily. Unlike ordinary flowers, this one reacts to offerings. When seeds are placed in its center, the flower gently closes its petals and dives into the river. Moments later, it resurfaces and blooms, returning a strictly greater quantity of seeds.

This flower's magic does not act at random; it applies a deterministic, unchanging rule to compute the return. The flower keeps no memory of previous offerings: the amount returned depends only on the quantity of seeds deposited in that particular offering. The sages of the forest guarantee that the transformation obeys the following natural laws, which together guarantee a single possible outcome for each offering:

- The Law of the Whole Seed: The flower operates exclusively with integer quantities of seeds.
- The Justice of the Waters: Strictly larger offerings always result in strictly larger returns.
- Moapy: In the language of the ancients, the word means "three". It describes the flower's supreme secret. If you take all the seeds returned by an initial offering and deposit them into the flower again, the final amount returned will be exactly three times that initial offering.

On your journey, you found the rare Victoria water lily and deposited exactly N seeds into it. The flower closed, sank, and bloomed exactly once (not twice!). However, the snap of a branch in the woods startled you, causing you to quickly stash the return in your pocket without having time to count it. Knowing the exact amount you offered at the start, you need to figure out how many seeds you now have with you.

Input

The input contains a single integer N ($1 \leq N \leq 10^7$), representing the initial quantity of seeds offered to the flower.

Output

Print a single integer representing the quantity of seeds the flower returned after blooming exactly once.

Sample input 1	Sample output 1
1	2

Sample input 2	Sample output 2
2	3

Sample input 3	Sample output 3
13	22

Problem G – Genes

A DNA sequence is represented by a string T. Several genes are also known, represented by strings and ordered from most relevant to least relevant. The genes are numbered sequentially, starting from 1.

Q queries are made. Each one delimits a segment of T, between indices L_i and R_i, inclusive. For each query, the goal is to determine the index of the most relevant gene that occurs entirely as a substring of that segment.

Input

The first line of input contains the string T ($1 \leq |T| \leq 10^5$). The second line contains the number G of genes ($1 \leq G \leq 5 \times 10^5$). Each of the following G lines contains a gene, from most to least relevant, one per line. Each gene is a non-empty string. It is guaranteed that the genes are distinct and that the sum S of their lengths is at most 5×10^5.

The next line contains the number of queries Q ($1 \leq Q \leq 10^5$). Each of the following Q lines contains two integers L_i and R_i ($1 \leq L_i \leq R_i \leq |T|$), representing the inclusive bounds of the query.

All strings contain only the letters `A`, `C`, `G`, and `T`.

Output

For each query, print the index of the most relevant gene that occurs entirely within the corresponding segment of the DNA. If no gene occurs in that segment, print -1.

Sample input 1	Sample output 1
ACAGACA 3 ACA G CA 3 1 7 2 6 1 2	1 2 -1

Explanation of sample 1:

In the first query, the segment is the entire sequence `ACAGACA`. The most relevant gene that occurs in it is `ACA` (index 1).

In the second query, the segment is `CAGAC`. The most relevant gene that occurs in it is `G` (index 2).

In the third query, the segment is `AC`. No gene occurs in this segment, so the answer is -1.

Sample input 2	Sample output 2
ACGT 2 ACGT GT 2 1 3 2 4	-1 2

Problem H – Traffic Lights

A street has N traffic lights, numbered from 1 to N. Each traffic light can be in one of two states: closed, represented by 1, or open, represented by 0.

To prepare the street for a special operation, M control devices have been installed. Each device acts on a contiguous stretch of the street: when activated, it toggles the state of every traffic light within its range. Thus, closed traffic lights are opened, and open traffic lights are closed.

The i-th device affects the traffic lights whose indices are between L_i and R_i, inclusive. Each device can be activated as many times as needed.

Is it possible to activate the devices so that all traffic lights end up open?

Input

The first line contains two integers N and M ($1 \leq N, M \leq 10^5$), representing, respectively, the number of traffic lights and the number of control devices.

