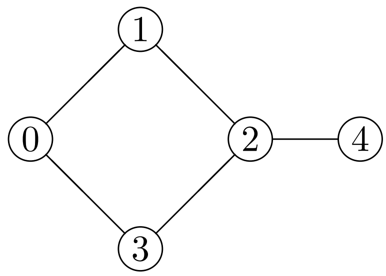
Autoforward Virus

Facenger, the premier social networking site was always a favourite target of cyber-bandits and hackers. A notorious group of hackers, the Synonymous has finally managed to identify a weakness in the security protocols. They developed a virus that spreads from user to user, amplifying over time and eventually overloading the messaging servers.

Normally, users have to manually send messages to one another, but the security weakness now allows the hackers to create messages that automatically spawn and forward more messages, without intervention. Specifically, any time a user gets an infected message, it immediately forwards the infected message to all its friends in the social network. Infected messages act independently: if a user receives multiple messages at any point, it broadcasts that many messages to all of its friends. It takes exactly one second for each message to arrive after being sent. For example, consider the following network:



If user 0 sends a message to all of its friends at time $T = 0$, then

- at time $T = 1$ users 1 and 3 get 1 messages each;
- at time $T = 2$ users 0 and 2 get 2 messages each; and
- at time $T = 3$, users 1 and 3 get 4 messages each and user 4 gets 2 messages.

Given the list of friendship relations and the user who sends the initial message at $T = 0$, you need to determine the total number of messages received by the users of the social network at some given time T. In the above example the number of messages would be 2, 4 and 10 at times $T = 1$, $T = 2$ and $T = 3$, respectively.

Input

The first line of the input contains four integers N, M, S and T, indicating the number of users ($1 \leq N \leq 100$), the number of friendship relations between users ($0 \leq M \leq \frac{N(N-1)}{2}$), the index of the user sending the initial message ($0 \leq S < N$), and the number of seconds ($0 < T < 10$).

Each of the following M lines contains two integers X and Y ($0 \leq X < Y < N$), indicating that users X and Y are friends. Friendship relations are symmetric and distinct.

Output

Output the number of messages sent at the specified time T.

Examples

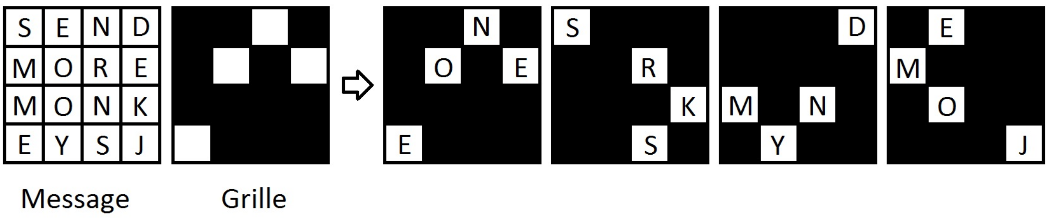
input	output
4 3 1 4 0 1 1 2 2 3	8
5 5 0 3 0 1 0 3 1 2 2 3 2 4	10

Boys and Grilles

Every boy dreams of writing a love letter to his true love, in a way that only they can understand. The *grille cipher* is an oldschool technique that allows them to accomplish this. In our version of the grille cipher, the message to be encoded is written on an $N \times N$ grid row-wise, top to bottom, and is overlaid with a card with a set of holes punched out of it (the so-called grille).

The message is encrypted by writing down the letters that appear in the holes, row by row, then rotating the grille 90 degrees clockwise, writing the new letters that appear, and repeating this process two more times. Of course, the holes in the grille must be chosen so that every letter in the message will eventually appear in a hole *exactly once* over the process.

An example is shown below, where a boy in love sent the message "Send more monkeys" to the ~~grille~~ girl of his dreams. This message is encrypted as "noeesrksdmnyemoj", after he added a random letter "j" to fill out the grid.



Your task is to take an encrypted message and the corresponding grille and decrypt it. However, the grille given to you might be invalid, i.e., the holes used do not allow every location in the grid to be visible exactly one time during the encryption process. If this is the case, then you must indicate that you can't decrypt the message.

Input

The first line of the input contains a positive integer $N \leq 10$ indicating the size of the grid and grille. The next N lines specify the grille, using '.' for a hole and 'X' for a non-hole. The last line contains the encrypted message, consisting solely of lowercase alphabetic characters. The number of characters in this line is N^2.

