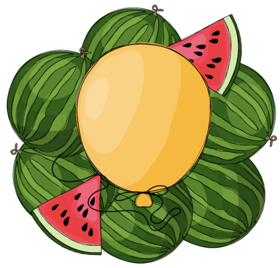
Judging Notes

1. Very few clarifications are issued during the contest, and they are only for ambiguities. Please read the problem statement and examine the sample test cases carefully before submitting any request for clarifications.
2. The following guidelines apply to handling input/output in programs:
   - All input comes from standard input.
   - All output goes to standard output.
   - Unless the problem explicitly states otherwise, the input for a problem consists of a single test case. If the input contains multiple test cases, then the problem will also state an upper bound on the number of test cases.
   - Your program may be run on multiple input files. Note that this means that if your program has more than one error (say, Time Limit Exceeded and Wrong Answer), then you can get either error as judgment.
   - Output formatting should follow the specification and sample output in the problem statement.
3. For problems with floating point output, the judging system will accept a range of answers as correct as long as they satisfy the constraints described in the problem statement. These constraints will be specified as an absolute and/or relative tolerance, which will be given.
4. There is no such thing as "Presentation Error" or "Format Error." If you misspell the word "impossible," for example, and the problem requires that word as output, then your submission will be judged as "Wrong Answer."
5. Unless a problem specifically indicates that uppercase or lowercase letters are important, then either will be accepted. For example, "Yes" or "yes" would be treated the same, but "yse" is Wrong Answer.
6. You should follow the sample output format, but extra whitespace within reason is acceptable. For example, if you print out a gigabyte of blanks, then the judging system will treat that as Wrong Answer; however an extra blank at the end of a line or an extra blank line between test cases is acceptable.
7. During the contest, input size constraints on test cases will be given as part of the problem statements.
8. If you submit a solution that has a Compile Error, then you will be notified of it (just as any other error). However, Compile Errors do not count toward penalty time.

Problem A: Yalda



Yalda is an ancient Persian celebration that marks the longest night of the year, corresponding to the winter solstice. The word Yalda originates from the Syriac language and means **birth**, signifying the rebirth of the sun as days gradually grow longer after the solstice.

On Yalda night, Iranian families come together to spend the night sharing stories, reading poetry (especially works of Persian poets like Hafez), and enjoying each other's company. The celebration is accompanied by traditional items, including:

- **Watermelon**: Representing the warmth of summer and warding off winter's chill.
- **Pomegranates**: Symbolizing the cycle of life and prosperity.
- **Nuts**: A treat that embodies abundance and blessings.

Mahya, busy organizing the ICPC regional contest, has missed celebrating Yalda night with her family. However, she doesn't want to miss out entirely and plans to celebrate with her friends at the university. To make the gathering more festive, Mahya wants to buy **exactly one** traditional item for the celebration. Due to her limited budget, she needs to carefully decide which item to buy.

The items should be selected based on their price and her preferences. Watermelon is the first choice, pomegranates are the second choice, and nuts are the third choice. If Mahya's budget does not allow her to buy any of the items, she will skip the purchase entirely.

Input

The input consists of the following:

- The first line contains an integer b ($0 \leqslant b \leqslant 10^6$), representing Mahya's budget in Rials.
- The next three lines contain the prices of watermelon, pomegranates, and nuts, respectively, each as a non-negative integer not exceeding 10^6.

Output

Print the name of the first item Mahya can afford from her preference list: "`Watermelon`", "`Pomegranates`", or "`Nuts`". If her budget is not sufficient for any of the items, output "`Nothing`".

Example

Standard Input	Standard Output
150 200 95 130	Pomegranates

Problem B: Divar's Salaries

Amin works in the financial department of Divar company. As part of his responsibilities, he needs to prepare the salary payment list for the company's employees. The salary of each employee is calculated based on a base hourly rate of x Rials. However, the actual pay rate varies depending on the type of working hours:

- **Normal Working Hours:** Paid at the standard rate of x Rials per hour.
- **Overtime Hours:** Any hours worked beyond the standard 140 hours per month are compensated at 1.5 times the base rate.
- **Holiday Hours:** Hours worked on recognized holidays are paid at twice the base rate, regardless of whether they fall within or beyond the standard 140 hours.