The second line contains N integers $v_1, v_2, \ldots, v_N$ ($v_i \in \{0, 1\}$), representing the initial state of the traffic lights.

Each of the following M lines describes a device. The i-th of these lines contains two integers L_i and R_i ($1 \leq L_i \leq R_i \leq N$), indicating that device i affects all traffic lights between L_i and R_i, inclusive.

Output

Print `YES` if it is possible to leave all traffic lights open, or `NO` otherwise.

Sample input 1	Sample output 1
5 4 0 1 1 0 1 3 5 3 3 1 3 2 4	YES

Problem I – Inside the Guinea Pig Playpen

Jaime, a friendly guinea pig, and their clever friend Churro, also a guinea pig, have won the International Cavy Programming Contest. To celebrate their victory, they are organizing a party. They have F other friends, numbered from 1 to F, and sent invitations to N of them (due to some budget limitations). Each invitation may cover several guests, such as the invited friend's family.

Each invited friend either accepted the invitation and specified their group's schedule, declined the invitation, or gave a tentative response that depends on whether another friend attends.

If a tentative response depends on a friend who declined, on a friend who was not invited, or on a circular chain of tentative responses, that friend will not attend.

Jaime and Churro will prepare a party playpen for the invited groups. At the minute when a group leaves, it no longer occupies the playpen, so its space can be used by groups arriving at that same minute. Determine the minimum guest capacity required for the playpen to accommodate all attending groups, excluding Jaime and Churro.

Input

The first line contains two integers F and N ($1 \leq F \leq 10^9$, $1 \leq N \leq \min(10^5, F)$), the total number of their friends and the number of invitations, respectively.

Each of the next N lines contains two integers k_i and p_i ($1 \leq k_i \leq F$, $1 \leq p_i \leq 10^5$). These values indicate that the i-th invitation was sent to friend k_i and covers a group of exactly p_i guests, including friend k_i. All values k_i are distinct.

Each of the next N lines contains the response to the corresponding invitation, in the same order as the invitations. A response has one of the following forms:

- `A` a_i t_i ($1 \leq a_i \leq 10^9$, $1 \leq t_i \leq 10^9$): friend k_i accepted; their group will arrive at minute a_i, stay for t_i minutes, and leave at minute $a_i + t_i$.
- `D`: friend k_i declined.
- `T` x_i ($1 \leq x_i \leq F$): friend k_i will attend if and only if friend x_i attends; in that case, both groups will arrive and leave at the same times.

Output

Print one integer: the minimum guest capacity required for the party playpen.

Sample input 1	Sample output 1
10 5 1 1 2 2 3 3 7 1 9 1 A 1 3 D T 2 T 1 T 7	3

Explanation of sample 1:

Friend 1 accepted the invitation, which covers only that friend. Friend 2 declined, so friend 3, whose response depends on friend 2, does not attend.

Friend 7 depends on friend 1, and friend 9 depends on friend 7. Therefore, friends 1, 7, and 9 all attend from minute 1 to minute 4. Their invitations cover 1 guest each, so the maximum number of guests in the playpen at the same time is 3.

Problem J – Phone Lines

Phone companies have recently changed the way they operate. Since you and your two teammates need to train together for online programming marathons, you want to calculate how much you will spend to contact them by phone.

There are N phone companies in total, numbered from 1 to N. Each company U may have multiple **directed** communication lines. Each such line starts at company U and reaches every company in an interval $[L, R]$. The communication cost varies as follows: the cost to connect company U to company L is X, to company $L+1$ is $X+K$, to company $L+2$ is $X+2\times K$, and so on, up to company R, for which the cost is $X+(R-L)\times K$. More formally, the cost to connect company U to company i ($L \leq i \leq R$) is $X+(i-L)\times K$.

The cost of a path is the sum of the costs of the connections that form it. The cost of a phone call from a company U to another company V is the smallest cost among all paths from company U to company V. A path may use no communication line at all; therefore, the cost of a call from a company to itself is 0.

