Problem A. Young Salvador

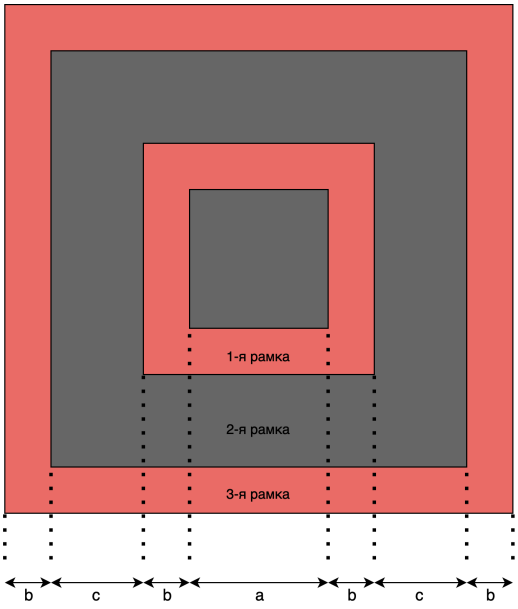
Time limit: 1 second

A child took an infinite grid sheet and began coloring the cells black and red.

To start, he drew a black square with a side length of a. Then he added a «frame» around the current drawing k times according to the following rules:

1. every odd time he retreated b cells from each side of the current drawing and colored the resulting frame red;
2. every even time he retreated c cells from each side of the drawing and colored the resulting frame black.

An example for $a = 3$, $b = 1$, $c = 2$, and $k = 3$ is shown in the figure below:



Determine how many cells will be colored black and red after the drawing is finished.

Input

The input consists of a single line containing four integers a, b, c, and k separated by spaces — the length of the side of the initial black square, the retreat lengths for the red and black frames, and the number of frames drawn around the initial square ($0 \le a, b, c \le 10^4$; $0 \le k \le 10^5$).

Output

Output the number of black cells first, followed by the number of red cells, separated by a space.

Scoring

Each group of tests will be evaluated only if the necessary groups of tests have been passed beforehand, and points are awarded if all tests in the group are passed. The tests from the statement are not evaluated. All tests are divided into groups with the following constraints:

Subtask	Additional constraints	Necessary subtasks	Points
1	$k \leq 1$		10
2	$k \leq 2$	1	13
3	$a = 0$		14
4	$c = 0$		14
5	$a, b, c \leq 4$; $k \leq 20$		12
6	$a, b, c \leq 4$; $k \leq 800$	5	17
7	none	$1 - 6$	20

Examples

standard input	standard output
3 1 2 3	65 56
1 1 1 6	97 72

Problem B. Dutton's Patterns

Time limit: 1 second

The hamster Dutton from Tatarstan was gifted a string s of length n.

He became curious about how many substrings of the string s exist in the form $a_1(k)a_2(i)a_3(j)$, where a_1, a_2, and a_3 denote lowercase Latin letters, and the numbers in parentheses after the symbol indicate how many times that symbol is repeated, but it is important that $a_1 \neq a_2$ and $a_2 \neq a_3$. For example, $i(3)n(2)o(1)$ describes the string $iiinno$.

Input

The first line of input contains an integer n — the length of the string s.

The second line contains the string s of length n, consisting of lowercase Latin letters.

The third line contains the numbers k, i, j, describing the number of symbols a_1, a_2, a_3 in the substrings, respectively.

Output

Output the number of substrings that can be represented in the form $a_1(k)a_2(i)a_3(j)$.

Scoring

Each group of tests will be evaluated only if the necessary groups of tests have been passed beforehand, and points are awarded if all tests of the group are passed. The tests from the statement are not evaluated. All tests are divided into groups with the following constraints:

Subtask	Constraints	Necessary subtasks	Points
1	$n \leq 100$	—	30
2	$n \leq 1000$	1	30
3	$n \leq 10^6$	1, 2	40

Example

standard input	standard output
10 llvvyywwww 2 2 2	2

Problem C. Cows Check Dutton

Time limit: 2 seconds

Dutton's ranch is an infinite grid where each cell initially contains the number 0.

The cows on Dutton's ranch decided to test their owner, so they decided to write instructions for the cow Bet on how she should move across the field.