Holiday hours always take precedence, and employees are compensated at the doubled rate for these hours, irrespective of the standard working hour limits.

Amin's task is to calculate the total monthly salary for each employee, taking into account their normal hours, overtime hours, and holiday hours. Help Amin determine the payment for all the employees in the company.

Input

The first line of input contains a single integer n ($1 \leqslant n \leqslant 1\,000$), representing the number of employees in the company. The following n lines each provide three integers x, k and h, which describe the details for each employee:

- x ($100 \leqslant x \leqslant 10^6$): The hourly wage in Rials, which is always a multiple of 10.
- k ($0 \leqslant k \leqslant 480$): The total number of hours worked during the current month.
- h ($0 \leqslant h \leqslant k$): The number of holiday hours worked.

Output

In the output, print the total monthly payment for each employee on a separate line. The payment must be formatted with commas separating every three digits, starting from the right.

Example

Standard Input	Standard Output
3 500 10 0 200 190 15 10000 150 20	5,000 44,500 1,700,000

Problem C: GPT DarkDown

Mahdieh is a developer working on a chatbot for Divar's open platform (Kenar Divar). She wants to use ChatGPT's API for her chatbot. This means the chatbot receives messages from ChatGPT and forwards them to the user. When the chatbot asks ChatGPT to generate a message, ChatGPT sends the message in chunks to the chatbot. So for each chunk, there is a time when Mahdieh's chatbot receives it, and there is a string it contains.

For a smooth user experience, Mahdieh wants her chatbot to simulate continuous typing, outputting 1 character per millisecond. Obviously, no character can be typed before it is received, so Mahdieh must ensure that the current typing character is available for her chatbot before typing it. To make the perfect smooth user experience she wants to see in her chatbot, the chatbot waits for a while without typing anything, and then starts typing non-stop. Now Mahdieh has all the chunks' information from a hypothetical message, and she needs your help to tell her the first time the chatbot can start to type.

However, ChatGPT (and therefore Mahdieh's chatbot) applies **Darkdown** formatting to the chunks of text, which includes **bold**, *italic*, `inline code`, and even emojis! 😄

So, there are formatting characters in the received message from ChatGPT that will not be rendered in the final text. You need to determine only the (visible) rendered content for smooth typing. For example, the Darkdown text "`[Let's] (Code!) :rocket:`" consists of 24 characters, but its rendered text is only 13 characters long: "*Let's* **Code!** 🚀".

Your task is to determine the earliest time Mahdieh can start typing the rendered output for a smooth experience.

Input

The input starts with a line containing a single integer k, as the number of chunks ($1 \leqslant k \leqslant 10^5$). The next k lines contain the complete input message, where every line represents a non-empty chunk. You can assume the ChatGPT message does not contain newline characters, and the chunks could start or end with spaces. It is guaranteed that the total number of characters in all the chunks combined will not exceed 10^5, and the final rendered message will not be empty. The last line contains k space-separated integers, $t_1, t_2, \ldots, t_k$, where t_i is the time the chatbot receives the i^{th} chunk from ChatGPT ($1 \leqslant t_1 < t_2 < \cdots < t_k \leqslant 10^9$).

The ChatGPT message has the following **Darkdown** formatting:

- **Inline Code Blocks**:
  - Inline code is given inside a pair of backtick characters, e.g. `` `code` ``. The backtick charcters themselves are not rendered in the final output. You can assume there is no backtick characters inside the inline code.
  - Formatting markers inside inline code blocks (like `(` or `\`) are rendered as literal characters and are not parsed as formatting.
- **Special Characters**:
  - In order to render the special characters (`(`, `)`, `[`, `]`, `:`, `\`, `` ` ``), it is sufficient to escape them by preceding them with a backslash character (e.g. `\(`, `\)`, `\[\]`, `\]`, `\:`, `\\`, `` \` ``).

