

Problem A. Antiprefix Function

Input file: **standard input**
Output file: **standard output**
Time limit: 2.5 seconds
Memory limit: 256 megabytes

Let the **antiprefix function** of the string s be an array $[q_1, q_2, \dots, q_n]$, where q_i is the greatest integer $j \in [0, i - 1]$ such that $s[1..j]$ differs from $s[i - j + 1..i]$ in **every** position.

For example, for the string 011001, its **antiprefix function** is $[0, 1, 1, 2, 3, 4]$.

You are given a **binary** string s of length n , find its antiprefix function.

Input

The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The first line of each test case contains one integer n ($1 \leq n \leq 10^5$) — the length of the binary string s .

The second line of each test case contains a binary string consisting of n characters.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output n integers : the antiprefix function of the string s .

Example

standard input	standard output
2	0 1 1 2 3 4
6	0 1 2 3 4 5 6 7 8
011001	
9	
010101010	

Problem B. GCD LCM SUM

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

In the land of Westeros, there lived two noble families, the Lannisters and the Starks. The Lannisters were known for their wealth and cunning, while the Starks were known for their honor and bravery.

One day, a wise maester approached both families with a mathematical problem. He told them that the solution to the problem would determine the worthiness of the families to rule the Seven Kingdoms.

The problem was to find two numbers, a and b , such that their greatest common divisor and their least common multiple would add up to a given number, x . Moreover, the difference between the two numbers should be as small as possible, and a must be less than or equal to b .

Input

The first line contains the number of test cases t ($1 \leq t \leq 100$)

The only line of each test case contains one integer x ($2 \leq x \leq 10^9$)

Output

For every test case, output a and b ($1 \leq a, b \leq x$) such that those two integers satisfy the conditions.

It can be proven that the solution exists for every x .

Example

standard input	standard output
6	3 6
9	3 4
13	1 16
17	4 5
21	9842322 18512939
777777779	130758 229141
809784331	

Problem C. Rooks

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 256 megabytes

You have an $n \times n$ chessboard and k rooks. Rows of this chessboard are numbered by integers from 1 to n from top to bottom and columns of this chessboard are numbered by integers from 1 to n from left to right. The cell (x, y) is the cell on the intersection of row x and column y for $1 \leq x \leq n$ and $1 \leq y \leq n$.

A rook attacks all cells that are in its row or column.

You have to remove some rooks from the board such that there is at least **one empty** cell which is not attacked by any rook.

What is the **minimum** number of rooks you have to remove ?

Input

The first line contains a single integer t ($1 \leq t \leq 100$) — the number of test cases.

The first line of each test case contains two integers n ($1 \leq n \leq 50$) the size of the chessboard and k ($0 \leq k \leq n^2$) the number of rooks.

Each of the next k lines of each test case contains two integers x and y ($1 \leq x, y \leq n$) — there is a rook in cell (x, y)

Output

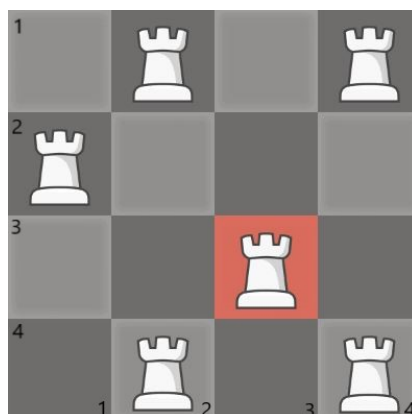
For each test case, output one integer m ($0 \leq m \leq k$), the minimum number of rooks to remove.

Example

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

Note

Consider example. The board will look like the following picture. You have to remove the rook colored in red in order to obtain a square which is not attacked by any rook.



Problem D. Casino I

Input file: **standard input**
Output file: **standard output**
Time limit: 12 seconds
Memory limit: 256 megabytes

This is an interactive problem.

Vertex Cover team got tired of problem solving and decided to go to the casino to have some fun and play Blackjack. In this casino, the goal of Blackjack is to collect cards that total to more points than the dealer, **without exceeding 1**.

Vertex Cover members are not good in betting, so they asked for your help.

You will play several rounds, In each round, each one of you and the dealer have initially a score of 0, the dealer will ask you if you want a card with a randomly and **uniformly** generated number between 0 and 1 written on it. If you say *hit*, the dealer will give you that card. If your total hand (the sum of numbers written on the cards) becomes **more than 1**, you get *bust* and lose. Otherwise, the dealer will ask you again. If you say *stand*, your turn is over with a total T , and dealer's turn starts. The dealer will get a card and do the following :

1. If his total points is less than or equal to T he will get another card and repeat again.
2. If his total points is strictly more than 1, you win the round.
3. If his total points is strictly more than T but no more than 1, you lose the round.

