icpc International Collegiate Programming Contest

icpc.foundation

icpc North America South Central

The 2025 ICPC North America South Central Regional Contest

Division 2

Official Problem Set

Jane Street

icpc titanium
multi-regional sponsor

JETBRAINS

This page is intentionally left blank.

Problem A
Snakey String

Time Limit: 1 second Memory Limit: 1GB

A *snakey string* is a fancy rendering of an otherwise normal string of text. When a string is made snakey, it is placed on a 2D grid such that the following conditions are met:

- The first character of the string is on the first column of the grid, the second on the second column, ..., and the last on the last column.
- Each character in the string occupies either the row directly above or directly below the row of the previous character.

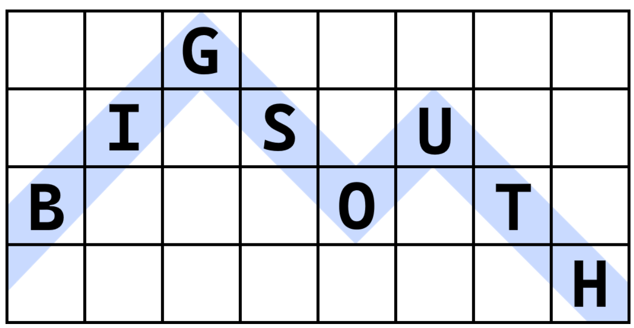

Figure A.1: The snakey string in the sample case.

Given a snakey string, can you recover the original string?

Input

The first line of input contains two integers r and c ($2 \leq r, c \leq 100$), the number of rows and columns of the grid.

The next r lines each contain c characters that form the 2D grid containing the snakey string. Empty cells are encoded with a period, while uppercase letters (`A`–`Z`) represent characters in the original string. Every column in the grid contains exactly one uppercase letter. It is possible that some rows do not contain any uppercase letters.

Output

Output a single string, the original string that was used to build this snakey string.

Sample Input 1	Sample Output 1

`4 8`
`..G.....`
`.I.S.U..`
`B...O.T.`
`.......H`
	`BIGSOUTH`

Problem B
Blind Bottles

Time Limit: 1 second Memory Limit: 1GB

You are playing the *Blind Bottles* game. There are n bottles, each with a unique color. The game host arranges them into one row, keeping their ordering a secret. Your goal is to determine the host's ordering by making guesses. In each guess, you provide one ordering of the bottles. The host will tell you the number of bottles in your ordering that are in the correct positions. You win the game when you make a guess that places all the bottles in their correct positions.

You want to win the game; however, you don't want to spend all day guessing. Therefore, you have decided that you'd like to win in at most 10^4 guesses.

Interaction

This is an interactive problem.

At the start of the input, you will receive a single integer n ($2 \leq n \leq 100$), the number of bottles. The bottle colors are labeled with unique integers from 1 to n.

After reading this integer, your program should begin making guesses. In each guess, output a single line of n space-separated integers, which is a permutation of the integers from 1 to n. This permutation represents one guess of the ordering. You will then receive a single integer k ($0 \leq k \leq n$), which is the number of bottles that are in the correct positions in your guess. If $k = n$, you have won the game and your program should terminate. If $k < n$, the game proceeds to the next guess. If you have not won the game after 10^4 guesses, your program should terminate, and you will receive a Wrong Answer verdict.

Do not forget to flush the output after printing each guess. You will not receive any further input after the game ends – either after you win or after you make 10^4 guesses.

The judge program is not adversarial, meaning that the ordering of the bottles is fixed at the start of the game and does not change during the game.