  - Double backslashes (\\) are rendered as a single literal backslash in the visible output.
  - It is guaranteed that every backslash in the input is followed by a special character, except in inline codes and after an escaped backslash (\\).
- **Bold and Italic Formatting**:
  - Bold formatting can be applied by enclosing the text with parentheses: `(bold)`.
  - Italic formatting can be applied by enclosing the text with square brackets: `[italic]`.
  - It is not permitted to have nested blocks of bold or italic text. Also, a block of text cannot be simultaneously bold and italic. For example, it is not allowed to have `[(bold and italic) italic]` or `((superbold) bold)`.
  - Inline codes or emojis are allowed inside a bold or italic text.
- **Emojis**:
  - An emoji is given as a non-empty string of lowercase English letters enclosed within a pair of colons (e.g. `:smile:`, `:rocket:`).
- **Punctuation and Spaces**:
  - All standard punctuation marks (`.`, `,`, `!`, `?`, `'`, `-`, `/`) and space characters are rendered normally.

It is guaranteed that the input adheres strictly to the Darkdown formatting rules described above, and the final rendered text is unique.

In order to find the answer, you have to consider only the visible (rendered) characters, which include:

- **Plain text**: All letters, numbers, spaces, escaped special characters, and standard punctuation marks not part of any formatting.
- **Emojis**: Represented as a single conceptual character. For example, `:smile:` is rendered as 😄. Note that the conceptual character for an emoji is ready for typing when the chatbot has received its ending colon (`:`).
- **Inline Code**: The content inside backtick characters.

Output

Output the earliest time Mahdieh's chatbot can start typing the message such that the chatbot prints the whole text smoothly.

Example

Standard Input	Standard Output
11 The (International Collegiate Programming Contest) \(`ICPC`\) is a global [competi tive programming contest] focusing on algorithm ic problem-solving and teamwork. [Le t's] test `some` (more) Darkdown (elemen ts) here. Visit the Of ficial ICPC Website\: \(https\://i cpc.global\) for more info! :rocket: 10 20 30 40 50 60 70 80 90 100 110	10
2 l ML5g(RsXFVBdO R)(`gwkz MV `) 10 1000	10

Note

For better understanding, these are the final visible rendered messages of the example inputs:

Sample 1:
The **International Collegiate Programming Contest** (`ICPC`) is a global *competitive programming contest* focusing on algorithmic problem-solving and teamwork. *Let's* test `some` **more** Darkdown *elements* here. Visit the Official ICPC Website: (https://icpc.global) for more info! 🚀

Sample 2:
l ML5g **RsXFVBdO R**`gwkz MV`

Problem D: Electromagnetic Attacks

In Barareh, a Point-to-Point (P2P) wireless network is used to connect base stations for private data services. In a P2P wireless network, each base station uses some directional antennas to connect with the other base stations. If base stations are modeled as points and communication links are modeled as line segments, the P2P network in Barareh surprisingly has some geometric properties. Specifically, the network is a planar graph whose outer boundary is a convex polygon and all interior faces are triangles.

In the recent world conflicts, an important question has occupied the mind of Khorzukhan, the Minister of Information and Communications Technology (ICT) of Barareh. He wants to know how resilient the network is to electromagnetic attacks. In an electromagnetic attack, noise is created in a certain region (so-called attack region), disrupting all communications passing through that region. The remaining network consists of every base station and every communication link that were strictly outside of the attack region. Specifically, Khorzukhan wants to know if the remaining network remains connected. To achieve this, he has instructed his ministry to simulate multiple electromagnetic attacks separately, testing the network's tolerance to interference and reporting whether the remaining network stays connected after each simulation.

Input

The first line of input contains n, m, and k ($3 \leqslant n \leqslant 10^5$, $3 \leqslant m \leqslant 3 \cdot 10^5$, $1 \leqslant k \leqslant 10^5$), which are the number of base stations, the number of communication links, and the number of attack simulations, respectively.

In the next n lines, the i^{th} line contains the x and y coordinates of the i^{th} base station, both of which are non-negative integers ($0 \leqslant x, y \leqslant 10^9$). It is guaranteed that not all base stations are collinear.

Each of the next m lines represents a communication link. Each line contains two integers i and j ($1 \leqslant i, j \leqslant n$), representing a communication link as a straight line segment between i^{th} and j^{th} base station. The m communication links form a planar graph. The outer boundary is a convex polygon and interior faces are all triangles.