These are the rules, You have to play $n = 10^5$ rounds, if you win at least 42% of the rounds, you will win and get the mega millions jackpot. Good luck !

Input

A single line containing a single integer $n = 10^5$: The number of games.

Interaction Protocol

After reading the number of games n . You will play 10^5 rounds, each round starts when the interactor outputs G .

In each turn of the **current** round, you have two possible movements:

- 1 : The interactor will generate a uniformly random variable between 0 and 1, and adds it to the sum.
 - If the current sum exceeds 1, the interactor will output L
 - Otherwise, it will output " $S\ r$ " where r is the current sum.
- 0: Your turn is over, and the dealer will play **optimally**. Then the interactor will give you the winner of the round as follow:
 - It will output L if you lose.
 - It will output W if you win.

You have to use a flush operation right after the movement. For example, in C++ you should use the function `fflush(stdout)` or `cout.flush()`, in Java — `System.out.flush()`, and in Python — `sys.stdout.flush()`.

Example

standard input	standard output
3	G
1 1 1 0	S 0.265346
1 1 1 1 0	S 0.789890
1 1 1	S 0.932484
	W
	G
	S 0.070631
	S 0.182233
	S 0.726913
	S 0.827251
	L
	G
	S 0.785957
	S 0.896195
	L

Note

- There's no limit to how many cards you can ask for, but once your hand total's higher than 1, you bust and the dealer gets your bet.
- The example above is **not a complete** example, as $n = 3$, and not $n = 10^5$, but it is included **to understand** how the game should work.
 - For the first game, you played 3 turns, less or equal than reaching a sum $S = 0.932484 \leq 1$. And with that, you skipped.
 - For the second game, you played 4 turns, reaching a sum $S = 0.827251 \leq 1$. And with that, you skipped
 - For the third game, you played 3 turns, but you reached a sum $S > 1$. This can be viewed as you lost before skipping.
- As the dealer's sum is not seen, and will not be either in other test cases, you **cannot see** how you won/lost. We will detail the scenarios:
 - For the first game, the dealer got $S_{\text{dealer}} = 0.549321$ after the first turn, and then $S_{\text{dealer}} = 0.86321$. Finally he took a card again, and got $S_{\text{dealer}} = 1.332513 > 1$, and so he lost, and you **won**.
 - For the second game, the dealer got $S_{\text{dealer}} = 0.836332 > S = 0.827251$ after the first turn, and so he won, and you **lost**.
 - In the third game, After getting $S = 0.896195$, but you chose to take a card again, and you got $S = 1.62371 > 1$. And with that, you **lost**.

Problem E. Casino II

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 256 megabytes

Casino I and II are two different problems, but you should read Casino I in order to understand the statement.

Oussama got angry because the mega millions jackpot was just a big box of candies. Taking revenge, he decided to break the dealer's system. Oussama got the leak and realized that the reals given by the dealer are hard-coded on a big array A .

In fact, starting from $i \leftarrow 0$ **before** the first game, each random number is generated by the dealer as follow:

1. generate A_i as the random number.
2. increment i : $i \leftarrow i + 1$

Oussama got the array and now knows the sequence of numbers that will get generated. He was wondering how many rounds he can win at most while knowing the values of the array if he plays exactly k rounds. Note that in each game, you cannot skip your first move : you should take at least one card.

Oussama is busy eating candies. Please help him !

Input

You will be given an integer n ($10 \leq n \leq 10000$) : The length of the array and k ($1 \leq k \leq 1000$) : The number of rounds you have to play.

Then you will be given n reals between 0 and 1. Note that the reals are randomly and uniformly generated.

Output

Output m : The maximum number of rounds you can win out of k rounds.

Examples

standard input	standard output
10 2 0.43436 0.14373 0.60797 0.02197 0.76351	0 0.52623 0.41605 0.08398 0.75709 0.67884
10 2 0.34985 0.81288 0.34235 0.54262 0.49512	1 0.62882 0.60371 0.26639 0.29891 0.42121

Note

- For each test case, the index i will not be reset after each game, It will in fact increment indefinitely.
- It is guaranteed that the hard-coded array is **sufficient** to play k rounds regardless of the strategy of playing.

