

Problem A. LCM Query

Input file: `stdin`
Output file: `stdout`
Time limit: 4 seconds
Memory limit: 256 megabytes

De Prezer loves *lcm* (Least Common Multiple). Ha has got a sequence $a_1, a_2, \dots, a_n$ but doesn't know how to calculate *lcm* of two numbers.

De Prezer also loves query. So he asks you to answer to m queries on this sequence.

In each query, he gives you number x and you should print the following number :

$$\min_{i=1}^{n-x+1} lcm(a_i, a_{i+1}, \dots, a_{i+x-1})$$

Answer can be very large, so print it modulo $10^9 + 7$.

Input

The first line of input consists of 2 integers n and m .

The second line of input contains n space separated integers $a_1, a_2, \dots, a_n$.

The next m lines, each line contains an integer x .

$$1 \leq n \leq 2 * 10^4$$

$$1 \leq m \leq 10^6$$

$$1 \leq a_i \leq 60 \text{ (For each } 1 \leq i \leq n \text{)}$$

$$1 \leq x \leq n$$

Output

Print m lines, each answer to one query.

Examples

stdin	stdout
5 5 1 2 3 4 5 1 2 3 4 5	1 2 6 12 60
5 5 2 3 1 4 5 1 2 3 4 5	1 3 6 12 60

Problem B. ShortestPath Query

Input file: `stdin`
Output file: `stdout`
Time limit: 1 second
Memory limit: 256 megabytes

De Prezer loves *troyic* paths. Consider we have a graph with n vertices and m edges. Edges are directed in one way. And there is at most one edge from any vertex to any other vertex. If there is an edge from v to u , then $c(v, u)$ is its color and $w(v, u)$ is its length. Otherwise, $c(v, u) = w(v, u) = -1$.

A sequence $p_1, p_2, \dots, p_k$ is a troyic path if and only if for each $1 \leq i \leq k$, $1 \leq p_i \leq n$ and if $i < k$, then $c(p_i, p_{i+1}) > -1$ and if $i + 1 < k$, then $c(p_i, p_{i+1}) \neq c(p_{i+1}, p_{i+2})$.

The length of such troyic path is $\sum_{i=1}^{k-1} w(p_i, p_{i+1})$ and it's called a $p_1 - p_k$ path.

In such graph, length of the shortest path from vertex v to u is the minimum length of all $v - u$ paths. (The length of the shortest path from any vertex to itself equals 0)

De Prezer gives you a graph like above and a vertex s .

De Prezer also loves query. So he gives you q queries and in each query, gives you number t and you should print the length of the shortest path from s to t (or -1 if there is no troyic path from s to t)

Input

The first line of input contains three integers n and m and C , the number of vertices, the numbers of edges and the number of valid colors.

The next m lines, each line contains 4 integers $v, u, w(v, u), c(v, u)$ ($1 \leq v, u \leq n$ and $v \neq u$ and $1 \leq w(v, u) \leq 10^9$ and $1 \leq c(v, u) \leq C$).

The line after that contains integer s and q .

The next q lines, each line contains information of one query, number t .

$$1 \leq n, m, C, q \leq 10^5$$

$$m \leq n(n-1)$$

$$1 \leq s, t \leq n$$

Output

For each query, print the answer.

Examples

stdin	stdout
5 4 1000 1 2 10 1 2 3 10 2 3 4 10 2 4 5 10 1 1 5 1 2 3 4 5	0 10 20 -1 -1
5 5 2 1 2 10 1 2 3 10 2 3 4 10 1 4 5 10 2 1 5 39 1 1 5 1 2 3 4 5	0 10 20 30 39

Problem C. Subrect Query

Input file: `stdin`
Output file: `stdout`
Time limit: 8 seconds
Memory limit: 512 megabytes

De Prezer loves rectangles. He has a $n \times m$ rectangle which there is a number in each of its cells. We show the number in the j -th column of the i -th row by $a_{i,j}$.

De Prezer also loves query. So he gives you q queries. In each query, he gives you number k and asks you to print the number of subrectangles of this rectangle that the difference between the maximum element and the minimum element in them is at most k .

Input

The first line of input contains 3 integers, n , m and q .

In the next n lines, there are informations of the rectangle. i -th line among them, contains m space separated integers, $a_{i,1}, a_{i,2}, \dots, a_{i,m}$.

The next q lines, each line contains a single integer k (for that query).

$$1 \leq n, m \leq 400$$

$$1 \leq q \leq 10$$

$$1 \leq a_{i,j} \leq 10^9 \text{ (for each } 1 \leq i \leq n \text{ and } 1 \leq j \leq m)$$

$$0 \leq k \leq 10^9 \text{ (for each query)}$$

Output

For each query, print the answer in a single line.

