Problem A. Candies

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

n bobo are playing a game about candies. bobo are labeled by $1, 2, \ldots, n$ for convenience. Initially, the i-th bobo has a_i candies in hand.

The game is played in m rounds. In each round, the bobo who has the least number of candies currently is awarded with x candies. If two or more bobo have the same number of candies, the bobo with the smallest label gets the prize.

The 1-st bobo is their leader. So he can get at most y more candies from some unknown source before the start of the game. Now he wonder the maximum number of candies he can have after the m rounds.

Input

The first line contains 4 integers n, m, x, y ($1 \le n, m \le 200000, 1 \le x, y \le 10^9$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$).

Output

A single integer denotes the maximum number of candies.

Examples

standard input	standard output
2 1 2 2 1 2	4

Problem B. Chessboard Game

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

bobo and yiyi are playing a game on a chessboard with $(n + 1)$ rows and $(m + 1)$ columns. Rows are numbered by $0, 1, \ldots, n$ from top to bottom, while columns are numbered by $0, 1, \ldots, m$ from left to right.

Cells $(0, 1), (0, 2), \ldots, (0, m), (1, 0), (2, 0), \ldots, (n, 0)$ are special. They may contain a "heaven gate" or "hell gate". People who enters a "heaven gate" immediately wins. However, the one who enters a "hell gate" dies and gives the victory to the other.

The game lasts for q rounds. In each round, a chess is placed on cell (x_i, y_i) initially. bobo and yiyi moves alternatively. bobo goes first. In one move, chess can be moved one cell upward or leftward.

Determine if bobo can win for each round. You know, bobo and yiyi are really clever guys ...

Input

The first line contains 3 integers n, m, q $(1 \le n, m, q \le 2 \cdot 10^5)$.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 1)$. If cell $(i, 0)$ contains a "heaven gate", then $a_i = 0$. If cell $(i, 0)$ contains a "hell gate" instead, then $a_i = 1$.

The third line contains m integers $b_1, b_2, \ldots, b_m$ $(0 \le b_i \le 1)$. If cell $(0, i)$ contains a "heaven gate", then $b_i = 0$. If cell $(0, i)$ contains a "hell gate" instead, then $b_i = 1$.

Each of the last q lines contains 2 integers x_i, y_i $(1 \le x_i \le n, 1 \le y_i \le m)$.

Output

For each rounds, print "Yes" if bobo can win. Print "No" otherwise.

Examples

standard input	standard output
2 2 4 10 11 1 1 1 2 2 1 2 2	No Yes Yes No

Problem C. Geometric Progression

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

bobo loves geometric progressions! So he wants to know the number of geometric progressions of length 3 in a sequence $a_1, a_2, \ldots, a_n$.

That is to say, count the number of (i, j, k) where $i < j < k$ and $a_i \cdot a_k = a_j^2$.

Input

The first line contains an integer n ($1 \le n \le 1000000$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_1 < a_2 < \cdots < a_n \le 1000000$).

Output

A single integer denotes the number of geometric progressions.

Examples

standard input	standard output
3 1 2 4	1
4 1 2 4 8	2

Problem D. Inverse KMP

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

bobo has just learnt Knuth-Morris-Pratt (KMP) algorithm.

For string $S = s_1 s_2 \dots s_n$, $\mathrm{KMP}(S) = (f_2, f_3, \dots, f_n)$ where f_i is the maximum $j < i$ where $s_1 s_2 \dots s_j = s_{i-j+1} s_{i-j+2} \dots s_i$.

Given $f_2, f_3, \dots, f_n$ and the size of alphabet, find out the number of strings S where $\mathrm{KMP}(S) = (f_2, f_3, \dots, f_n)$ modulo $(10^9 + 7)$.

Input

The first line contains 2 integers n and c, which denotes the length of the string and the size of alphabet, respectively $(2 \le n \le 2 \cdot 10^5, 1 \le c \le 10^9)$.

The second line contains $(n-1)$ integers $f_2, f_3, \dots, f_n$ $(0 \le f_i < i)$.

It is guaranteed that there exists at least one solution.

Output

A single integer denotes the number of strings.

Examples

standard input	standard output
3 3 0 0	12
5 1000000000 1 2 3 4	1000000000

Problem F. Walls

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

Poor bobo is trapped in a maze!

The maze is divided into n rows and m columns. Each cell of the maze contains a wall across the diagonal. Thus, there are only two types of cells.

Thanks to bobo's magic power, he can change the type of cell (i, j) with cost $c_{i,j}$. As a kind magician, bobo would like to make the maze unable to trap people anymore. That is to say, there will be no closed area surrounded by walls.

Find the minimum total cost for bobo to achieve the goal.