Problem F. Expected Runtime

Input file: **standard input**
Output file: **standard output**
Time limit: **3.5 seconds**
Memory limit: **512 megabytes**

Rami was playing with the following algorithm. He chooses a positive integer $n \in \mathbb{N}^*$. He begins with $R = 1$ and $k = 0$ and while $R \bmod n \neq 0$, he:

1. Chooses a random number $a \in \{0, \dots, n-1\}$
2. He sets $R \leftarrow R \times a$
3. He increments $k : k \leftarrow k + 1$

As k denotes the number of iterations on the loop, he wants to know the expected value of k .

Input

- The first line contains an integer $1 \leq T \leq 10^3$: the number of test cases.
- Each test case is represented by a line containing a single integer $1 \leq n \leq 10^6$

Output

For each test case, output a single real k : the expected runtime of the algorithm.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-6} .

Formally, let your answer be a , and the jury's answer be b . Your answer is accepted if and only if $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-6}$.

Example

standard input	standard output
6	2.000000
2	3.000000
3	5.333333
10	9.000000
25	9.296296
100	11.000000
1024	

Problem G. Find The Array

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

You are given four integers a, b, c, d . Find any array satisfying those 4 conditions :

1. The **maximum** of its prefix sum array is a .
2. The **minimum** of its prefix sum array is b .
3. The **maximum** of its suffix sum array is c .
4. The **minimum** of its suffix sum array is d .

The prefix sum array of an array $a[1..n]$ of size n is an array of size $n + 1$ defined as follow :

- $Pr[1] = 0$
- $Pr[i] = \sum_{k=1}^{i-1} a[k]$ for all $2 \leq i \leq n + 1$

The suffix sum array of an array $a[1..n]$ of size n is an array of size $n + 1$ defined as follow :

- $Sf[n + 1] = 0$
- $Sf[i] = \sum_{k=i}^n a[k]$ for all $1 \leq i \leq n$

Input

The first line contains one integer t ($1 \leq t \leq 10^3$) — the number of test cases.

Each test case contains 4 integers a, b, c, d ($-10^9 \leq a, b, c, d \leq 10^9$)

Output

For each test case :

- Output k ($1 \leq k \leq 1000$) : The size of the array
- Then, output an array a of size k ($-2 \cdot 10^9 \leq a[i] \leq 2 \cdot 10^9$) : Any array satisfying the conditions above.

If there is no answer output -1 .

Example

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

Note

In the second test case, the prefix sum array is : $[0, 3, -5, 1, 3, 2, 6, 0, 5]$ and the suffix sum array is : $[5, 2, 10, 4, 2, 3, -1, 5, 0]$.

Problem H. Balance The sequence

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

A sequence of brackets is called balanced if one can turn it into a valid math expression by adding characters '+' and '1'. For example, sequences $((()))$, $()$, and $((()()))$ are balanced, while $)()$, $((()$, and $((()))($ are not.

Yessine received a balanced sequence of brackets s of length n , unfortunately, someone took a contiguous subsequence $s[i..j]$ and reversed it.

Yessine wants to rebalance the sequence again, please help him to find any pair (i, j) such that if you reverse the subsequence $s[i..j]$, the sequence becomes balanced.

For example, if $n = 8$ and $s = ()()()$, to make s balanced, you can choose $i = 2$ and $j = 7$, s will become $((()()))$.

Input

The first line contains the number of test cases t ($1 \leq t \leq 10^3$)

The first line of each test case contains one **even** integer n ($2 \leq n \leq 5 \cdot 10^3$) — the length of the sequence s .

The second line of each test case contains a string consisting of n brackets.

It is guaranteed that the sum of n over all test cases does not exceed 10^4 .

Output

For each test case, print one line containing two integer : i and j ($1 \leq i < j \leq n$)

If there are multiple solutions, print any.

Example

standard input	standard output
2	2 7
8	1 8
$()()()$	
8	
$))))((($	

Problem I. Hall of Faces

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

The members of the Faceless Men possess the unique capability to physically change their faces, shapeshifting so that they appears as an entirely new person. Their headquarters are located in the House of Black and White, where the Hall of Faces exists.

Inside Hall of Faces there exists rows of skinned faces belonging to deceased individuals. Those faces are displayed and labeled according to their defining characteristics (a label example could be: fatty , silver-haired , one-eyed etc). Note that it's possible for the hall of faces to have multiple faces with the same defining characteristic (thus same label).

After accepting a certain mission Jaqen H'ghar went to the hall of faces to get the corresponding face for his new persona

To find the face he wanted, Jaqen H'ghar must first return the face he currently uses to his position in the hall of faces then pick a new face based on the defining characteristic he need for completing the job.

For the purpose of this problem, let's assume that the Hall of Faces is designed in a circular fashion and that all faces are placed in a single row on the wall. Can you determine the minimum number of faces Jaqen H'ghar must inspect before finding the face he wants?