Bet starts at cell $(0; 0)$. The instruction is a string s consisting of n characters. Each character is a command. There are four basic commands available for execution:

- U — move from cell $(x; y)$ to $(x; y + 1)$;
- D — move from cell $(x; y)$ to $(x; y - 1)$;
- L — move from cell $(x; y)$ to $(x - 1; y)$;
- R — move from cell $(x; y)$ to $(x + 1; y)$.

On the first step, Bet writes the number 1 in cell $(0; 0)$, on the second step — the number 2 in the cell where she ends up after the first command, and so on. If she ends up in a cell where she has been before, she will overwrite the number with a new one.

After Bet walks across the field, the cows will ask q queries. There are two types of queries:

- 1 x — you need to report the sum of all numbers in row x;
- 2 y — you need to report the sum of all numbers in column y.

Input

The first line contains an integer n ($1 \le n \le 10^5$) — the number of commands.

The second line contains a string s of n characters U, D, L or R, representing the commands.

The third line contains an integer q ($1 \le q \le 10^5$) — the number of queries.

The following q lines contain queries of the form «1 x» and «2 y», requiring the answer - the sum of numbers in the row or column respectively.

Output

You need to output q numbers — the answers to the queries, each on a separate line.

Scoring

Each group of tests will be evaluated only if the necessary groups of tests have been passed beforehand, and points are awarded if all tests of the group are passed. The tests from the statement are not evaluated. All tests are divided into groups with the following constraints:

Subtask	Constraints	Required subtasks	Points
1	$n \le 100; q \le 100$	—	30
2	$n \le 500; q \le 10^5$	1	30
3	$n \le 10^5; q \le 10^5$	1, 2	40

Example

standard input	standard output
10 DDURULLDRU 5 1 0 2 2 1 1 2 0 1 -3	24 0 11 25 0

Problem D. Segments

Time limit: 1 second

Dutton learned that a total of n fences were installed. Each fence is described as a segment $(l; r)$.

Dutton, the mathematician hamster, gave Jamie n segments, denoted as (l_i, r_i), where $1 \le l_i \le r_i \le m$. Jamie wants to add one new segment (l, r), where $1 \le l \le r \le m$.

However, Dutton set a condition: the new segment (l, r) can only be added if none of the existing segments (l_i, r_i) is completely contained within the new segment (l, r).

You need to determine the number of different ways to choose such a segment (l, r) that satisfies the specified condition.

Input

The first line contains two integers n and m $(1 \le n, m \le 10^5)$ — the number of segments and their maximum possible coordinate, respectively.

In the following n lines, the segments themselves are described by integers l_i and r_i $(1 \le l_i \le r_i \le m)$ — the left and right boundaries of the segment, respectively.

Output

Output a single number — the number of different ways to choose a new segment.

Scoring

Each group of tests will be evaluated only if the necessary groups of tests have been passed beforehand, and points are awarded if all tests of the group are passed. The tests from the statement are not evaluated. All tests are divided into groups with the following constraints:

Subtask	Constraints	Necessary Subtasks	Points
1	$n \le 100$; $m \le 100$	—	30
2	$n \le 10^5$; $m \le 100$	1	35
3	$n, m \le 10^5$;	1, 2	35

Examples

standard input	standard output
2 10 1 2 3 4	32
3 10 1 5 5 10 2 5	40
4 10 1 5 5 10 2 5 3 7	37
5 10 1 5 5 10 2 5 3 7 7 9	31

Problem E. Basements and Mathematicians

Time limit: 1 second

VK developer Rustem loves to play board games, especially during breaks between creating new features for his services. This time he decided to organize a battle in the game "Basements and Mathematicians."Rustem gathers friends to compete in the game Basements and Mathematicians.

For each game, a new board needs to be created, which is a regular k-gon. Rustem knows the dimensions of the rectangular table, which is n in length and m in width, where their game will take place.

His board must fit entirely on the table and should have the maximum possible area. However, there is one peculiarity: to prevent the board from moving on the table, its bottom left corner is attached to the bottom left corner of the table so that the board does not extend beyond the table's boundaries. Help Rustem find the maximum possible area of the board or state that such a board cannot exist. If it is impossible to create the board, output the number -1.