Output

Output the decrypted text in a single line with no spaces, or the phrase "`invalid grille`" if the grille is invalid.

Examples

input	output
4 XX.X X.X. XXXX .XXX noeesrksdmnyemoj	sendmoremonkeysj
4 .XX. XXXX XXXX .XX. abcdefghijklmnop	invalid grille
2 X. XX aybb	baby

Challenge of Zelda

Link, in order to please princess Zelda, has to solve one final challenge proposed by the princess. The challenge is a puzzle that takes place in a room full of gargoyles (stone statues with two faces, on two opposing sides of their head), mirrors, and obstacles. There is a door that leads to Zelda, and Link must unlock it. On that door is written, in an ancient tongue, the secret to opening the door:

"Every face of every gargoyle shall see a face of a gargoyle."

This means that the gargoyles must be rotated in such a way that there is a path for a beam of light to connect each gargoyle's face to another gargoyle's face (possibly its own). The beam of light is reflected by mirrors.

The floorplan of the room can be described as a rectangular $N \times M$ grid of cells:

- A dot ('.') represents an empty cell.
- A hash ('#') represents an obstacle.
- A slash ('/') represents a double-sided mirror, as does a backslash ('\') .
- A character 'V' represents a gargoyle with two faces facing top and bottom.
- A character 'H' represents a gargoyle with two faces facing left and right.

In addition to the '\' and '/' mirrors, the room is surrounded by walls of mirrors. The following common sense about light is assumed:

- Light travels in a straight line through empty cells.
- Two beams of light can intersect without interfering with each other.
- A '\' mirror reflects light coming from the top/bottom/left/right to the right/left/bottom/top. A '/' mirror reflects light coming from the top/bottom/left/right to the left/right/top/bottom.
- Light is reflected by 180 degrees when it hits a wall (walls are all mirrors).
- Light is blocked by obstacles and gargoyles.

Link may rotate any gargoyle by 90 degrees. As time is running short, he wants to know the minimum number of gargoyles that have to be rotated in order to unlock the door.

Input

The first line of input contains two space-separated integers N and M ($1 \leq N, M \leq 500$), the dimensions of the room.

Each of the next N lines contains a string s of length M with the characters described above. This is the floorplan of the room.

Output

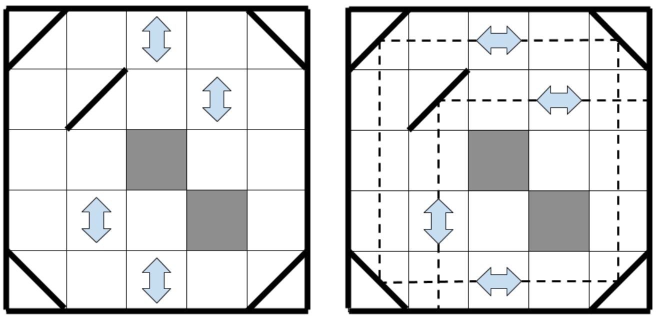
Output a single integer, which is the minimum number of gargoyles that have to be rotated in order to unlock the door. If the puzzle has no solution, output `-1`.

Examples

input	output
5 5 /.V.\ ./.V. ..#.. .V.#. \\.V./	3
2 5 V...\ H...V	-1
2 2 VV VV	0

Explanation

The puzzle from the first sample input is displayed in the following figure, with the initial configuration on the left and a solution on the right. Three gargoyles are rotated to solve this puzzle.



Dadud Inudaed

You and your friend C.J. Rofling are studying magic arts at the Hogwarts School of Witchcraft and Wizardry. After lots of practice and learning, the two of you discovered that the same magic spell effect can be produced by not just one, but in fact by many different magic words.

If two magic words produce the same effect, they are called *harmonic*. Your conjecture is that two (unordered) magic words A and B are harmonic if and only if they are of equal length and the following conditions hold for all characters in the words:

- If two characters of A are equal, then the corresponding characters of B are also equal. Formally, whenever we have $A_i = A_j$ for two positions i and j in word A, then $B_i = B_j$ must also hold.
- If two characters of A are different, then the corresponding characters of B are also different. Formally, whenever we have $A_i \neq A_j$ for two positions i and j, then $B_i \neq B_j$ must also hold.

Now you are wondering how many of the spells that you've learned produce the same magical effect. You are given a list of common magic words, and your task is to count the number of harmonic word pairs.