If the face desired by Jaqen H'ghar is not present inside the hall of face output -1.

Note: Jaqen view faces in order, going either in the clockwise or counter clockwise direction before finding the one he wants

Input

The first line contains a single integer t ($1 \leq t \leq 100$) — the number of test cases.

The first line of each test case contains an integer n ($1 \leq n \leq 10^5$) the number of faces in the hall of faces along with string that represent the label of the face that Jaqen want.

Each face is represented by a corresponding label (different faces may have the same label). different label are seperated by a white space. Among the different label you are guaranteed to have only one having as value "Jaqen". This indicate the position of the face that Jaqen is currently using.

The sum of number of characters over all test case is guaranteed to be less than 10^6 and the length of each label is guaranteed to be less than 20 character long.

Output

After returning the face he is currently using to his position, output the minimum number of faces Jaqen H'ghar has to consider before finding the face he wants.

Example

standard input	standard output
3	2
5 one-eyed	2
red-haired Jaqen silver-haired one-eyed	red-haired
6 brown-eyed	
brown-eyed brown-eyed silver-haired one-eyed Jaqen long-nose	
3 thick-eyebrows	
Jaqen brown-eyed long-nose	

Problem J. Multiplicative Array

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 256 megabytes

Rami has a multiplicative array $A = [A_1, \dots, A_n]$ of size n . A multiplicative array is an array on which:

$$\forall u, v \in \{1, \dots, n\} / \gcd(u, v) = 1, \quad A_{uv} = (A_u \cdot A_v) \bmod M$$

with $M = 10^9 + 7$

He will apply the following algorithm:

- Set $A^{(0)} \leftarrow A$
- For $i \in \{1, \dots, m\}$:

$$\forall k \in \{1, \dots, n\}, \quad A_k^{(i)} \leftarrow \sum_{d|k} A_d^{(i-1)}$$

He wants to know the content of each element of the final array $A^{(m)}$, modulo $M = 10^9 + 7$.

Input

- The first line contains two integers n, m with:
 - $1 \leq n \leq 10^6$: The size of the array
 - $1 \leq m \leq 10^{18}$: The number of iterations.
- $0 \leq A_1, \dots, A_n \leq 10^9$: The content of the multiplicative array.

It is guaranteed that there exists i such that $A_i \neq 0$

Output

n integers $B_1, \dots, B_n$ with $B_i = A_i^{(m)}$ the content of the final array.

Examples

standard input	standard output
10 2 1 0 0 0 0 0 0 0 0 0	1 2 2 3 2 4 2 4 3 4
10 5 1 2 3 4 5 6 7 8 9 10	1 7 8 29 10 56 12 93 39 70

Note

- It is **guaranteed** that the given array will be multiplicative.
- In the first test case, *Rami* will apply two iterations of the algorithm above.
 1. In the first iteration, he will get the array $A^{(1)} = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$.
 2. In the second iteration, he will have:

$$A_k^{(2)} = \sum_{d|k} A_d^{(1)} = \sum_{d|k} 1$$

And with that, $\forall k \in \{1, \dots, n\}$, $B_k = A_k^{(2)}$ is equal to the number of divisors of k .

Problem K. Splitting Game

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

In the world of Westeros, two skilled knights, *Rami* and *Yessine*, were known for their strategic prowess and their love for games of wit and strategy. One day, they decided to test their skills against each other in the Splitting Game.

This game is a 2 players turn-based game played on some array A of size n , with each turn as follow:

- A player must choose $A_k > 0$ and remove it from the array.
- After removing A_k , the same player must create some array B whose elements are all **positive** and **strictly less** than A_k . Note that B can be **empty**.
- Finally, he will concatenate B to A

Rami starts first, and a player loses if he cannot make a move.

Given the array A , and assuming both players play optimal moves. Who will win?

Input

- $0 \leq n \leq 2 \times 10^5$
- $0 \leq A_1, \dots, A_n \leq 10^{18}$

Output

A single line containing:

- `Rami` if Rami will win.
- `Yessine` otherwise.

The judge system is case sensitive so make sure that your output follows the format indicated above

Examples

standard input	standard output
0	Yessine
5 1 1 1 1 1	Rami
3 1 2 2	Rami

Note

It can be proven that no matter the strategy of each player, the game will always be **finite**.

Problem L. Good Transformations

Input file: **standard input**
 Output file: **standard output**
 Time limit: 4 seconds
 Memory limit: 256 megabytes

Rami has started to learn 2 dimensional discrete geometry.