You use the services of company 1, while your two teammates use the services of companies A and B. You need to calculate the minimum total cost to call both of your teammates, that is, the sum of the minimum costs of the calls from company 1 to companies A and B. You can only train together if it is possible to call both teammates.

Input

The first line contains four integers N, M, A, and B ($1 \leq N \leq 10^5$, $1 \leq M \leq 10^5$, $1 \leq A, B \leq N$), representing, respectively, the number of companies, the number of communication lines, and the company numbers of your two teammates. Companies A and B may be equal to each other and may also be equal to company 1.

Each of the following M lines contains five integers U, L, R, X, and K ($1 \leq U \leq N$, $1 \leq L \leq R \leq N$, $0 \leq X \leq 10^{13}$, $-10^8 \leq K \leq 10^8$), describing a communication line that starts at company U and reaches the companies in the interval $[L, R]$. It is guaranteed that $X+(R-L)\times K \geq 0$.

It is possible that U belongs to the interval $[L, R]$.

Output

Print a single line containing an integer representing the minimum total cost to call both teammates. If it is not possible to call at least one of them, print -1.

Sample input 1	Sample output 1
8 4 2 8 1 4 6 1 1 4 2 3 2 2 2 6 8 0 5 6 7 8 6 1	13

Explanation of sample 1:

The smallest cost to call from company 1 to company 2 is 3, via the path $1 \to 4 \to 2$, whose connections cost 1 and 2. The smallest cost to call from company 1 to company 8 is 10, via the path $1 \to 6 \to 8$, whose connections cost 3 and 7.

Therefore, the total cost to call both teammates is $3+10=13$.

Sample input 2	Sample output 2
8 4 3 7 1 4 7 10 -2 6 5 8 0 1 4 1 2 3 0 3 1 5 4 6	-1

Explanation of sample 2:

In this case, there is no path from company 1 to company 3, so the answer is -1.

Sample input 3	Sample output 3
6 3 5 2 1 2 6 2 3 3 5 6 1 0 6 5 5 1 1	8

Sample input 4	Sample output 4
1 1 1 1 1 1 1 0 0	0

Explanation of sample 4:

Both teammates use the services of company 1. Since the minimum cost to call from company 1 to itself is 0, the total cost is 0.

Problem K – Contingency

A large automaker located in the Manaus Industrial Pole (PIM) manages a vast warehouse with N different types of electronic components. The primary parts-picking system is efficient, with autonomous robots that fetch the needed components from the warehouse, based on optical sensors that identify the part type, and carry them to the main sorting conveyor, where they are sorted and sent to the assembly line.

Unfortunately, sensor failures can occur. In such situations, the factory relies on a contingency protocol: emergency robots remove components from the warehouse one at a time, but completely "blindly", without identifying which type of component they are picking up until it reaches the main sorting conveyor. As a result, in some situations, the robots may remove more units of certain components than necessary.

The factory has production targets that require K_i units of component type i. Currently, the stock available in the warehouse has C_i units of component i.

The logistics manager needs to know: in the worst possible scenario, what is the minimum number of components the robots must remove from the warehouse to guarantee that all the assembly line's requirements are fully met? If the stock is insufficient to guarantee that all targets can be met, the system must signal the impossibility of the operation by printing -1.

Input

The first line contains an integer N ($1 \le N \le 2 \times 10^5$), the number of component types.

The second line contains N integers $C_1, C_2, \ldots, C_N$ ($1 \le C_i \le 10^4$), representing the available stock quantity of each component.

The third line contains N integers $K_1, K_2, \ldots, K_N$ ($1 \le K_i \le 10^4$), representing the required target for each component.

Output

Print a single line containing the minimum number of removals needed in the worst case to guarantee all targets, or -1 if it is impossible to guarantee that the targets can be met with the current stock.