Input

The first line of the input contains an integer N ($1 \leq N \leq 100\,000$), the number of magic words. After this, there are N lines, each containing a string: the magic words. Each string consists of characters A-Z and has a length of at most 50 characters.

Output

Print a single integer: the number of harmonic pairs.

Examples

input	output
6 AAB ABKA SSG TSGT ZZZZ KEAK	4
2 DADUDINUDAED AVADAKEDAVRA	1

Explanation

The harmonic pairs in the first sample input are (`AAB`, `SSG`), (`ABKA`, `TSGT`), (`ABKA`, `KEAK`) and (`TSGT`, `KEAK`).

Edit Peptide

Palindromic peptides prevail! At least according to Anna, but what else can you expect from a chemist with a palindromic name. Peptides are chemical compounds consisting of $N \geq 2$ amino acids linked in a chain. We number the amino acids from 0 to $N-1$ according to their position in the chain.

The reason why Anna is so enthusiastic about palindromic peptides is that their special spatial structure allows chemists to induce a reaction which swaps a pair of amino acids in the chain: acid 0 can be swapped with acid $N-1$, acid 2 can be swapped with acid $N-2$, and so on. However, editing the molecule structure too much may make the peptide unstable, thus it is not allowed to perform more than K swaps.

Each amino acid has some potential energy P_i, represented by a nonnegative integer. Anna wants to perfom Q experiments. In each experiment, she wants to maximize the total potential energy of the amino acids over some contiguous segment of the chain. That is, for a given interval $[l, r]$ $(0 \leq l \leq r < N)$, she wants to maximize the sum $P_l + P_{l+1} + \ldots + P_r$ by performing at most K palindromic swaps.

Anna starts each experiment with a new peptide molecule, and she uses the same type of molecule for each experiment. Given the description of the peptide and Q intervals, your task is to compute the maximum potential energy that Anna can achieve in each case. You have to compute the answers *online*, that is, you have to solve the problem for an interval first before learning the next interval (see the Input section for details).

Input

The first line of the input contains three integers N, K and Q $(2 \leq N, Q \leq 200\,000$ and $0 \leq K \leq N)$, the size of the peptide chain, the maximum number of swaps and the number of experiments.

The second line contains N integers P_i $(0 \leq P_i \leq 10^9)$, the potential energies of the acids.

Each of the next Q lines contains two integers.

- The first line contains the boundaries l and r for the first experiment $(0 \leq l \leq r < N)$.
- In the j-th line $(2 \leq j \leq Q)$ there are two integers s_j and t_j $(0 \leq s_j, t_j \leq 10^{18})$. Let a_{j-1} denote the correct answer to the $(j-1)$-th question. Then, the value of l and r for the j-th experiment are $s_j \oplus a_{j-1}$ and $t_j \oplus a_{j-1}$, respectively.

Here, $\oplus$ denotes the binary XOR operator.

Output

Print Q lines, where the j-th line $(1 \leq j \leq Q)$ contains the integer a_j, the maximum sum of potentials in the j-th experiment.

Examples

input	output
7 1 5 4 4 0 6 1 1 0 0 2 8 15 18 22 12 10 20 20	9 16 12 16 1

Explanation

The first interval is $[l, r] = [0, 2]$. The sum of potentials without any swaps is $4 + 4 + 0 = 8$ over this segment, which can be improved to $a_1 = 4 + 4 + 1 = 9$ by swapping acids 2 and 4.

The second interval is $[l, r] = [8 \oplus 9, 15 \oplus 9] = [1, 6]$. The sum of potentials without swaps is 12 which can be improved to $a_2 = 16$ by swapping acids 0 and 6.

The third interval is $[l, r] = [18 \oplus 16, 22 \oplus 16] = [2, 6]$. The sum of potentials without swaps is 8. There are two ways to improve this, either by swapping acids 0 and 6 or by swapping 1 and 5. Since $K = 1$, we are only allowed to make one swap, and swapping 0 and 6 is better, so $a_3 = 12$.

The fourth inteval is $[l, r] = [12 \oplus 12, 10 \oplus 12] = [0, 6]$, the whole peptide chain. The sum of potentials is 16 which cannot be improved by swapping a pair of acids, so $a_4 = 16$.

The fifth interval is $[l, r] = [20 \oplus 16, 20 \oplus 16] = [4, 4]$. The potential of acid 4 is 1, which cannot be improved by swapping acids 4 and 2. So we have $a_5 = 1$.