Given an infinite grid G , he will apply a linear transformation T to each point (i, j) .

A linear transformation T is a function that sends $(i, j) \rightarrow (i', j')$ with:

$$\begin{cases} i' = ai + bj \\ j' = ci + dj \end{cases}$$

With $a, b, c, d \in \mathbb{Z}$ its parameters which do not depend on (i, j) .

Also he divided the grid into square blocks each with dimensions $m \times m$, with the $(i, j)^{\text{th}}$ block defined as:

$$B_{i,j} \text{ is the set } \{(i \cdot m + u, j \cdot m + v), \quad u, v \in \{0, \dots, m-1\}\}$$

Furthermore, he defined the $(u, v)^{\text{th}}$ point of a block $B_{i,j}$ as the point $(i \cdot m + u, j \cdot m + v)$

The m -score $\chi_m(T)$ of a linear transformation T is defined as the number of pairs (u, v) such that:

1. $u, v \in \{0, \dots, m-1\}$
2. There exists integers $i, j, p, q \in \mathbb{Z}$ such that $T(p, q)$ is the $(u, v)^{\text{th}}$ point of the block $B_{i,j}$

Now, *Rami* is interested on a family of linear transformations that he called m -good transformations.

In fact, an m -good transformation is defined as a linear transformation T that:

- Maximizes $\chi_m(T)$ over all linear transformations.
- Has as parameters $0 \leq a, b, c, d < m$.

Rami wants to count the number of m -good transformations for a given m modulo $M = 10^9 + 7$.

Please help him.

Input

- $1 \leq T \leq 10^6$ the number of test cases.
- Each test case consists of a single line containing a single integer $1 \leq m \leq 10^6$

Output

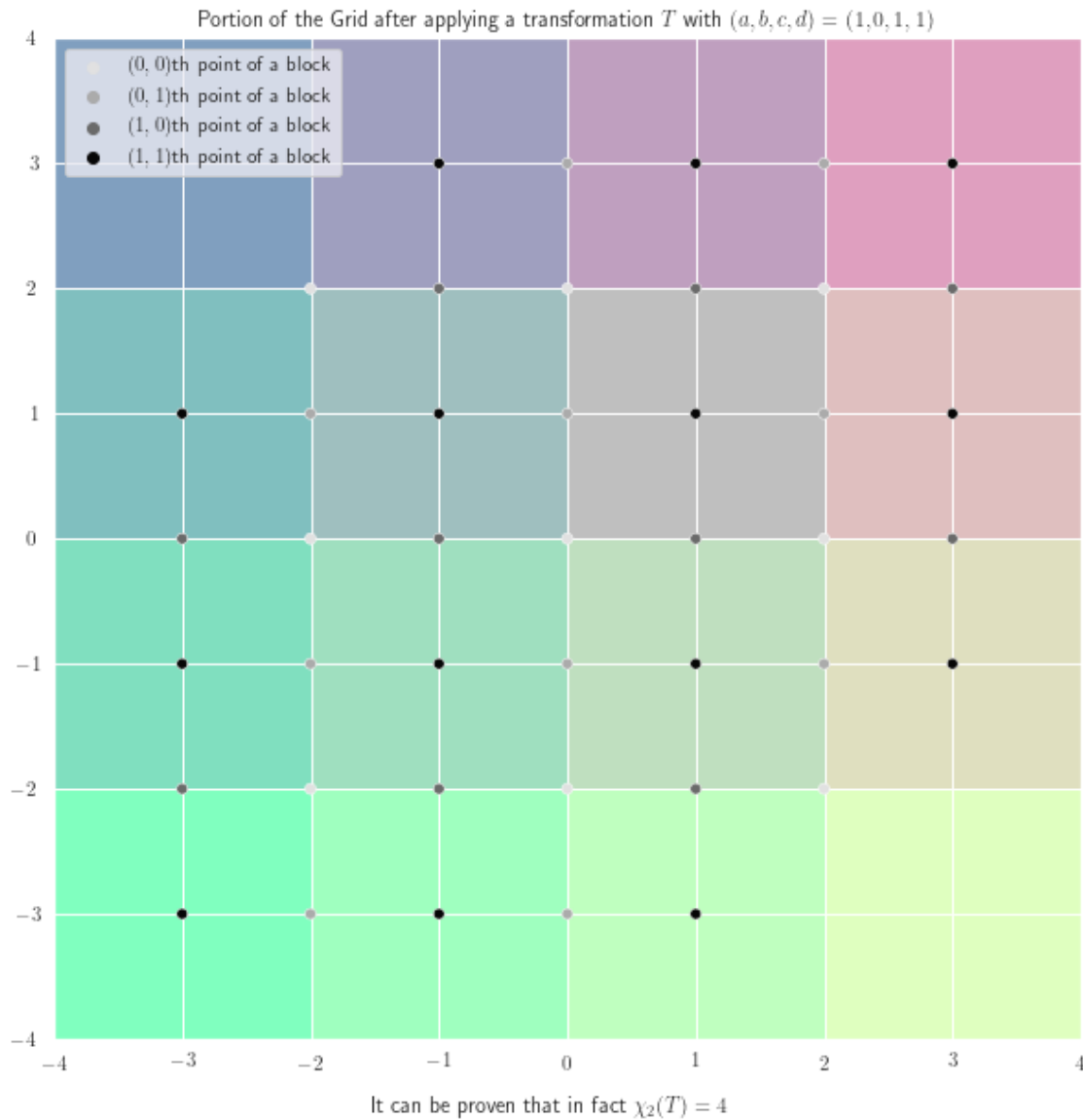
For each test case, output the number of m -good transformations modulo $M = 10^9 + 7$