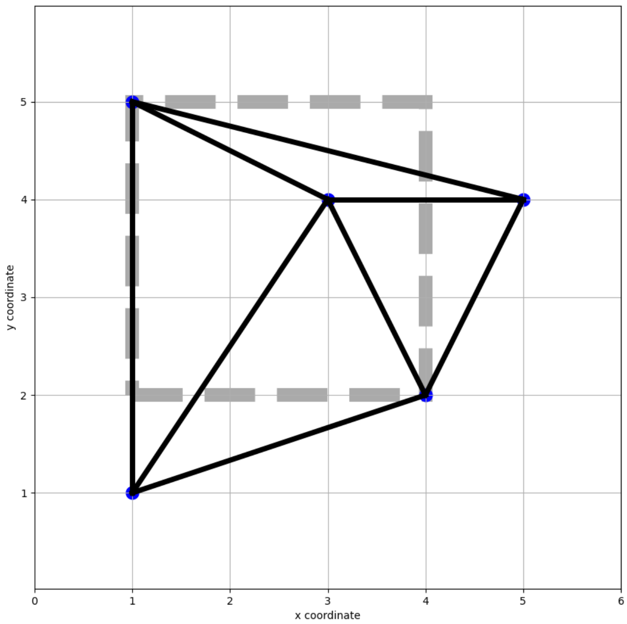
At the end, the attack regions come in k lines. Each attack region is a non-empty rectangle, represented by the coordinates x_1, y_1, x_2, y_2 of its lower-left and upper-right corners ($0 \leqslant x_1 < x_2 \leqslant 10^9$, $0 \leqslant y_1 < y_2 \leqslant 10^9$). The sides of all rectangles are parallel to the coordinate axes. Note that if a base station or some part (even one point) of a link lies inside the attack region (including the boundary), it is not usable during the attack.

Output

In k lines, for each attack simulation, print `Yes` if the remaining network resulting from the attack is connected; otherwise print `No`. If all base stations are within the attack region, the remaining network becomes empty and is still considered connected.

Example

Standard Input	Standard Output
5 8 2 1 1 1 5 5 4 4 2 3 4 1 2 2 3 3 4 4 1 5 1 5 2 5 3 5 4 1 2 4 5 2 3 3 4	No Yes



In the first attack simulation, the first and third base stations are outside the attack region; however, they are not connected to each other.

Problem E: PCB

In designing a printed circuit board (PCB), each consumer must be connected to a power supply via conductive wires. The PCB is a rectangle of width W and height H. It is represented as a grid of integer coordinates from $(0, 0)$ to $(W + 1, H + 1)$.

There are n power supplies along the left edge of the board and n consumers each located somewhere inside the board. The i^{th} power supply is located at position $(0, h_i)$ and the i^{th} consumer is located at position (x_i, y_i). Each power supply must connect to exactly one consumer and vice versa.

Each wire must run along the grid lines, bending at most once. i.e., each wire is either a straight vertical or horizontal line or makes exactly one 90-degree turn, forming an "L" shape. Wires cannot cross or overlap with each other anywhere along their paths.

Your task is to determine a matching between power supplies and consumers such that the total length of all wires is minimized.

Input

The input consists of several lines:

- The first line contains three integers W, H and n ($1 \leqslant W, H \leqslant 10^8$; $1 \leqslant n \leqslant 10^6$).
- Each of the next n lines contains an integer h_i ($1 \leqslant h_i \leqslant H$).
- Each of the next n lines contains two integers x_i and y_i ($1 \leqslant x_i \leqslant W$; $1 \leqslant y_i \leqslant H$).

It is guaranteed that each point in the board contains at most one power supply or consumer. Moreover, no two consumers i and j exist where $x_i = x_j$.

Output

If it is not possible to find such a matching under the given constraints, output a single line containing -1.

Otherwise, output a single line containing n space-separated integers. The i^{th} integer describes p_i, indicating that power supply i is connected to consumer p_i.

Example

Standard Input	Standard Output
5 5 2 2 4 3 2 5 4	1 2
10 10 5 9 6 2 8 1 2 3 5 8 3 8 4 8 1 2	2 4 5 3 1

Problem F: I am Sherlocked

Detectives of Scotland Yard, the headquarters of the Metropolitan Police, informed Sherlock Holmes about a cyber attack on the Central Bank by James Moriarty. They have also found that there is a secret code that can stop the attack. According to their top Agent, Moriarty has hidden the code in one of his clients' phonebook. Interestingly, Sherlock has access to this mysterious client!