Fiji Tours

After successfully competing at ICPC Hungary 2023, you finally decided to go on a vacation to Fiji Island. There are N locations of interest in Fiji that you are planning to visit over your stay. Locations are numbered from 0 to $N-1$. The locations are connected by $N-1$ bidirectional roads. It is possible to get from every location to any other location by walking the roads.

You want to utilize your time well by visiting a set of locations right on the day of your arrival. You will arrive at location X and by the end of the day you have to get to your hotel at location Y. You want to take a tour starting at location X and ending at location Y, traveling along the roads and visiting some locations. It is allowed to traverse the same road multiple times, and visit the same location multiple times (including visiting locations X and Y multiple times).

You are wondering: how many different tours can you plan for this day? You don't mind visiting the same location many times, so two tours are considered different if the two *sets* of visited locations are different.

Input

The first line of the input contains three integers N $(1 \leq N \leq 200\,000)$, X and Y $(0 \leq X, Y < N)$.

The following $N-1$ lines describe the roads. Each row contains two integers U and V $(0 \leq U, V < N, U \neq V)$, indicating that there is a road between locations U and V.

Output

Print a single line containing the number of different tours. Since the answer can be large, output it modulo $10^9 + 7$.

Examples

input	output
7 1 4 1 6 4 6 1 0 4 2 1 5 3 4	16

Explanation

Every tour from 1 to 4 contains locations 1, 6 and 4. It is optional to visit some of locations 0, 5, 3 or 2, independently of each other. Therefore, the number of different tours is $2^4 = 16$.

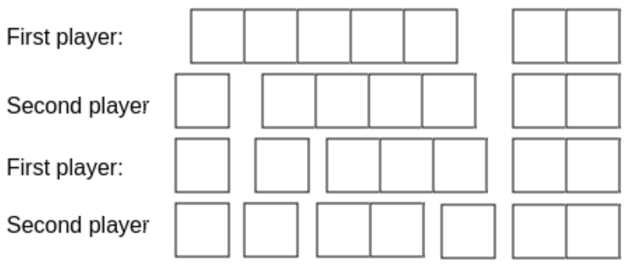
Note that tours like $1 \to 6 \to 4$ and $1 \to 6 \to 1 \to 6 \to 4$ are not considered different, as they visit the same set of locations $\{1, 4, 6\}$.

Game of Strips

At Elwyn's university, visiting classes is mandatory and he is not allowed to skip any of them. Luckily for him, his lecturer, Professor Tibor von Minerale does not mind if students are not paying full attention in his class. They take full advantage of this by playing games with each other instead of studying.

Elwyn's favourite game is played by two players. Initially, there is a paper strip of size $1 \times A$ on the table which contains $A \geq 3$ square-shaped cells in a single row. The players take turns in choosing a paper strip from the table which has at least 3 cells, and tear it into two paper strips along an edge between two cells. The game ends when the next player is not able to make a valid move, and the player who made the last move wins.

However, Elwyn quickly figured out that this game does not make much sense, as the same player (the first or the second, depending on A) will always win, irrespective of how they play. So he made an improvement: the players are not allowed to tear a paper strip into two equal parts. Other than that, they are free to choose the dividing edge. Specifically, it is not allowed to tear a strip of size 1×2 into smaller pieces. For example, let $A = 7$. A possible sequence of moves is shown in the following figure:



Following the last move of the second player, there is no valid move for the first player to make. Thus, the second player is the winner of this game. Note that after the second move, the first player is not allowed to tear the strip of size 4 into two strips of size 2 each and win the game.

There is a reason why Elwyn likes this game so much: he's already figured out that one of the two players always has a winning strategy: they can force a win no matter what moves the other player makes. In fact, he knows very well that for $A = 7$, the second player can force a win irrespective of how the first player plays. But sometimes it is worth playing first, for example, for $A = 6$, it can be proved that the first player has a winning strategy.

Today Elwyn is even more bored than usual at the lecture, so he decided to make the game a bit more interesting: initially, there will be two strips of paper with sizes $1 \times A$ and $1 \times B$ on the table (otherwise, the rules are unchanged).

He looks at his exercise book and finds that the maximum size of the strips he can produce is N. Now he wonders: how many (ordered) pairs (A, B) exist such that $3 \leq A, B \leq N$, and when playing the game on two strips of sizes A and B, the second player has a winning strategy. Hopefully you've payed more attention in your classes than Elwyn did, and can answer this question for him.

Input

The input is a single line containing the integer N ($3 \leq N \leq 250\,000$), the maximum size of the two paper strips.