Input

The input consists of a single line containing integers n and m ($1 \le n, m \le 1000$), representing the length and width of the table, respectively, and an integer k ($3 \le k \le 10^5$) — the number of vertices of the regular polygon that serves as the board.

Output

Output a real number — the area of the board that meets the conditions, with a precision of at least 10^{-6}. That is, your answer should differ from the required value by no more than 10^{-6}. If no solution exists, you must output only the number -1.

Scoring

Each group of tests will be evaluated only if the necessary groups of tests have been passed beforehand, and points are awarded if all tests in the group are passed. The tests from the problem statement are not evaluated. All tests are divided into groups with the following constraints:

Subtask	Constraints	Required subtasks	Points
1	$k = 4$	—	15
2	$n = m; 3 \le k \le 4$	1	30
3	$k = 3$	—	35
4	$k \le 10^5$	1, 2, 3	20

Examples

standard input	standard output
10 11 4	100.00000000000000000000
10 10 3	46.41016151377545652726

Problem F. Dutton's Night Snack

Time limit: 1 second

Dutton the hamster has n switches that control the lights. He is afraid of the dark, so he asks you to write a program to help him.

His home looks like a corridor consisting of n rooms connected in a row. In each room, there is a light bulb that can initially be either on "1"or off "0".

The initial states of the light bulbs are described by an array L. Dutton often wakes up at night to go eat, so before going to bed, he wants to switch all the light bulbs so that he can walk through the corridor from his bedroom to the kitchen in such a way that the light bulb in each room he passes through is on.

Since he wants to save on electricity, all other rooms should have their lights off.

The kitchen is located in the room numbered A. If $A = 0$, he can go directly to the kitchen from his bedroom, and all the light bulbs in the corridor must be off. If $A > 0$, then the light bulbs in the rooms numbered from 1 to A must remain on.

He can switch the light bulbs as follows: choose k consecutive rooms in the corridor, and in all k selected rooms, if the light bulbs were on, they will turn off, and vice versa. That is, "0"will change to "1 and "1"will change to "0".

Dutton is tired after a hard day's work, so he wants to make as few of these switches as possible.

Your task is to write a program that finds the minimum number of actions Dutton needs to take or states that his request cannot be fulfilled.

Input

The first line contains the numbers n ($1 \le 10^5$) — the number of rooms in the corridor and k ($1 \le n$) — the number of consecutive light bulbs that can be switched in one move.

The second line contains an array L of size n — the initial states of the light bulbs, where the light bulb in room number i is on "1"or off "0".

The third line contains a single number A ($0 \le A \le n$) — the room number where the kitchen is located.

Output

Output a single number — the minimum number of actions required to switch the light bulbs as described above. If it is impossible to do so with any number of actions and any sequence, output -1.

Scoring

Each group of tests will be evaluated only if the necessary groups of tests have been passed beforehand, and points are awarded if all tests in the group are passed. The tests from the problem statement are not evaluated. All tests are divided into groups with the following restrictions:

Subtask	Constraints	Required Subtasks	Points
1	$k = 1$	—	10
2	$n \le 1000$; $A = 0$	—	20
3	$n \le 10^5$; $A = 0$	2	30
4	$n \le 10^3$	1	20
5	$n \le 10^5$	1, 2, 3, 4	20

Examples

standard input	standard output
10 3 1 0 0 0 0 0 1 1 1 0 1	1
10 3 1 0 0 0 1 1 0 1 1 0 1	2

Problem G. Ilnur and the Array

Time limit: 1 second

Ilnur is given an array of numbers of length n, where the i-th number of the array is a_i. For two numbers with indices i and j, Ilnur defines the following quantity: $P(i, j) = \lceil \sqrt{a_i \cdot a_j} \rceil$, where $\lceil \cdot \rceil$ denotes rounding up.

Ilnur calls the array good if there are no two distinct pairs of numbers with the same value of P. Help Ilnur determine whether his array is good.

Input

The first line contains a single positive integer n — the size of the array. The second line contains n positive integers $a_1, \ldots, a_n$.