It is known that each correct phone number has 11 digits and starts with a leading zero digit. Moriarty's client may separate the digits of each phone number with hyphen ("`-`") or space characters. Even the leading zero digit might be missing for some phone numbers in the notebook. For example, phone number "`09163264128`" might be written like "`916 32 64 128`", or even "`- 0916-16-32--64 - 128-`". Sherlock is not aware of the exact content of the phonebook, but due to his previous knowledge, he knows how the owner writes phone numbers.

Sherlock arranged a friendly conversation with the owner of the phonebook in a cafe. Meanwhile, Dr. John Watson, Sherlock's colleague, took the phonebook sneakily. Sherlock instructed Dr. Watson to clean up the phonebook into a new cleaned sequence $\mathcal{C}$ of numerical characters in his computer. The sequence is zero-indexed, i.e. the index of the first element is zero. To this end, Dr. Watson should type the phone numbers consecutively (without any non-digit/separating characters) into his computer, in the same order of appearance as in the phonebook. In order to clean the phone numbers, he should remove all non-digit characters and add the leading zero if necessary.

According to their plan, Sherlock is supposed to have an informal conversation with the mysterious client. As soon as he finds any information about the code, he is going to text it to Dr. Watson. To this end, Dr. Watson has already prepared the cleaned sequence $\mathcal{C}$ and has put the cursor at the beginning character. This cursor can be placed before any character of $\mathcal{C}$. He should wait for Sherlock's instrucitons to produce the output sequence $\mathcal{S}$.

Sherlock assumes that Dr. Watson has made the cleaned sequence $\mathcal{C}$ and send one of the following instructions based on that:

1. `go` i: Move the cursor to the beginning of the i^{th} cleaned phone number in $\mathcal{C}$. For example, to jump to the first phone number in $\mathcal{C}$ he would use "`go 0`".
2. `forward` i: Move the cursor forward by i digits.
3. `backward` i: Move the cursor backward by i digits.
4. `next` i: Write the next i digits starting at the current position into $\mathcal{S}$. More specifically, if Dr. Watson's cursor is before position c, he should pick digits $c, c+1, \ldots, c+i-1$ but the position of the cursor remains unchanged.
5. `pick` i j: If $i < j$, write digits into $\mathcal{S}$ from the position i to j $(i, i+1, \ldots, j)$ of the current phone number, which is the one that the digit after the cursor belongs to. Otherwise he should write $(i, i-1, \ldots, j)$. Again, the position of the cursor remains the same. Note that, $0 \leqslant i, j \leqslant 10$ and "`pick 0 0`" will pick the first digit.

For the sake of any correction, Sherlock may also send the following instruction:

- `delete` i: Delete the last i digits from the end of $\mathcal{S}$.

Input

The input consists of the phonebook's content and Sherlock's instructions. The phonebook contains n phone numbers ($1 \leqslant n \leqslant 1\,000$), separated by commas or newlines. Each phone number in the phonebook is a string, consisting of 10 or 11 digits (`0-9`) and possibly hyphens ("`-`") and/or spaces characters. The size of the phonebook, i.e. the number of characters in the phonebook, will not exeed 10^6 characters. The list of the phone numbers terminates with a line containing the single character "`#`".

Each of the rest of the following lines contains one of Sherlock's instructions. He will send at most 10 000 instructions. All instruciton's arguments are non-negative integers not greater than 20 000. It is guaranteed that all Sherlock's texts are in the form of one of the six mentioned instructions.

Output

Print the extracted secret code (possibly empty), which is a string of digits. If the extracted secret code has more than 10 000 digits, just print the first 10 000 first digits. Sherlock might lose his attention and send malfunctioning instruction. He might jump to a non-existing phone number (with "`go`") or send an invalid argument for "`forward`", "`backward`", "`next`", "`pick`", and "`delete`" i.e. the instruction refers to a digit or phone that doesn't exist in cleaned sequence $\mathcal{C}$. In such cases you find that Moriarty has succeeded and print "`MISS ME!`" with capital letters.