Output

Print a single line containing an integer: the number of (A, B) pairs for which the second player can win.

Examples

input	output
3	1
6	6

Explanation

In the first sample case, the only possible pair of strip sizes is $(A, B) = (3, 3)$. In this case, the second player has a winning strategy, so the answer is 1.

In the second sample case, there are $4 \cdot 4 = 16$ possible (A, B) pairs. Out of these, the second player has a winning strategy in the following 6 cases: $(3, 3), (3, 6), (4, 4), (5, 5), (6, 3), (6, 6)$.

Handball Training

Handball is a game of skills and brain. The innovative coaching staff of Pick Szeged developed a new training exercise to improve the fast thinking and passing accuracy of their players. There are N players numbered from 0 to $N-1$. The players are standing clockwise around a circle, with player 0 holding a ball.

The coach repeadetly calls out one of the following two instructions:

1. a number T, indicating that the ball is to be passed to the player who is T positions clockwise from the current player holding the ball, wrapping around the circle if necessary. If T is negative, then the pass is to the counter-clockwise direction. If T is 0 (or $N, -N, 2N, -2N$, etc.) then the current player throws up the ball and then catches it. This still counts as a pass.
2. the phrase `undo` M, indicating that the last M passes should be undone and the ball should be passed back to the player who was holding it M passes before. Note that `undo` commands never undo other `undo` commands; they just undo instructions of the first type.

For example, if there are 5 players, and the coach calls out the instructions `8 -2 3 undo 2`, the ball is passed from player 0 to player 3; then from player 3 to player 1; then from player 1 to player 4. Finally, the `undo 2` instruction results in the ball being thrown back from player 4 to player 1 and then from player 1 back to player 3.

Your task is to write a program that determines the last player holding the ball from a list of instructions by the coach.

Input

The first line of the input contains two positive integers N and K ($1 \leq N \leq 30$, $1 \leq K \leq 100$) indicating the number of players and the number of instructions, respectively.

The second line contains K instructions. Each instruction is either an integer T ($-10\,000 \leq T \leq 10\,000$), indicating how many positions to throw the ball clockwise, or `undo` M ($M \geq 1$), indicating that the last M throws should be undone. The commands never undo beyond the start of the training.

Output

Print the player with the ball at the end of the training.

Examples

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

Increasing Elo

Hans is an avid chess player who would do *anything* (well, anything except cheating, right?) to increase his Elo rating, together with his skill in the game. Monitoring the long-term improvement of one's playing strength is not an easy task: one may accidently have some bad games, drop some Elo rating points, which they would immediately win back in a subsequent set of games.

Hans came up with a new system to measure his performance over time: given N chess games played by Hans, he assigns a *unique* score to each game, which is a number between 1 and N (inclusive). The scores are assigned so they represent the relative differences in Hans' Elo ratings following each game. For example, if $N = 3$ then the scores $1, 3, 2$ depict that his Elo was the lowest after the first game, the highest after the second game and the second highest after the third game.

Hans then chooses a positive integer K, representing his tolerance for stagnating Elo ratings. He claims that his improvement is satisfactory over the N games if the score of any game is strictly greater than the *minimum* score of the previous K games. Formally, denote the scores of the games by $s_1, s_2, \ldots, s_N$. Hans is satisfied if for every i from $k + 1$ to N (inclusive) $s_i > \min(s_{i-1}, s_{i-2}, \ldots, s_{i-k})$ holds.

For example, the scores $1, 3, 2$ are satisfactory for $K = 2$ as $2 > \min(1, 3)$. However, the same scores are not satisfactory for $K = 1$ as out of $3 > \min(1)$ and $2 > \min(3)$, the second inequality does not hold.

Hans is wondering: given the value of N and K, how many sequences of scores exist which are satisfactory.

Input

The input is a single line containing N and K ($1 \le K \le N \le 1\,000\,000$), the number of games and the stagnation tolerance.

Output

Print a single line containing the number of different satisfactory score sequences. Since the answer can be large, output it modulo $10^9 + 7$.

Examples

input	output
3 2	4
3 1	1
4 2	10

Explanation

In the first sample case, the four satisfactory sequences are $1, 2, 3$; $1, 3, 2$; $2, 1, 3$; and $3, 1, 2$. Note that the scores in each sequence must be unique.

Journey Discounts

You are planning to spend your holidays touring Europe, staying each night in a different city for N consecutive nights. Luckily for you, your favourite website for booking accomodations offers special discounts for such journeys.

