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 $S = s_1 s_2 \dots s_n$, $\text{KMP}(S) = (f_2, f_3, \dots, f_n)$ 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 $\text{KMP}(S) = (f_2, f_3, \dots, f_n)$ 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 E. IQ Test

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

As a truly clever guy, bobo has never entered any kind of IQ tests. But here comes one.

The test consists of n questions, which are numbered conveniently by $1, 2, \ldots, n$. Each question has two options – namely options "A" and "B". The i-th question is "How many questions among questions $1, 2, \ldots, (i-1)$ are answered by option t_i?". (t_i is either "A" or "B".) Option "A" says there are x_i questions while option "B" says y_i.

bobo soon notices that the test is poorly-designed, so he wonder how many questions he can answer correctly at most.

Input

The first line contains an integer n ($1 \le n \le 200000$).

Each of the following n lines contains a character t_i and 2 integers x_i, y_i ($t_i \in \{A, B\}, 0 \le x_i, y_i \le n$).

Output

A single integer denotes the maximum number of questions he can answer correctly.

Examples

standard input	standard output
2 A 0 1 B 0 1	2
2 A 1 2 B 0 1	1

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 G. Random Spanning Tree

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

Yuuka lives in Moe Country. The road system in Moe Country is a connected graph G. Each edge has a random (real) length, which is uniformly random in $[0, 1]$.

Now Yuuka is eager to know the expectation of minimum spanning tree of G.

Input

The first line contains 2 integers n, m, which denotes the number of vertices and edges of G, respectively ($2 \le n \le 8, n - 1 \le m \le \frac{n(n-1)}{2}$).

The vertices in G are conveniently labeled by $1, 2, \dots, n$.

Each of the following m lines contains 2 integers a_i, b_i, which denotes an edge between vertices a_i and b_i ($1 \le a_i, b_i \le n$).

It is guaranteed that the graph G is connected, without self loops and parallel edges.

Output

A single fraction p/q denotes the expectation.

Examples

standard input	standard output
3 2 1 2 2 3	1/1
3 3 1 2 2 3 3 1	3/4

Problem H. Tree Embedding

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

bobo has a tree with n vertices. bobo would like to assign an m-dimension vector $\mathbf{p}(v)$ to vertex v, such that for all a, b, $\text{dist}(a, b) = \langle \mathbf{p}(a), \mathbf{p}(b) \rangle$.

Note that $\text{dist}(a, b)$ is the length of the shortest path between vertices a and b. For two vectors $\mathbf{u} = (u_1, u_2, \ldots, u_m)$ and $\mathbf{v} = (v_1, v_2, \ldots, v_m)$, $\langle \mathbf{u}, \mathbf{v} \rangle = \max\{|u_1 - v_1|, |u_2 - v_2|, \ldots, |u_m - v_m|\}$.

Input

The first line contains an integer n ($2 \le n \le 1000$).

Vertices are numbered by $1, 2, \ldots, n$ for convenience.

Each of the following $(n - 1)$ lines contains 3 integers a_i, b_i, c_i, which denotes an edge between vertices a_i and b_i with length c_i ($1 \le a_i, b_i \le n, 1 \le c_i \le 100000$).

Output

The first line contains an integer m, which denotes the dimension of vectors ($1 \le m \le 16$).

Each of the following n lines contains m integers which denotes the vector $\mathbf{p}(i)$. The coordinates should be in $[-10^9, 10^9]$.

Any appropriate solution will get accepted.

Examples

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

Problem I. Tri-color Spanning Tree

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

bobo has got an undirected graph G, whose edges are colored in red, green and blue.

He would like to count the number of spanning trees with at most g green edges and b blue edges modulo $(10^9 + 7)$.

Input

The first line contains 4 integers n, m, g, b. n and m denote the number of vertices and edges of G, respectively ($1 \le n \le 40, 0 \le m \le 10^5, 0 \le g, b < n$).

The vertices are conveniently numbered by $1, 2, \ldots, n$.

Each of the following m lines contains 3 integers a_i, b_i, c_i, which denotes an edge between vertices a_i and b_i ($1 \le a_i, b_i \le n, a_i \ne b_i, 1 \le c_i \le 3$). $c_i = 1, 2, 3$ denotes that the color of the i-th edge is red, green or blue, respectively.

Output

A single integer denotes the number of spanning trees.

Examples

standard input	standard output
2 3 0 0 1 2 1 1 2 2 1 2 3	1
3 6 1 0 1 2 1 1 2 1 2 3 1 2 3 2 3 1 2 3 1 2	10

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