Example

Standard Input	Standard Output

`0912 358 8908`
`0872-3344567,0989112-2345`
`9899988782`
`#`
`go 0`
`pick 0 3`
`next 4`
`forward 12`
`next 2`
`backward 2`
`pick 0 1`
`go 3`
`pick 0 1`
`delete 1`
	`0912091287090`

`09242424024`
`00188990376`
`#`
`go 2`
`next 4`
`delete 1`
	`MISS ME!`

Problem G: Double Radars

Ali recently found a treasure and has arranged the coins from the treasure around a circle. There are k houses around the circle with equal distances. Houses are numbered 1 through k consecutively in the clockwise direction. The treasure contains n coins, where the i^{th} coin (for $1 \leqslant i \leqslant n$) has the value w_i and is located at the house x_i.

To protect the treasure, Ali has installed two radars stationed at the center of the circle, monitoring its circumference. Radar i (for $i \in \{1, 2\}$) starts by monitoring house r_i and moves $\frac{1}{v_i}$ houses per minute. More intuitively, every v_i minutes, the Radar i goes one house forward. Initially, the first radar moves clockwise, and the second radar moves counterclockwise. Whenever the two radars meet, they both reverse their directions. Note that this can happen in the area between two adjacent houses.

Gholi, who wants to steal as many coins as possible, plans to start at an arbitrary house on the circle and move at most $\frac{1}{v}$ houses per minute in either direction (clockwise or counterclockwise). He starts moving at the time zero. He can reverse his direction anytime or stay still for a while. If Gholi crosses paths with one of the radars at any moment, he will be immediately caught and sent to jail. He cannot steal a coin if this happens at a house.

Help Gholi to maximize the total value of the coins he can steal before being detected by the radars.

Input

The first line contains three integers, n $(1 \leqslant n \leqslant 10^5)$ number of coins, k $(1 \leqslant k \leqslant 10^9)$ number of houses around the circle, and v $(1 \leqslant v \leqslant 10^4)$ speed of Gholi.

The second line contains the starting monitoring house r_1 and speed v_1 of the first radar $(1 \leqslant r_1 \leqslant k, 1 \leqslant v_1 \leqslant 10^4)$.

The third line contains the starting monitoring house r_2 and speed v_2 of the second radar $(1 \leqslant r_2 \leqslant k, 1 \leqslant v_2 \leqslant 10^4)$. It is guaranteed that $r_1 \neq r_2$.

The fourth line contains n distinct integers, $x_1, x_2, \ldots, x_n$, representing the houses where the coins are located $(1 \leqslant x_i \leqslant k)$.

The fifth line contains n integers, $w_1, w_2, \ldots, w_n$, representing the value of each coin $(1 \leqslant w_i \leqslant 10^9)$.

Output

Output the maximum total value of coins Gholi can steal before being detected by the radars.

Example

Standard Input	Standard Output
3 5 1 1 2 2 3 1 2 4 1 2 4	6

Problem H: Anti-Missile

We want to execute a strategic missile strike against an enemy's critical resources. The enemy has deployed air defence systems to protect these resources. However, their defence setup has certain vulnerabilities, and your mission is to exploit them effectively.

Each air defence system can protect all strategic resources and air defence systems within its radius of operation, but cannot defend itself. Due to technical limitations, each critical resource or air defence system is protected by at most one other air defence system.

Missiles can be used to destroy either an undefended strategic resource or an air defence system.

A resource is considered undefended if no active air defence system protects it. When an air defence system is destroyed, it can no longer protect any resources or other air defence systems. Your goal is to maximize the number of strategic resources destroyed.

Input

The input consists of the following:

The first line contains three integers m (number of missiles), n (number of strategic resources), and d (number of air defence systems), where $(0 \leqslant m, n, d \leqslant 5\,000)$.

The next n lines contain two integers x_i and y_i $(0 \leqslant x_i, y_i \leqslant 10^9)$ the coordinates of the i^{th} strategic resource.

The following d lines each contain three integers x_j, y_j and r_j $(0 \leqslant x_j, y_j \leqslant 10^9;\ 0 \leqslant r_j \leqslant 10^9)$ the coordinates of the j^{th} air defence system and its radius of protection.