Example

standard input	standard output
5	6
2	48
3	2880
10	300000
25	28800000
100	

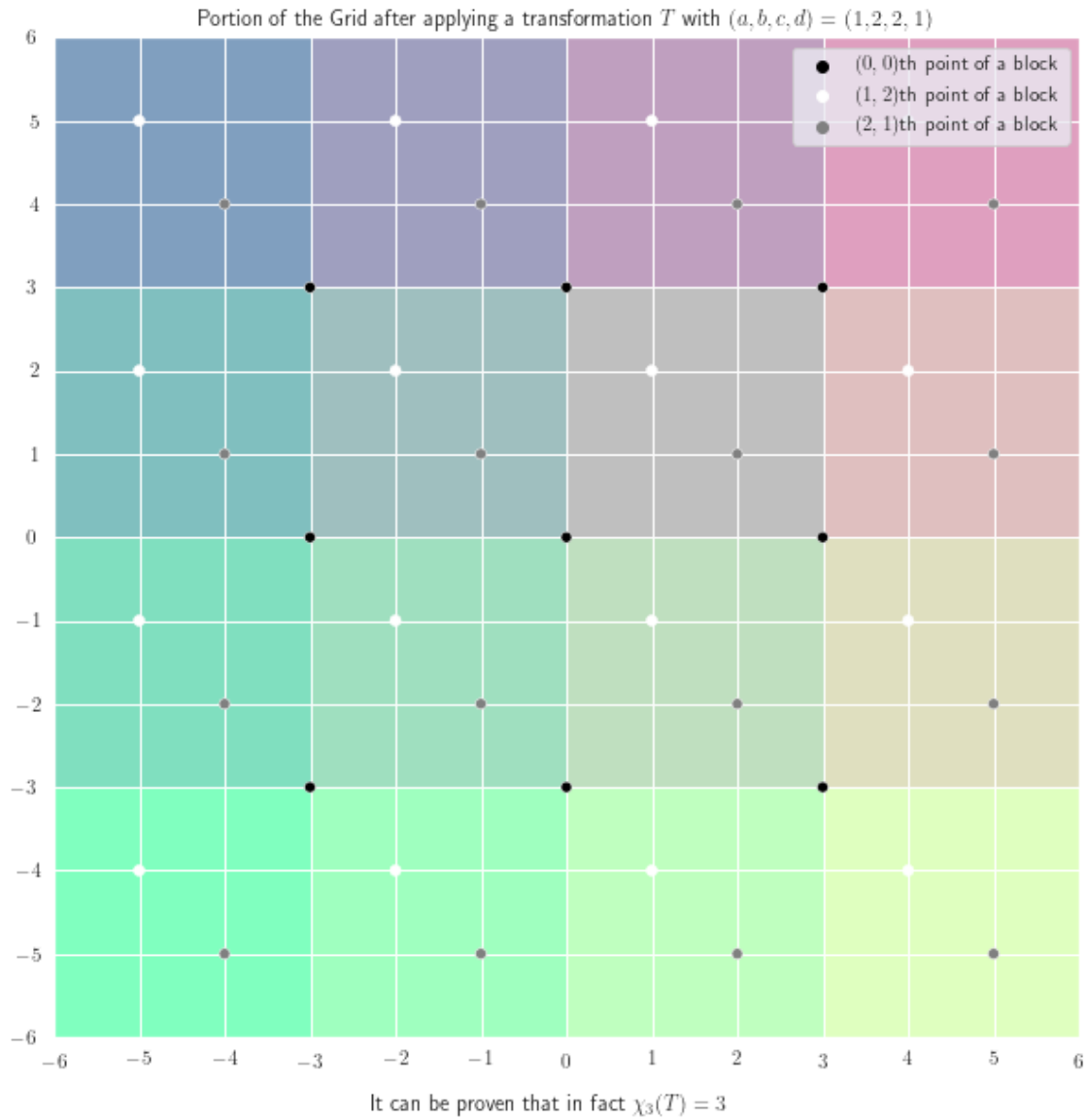
Note

This is a visualisation of a linear transformation T for $m = 2$. Each block has its own colour. Also the points generated by this linear transformation are shown.



It can be proven that this transformation is m -good.

This is another visualisation of a linear transformation T for $m = 3$.



It can be proven that there exists a linear transformation S with $\chi_3(S) > 3$. And so this linear transformation is not m -good.

Problem M. Colour the Banners

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

In the world of Westeros, the Kingdoms of Westeros were constantly vying for power and control, each represented by a different color on the banners that flew high above their castles. But when a strange and powerful enemy threatened to destroy them all, the lords and ladies of the kingdoms realized that they needed to work together if they were to survive.

To do so, they turned to an ancient and mysterious mage, known by the name *IronByte*, who was said to have the power to solve any problem. The mage presented them with a strange riddle, stating that they needed to choose the colors of their banners in such a way that there were no adjacent banners with the same colours, or face the consequences.

The problem with that, is that many kingdoms have already installed their banners, and they cannot be altered. The only solution is to set the colours of the banners for the remaining kingdoms so that there will not be a pair of adjacent kingdoms with the same colour.

Now, the banners are represented by a string S where S_i is:

- The colour of the banner of the i^{th} kingdom if S_i is an uppercase latin character
- A kingdom without a banner if $S_i = .$

Also, the banners form a **straight line**, so the adjacent banners are represented by the pairs (S_i, S_{i+1}) .

A final critical problem, is the budget Kingdoms is limited, so they should use as few colours as possible.

Please help them to determine the **minimal** number of colours they need to solve the riddle, or -1 if they will face imminent destruction.

Input

- The first line consists of an integer $1 \leq n \leq 10^5$, representing the number of Kingdoms
- The second line consists of a string S with length n , where:
