

Problem A. Traffic Light

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 512 megabytes

First-grader Pasha has taken the wheel of a car once again. Although he cannot reach the pedals or the steering wheel, this is not a problem — he has placed some textbooks on the gas pedal, causing the car to move at a constant speed of v .

From his vantage point in the seat, he sees an intersection ahead with a traffic light. The traffic light will turn red after some time has passed since a pedestrian pressed the button, which has just happened.

Pasha knows that he is s meters away from the intersection, and that the red light will activate in t seconds, remaining illuminated for T seconds. Since no other pedestrians are visible on the horizon, we can assume that the red light will only be triggered this one time, and Pasha will have enough time to pass before any subsequent pedestrian arrives.

Your task is to help Pasha determine whether he needs to take control of the vehicle and apply the brakes, or if he can simply enjoy the ride without running a red light.

Input

The input consists of a single line containing four integers s , v , t , and T ($1 \leq s, v, T \leq 100$, $0 \leq t \leq 100$) — the distance to the traffic light, the speed of the car, the time until the traffic light turns red, and the duration for which the traffic light will remain red. You may consider the car as a point mass. It is guaranteed that the car will not approach the traffic light exactly when it switches.

Output

Print a single line containing “Yes” if Pasha needs to brake, and “No” otherwise.

Examples

standard input	standard output
9 2 4 2	Yes
9 2 5 2	No

Problem B. Knight's Revenge

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

It is commonly said that one warrior in the field is not a warrior at all. However, the legendary hero Evpatiy Kolovrat questioned the validity of this assertion and ventured into a vast 10×10 field to confront the enemy forces, riding his loyal white steed and wielding his trusty sling.

Evpatiy wishes to select an empty cell (x, y) within the field and remain there, aiming to defeat as many adversaries as possible. His prowess with the sling allows him to target enemies located in cells that can be reached by a knight's move from (x, y) . The task is to determine the maximum number of enemies that Evpatiy can defeat from any given position.

Input

The input consists of a description of the field, containing 10 lines, each with 10 characters. Empty cells are represented by the character ".", while cells occupied by enemies are marked by the character "h". It is guaranteed that there is at least one empty cell in the field.

Output

Output a single integer: the maximum number of enemies that Evpatiy can defeat from a strategically chosen position.

Examples

standard input	standard output
.hhhhhhhhh hhhhhhhhh hhhhhhhhh hhhhhhhhh hhhhhhhhh hhhhhhhhh .hhhhhhhhh .hhhhhhhhh hhhhhhhhh hhhhhhhhh.	4
hhhhhhhhh hhhhhhhhh hhhhhhhhh hhhhhhhhh hhhhhhhhh hhhh.hhhhh hhhhhhhhh hhhhh.hhhh hhhhhhhhh hhhhhhhhh	7

Problem C. The Thermotaur Labyrinth

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

Have you ever heard of the Thermotaur Labyrinth? It is a square, single-story building partitioned into identical square rooms. However, it is not quite a labyrinth — one can move from any room to any adjacent room on a side, but no one can exit the boundaries of the labyrinth. Each room has a distinct temperature, which can vary from one room to another.

A Mad Scientist conducts experiments on minotaurs within this Thermotaur Labyrinth. He releases them one by one into specific rooms and observes where they ultimately go. It turns out that minotaurs do not favor high temperatures; thus, they will always leave their current room for an adjacent room with the lowest temperature. Naturally, if all adjacent rooms have a higher temperature than their current one, the minotaur will choose to remain in its current room.

Upon learning about the experiments of the Mad Scientist and the minotaurs' behavior, you took pity on them and declared that you could write a program that determines the final room that a minotaur will occupy, given any starting position.

Input

The first line of the input contains the size of the labyrinth N ($1 \leq N \leq 500$). The following N lines each contain N integers — representing the temperatures of the rooms in the labyrinth. All temperatures are distinct positive integers, not exceeding N^2 .

The next line contains the number of queries Q ($1 \leq Q \leq 20\,000$). Each of the following Q lines contains two integers indicating the row and column of the room where the minotaur is initially placed.

Output

For each query, output on a separate line the temperature of the room where the minotaur ultimately comes to rest.