Each group of tests will be evaluated only if the necessary prerequisite groups of tests have been passed, and points are awarded if all tests in the group are passed. All tests are divided into groups with the following constraints:

Subtask	Constraints	Required Subtasks	Points
1	$n \le 10$; $a_i \le 100$	—	10
2	$n \le 10^3$; $a_i \le 10^6$	1	15
3	$n \le 10^6$; $a_i \le 10^6$	1, 2	75

Output

Print `YES` if Ilnur's array is good, `NO` otherwise.

Examples

standard input	standard output
3 3 2 1	NO
3 1 3 5	YES

Problem H. Dutton's Night Snack 7-8

Time limit: 1 second

Dutton the hamster has n switches that control the lights. He is afraid of the dark, so he asks you to write a program to help him.

His house looks like a corridor consisting of n rooms connected in a row. In each room, there is a light bulb that can initially be either on «1» or off «0».

The initial states of the light bulbs are described by the array L.

Dutton often wakes up to go to the kitchen, so before going to bed, he wants to toggle all the light bulbs so that he can walk through the corridor from his bedroom to the kitchen with every room having the light bulb turned on.

The kitchen is located in the last room, meaning that the light bulbs in rooms numbered from 1 to n must be turned on.

He can toggle the light bulbs as follows: choose k consecutive rooms in the corridor, and in all k selected rooms, if the light bulbs were on, they will turn off, and vice versa. That is, «0» will change to «1», and «1» will change to «0».

Dutton is tired after a hard day's work, so he wants to make as few toggles as possible.

Your task is to find the minimum number of actions Dutton needs to take or state that his request is impossible. If it is impossible to turn on all the light bulbs regardless of the number of actions and their sequence, output -1.

Four test arrays and numbers k are given for which you need to find the minimum number of actions. If you do not know the answer for a test or it is impossible to turn on all the bulbs, write -1.

As an answer to this problem, attach a **txt** file. The first line of the file should contain the answer for the first test, the second line should contain the answer for the second test, and so on.

No more than three files will be accepted for this problem.

Each test is worth 25 points.

Test Number	**Value k**	**Array L**	**Answer**
1	2	$[0, 0, 1, 1, 0, 1, 0]$	(answer field)
2	3	$[1, 0, 0, 1, 0, 1, 1, 1, 0, 0]$	(answer field)
3	3	$[1, 1, 1, 0, 1, 1, 1, 0, 0, 1]$	(answer field)
4	7	$[1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 0, 0, 1, 1, 0, 1]$	(answer field)

Problem I. Primitive Tetris

Time limit: 2 seconds

As is well known, the game of Tetris takes place on a grid of size $n \times m$ (n rows and m columns), where at each unit of time, tetrominoes of several types appear from the top and start moving downwards at a fixed speed.

The player's task is to rotate or move these pieces in such a way as to fill the grid as evenly as possible. In the original game, as soon as a row of the grid is completely filled with pieces, it is removed, allowing the game to continue indefinitely.

In our version, no cells will be removed, and the game will end as soon as the top cell of any piece goes beyond the upper boundary of the grid. You are also not required to implement a player's strategy—it's enough to report, based on a sequence of commands like «place piece F_i in column j_i», after which action the game will end and what the grid will look like.

Each piece consists of four cells connected by edges (standard Tetris pieces) and is represented by a "drawing"made up of characters '.' (empty cell) and '#' (piece cell). For example,

```
##      .#      .##      ####      ###
##      ##      ##.               ..#
        .#
```

If piece F_i occupies multiple columns, j_i will indicate the column of the game grid occupied by the leftmost cell of F_i. Each piece moves down the grid until it touches another piece or the bottom boundary of the grid, after which it stops and never moves again.

Initially, the grid is empty. Determine the number of the piece on which the game will end, and what the grid will look like at the end of the game.

Input

The first line contains three integers n, m, and q—the dimensions of the game grid and the total number of pieces to be processed ($1 \le n \le 10^4$; $4 \le m \le 50$; $1 \le q \le 4 \cdot 10^5$).

Following this are the descriptions of q pieces.

The first line of the piece description contains integers r_i, c_i, j_i—the dimensions of the grid containing the piece and the column number of the game grid where this piece will appear ($1 \le r_i, c_i \le 4$; $1 \le j_i \le m - c_i + 1$).