 - $S_i \in \{A, \dots, Z\}$ if the banner of the i^{th} kingdom is installed and has as colour S_i
 - $.$ otherwise if the banner of the i^{th} kingdom is not yet installed.

Output

A single integer n representing the minimal number of colours, or -1 if it is impossible to have adjacent pairs with different colours.

Examples

standard input	standard output
5 RG..R	2
5 RG.BG	3

Problem N. The Volcano And the Dragon Egg

Input file: standard input
Output file: standard output
Time limit: 4 seconds
Memory limit: 256 megabytes

A member of the Warlocks of Qarth named Kovarro Khalil stumbled upon a rare opportunity while exploring the Freehold of Valyria. He discovered inside a volcano pit a dragon egg laying on a safe island of rock surrounded from all direction by magma.

Overjoyed with his finding the mage started casting a telekinesis spell to bring the egg to him, but unfortunately, the volcano suddenly became active and the magma level started rising at a rate of $1mm$ per millisecond.

The magma stocked inside the volcano is infinite and it will keep rising at the same rate until it reaches the dragon egg.

The warlock is aware that if the magma touches the egg, it may hatch and the telekinesis spell will be ineffective.

Given a bird eye view detailing the topology of the volcano pit and the position of the dragon egg, help the warlock determine how much time he still has before the magma touch the dragon egg.

The topology of the volcano pit will be represented by $n \times m$ grid where each cell indicate the elevation of that particular point above the magma level in mm (cells containing 0 represents the magma level at the time when the warlock started casting his spell)

notice that the egg can be placed in any altitude on top of the rocky island and it may be surrounded by a wall of rock in this case the only way that the magma reach the egg is when the magma level goes over the wall of rock.

Due to the magical properties of the rocky island, the rock won't melt when touched by magma.



Input

The first line contains 2 integer n and m ($3 \leq n, m \leq 10^3$) respectively the length and width of the provided view of the volcano pit.

The second line contains 2 integer x ($2 \leq x \leq n - 1$) and y ($2 \leq y \leq m - 1$) the coordinates of the rock on top of which the dragon egg is placed.