Example

standard input	standard output
4	1
1 2 3 4	1
7 14 12 11	1
6 13 10 15	5
5 8 9 16	5
7	5
1 1	1
1 4	
2 1	
3 1	
4 4	
3 3	
2 3	

Problem D. Fibonacci Grouping

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 512 megabytes

In a newly organized event, a group of n individuals has lined up in a single row, numbered sequentially starting from one. To make the event more interesting, the participants are grouped based on a unique selection method. Each time a grouping occurs, individuals standing at positions that correspond to Fibonacci numbers are selected to form a group. The individuals selected in this round are then taken aside, while the remaining individuals are regrouped in a similar manner.

Your task is to find out how many distinct groups will be formed from the original lineup of n individuals.

Input

The input consists of a single line containing an integer n ($1 \leq n \leq 10^{18}$) — the initial number of individuals.

Output

Output a single integer k — the total number of groups created through this selection process.

Examples

standard input	standard output
2	1
5	2
12	3

Problem E. Uniform Square Tiling

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 512 megabytes

A family has recently moved into a new apartment. One of the square walls in their living room is adorned with square tiles made of gold and silver. These tiles are arranged in a seemingly random fashion, without any discernible pattern. However, the family desires the wall to be uniform.

Unfortunately, shortly after moving in, they do not have the funds to purchase new tiles or hire professionals experienced in working with such valuable materials to re-tile the entire wall. As a temporary solution, the head of the family decided to cover a large portion of the wall with paintings, cabinets, and other wall decor. Nonetheless, they do not wish to conceal the entire wall. Instead, they opted to leave a square section of the wall visible, upon which they plan to place vases with flowers. To achieve a visually pleasing appearance, they will need to remove some tiles and replace them with tiles of the other material so that the visible square is completely uniformly tiled.

Now, they need to determine which section of the wall to keep visible. Naturally, they would like to select the section that requires the minimal number of tile replacements. The family needs to first ascertain how many tiles will need to be replaced in order to facilitate this process.

Input

The first line of the input file contains two integers n and k ($1 \leq n \leq 1000$, $1 \leq k \leq 100$, $k \leq n$), separated by a space. Here, n represents the dimension of the wall in tiles, and k represents the dimension of the desired visible square in tiles.

The next n lines contain n characters each, representing the layout of the tiles on the wall. The character “G” denotes a gold tile, while the character “S” indicates a silver tile.

Output

Output a single integer m , which denotes the minimum number of tile replacements required to form a $k \times k$ square on the wall consisting entirely of identical tiles.

Examples

standard input	standard output
3 2 GGG SSG GGG	1
5 3 GGGGG SSSSS GGGGG SSSSS GGGGG	3

Problem F. Optimal Arithmetic Sequence

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 512 megabytes

Nikita and Olya quickly grew bored during the preliminary round and decided to engage in a more interesting activity. Olya drew several cards, each containing text of the form “ $c\ n$ ”, where c is one of the operations “+”, “-”, “*” or “/”, and n is a number ranging from 1 to 9. Now, Nikita must arrange these cards in such a way that the sequential application of the operations to the number one yields the maximum possible value. Help Nikita achieve this.

Input

The first line of the input file contains a single integer n ($1 \leq n \leq 100$) — the number of cards drawn by Olya. Each of the following N lines contains the operation symbol and the associated number written on the respective card, separated by a space.

Output

Output the descriptions of the cards in the optimal order, each on a separate line, maintaining the same format.

Example

standard input	standard output
8	/ 2
+ 1	/ 3
+ 5	+ 1
- 1	+ 5
- 2	* 2
* 2	* 4
* 4	- 1
/ 2	- 2
/ 3	

Problem G. Minecraft

Input file: **standard input**
Output file: **standard output**
Time limit: **1 second**
Memory limit: **512 megabytes**

The side view of the Minecraft game world is represented as a rectangular grid of dimensions $h \times w$ — a vertical cross-section of the ground. Each cell in this grid can be empty, occupied by a dirt block, or occupied by a gold block. Beneath the bottom edge of the grid lies lava. Each block is supported either by another block or by lava. Additionally, gravity operates in Minecraft; thus, an object will fall downward if it lacks support.

Initially, the player stands on an empty cell located above the leftmost occupied cell.