Sample input 1	Sample output 1
3 10 20 30 2 5 10	52

Explanation of sample 1:

Note that, acting blindly, the robots could remove all the components of types 2 and 3 before removing any unit of the first component. In this case, 52 components would be removed before the minimum required quantity of all of them is reached. Analyzing all the possibilities, it is possible to verify that this is the worst case.

Sample input 2	Sample output 2
2 5 5 1 10	-1

Problem L – Maze

Yan is starting a game company! His first creation is a maze puzzle game, where the player needs to reach the target cell in the smallest possible number of steps. The maze is represented by an $N \times M$ grid and has several types of cells:

- Free cells (`.`): cells one can freely walk through.
- Walls (`#`): blocked cells that cannot be walked through.
- Starting position (`*`): where the player begins.
- Final position (`F`): the goal to be reached.
- Diagonal walls (\ or `/`): cells with a wall connecting two opposite corners. It is always possible to enter a cell of this type, but one cannot cross the diagonal wall, which restricts which neighboring cells can be walked to next.

  More specifically, upon entering a cell with a diagonal wall, the possible moves depend on the wall's orientation:

  - In a \ cell, it is possible to move between up and right (`U` $\leftrightarrow$ `R`) or between down and left (`D` $\leftrightarrow$ `L`).
  - In a `/` cell, it is possible to move between up and left (`U` $\leftrightarrow$ `L`) or between down and right (`D` $\leftrightarrow$ `R`).
- Switches (`X`): for each diagonal wall, there is exactly one corresponding switch that, when triggered, flips the orientation of that wall (\ becomes `/` and vice versa). A switch can only be triggered if you are standing on its cell.

The k-th switch `X` (in reading order, row by row from left to right) is associated with the k-th diagonal wall (also in reading order).

It is allowed to move in the four cardinal directions (`U`, `D`, `L`, `R`) or trigger the switch on the current cell (`X`). Each of these actions counts as one move.

Yan wants the final stage to be extremely challenging! Help him by creating a solvable maze of dimensions $N \times M$ ($1 \leq N, M \leq 100$) containing at most 10 diagonal walls, such that the minimum number of moves needed to solve it is at least 10^5 steps.

Input

This problem has no input.

Output

On the first line, print two integers N and M ($1 \leq N, M \leq 100$).

On each of the following N lines, print a string of M characters representing each row of the maze.

The maze must contain exactly one `*` cell and exactly one `F` cell. The number of switches must equal the number of diagonal walls, and this number must not exceed 10.

The maze must be solvable, and the minimum number of moves needed to reach the final position must be at least 10^5.

Any maze that satisfies all the conditions will be accepted.

Problem M – Microwave

The company Charliebot trades several high-frequency investment strategies on the SBC (Sociedade de Bolsas Computadorizadas). The company cares a great deal about network latency, since the lower it is, the higher the expected return of the strategies.

The fiber optic network currently in use has been mapped into N points of presence and M bidirectional links. Charliebot's buy and sell signals originate at its data center (point 1) and need to reach the SBC's co-location facility (point N). For each pair (i, j) of directly connected points of presence, the latency f_{ij} in microseconds has been measured.

The network provider has recently installed a bidirectional microwave network between some points of presence and measured its latency as w_{ij} microseconds. This introduced a new option to reduce latency. However, this network is more expensive, and Charliebot's budget allows hiring at most K microwave links.

Since a microwave link may need to go around the terrain where it was installed, there is no guarantee that $w_{ij} < f_{ij}$. Furthermore, the network provider guarantees that it will always be possible to find a path in the network even when the budget for microwave links is 0.

What is the smallest possible end-to-end latency, in microseconds?

Input

The first line contains three integers N, M, and K ($2 \leq N \leq 10^5$, $1 \leq M \leq 2 \times 10^5$, $0 \leq K \leq 10$).