Then follow r_i lines of c_i characters '.' or '#' in each, describing the piece. It is guaranteed that the piece described in this way is connected and consists of exactly four cells (i.e., it is a valid Tetris piece).

Output

In the first line, output the number of the first piece (from 1 to q) that will go beyond the upper boundary of the grid. If all q pieces fit on the grid, output $q + 1$.

Then output n lines of m characters '.' or '#' in each— the state of the grid at the moment the game ends. If the last piece went beyond the upper boundary of the grid, do not output its part that is beyond the grid.

Scoring

Each group of tests will be scored only if the necessary groups of tests have been passed beforehand, and points are awarded if all tests in the group are passed. The tests from the statement are not scored. All tests are divided into groups with the following constraints:

Subtask	Additional constraints	Necessary subtasks	Points
1	$n = 1$; $m = 4$		5
2	$n \leq 3$; $m = 4$	1	5
3	$c_i = 1$ for all i		8
4	$m \bmod 2 = 0$; $c_i = r_i = 2$ and $j_i \bmod 2 = 0$ for all i		13
5	$r_i = 1$ for all i		11
6	$n, m \leq 15$; $q \leq 50$	1, 2	16
7	$n, q \leq 800$	1, 2, 6	19
8	none	1 – 7	23

Example

standard input	standard output
3 5 4 1 4 1 #### 3 2 4 #. ## .# 2 2 1 ## ## 2 3 2 ### .#.	4 ####. ##.## #####

Problem J. Primitive Tetris 7-8

Time limit: 2 seconds

As is well known, the game of Tetris takes place on a grid of size $n \times m$ (n rows and m columns), where at each unit of time, grid shapes of several types appear from the top and start moving downwards at a fixed speed.

The player's task is to rotate or move these shapes in such a way as to fill the grid as evenly as possible. In the original game, as soon as a row of the grid is completely filled with shapes, it is removed, allowing the game to continue indefinitely.

In our version, no cells will be removed, and the game will end as soon as the top cell of any shape goes beyond the upper boundary of the grid. You are also not required to implement a player's strategy—it's enough to determine how the grid will look at the end of the game based on a sequence of commands like "place shape F_i in column j_i."

Each shape consists of four cells connected by sides (standard Tetris shapes) and is represented by a "drawing"of characters '`.`' (empty cell) and '`#`' (shape cell). For example,

```
##      .#      .##      ####      ###
##      ##      ##.               ..#
        .#
```

If shape F_i occupies multiple columns, j_i will indicate the column of the game grid occupied by the leftmost cell of F_i. Each shape moves down the grid until it comes into contact with another shape or the bottom boundary of the grid, at which point it stops and never moves again.

Initially, the grid is empty. Determine how the grid will look at the end of the game.

As a response to this task, attach a **txt** file. You should send a **txt** file where the first free line contains the answer for the first test, the second free line contains the answer for the second test, and so on.

For this task, no more than five files will be accepted for verification.

Input

The first line of the input data contains the number 4— the number of input tests.

The following lines describe the tests. The first line contains three integers n, m, and q— the dimensions of the game grid and the total number of shapes to be processed ($1 \le n \le 10^4$; $4 \le m \le 50$; $1 \le q \le 4 \cdot 10^5$).

Next are the descriptions of q shapes.

The first line of the shape description contains integers r_i, c_i, j_i— the dimensions of the grid containing the shape and the column number of the game grid where this shape will appear ($1 \le r_i, c_i \le 4$; $1 \le j_i \le m - c_i + 1$).

Then follow r_i lines of c_i characters '`.`' or '`#`' in each, describing the shape. It is guaranteed that the shape defined in this way is connected and consists of exactly four cells (i.e., it is a valid Tetris shape).

Output

Output n lines for each test, each containing m characters '`.`' or '`#`'— the state of the grid at the moment the game ends. If the last shape goes beyond the upper boundary of the grid, do not output its part that is outside the grid.

Scoring

There are four tests in the problem, each scored at 25 points. For each test, you will receive $\max(0, 25-2X)$ points, where X is the number of cells in which your answer differs from the correct answer.