You have already chosen the hotel you want to stay in for each city, so you know the price P_i of the room you'll be staying at during the i-th night of your holidays ($1 \le i \le N$). After staying for a night in a hotel you booked through the website, you are awarded one bonus point, and you can exchange K of these points in your account at any time for a free night in any hotel. The night you spend for free using bonus points won't give you another bonus point. There is no limit on the number of bonus points you can collect in your account.

For example, consider the case with $N = 6$ and $K = 2$ where the prices for the rooms are $P_1 = 10, P_2 = 3, P_3 = 12, P_4 = 15, P_5 = 12$ and $P_6 = 18$. After paying for the first four nights you would have four points in your account, which you could exchange to stay for free the remaining two nights, paying a total of $P_1 + P_2 + P_3 + P_4 = 40$ for your accommodation. However, if after the first three nights you use two of the three points you earned to stay the fourth night for free, then you can pay for the fifth night and use the final two points to get the sixth one for free. In this case, the total cost of your accommodation is $P1 + P2 + P3 + P5 = 37$, so this option is actually more convenient.

You want to make a program to find out what the minimum possible cost for your holidays' accommodation is.

Input

The first line of the input contains two integers N and K ($1 \le N, K \le 100\,000$), representing the number of nights and the number of points you need in order to get a free night.

The second line contains N integers $P_1, P_2, \ldots, P_N$ ($1 \le P_i \le 10^4$), representing the prices of the rooms you will be staying at during your holidays.

Output

Print a line with one integer, the minimum cost of your accommodation for all nights.

Examples

input	output
6 2 10 3 12 15 12 18	37
6 1 10 3 12 15 12 18	25
5 5 1 2 3 4 5	15

Keeping It Balanced

Four friends are playing foosball. Each of them has a skill level which is represented by an integer number: the higher the number, the better the player is.

The four friends want to form two teams of two players each. For the game to be more exciting, they want the skill level of the teams to be as close as possible. The skill level of a team is the sum of the skill levels of the players in that team.

Although they are very good foosball players, these friends are not so good at other things, like Math or Computing. Can you help them find the smallest possible difference between the two teams' skill levels?

Input

The input consists of a single line that contains four integers A, B, C and D, representing the skill levels of the four players ($0 \leq A, B, C, D \leq 10^4$).

Output

Output a line with an integer representing the smallest difference between the skill levels of the two teams.

Examples

input	output
4 7 10 20	7
0 1000 0 1	999
2 1 4 3	0

Lazy World Map

The new expansion to your favourite RPG game has just been released! It is called *Infernal Creatures' Paradise: Clanwars* (ICPC), and this new expansion finally enables the players to join factions and participate in cool social activities like massacres and mass slaughters.

The world map of the game can be modeled as a grid of square cells, where each grid belongs to one of the factions. Traveling in the game world takes a lot of time, so you don't even know the exact number of factions present in the game. However, after playing for a while, you realized that the developers were lazy, and just procedurally generated all of the factions using a simple algorithm:

- First, they took an $N \times M$ grid of cells and assigned each square a character between A and Z.
- Then, based on this map piece, an $NA \times NB$ grid was formed by placing copies of the original grid A times below and B times in parallel to the original piece.
- Finally, the factions are formed using the following rule: two squares belong to the same faction if and only if there is a path between them consisting of horizontally or vertically adjacent squares, with each square having the same character.

Knowing the size of the world map and the layout of the original $N \times M$ grid piece, you want to compute the total number of factions present in the game.

Input

The first line contains four integers N, M, A and B ($1 \leq N, M \leq 20$, $1 \leq A, B \leq 10^9$).

The following N lines describe the original piece of the grid. Each row contains a string of length M with uppercase latin characters.

Output

Print a single line containing the number factions. Since the answer can be large, output it modulo $10^9 + 7$.

Examples

input	output
3 2 2 5 AB BA AA	27

Explanation

The world map is the following grid of 6×10 cells:

```
ABABABABAB
BABABABABA
AAAAAAAAAA
ABABABABAB
BABABABABA
AAAAAAAAAA
```

Each of the 20 B characters belong to a different faction, similarly to the 5 A characters in the first row. There are two more factions, one consists of every cell with character A in the second, third and fourth rows, and the other consists of the cells with character A in the last two rows.

So there are a total of $20 + 5 + 2 = 27$ factions.