Examples

stdin	stdout
5 4 6 451 451 452 452 452 452 452 452 451 452 450 450 451 451 451 451 452 452 450 450 0 2 773726 724963313 1 1	42 150 150 150 88 88
4 5 8 1314 1287 1286 1290 1295 1278 1271 1324 1317 1289 1305 1305 1284 1300 1309 1318 1296 1301 1274 1315 976296835 12 13 38 16 40 665711658 35	150 34 35 82 37 92 150 77

Problem D. TROY Query

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

De Prezer loves *TROY*s. A *TROY* is a $10^{18} \times 10^{18}$ square and there is either $+1$ or -1 in each cell of this square (actually it's a grid).

There are two types of operation :

1. Multiply all numbers in a row by -1 .
2. Multiply all numbers in a column by -1 .

De Prezer has found two *TROY*s, the number in the j - *th* column of the i - *th* row of the first *TROY* is $a_{i,j}$ and in the second *TROY* is $b_{i,j}$.

De Prezer also loves query, so he gives you some queries.

First of all, you don't have any information about numbers in these two *TROY*s. Each query, gives the values $x, y, a_{x,y}, b_{x,y}$ (that you didn't get before), and you should tell him if it is can be possible to transform the first *TROY* to the second one (the values in the cells that we don't know, could be the way that we can transform the first *TROY* to the second one) using the operations above, with the information you got so far.

Input

The first line of input contains integer n , the number of queries.

Each of the next n lines, contain 4 integers $x, y, a_{x,y}, b_{x,y}$.

$$1 \leq n \leq 10^5$$

$$1 \leq x, y \leq 10^{18} \text{ and } a_{x,y}, b_{x,y} \in \{-1, +1\}$$

Output

For each query, print a single string in a line, "Yes" or "No" (Without quotes).

Examples

stdin	stdout
3 829054240386762533 576622723736087196 +1 -1 659693256240999920 576622723736087196 +1 +1 395697514003384346 576622723736087196 +1 +1	Yes Yes Yes
4 1 1 +1 -1 1 2 +1 -1 2 1 +1 -1 2 2 +1 +1	Yes Yes Yes No

Problem E. Palindrome Query

Input file: `stdin`
Output file: `stdout`
Time limit: 4 seconds
Memory limit: 256 megabytes

De Prezer loves palindrome strings. A string $s_1s_2...s_n$ is palindrome if and only if it is equal to its reverse.

De Prezer also loves queries.

You are given string s of length n and m queries. There are 3 types of queries :

- 1 $p\ x$: Modify $s_p = x$ where $1 \leq p \leq n$ and x is a lower case English letter.
- 2 p : Print the length of the largest palindrome substring of s like $s_ls_{l+1}...s_r$ such that $l \leq p \leq r$ and $r - p = p - l$. ($1 \leq p \leq n$)
- 3 p : Print the length of the largest palindrome substring of s like $s_ls_{l+1}...s_r$ such that $l \leq p$ and $p + 1 \leq r$ and $r - p - 1 = p - l$. ($1 \leq p \leq n - 1$) or -1 if there is no such substring.

Input

The first line of input contains s and m .

Next m lines contain queries.

$1 \leq n, m \leq 10^5$

s only contains lower case English letters.

Output

For each query of type 2 and 3 print the answer in a single line.

Examples

stdin	stdout
abcd 3 3 1 1 2 c 3 2	-1 2
abcba 6 2 3 1 3 b 2 3 3 2 1 1 b 3 2	5 5 2 4

Problem F. Tree Query

Input file: `stdin`
Output file: `stdout`
Time limit: 2 seconds
Memory limit: 256 megabytes

De Prezer loves trees. A tree of order n is a connected graph with n vertexes and $n - 1$ edges.

De Prezer also loves query.

De Prezer has a weighted tree of order n (each edge has a length). The distance between two vertexes is the sum of length of edges between them and we show it by $d(v, u)$ (distance between vertexes u and v).

De Prezer ask you to perform q queries on this tree.

Each query gives you numbers v, l and you should tell him the number of vertexes like u such that $d(v, u) \leq l$ (including v itself).

Input

The first line of input contains integers n and q .

The next $n - 1$ lines contain edges. Each line contains an edge a, b, w , two endpoints and the weight.

The next q lines, contain queries.

$$1 \leq n, q \leq 10^5$$

$$1 \leq v \leq n$$

$$1 \leq l \leq 10^{14}$$

$$1 \leq a, b \leq n$$

$$1 \leq w \leq 10^9$$

Output

For each query, print a single integer in a single line.

Examples

stdin	stdout
4 6 1 2 3 2 3 10 3 4 51 1 2 1 10 3 30 3 51 2 60 2 61	1 2 3 4 3 4
5 5 1 2 100 1 3 100 1 4 10 1 5 200 1 99 1 199 1 299 4 100 4 110	2 4 5 2 4