The player may perform the following actions:

1. Mine the block directly below if there is no lava beneath that block. After this action, the player moves down one cell.
2. Mine the block to the right. If this block is not the uppermost in its column, all blocks above it will fall down. The player does not move to the right.
3. Place a dirt block in an empty cell to the right. If there is an empty cell beneath this block, it will fall down.
4. Move into an empty cell to the right. If the cell beneath the player is empty, the player will fall down. The player may safely fall from any height.
5. If the player is not at the upper boundary, they may jump up one cell. While in the air, the player can place a dirt block underneath themselves or perform any of the actions listed above.

The objective of the player is to mine all gold blocks and reach the right edge of the grid using the minimum number of block creation and destruction operations.

Input

The first line of the input file contains two integers h and w ($5 \leq h, w \leq 100$) — the height and width of the grid, respectively. The next h lines contain w characters each, describing the grid: the character “.” denotes an empty cell, the character “#” denotes a dirt block, and the character “G” denotes a gold block. There are no gold blocks on the left and right boundaries. All cells at the top boundary are empty. All cells at the bottom boundary are occupied by dirt blocks. There is no empty block located underneath any occupied block.

Output

The output file should contain a single integer — the minimum number of block creation and destruction operations required to achieve the objective.

Examples

standard input	standard output
5 5 #..#. ##.## #GGG# #####	5
5 5 #..G. #..## ##### #####	1
8 5### .#.G# ##### #####	4

Problem H. The Hard Question

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 512 megabytes

Dance parties at the programming school are quite unique.

The boys, numbered with natural numbers from 1 to n , take turns approaching the girls to invite one of the girls who is not currently dancing. Specifically, the boy with number 1 invites a partner first, followed by the boy with number 2, and so forth. Naturally, a boy will only invite a girl he finds attractive, which means a boy might end up with no one to invite, and thus remain without a dance partner.

During their time between dances, when grouped without girls, the boys start to worry about the possibility that they may all end up without partners for the next dance. Your task is to help them determine the answer to this question.

Input

The first line of the input file contains two integers n and k ($2 \leq n, k \leq 50$) — the number of boys and the number of girls, respectively. Each of the following n lines describes the preferences of the i -th boy: the integer m_i (the number of girls he likes) followed by m_i distinct integers from 1 to k , representing the indices of these girls in ascending order.

The next line contains the number of inquiries q ($1 \leq q \leq 1000$). Each of the following q lines starts with the integer t_j (the number of boys in a particular group), followed by t_j distinct integers from 1 to n , indicating the indices of the boys in ascending order.

Output

For each query, output a single line containing “Yes” if all boys in this group could end up without partners, and “No” otherwise.

Examples

standard input	standard output
3 3 2 2 3 2 1 2 2 2 3 3 1 1 1 2 1 3	No No Yes
3 3 2 2 3 2 2 3 2 1 2 3 1 1 1 2 1 3	No No No
4 4 4 1 2 3 4 4 1 2 3 4 2 1 2 2 1 2 4 1 2 1 3 1 4 2 3 4	No Yes Yes Yes

Problem I. Dormitory Mapping

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 512 megabytes

Alexey, a first-year student, has settled into his new surroundings at the National University of Berland. He attends classes in the mornings and spends his evenings relaxing by playing video games or browsing social media. However, when he runs out of games and his internet access is cut off, he seeks new challenges.

One day, Alexey found a map of the university's dormitories, which consists of n dormitories numbered from 1 to n , connected by bidirectional roads. The roads allow students to easily visit one another, ensuring that there is a path between any pair of dormitories.

In an effort to practice his arithmetic skills, Alexey wrote the sum of the dormitory numbers connected by each road on the map and then set it aside.

Upon revisiting the map, Alexey discovered that the dormitory numbers have faded, but the sums written on the roads are still legible. He has assigned new numbers to the dormitories, labeled from 1 to n . However, he soon realized that navigating the campus based on the new numbering could lead to confusion, especially when trying to find the dormitory of a third-year student who is celebrating a birthday.

To avoid getting lost, Alexey has asked for your help in recovering the original dormitory numbers. Given the sums on the roads and the new numbering, he hopes to restore the original identifiers before the exams.

Input

The first line contains two integers n and m ($1 \leq n \leq 10^5$, $0 \leq m \leq 3 \cdot 10^5$) — the number of dormitories and the number of roads connecting them, respectively. The following m lines provide the details of the roads in the format $a \ b \ s$ ($1 \leq a, b \leq n$, $1 \leq s \leq 2 \cdot n$) — where a and b are the new numbers of the road endpoints and s is the sum inscribed on that road. No road connects a dormitory with itself. There are at most one road between two dormitories.