Output

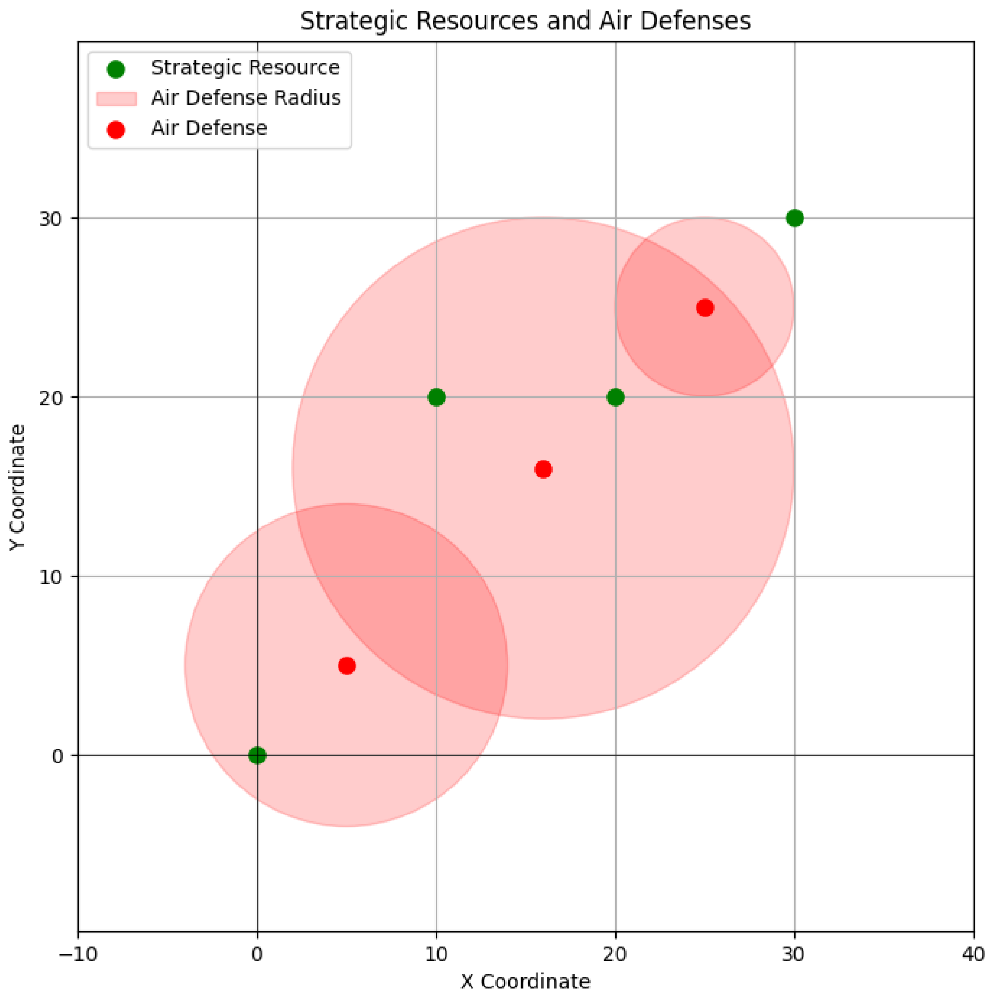
Output the maximum number of strategic resources that can be destroyed.

Example

Standard Input	Standard Output
3 4 3 0 0 10 20 20 20 30 30 5 5 9 16 16 14 25 25 5	2

Note

The example test case is depicted by the figure below:



Problem I: Conference Rides

There are n attendees in a conference, numbered 1 through n. Each of the first m attendees (numbered 1 through m) has a car to drive home after the event. The remaining $n - m$ attendees who do not have a car, are going to get a ride to their homes with the help of the first m attendees. Each of the first m attendees can pick up at most one other attendee (from attendees $m+1$ to n) and drive them to their house before going to their own home. You are given the distance matrix D of the $n+1$ locations (the conference hall and n attendees' homes). Find a way for attendees with cars to drive the attendees without cars home, such that the time it takes for all attendees to arrive at their homes is minimized. The distance matrix D is an $(n+1) \times (n+1)$ matrix where $D[i][j]$ denotes the estimated time of transportation from location i to location j. Location i (for $1 \leqslant i \leqslant n$) denotes the home of the i^{th} attendee and the conference hall is positioned at location $n+1$.