Each of the following M lines contains four integers i, j, f_{ij}, and w_{ij} ($1 \leq i, j \leq N$, $i \neq j$, $1 \leq f_{ij} \leq 10^9$, $w_{ij} = -1$ or $1 \leq w_{ij} \leq 10^9$), describing a direct connection between points of presence i and j with latency f_{ij} on the fiber link and w_{ij} on the microwave channel. If $w_{ij} = -1$, there is no microwave channel on that connection. Each pair of points appears in at most one line.

Output

Print a single line containing an integer corresponding to the minimum latency, in microseconds, between Charliebot's data center and the SBC, using at most K microwave links.

Sample input 1	Sample output 1
4 5 1 1 2 10 3 2 4 10 4 1 3 5 -1 3 4 7 2 1 4 30 8	7

Explanation of sample 1:

Since at most one microwave link is allowed, the best route is to go from point 1 to point 3 via the fiber optic link, with a latency of 5 microseconds, and from point 3 to point 4 via the microwave link, with a latency of 2 microseconds. Thus, the total latency is 7 microseconds.

Sample input 2	Sample output 2
3 3 0 1 2 5 2 2 3 7 1 1 3 20 3	12

Problem N – Nlogônia's Keys

Nlogônia is implementing a new data transmission system between its N research stations, numbered from 1 to N. The network infrastructure is economical: there are exactly $N - 1$ bidirectional cables connecting the stations, so that there is always a single simple path between any pair of stations.

To ensure that messages reach only those who are actually part of the route, the engineers created a mechanism based on access keys:

1. Each research station will receive a list of access keys.
2. When a station u sends a message to another station w, the message is signed with all the keys that u and w have in common.
3. A station v can only read and authorize the passage of the message if it possesses all the keys used in that signature. If two stations share no key, any station in the network will be able to read the message.

The security committee established the following rules for the distribution of keys:

- *Authorization along the path:* Every station v that lies on the path between u and w (including the stations u and w themselves) must possess all the keys shared by u and w.
- *Blocking off the path:* If a station v does not belong to the path between u and w, there must be at least one key shared by u and w that station v does not possess (preventing it from reading the message).
- *Economy:* The global catalog may contain at most $2N$ distinct key types in the entire network.

Your task is to help the engineering team determine which keys each station should receive in order to satisfy all the requirements.

Input

The first line of input contains a single integer N ($2 \leq N \leq 1000$), representing the number of research stations.

The following $N - 1$ lines describe a connection in the network. Each line contains two integers u and v ($1 \leq u, v \leq N$, $u \neq v$), indicating that there is a cable directly connecting stations u and v. It is guaranteed that the network forms a tree.

Output

On the first line, print a single integer K ($0 \leq K \leq 2N$), indicating the total number of distinct keys created. The keys are identified by integers from 1 to K.

In the next N lines, print the key list of each station from 1 to N. The i-th of these lines must begin with an integer Q_i ($0 \leq Q_i \leq K$) indicating the number of keys held by station i, followed by Q_i distinct integers representing the keys given to station i.

It is guaranteed that at least one valid key assignment exists. If more than one exists, any of them will be accepted.

Sample input 1	Sample output 1
3 1 2 2 3	4 2 1 3 2 2 3 2 2 4

Explanation of sample 1:

The network is a line $1 - 2 - 3$ with $N = 3$, allowing up to $2(3) = 6$ keys (the example uses 4 keys):

- Station 1 has the keys $\{1, 3\}$.
- Station 2 has the keys $\{2, 3\}$.
- Station 3 has the keys $\{2, 4\}$.

Testing the communications between each pair of stations:

- Communication between 1 and 3: The list of common keys is empty. Since the signature contains no keys, station 2 is automatically authorized to process the message (and indeed 2 belongs to the path between 1 and 3).
- Communication between 1 and 2: The only common key is key 3. Station 3 does not have key 3, so 3 is blocked (and indeed 3 does not belong to the path between 1 and 2).
- Communication between 2 and 3: The only common key is key 2. Station 1 does not have key 2, so 1 is blocked (and indeed 1 does not belong to the path between 2 and 3).