Output

In the first line, output k — the number of valid ways to restore the original dormitory numbers. If $k > 0$, output any one valid permutation of numbers from 1 to n on the second line, where the i -th number indicates the original dormitory number corresponding to the new number i . Note that Alexey may have made errors when calculating the sums, and it might not always be possible to restore the original numbers.

Examples

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

Problem J. Kings' Dominion

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

Recently, Pasha became interested in a chess puzzle. He placed k kings on a 100×100 chessboard. In a single move, a king can be moved to one of the neighboring squares. Two squares are considered neighboring if they share an edge or a corner. If the target square is occupied by another king, the latter is removed from the board. Pasha can choose any king and will continue to move it until he captures another king. What is the minimum number of moves required for Pasha to leave exactly one king remaining on the board?

Input

The first line of the input file contains an integer k ($2 \leq k \leq 100$) — the number of kings on the chessboard. The following k lines each contain two integers r_i and c_i — the row and column coordinates of the i^{th} king, respectively ($1 \leq r_i, c_i \leq 100$).

Output

The output file should contain a single integer — the minimum number of moves needed for exactly one king to remain on the board.

Example

standard input	standard output
5 1 1 1 3 3 2 4 1 4 3	6

Problem K. Dima and Petya's Café

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

Dima and Petya are two friends who have decided to open a café chain named “DimaPetya.”

The city where they plan to open their cafés has a rectangular layout, divided by $n + 1$ streets running west to east and $m + 1$ avenues running north to south. Each street and avenue spans the entire city without any turns, and no two streets or avenues coincide. The streets are numbered from 1 to $n + 1$ from north to south, while the avenues are numbered from 1 to $m + 1$ from west to east.

To fund their venture, Dima and Petya calculated their earnings and realized they could open exactly two cafés in different city blocks. They plan to study the local customer base before making their decision.

For each of the $n \times m$ blocks, they determined the number of potential customers residing in each. These customers will not visit a café if it requires crossing more than one street or avenue. If a customer can reach both cafés, they will choose only one.

Dima and Petya seek a placement where both cafés attract at least k customers.

Your task is to help Dima and Petya compute the number of ways to position their cafés.

Input

The first line of the input file contains three integers: n, m, k ($3 \leq n, m \leq 50$, $1 \leq k \leq 10^6$).

This is followed by n lines, each containing m integers that represent the number of potential customers in each block. The i -th line and j -th integer indicate the number of potential customers in the block between streets i and $i + 1$ and avenues j and $j + 1$. Each block contains at least one and at most 100 000 potential customers.

Output

The output should contain a single integer: the number of ways to select two distinct blocks for the cafés.

Example

standard input	standard output
4 3 9 2 1 1 1 1 1 1 1 1 3 1 2	32

Problem L. Optimal Milk Temperature

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 512 megabytes

Vova is a boy with a great affection for milk, particularly warm milk. However, he absolutely dislikes milk with foam, and under no circumstances will he drink it.

Vova has a good relationship with the milkman, who delivers fresh milk to him every day. For the next n days, Vova knows in advance how delicious each milk batch will become when heated to a certain temperature. Specifically, for each day i , he knows the flavor enhancement per degree of heating, denoted as b_i , as well as the maximum temperature t_i (in degrees) at which the milk will still be foam-free.

The milkman has a busy schedule, yet he is willing to warm Vova's milk to exactly T degrees every day.

Now, Vova faces a challenging problem: he must determine the optimal temperature T to request for heating the milk so that the total flavor enhancement from all the milk he drinks is maximized.

Input

The first line contains a single integer n ($1 \leq n \leq 100\,000$).

The second line contains n integers b_i ($1 \leq b_i \leq 10^5$).

The third line contains n integers t_i ($1 \leq t_i \leq 10^9$).

Output

The output should consist of a single integer — the maximum total flavor enhancement from the milk Vova drinks given the optimal choice of temperature T .

Examples

standard input	standard output
3 1 1 1 2 2 7	7
2 3 42 10 1	45

Note

In the first example, Vova will only drink the third batch of milk, heated to 7 degrees. The first two batches will not be consumed because they would develop foam when heated to 7 degrees.

In the second example, he will drink both milk batches, each heated to 1 degree.