Input

The input starts with a line containing two integers n and m, ($1 \leqslant n \leqslant 500$ and $1 \leqslant m \leqslant n$). It is guaranteed that $2m \geqslant n$.

The following $n+1$ lines specify the distance matrix D, each containing $n+1$ integers. The j^{th} number from the $i+1^{\text{th}}$ line of the input (for $1 \leqslant i, j \leqslant n+1$) specifies $D[i][j]$ ($0 \leqslant D[i][j] \leqslant 10^8$). It is guaranteed that $D[i][k] \leqslant D[i][j] + D[j][k]$ for any $1 \leqslant i, j, k \leqslant n+1$, and also $D[i][j] = 0$ for $i = j$, but $D[i][j]$ is not necessarily equal to $D[j][i]$.

Output

In the first line of output, print the minimum time it takes for all attendees to arrive at their homes. In the next m lines, each line i (for $1 \leqslant i \leqslant m$) should contain a single non-negative integer t_i, denoting the driving schedule of the i^{th} attendee. If $t_i = 0$, the attendee drives directly to their home without picking up any other attendees. Otherwise ($t_i > 0$), the i^{th} attendee picks up the attendee t_i and takes them to their home before driving to their own home. Each attendee must be transferred by exactly one car.

Example

Standard Input	Standard Output
3 2 0 1 1 2 2 0 1 3 4 2 0 4 4 3 2 0	4 0 3

Problem J: Parking Theory

Sharif University has a rectangular parking lot with $n \times m$ spaces for cars. Each row and column of the parking lot has entrances at both ends.

The parking lot is full, and the order in which the cars entered is given for each parking space. Specifically, a cell with the number 1 is the first car that entered the parking lot, and a cell with the number $n \cdot m$ is the last one to enter.

Abolfazl has a theory about how cars park in this lot. He believes that any car entering the parking lot from a specific side (row or column) moves straight until it finds its parking spot and never changes direction. Moreover, a car cannot pass through a cell that already contains a parked car.

Abolfazl wants to count the number of subgrids in the parking lot that satisfy this condition. A subgrid is valid if all cars in that subgrid can park without violating the above rules, considering only the cars within the subgrid.

Help Abolfazl determine the number of such valid subgrids.

Input

The first line of input contains two integers n and m ($1 \leqslant n, m \leqslant 500$), the number of rows and columns of the parking lot. Each of the following n lines contains m integers, indicating the order of entry of the cars. It is guaranteed that numbers are different between 1 and $n \cdot m$.

Output

Print a single integer, the number of valid subgrids in the parking lot.

Example

Standard Input	Standard Output
2 3 1 2 5 3 4 6	18

Problem K: Boat

A river separates Upper Barareh from Lower Barareh. To transport people between these two towns, a **two-seater** boat (a boat that can carry at most two people) with a certain weight capacity has been provided. This boat must be steered by at least one person. i.e. it can not move across the river without any passengers.

The National Barareh Festival is scheduled to be held in Upper Barareh. All Lower Barareh residents want to participate in this celebration and need to move to Upper Barareh as quickly as possible. Your task is to help them move to Upper Barareh with the minimum number of boat trips across the river.

Input

The first line of the input contains two integers n and w, where n is the number of Lower Barareh residents ($1 \leqslant n \leqslant 1\,000$), and w is the maximum weight the boat can carry ($1 \leqslant w \leqslant 10^6$). The next line contains n space-separated integers, describing the weights of the residents of Lower Barareh. All the weights are positive integers not exceeding 10^6.

Output

If it is not possible to transfer all the residents of Lower Barareh, print a single line containing "`-1`" in the output. Otherwise, print the minimum number of times the boat must travel between Lower Barareh and Upper Barareh (in both directions) in order to transfer all residents of Lower Barareh to Upper Barareh.

Example

Standard Input	Standard Output
3 7 1 3 4	3
3 4 2 3 4	-1