Codeforces Round #284 (Div. 1)

A. Crazy Town

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Crazy Town is a plane on which there are n infinite line roads. Each road is defined by the equation $a_i x + b_i y + c_i = 0$, where a_i and b_i are not both equal to the zero. The roads divide the plane into connected regions, possibly of infinite space. Let's call each such region a block. We define an intersection as the point where at least two different roads intersect.

Your home is located in one of the blocks. Today you need to get to the University, also located in some block. In one step you can move from one block to another, if the length of their common border is nonzero (in particular, this means that if the blocks are adjacent to one intersection, but have no shared nonzero boundary segment, then it are not allowed to move from one to another one in one step).

Determine what is the minimum number of steps you have to perform to get to the block containing the university. It is guaranteed that neither your home nor the university is located on the road.

Input

The first line contains two space-separated integers x_1, y_1 ($-10^6 \le x_1, y_1 \le 10^6$) — the coordinates of your home.

The second line contains two integers separated by a space x_2, y_2 ($-10^6 \le x_2, y_2 \le 10^6$) — the coordinates of the university you are studying at.

The third line contains an integer n ($1 \le n \le 300$) — the number of roads in the city. The following n lines contain 3 space-separated integers ($-10^6 \le a_i, b_i, c_i \le 10^6$; $|a_i| + |b_i| > 0$) — the coefficients of the line $a_i x + b_i y + c_i = 0$, defining the i-th road. It is guaranteed that no two roads are the same. In addition, neither your home nor the university lie on the road (i.e. they do not belong to any one of the lines).

Output

Output the answer to the problem.

Examples

input
1 1 -1 -1 2 0 1 0 1 0 0
output
2

input
1 1 -1 -1 3 1 0 0 0 1 0 1 1 -3
output
2

Note

Pictures to the samples are presented below (A is the point representing the house; B is the point representing the university, different blocks are filled with different colors):

B. Name That Tune

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

It turns out that you are a great fan of rock band AC/PE. Peter learned that and started the following game: he plays the first song of the list of n songs of the group, and you have to find out the name of the song. After you tell the song name, Peter immediately plays the following song in order, and so on.

The i-th song of AC/PE has its recognizability p_i. This means that if the song has not yet been recognized by you, you listen to it for exactly one more second and with probability of p_i percent you recognize it and tell it's name. Otherwise you continue listening it. Note that you can only try to guess it only when it is integer number of seconds after the moment the song starts playing.

In all AC/PE songs the first words of chorus are the same as the title, so when you've heard the first t_i seconds of i-th song and its chorus starts, you immediately guess its name for sure.

For example, in the song Highway To Red the chorus sounds pretty late, but the song has high recognizability. In the song Back In Blue, on the other hand, the words from the title sound close to the beginning of the song, but it's hard to name it before hearing those words. You can name both of these songs during a few more first seconds.

Determine the expected number songs of you will recognize if the game lasts for exactly T seconds (i. e. you can make the last guess on the second T, after that the game stops).

If all songs are recognized faster than in T seconds, the game stops after the last song is recognized.

Input

The first line of the input contains numbers n and T ($1 \le n \le 5000$, $1 \le T \le 5000$), separated by a space. Next n lines contain pairs of numbers p_i and t_i ($0 \le p_i \le 100$, $1 \le t_i \le T$). The songs are given in the same order as in Petya's list.

Output

Output a single number — the expected number of the number of songs you will recognize in T seconds. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Examples

input
2 2 50 2 10 1
output
1.500000000

input
2 2 0 2 100 2
output
1.000000000

input
3 3 50 3 50 2 25 2
output
1.687500000

input
2 2 0 2 0 2
output
1.000000000

C. Array and Operations

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

You have written on a piece of paper an array of n positive integers $a[1], a[2], \ldots, a[n]$ and m *good* pairs of integers $(i_1, j_1), (i_2, j_2), \ldots, (i_m, j_m)$. Each *good* pair (i_k, j_k) meets the following conditions: $i_k + j_k$ is an odd number and $1 \le i_k < j_k \le n$.

In one operation you can perform a sequence of actions:

- take one of the *good* pairs (i_k, j_k) and some integer v ($v > 1$), which divides both numbers $a[i_k]$ and $a[j_k]$;
- divide both numbers by v, i. e. perform the assignments: $a[i_k] = \frac{a[i_k]}{v}$ and $a[j_k] = \frac{a[j_k]}{v}$.