Sample Interaction 1

Read	Write
5	
	3 2 5 1 4
2	
	3 5 2 1 4
3	

```
3 4 2 1 5
```

```
5
```

Problem C
One Way Only

Time Limit: 1 second Memory Limit: 1GB

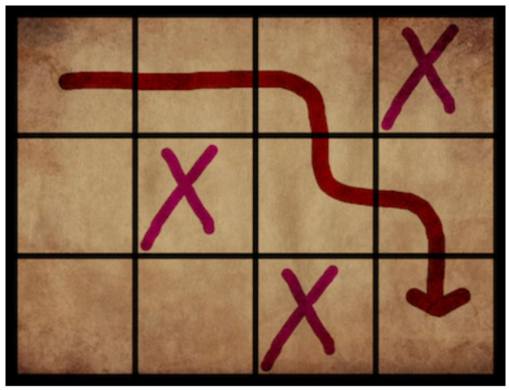

Figure C.1: The preferred path in the sample case. Three tiles are X-ed out to enforce this path.

You've been tasked with mapping out one of the most dangerous hiking routes deep within the Amazon rainforest. This hiking route exists on a rectangular area of land that has been tiled into a 2D grid with r rows and c columns. Hikers who dare attempt this trail begin on the top-left tile $(1, 1)$ and work their way down to tile (r, c). In order to finish as fast as possible, every hiker will move only rightward or downward during their hike.

After a thorough examination of the entire land area, you've uncovered just how unforgiving this jungle can be. With many tiles containing extreme hazards (from apex predators to mangrove swamps), you've determined that there is exactly one path through the jungle consisting of only rightward and downward moves that also avoids all of the major obstacles. You call this path your *preferred path*.

However, hikers who try this route may feel adventurous and follow a different path (still consisting of only rightward and downward moves) instead of the preferred path. To discourage the hikers from doing this, you plan to X out some tiles on the map, knowing that a hiker will not pass through an X-ed out tile under any circumstances. Consequently, if the only path from $(1, 1)$ to (r, c) that does not pass through an X-ed out tile is your preferred path, then the hikers will have no choice but to follow it!

Given the map dimensions and your preferred path, can you determine the minimum number of tiles you must X out to force all hikers to take your preferred path?

Input

The first line of input contains two integers r and c ($1 \leq r, c \leq 10^5$, $r \cdot c \geq 2$), the number of rows and columns on the map.

The next line has a string of length $r + c - 2$ which contains exactly $r - 1$ 'D's and $c - 1$ 'R's: the sequence of moves that follows your preferred path through the jungle. A 'D' represents a downward move on the map, while an 'R' represents a rightward move.

Output

Output a single integer, the minimum number of tiles you must X out on the map to guarantee that all future hikers will take your preferred path through the jungle.

Sample Input 1	Sample Output 1
3 4 RRDRD	3

Problem D
GUID Generator

Time Limit: 5 seconds Memory Limit: 2GB

Anna is preparing a new database for her web application, in which a Globally Unique Identifier (GUID) is assigned to each table row. Getting tired of the traditional random GUID generation method, Anna decides to try something new this time.

Anna finds a tree with n nodes, where between each pair of nodes there exists one unique simple path (A simple path visits each node at most once). She assigns each node one hexadecimal character (`0-9` or `a-f`). To generate a GUID, she selects two nodes s and t (not necessarily distinct), and constructs a string by concatenating all the hexadecimal characters along the simple path from s to t. This resulting hexadecimal string is then used as a GUID. Note that the GUIDs generated by this method do not have a fixed length. Besides, the GUID produced by the path from s to t may not be the same as the GUID produced by the path from t to s.

Anna can generate different GUIDs by choosing different nodes s and t. To ensure the quality of her GUID generation method, she wants to know how many unique GUIDs can be generated from the given tree.

Input

The first line of input has a single integer n ($2 \leq n \leq 2\,000$), the number of nodes in the tree. The nodes are numbered from 1 to n. The next line contains a string of n hexadecimal characters (`0-9` or `a-f`), where the ith character is the hexadecimal character assigned to node i.

The following $n-1$ lines each contain two integers u and v ($1 \leq u, v \leq n$, $u \neq v$), indicating that there is an edge between nodes u and v. It is guaranteed that the given edges form a tree.

Output

Output a single integer, the number of unique GUIDs that can be generated from the given tree.

Explanation of Sample Case 1

Starting from node 1 or node 3, you can obtain 4 unique GUIDs: `a`, `ab`, `aba`, and `abb`. Starting from node 2, you can obtain 3 unique GUIDs: `b`, `ba`, and `bb`. Starting from node 4, you can obtain 3 unique GUIDs: `b`, `bb`, and `bba`. In total, there are 8 unique GUIDs that can be obtained.

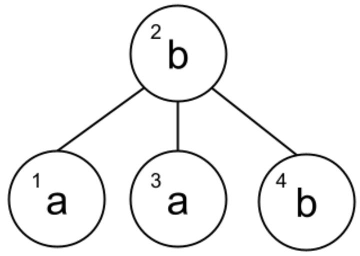

Figure D.1: Illustration of the first sample case. The node numbers are marked at the top-left of each node, and the hexadecimal characters are shown in the center.

Sample Input 1	Sample Output 1
4 abab 1 2 3 2 2 4	8

Sample Input 2	Sample Output 2
6 01aa10 1 2 2 3 2 4 4 5 5 6	18

Problem E
Photo Encoding

Time Limit: 1 second Memory Limit: 1GB

You've been tasked with printing some of your old family photos. These photos are ancient – not only are they black and white, but they are also incredibly pixelated! In fact, each photo can be represented as an $n \times n$ grid of pixels, where each pixel is either black or white.

Before you print each photo, you want to buy a frame that perfectly fits an $n \times n$ photo. However, you realize that you do not know n (the dimensions of the photo are unknown)! To make things worse, your computer stores the photo in an unreadable binary format – the only information you can recover about this photo is the list of Manhattan distances of each black pixel from the top-left pixel. The Manhattan distance between two pixels at (r_1, c_1) and (r_2, c_2) is $|r_1 - r_2| + |c_1 - c_2|$.

For example, the 5×5 photo in Figure I.1 corresponds to the list $[1, 4, 4]$. You notice that there are multiple possible photos that could correspond to the same list. As an example, the 4×4 photo shown in Figure I.2 also corresponds to the list $[1, 4, 4]$.

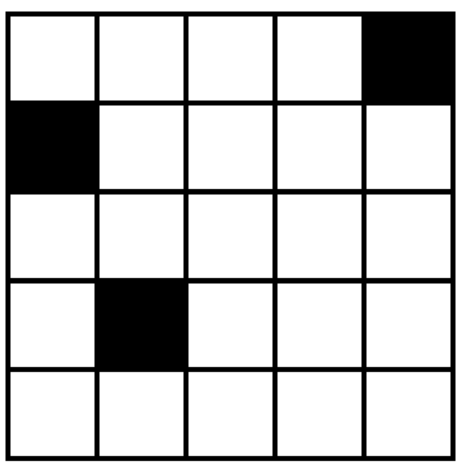

Figure E.1: A 5×5 photo that corresponds to $[1, 4, 4]$.

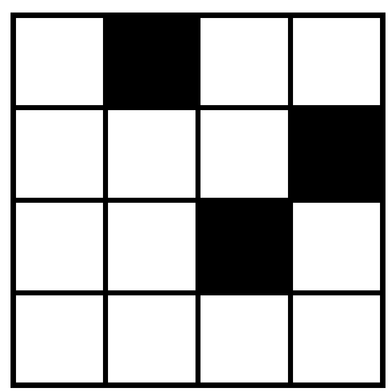

Figure E.2: A 4×4 photo that also corresponds to $[1, 4, 4]$.

With this in mind, you want to be prepared when buying the picture frame. Given a list of Manhattan distances, can you determine the smallest possible n such that there exists an $n \times n$ photo corresponding to this list?

Input

The first line of input contains one integer m ($1 \leq m \leq 1\,000$), the number of black pixels in the photo.

Each of the next m lines contains a single integer between 0 and 200 (inclusive), representing the Manhattan distance from one black pixel to the top-left pixel of the photo. The distances are given in ascending order and are guaranteed to correspond to a valid photo.

Output

Output a single integer, the minimum side length n such that there exists an $n \times n$ photo corresponding to the input list.

Sample Input 1	Sample Output 1
3 1 4 4	4

Sample Input 2	Sample Output 2
8 0 1 3 5 5 5 5 6	5

Problem F
Breakout

Time Limit: 1 second Memory Limit: 1GB

Uh oh! After you gave a low rating to one of Merlin's riddles, he has taken revenge by locking you inside his magical dungeon! This dungeon can be modeled as a circle with n rooms arranged along its circumference. The rooms are labeled from 1 to n in clockwise order.

You start at room 1 and know that there exists an exit in one of the rooms. To get to the exit room, you can walk either clockwise or counterclockwise to a neighboring room. However, Merlin is smart. To impede your escape, Merlin has placed wooden crates in several hallways connecting two adjacent rooms. To pass through a hallway, you must break any wooden crates blocking it.

You want to make it out to the exit room as quickly as possible. What is the minimum number of crates you must break through to escape?

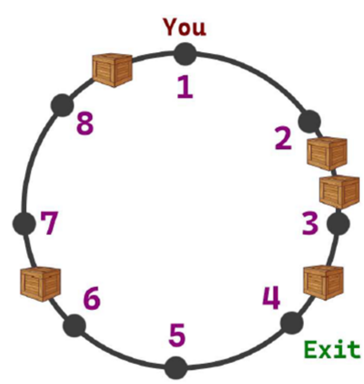

Figure F.1: Illustration of the sample case.

Input

The first line of input contains three integers n, x, and m ($2 \leq n \leq 1\,000$, $1 \leq x \leq n$, $0 \leq m \leq 1\,000$), meaning that there are n rooms with room x being the exit room, and there are m crates.

Each of the next m lines contains a single integer r ($1 \leq r \leq n$), indicating that there is a crate in the hallway clockwise between room r and the next room along the circumference. If $r = n$, it means that the crate is in the hallway between room n and room 1. These integers may contain duplicates, meaning that multiple crates are placed in the same hallway.

Output

Output a single integer, the minimum number of crates you must destroy to breakout.

Sample Input 1	Sample Output 1
8 4 5 2 6 3 2 8	2

Problem G
Shinjuku Station

Time Limit: 1 second Memory Limit: 1GB

Audrey is planning her first day of vacation in Tokyo! Her first stop is a Honkai: Star Rail pop-up shop. She has heard from her friends that the trains in Japan are a lot like her favorite train, the Astral Express, so she decides to take a series of trains to the pop-up. As luck would have it, she has to transfer at Shinjuku Station, the largest train station in Tokyo.

Audrey is thinking of taking one of the n trains that will arrive at Shinjuku Station, and then transferring to one of the m trains that will depart from Shinjuku Station. The trains in Tokyo are extremely reliable, and Audrey knows to the second when each train will arrive at or depart from Shinjuku Station.

Audrey is a bit nervous about transferring at Shinjuku Station, so she does not want to have a tight transfer that takes *strictly* fewer than s seconds. However, neither does Audrey want to spend too long transferring at Shinjuku Station. Help Audrey determine the minimum transfer time possible. Audrey will not consider any itineraries that cross the boundary of a calendar day.

Input

The first line of input contains two integers n and m ($1 \leq n, m \leq 50$), the number of trains entering and leaving Shinjuku Station, respectively.

The next n lines each contain a string of the form `HH:MM:SS`, indicating the exact arrival time of a train entering Shinjuku.

The next m lines each contain a string of the form `HH:MM:SS`, indicating the exact departure time of a train leaving Shinjuku.

It is guaranteed that each of these times is a valid time. Specifically, `HH` is an integer between 5 and 23 inclusive, and both `MM` and `SS` are non-negative integers between 0 and 59. All values are represented using two digits, possibly with a leading zero.

The last line contains a single integer s ($1 \leq s \leq 600$), representing the minimum transfer time in seconds required by Audrey.

Output

Output a single integer, the minimum number of seconds that Audrey needs for a valid transfer. If no such transfer exists, output -1.

Sample Input 1	Sample Output 1
2 2 10:00:00 11:00:00 10:05:00 11:04:00 300	300

Sample Input 2	Sample Output 2
1 1 23:13:23 13:23:13 23	-1

Problem H
Memories of Passport Control

Time Limit: 1 second Memory Limit: 1GB

After getting your latest passport, you are back to traveling internationally. This time, officers at passport control have been stamping only the earliest unstamped pages in your passport, so you have not needed to apply for a new passport early.

Due to new international aviation laws, when you pass through passport control, one of two things happens: either a single page is stamped, or exactly k pages are stamped. You pass through passport control exactly once per trip.

Your memory is foggy, and you no longer remember how many trips you have taken. Compute the minimum number of trips you could have taken to end up with exactly the given number of stamped pages in your passport.

Input

The first and only line of input contains two integers k and s ($2 \leq k \leq 50$, $1 \leq s \leq 50$), where k is the number of pages passport control stamps when stamping multiple pages at once, and s is the total number of pages that have been stamped in your passport.

Output

Output a single integer, the minimum number of trips you could have taken. It can be proven that it is always possible to have exactly s stamped pages.

Sample Input 1	Sample Output 1
13 23	11

Sample Input 2	Sample Output 2
13 39	3

This page is intentionally left blank.

Problem I
Inazuma Cube Device
Time Limit: 6 seconds Memory Limit: 1GB

E.T. the gray alien loves playing Genshin Impact but he is not very good at the game. He is currently trying to solve a cube device puzzle in the Inazuma region but has been stuck for several hours. He is now requesting your help to solve it.

Figure I.1: A cube device puzzle.

Figure I.2: A player striking a cube device.

There are n cubes scattered across the puzzle ground, numbered from 1 to n. Each cube has a flower with k petals. At any time, a cube can have anywhere from 1 to k of its petals lit. When the player strikes a cube, the number of lit petals increases by one – unless all k petals are already lit, in which case the cube resets to having only one lit petal. The puzzle is solved when every cube has all k of its petals lit.

Some pairs of cubes have a one-directional chain reaction from one cube to another: striking one cube also increases the number of lit petals on the other cube, but not vice versa. For any such chain reaction from cube a to cube b (where striking cube a increases the number of lit petals on both cube a and cube b), the cube numbers must satisfy $a < b$, meaning that all chain reactions go from a smaller-numbered cube to a larger-numbered cube. Note that if there is a chain reaction from cube a to cube b, and another chain reaction from cube b to cube c, striking cube a does not affect cube c (unless there is another chain reaction from cube a to cube c directly).

E.T. would like to know the minimum number of times he must strike the cubes to solve the puzzle.

Input

The first line of input contains three integers n, m, and k ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq 5 \cdot 10^5$, $m \leq \frac{n(n-1)}{2}$, $1 \leq k \leq 10^9$), the number of cubes, the number of directed relations, and the number of petals per cube.

The next n lines each contain a single integer. The integer on the ith line is the initial number of lit petals on cube i. All those integers are between 1 and k, inclusive.

Each of the next m lines contains two integers a and b ($1 \leq a < b \leq n$), describing a one-directional chain reaction from cube a to cube b. All chain reactions are unique.

Output

Output a single integer, the minimum number of times E.T. the alien must strike the cubes, or -1 if this puzzle is impossible to solve.

Explanation of Sample Case 1

We can solve the puzzle in 22 strikes:

1. Strike cube 5 4 times. Then, cube 5 has 10 lit petals.
2. Strike cube 2 9 times. Then, cube 2 has 8 lit petals, cube 3 has 5, and cube 4 has 8.
3. Strike cube 3 5 times. Then, cube 3 has 10 lit petals.
4. Strike cube 1 2 times. Then, both cube 1 and cube 2 have 10 lit petals. Note that striking cube 1 does not increase the number of lit petals on cube 3 or cube 4.
5. Strike cube 4 2 times. Then, all cubes have 10 lit petals.

Sample Input 1	Sample Output 1
5 3 10 8 9 6 9 6 1 2 2 3 2 4	22

Problem J
Game of Nines

Time Limit: 1 second Memory Limit: 1GB

You are playing a simple game of adding digits. You are given a list of single digits between 0 and 8 (inclusive). In each move, you may choose any two digits a and b, add a to b, and replace b with the sum $a + b$. If $a + b \geq 10$, keep only the units digit (e.g. $5 + 8$ becomes 3; $4 + 6$ becomes 0). Whenever the sum is 9, the result is eliminated immediately and cannot participate in further additions.

Your goal is to eliminate as many digits as possible, using any number of moves.

Input

The first line of input contains a single integer n ($2 \leq n \leq 1\,000$), the number of digits. Each of the next n lines contains a single digit between 0 and 8, forming the initial list of digits.

Output

Output a single integer, the minimum number of digits that will remain if you play optimally to eliminate as many digits as possible.

Explanation of Sample Case 1

Add 3 to 6 and eliminate one digit ($3 + 6$ becomes 9). The remaining digits are 2 and 3. Then add 2 to 3 three times ($3 + 2$ becomes 5, $5 + 2$ becomes 7, $7 + 2$ becomes 9). The single digit remaining is 2. It is also possible to eliminate two digits with a different sequence of moves.

Sample Input 1	Sample Output 1
3 2 3 6	1

Sample Input 2	Sample Output 2
2 4 4	2

This page is intentionally left blank.

Problem K
Swap for Palindrome

Time Limit: 2 seconds Memory Limit: 1GB

Eric is a cool kid who just learned that a palindrome is a string that reads the same forward and backward. Recently, he found a string consisting of n lowercase English letters. He immediately started looking for palindromic substrings in it (a substring is a contiguous sequence of characters within a string). He soon realized that the string did not contain very long palindromic substrings.

Eric wants to improve the situation by performing exactly one swap of two letters in the string (the letters must be at different positions). What is the length of the longest palindromic substring that can be obtained after Eric performs exactly one swap?

Input

The input consists of a single string containing at least 2 and at most $5\,000$ lowercase English letters (`a`–`z`).

Output

Output a single integer, the length of the longest palindromic substring that Eric can obtain after exactly one swap.

Explanation of Sample Case 1

Eric can swap the first `c` with the first `a`, i.e., `ccbaada`. The resulting string is `acbcada`, which has a palindromic substring of length 5: `acbca`.

Sample Input 1	Sample Output 1
`ccbaada`	`5`

Sample Input 2	Sample Output 2
`ab`	`1`

This page is intentionally left blank.

Problem L
Moonlit Time Machine

Time Limit: 1 second Memory Limit: 1GB

After decades of research in alchemy and magic, you have finally created your magnum opus – a room that, when bathed in the glow of the moon, sends you several days into the future. You have given this room a very classy name: the *Moonlit Time Machine*.

Whenever you use the time machine, you are projected forward some number of days. You don't know the number, but you know that it is between 1 and 28 (inclusive), and that the number is always the same every time you use the machine. As an alchemist, you wish to find out this number! While there is no technology to tell you the date, you can look at the moon's cycle. You know that every 28 days, the moon cycles through a list of phases, as defined below:

Day(s)	Phase
0	New
1 to 4	Crescent
5 to 8	Quarter
9 to 13	Gibbous
14	Full
15 to 19	Gibbous
20 to 22	Quarter
23 to 27	Crescent

To figure out this number of days, you wait until you see a new moon, then use the time machine several times, recording the moon's phase after each use. Given your observations, can you determine a possible number of days that the Moonlit Time Machine sends you forward each time?

Input

The first line of input contains a single integer n ($3 \leq n \leq 100$), the number of observations you have made.

Each of the next n lines contains a single string describing an observed phase of the moon. Each string is one of `New`, `Crescent`, `Quarter`, `Gibbous`, or `Full`, with its first letter capitalized and the remaining letters lowercase. It is guaranteed that these are valid observed phases of the moon after using your time machine.

Output

Output a single integer, the number of days the Moonlit Time Machine sends you into the future with each use. If there are multiple answers, output the smallest one. It is guaranteed that at least one valid answer exists.

Sample Input 1	Sample Output 1
4 Crescent Quarter Gibbous Gibbous	3

Problem M
Much Room for Mushrooms

Time Limit: 1 second Memory Limit: 1GB

When botany proves too challenging, you can always switch to *mycology* – the study of mushrooms! As one of your first experiments, you plan to grow many mushrooms inside an infinitely large 2D exhibit, which can be modeled as a grid of tiles.

As it turns out, this specific species of mushroom that you are growing has a consistent shape. Each mushroom can be represented as a vertical stem (of any height) that is exactly one tile wide. Then, the very top tile in the stem juts outwards by one tile to the left, right, and up to form the *cap* of the mushroom. The figure below shows three such mushrooms:

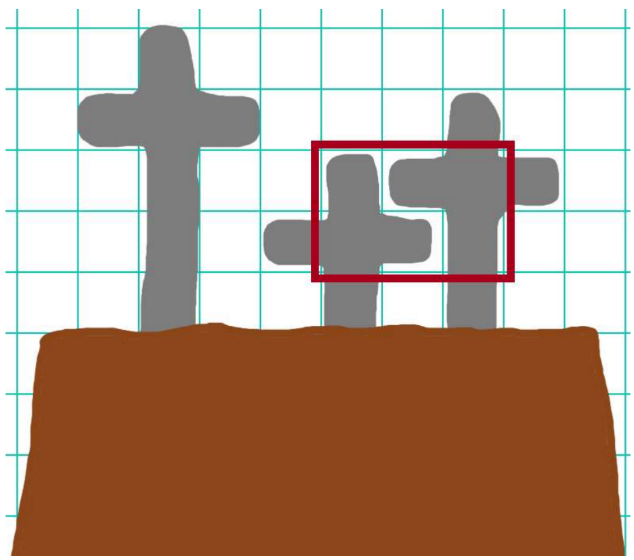

Figure M.1: A visualization of the first sample case.

You've noticed that for each mushroom to survive, two conditions must be met in your exhibit:

1. No tile can be shared by more than one mushroom.
2. Every mushroom must begin growing on the same row.

To make your exhibit more interesting, you'd like to grow mushrooms so that there exists an $r \times c$ rectangle in your exhibit where every tile is occupied by a mushroom. But in this exhibit, there is *not much room for mushrooms*. What is the minimum number of mushrooms needed to completely fill an $r \times c$ rectangle with mushrooms, assuming you can control the position and height of each mushroom you plant?

Input

The first line of input contains two integers r and c ($1 \leq r, c \leq 1\,000$), the number of rows and columns of the rectangle in your exhibit you want to completely fill with mushrooms.

Output

Output a single integer, the minimum number of mushrooms that are needed to completely fill an $r \times c$ rectangle, or -1 if it is not possible.

Sample Input 1	Sample Output 1
`2 3`	`2`

Sample Input 2	Sample Output 2
`4 1`	`1`

Sample Input 3	Sample Output 3
`100 100`	`-1`

This page is intentionally left blank.

2025 ICPC South Central USA Regional Division 2

Back Cover