Then a grid of $n \times m$ numbers will follow representing the topology of the volcano pit at the time when the warlock started casting his spell ($0 \leq a[i][j] \leq 10^9$)

Each number represent the elevation of that point above the magma level while 0 represents points already swallowed by magma

It is guaranteed that the outer layer of the volcano pit contains only lava.

Output

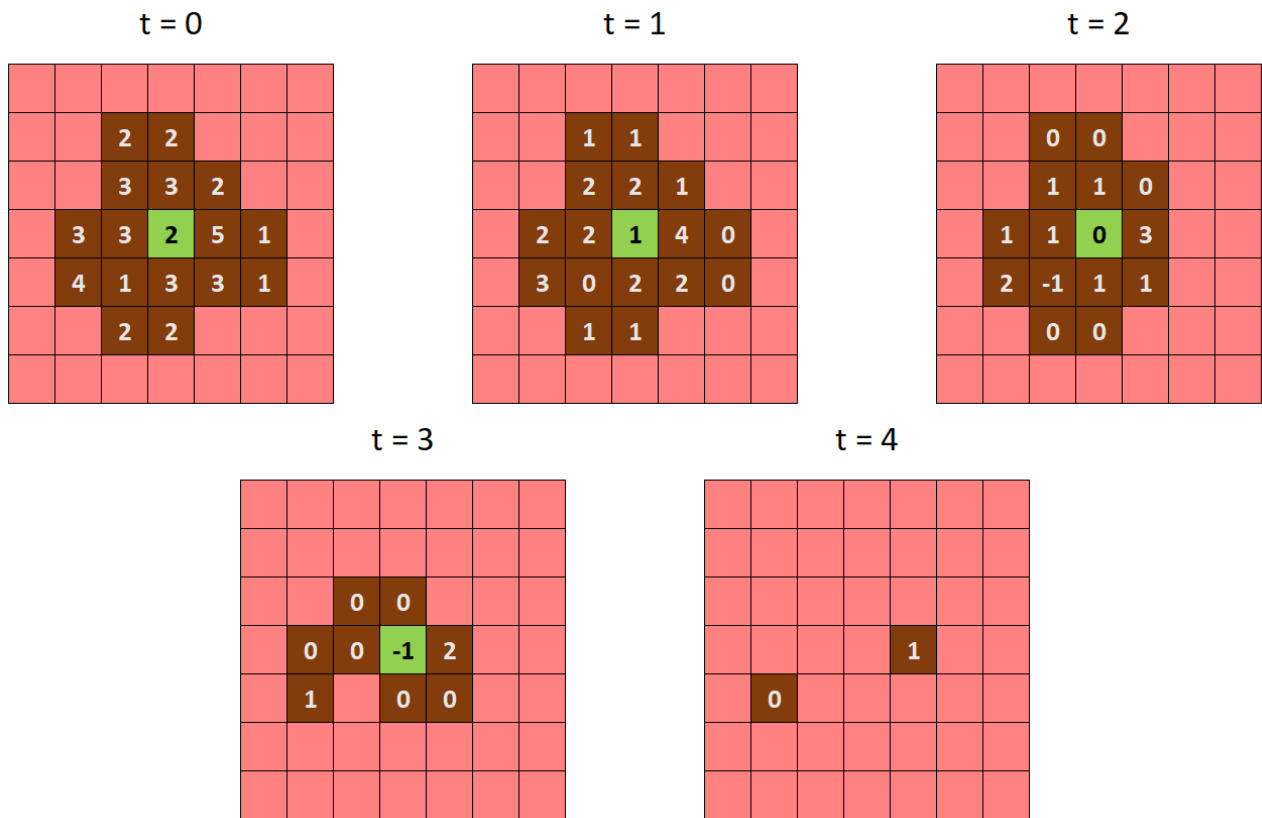
Help the warlock determine how much time he still has before the magma touch the dragon egg.

Examples

standard input	standard output
5 5 3 3 0 0 0 0 0 0 0 2 0 0 0 2 1 2 0 0 0 2 0 0 0 0 0 0 0	3
3 4 2 2 0 0 0 0 0 1 0 0 0 0 0 0	2
5 5 3 3 0 0 0 0 0 0 2 2 1 0 0 2 3 1 0 0 2 2 1 0 0 0 0 0 0	4
7 7 4 4 0 0 0 0 0 0 0 0 0 2 2 0 0 0 0 0 3 3 2 0 0 0 3 3 2 5 1 0 0 4 1 3 3 1 0 0 0 2 2 0 0 0 0 0 0 0 0 0 0	4

Note

Below is a simulation of the 4th test example



The answer is 4 since after exactly 4 millisecond the egg laying on top of the cell (4,4) will be submerged by magma