Input

The first line contains 2 integers n, m ($1 \le n, m \le 1000$).

Each of the following n lines contains m characters, which denotes the direction of wall in the cell.

Each of the last n lines contains m integers $c_{i,1}, c_{i,2}, \ldots, c_{i,m}$ ($1 \le c_{i,j} \le 1000$).

Output

A single number denotes the minimum of cost.

Examples

standard input	standard output
3 3 /\/ \/\ /\/ 1 3 3 3 1 3 3 3 3	2
2 2 \/ /\ 1000 1000 1000 1000	0

Problem J. XOr

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

bobo has a sequence of integers $a_1, a_2, \ldots, a_n$. He decides to divide the sequence into exactly m consecutive parts.

The cost of each part is its xor sum (bitwise exclusive-or), while the cost of division is bitwise or-sum of its parts' costs.

Help bobo find the minimum cost.

Input

The first line contains 2 integers n, m ($1 \le n \le 200000, 1 \le m \le n$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$).

Output

A single integer denotes the minimum cost.

Examples

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

Problem K. Barcode

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

Barcode consists of black and white vertical bars transformed to ones and zeroes by an optical reader. White and black bars alternate and can be thick or thin. A thin bar represents 0 and a thick bar represents 1, regardless of color. Thus, each barcode represents a sequence of digits.

Each bar in a barcode of a product appears as five "squares" high column. The width of a thin bar is one and the width of a thick bar is two "squares".

The reader used in this problem fell on the floor and since then it was unable to properly recognize the color of some "squares" of a barcode.

Write a program that from a given scanning of a bar code with our faulty reader will determine the binary sequence if at all possible.

Input

The first line of the input file contains an integer N, $1 \le N \le 100$, representing the total width of the scanned bar code.

Eachof the next five lines contains N characters, each of which can be either '`X`', '`.`' (dot), '`?`' (question mark). '`X`' represents successfully recognized black "square", a dot represents successfully recognized white "square" and a question mark means that the reader couldn't determine the color of the "square".

Output

The only line of the output should contain a sequence of binary digits without separators that represent the barcode or -1 if a unique binary sequence cannot be determined.

Example

standard input	standard output
4 .X?? .??. ??.? ?X.? .X?.	001

Problem L. Black Box

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

You have the black box, and whenever you have any spare coins, you throw them all into it. Now, you would like to determine how much money is inside.

Unfortunately the coins cannot be removed without destroying the box, which you don't want. The only possibility is to weigh the box and try to guess how much money is inside by the weight.

Assume that the weight of an empty box is negligible (i.e., zero), and we are able to know the total weight of the coins inside. Also assume that we know the weights of the different types of coins. Your task is to determine the minimum possible amount of cash inside the black box.

Input

The first line contains one integer W ($1 \le W \le 10^4$) which is the weight (in grams) of the cash inside the black box. The second line contains one integer N ($1 \le N \le 500$) which is the number of different types of coins. Then N lines follow, each specifying one coin type with two integers, P and W ($1 \le P \le 5 \cdot 10^4$, $1 \le W \le 10^4$), where P is the value of the coin in monetary units and W is its weight in grams.

Output

Print a single positive integer representing the minimum amount of money that can be produced using coins with the given total weight. If no combination of coins can add up to the given weight exactly, print -1.

Example

standard input	standard output
100 2 1 1 30 50	60

Problem M. Game

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

This is a simple but ancient game. You are supposed to write down the numbers 1, 2, 3, ..., $2n - 1$, $2n$ consecutively in clockwise order on the ground to form a circle, and then, to draw some straight line segments to connect them into number pairs. Every number must be connected to exactly one another. And, no two segments are allowed to intersect.

It's still a simple game, isn't it? But after you've written down the $2n$ numbers, can you tell me in how many different ways can you connect the numbers into pairs? Life is harder, right?

Input

Each line of the input file will be a single positive number n, except the last line, which is a number -1. You may assume that $1 \le n \le 100$.

Output

For each n, print in a single line the number of ways to connect the $2n$ numbers into pairs.

Example

standard input	standard output
1 2 3 4 5 -1	1 2 5 14 42

Problem N. Ordered Sequences

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

Consider sequences of 'A' and 'B', starting with 'B'. Lets sort them in next order: shortest sequences are coming first, if two sequences have the same length, lexicographically smallest is coming first and then lets number them starting from 1.

Given the sequence, find its number.

Input

First line of the input contains one integer n ($1 \le n \le 1000$) — number of test cases. Then n lines follow, each contains non-empty string containing only of 'A' and 'B', starting from 'B' and no more than 24 characters long — the given sequence.

Output

For each test case print number of the given sequence.

Example

standard input	standard output
3 B BA BB	1 2 3