Determine the maximum number of operations you can sequentially perform on the given array. Note that one pair may be used several times in the described operations.

Input

The first line contains two space-separated integers n, m ($2 \le n \le 100$, $1 \le m \le 100$).

The second line contains n space-separated integers $a[1], a[2], \ldots, a[n]$ ($1 \le a[i] \le 10^9$) — the description of the array.

The following m lines contain the description of *good* pairs. The k-th line contains two space-separated integers i_k, j_k ($1 \le i_k < j_k \le n$, $i_k + j_k$ is an odd number).

It is guaranteed that all the *good* pairs are distinct.

Output

Output the answer for the problem.

Examples

input
3 2 8 3 8 1 2 2 3

output
0

input
3 2 8 12 8 1 2 2 3

output
2

D. Traffic Jams in the Land

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Some country consists of $(n + 1)$ cities, located along a straight highway. Let's number the cities with consecutive integers from 1 to $n + 1$ in the order they occur along the highway. Thus, the cities are connected by n segments of the highway, the i-th segment connects cities number i and $i + 1$. Every segment of the highway is associated with a positive integer $a_i > 1$ — the period of traffic jams appearance on it.

In order to get from city x to city y ($x < y$), some drivers use the following tactics.

Initially the driver is in city x and the current time t equals zero. Until the driver arrives in city y, he perfors the following actions:

- if the current time t is a multiple of a_x, then the segment of the highway number x is now having traffic problems and the driver stays in the current city for one unit of time (formally speaking, we assign $t = t + 1$);
- if the current time t is not a **multiple** of a_x, then the segment of the highway number x is now clear and that's why the driver uses one unit of time to move to city $x + 1$ (formally, we assign $t = t + 1$ and $x = x + 1$).

You are developing a new traffic control system. You want to consecutively process q queries of two types:

1. determine the final value of time t after the ride from city x to city y ($x < y$) assuming that we apply the tactics that is described above. Note that for each query t is being reset to 0.
2. replace the period of traffic jams appearing on the segment number x by value y (formally, assign $a_x = y$).

Write a code that will effectively process the queries given above.

Input

The first line contains a single integer n ($1 \le n \le 10^5$) — the number of highway segments that connect the $n + 1$ cities.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($2 \le a_i \le 6$) — the periods of traffic jams appearance on segments of the highway.

The next line contains a single integer q ($1 \le q \le 10^5$) — the number of queries to process.

The next q lines contain the descriptions of the queries in the format c, x, y (c — the query type).

If c is character 'A', then your task is to process a query of the first type. In this case the following constraints are satisfied: $1 \le x < y \le n + 1$.

If c is character 'C', then you need to process a query of the second type. In such case, the following constraints are satisfied: $1 \le x \le n$, $2 \le y \le 6$.

Output

For each query of the first type output a single integer — the final value of time t after driving from city x to city y. Process the queries in the order in which they are given in the input.

Examples

input
10 2 5 3 2 3 5 3 4 2 4 10 C 10 6 A 2 6 A 1 3 C 3 4 A 3 11 A 4 9 A 5 6 C 7 3 A 8 10 A 2 5
output
5 3 14 6 2 4 4

E. Stairs and Lines

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given a figure on a grid representing stairs consisting of 7 steps. The width of the stair on height i is w_i squares. Formally, the figure is created by consecutively joining rectangles of size $w_i \times i$ so that the w_i sides lie on one straight line. Thus, for example, if all $w_i = 1$, the figure will look like that (different colors represent different rectangles):

And if $w = \{5, 1, 0, 3, 0, 0, 1\}$, then it looks like that:

Find the number of ways to color some borders of the figure's inner squares so that no square had all four borders colored. The borders of the squares lying on the border of the figure should be considered painted. The ways that differ with the figure's rotation should be considered distinct.

Input

The single line of the input contains 7 numbers $w_1, w_2, \ldots, w_7$ ($0 \le w_i \le 10^5$). It is guaranteed that at least one of the w_i's isn't equal to zero.

Output

In the single line of the output display a single number — the answer to the problem modulo $10^9 + 7$.

Examples

input
0 1 0 0 0 0 0
output
1

input
0 2 0 0 0 0 0
output
7

input
1 1 1 0 0 0 0
output
9

input
5 1 0 3 0 0 1
output
411199181

Note

All the possible ways of painting the third sample are given below: