Problem A. Anime

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

Aizhan is watching the last episode of the JJK. The duration of the last episode is exactly n seconds. For every moment, she knows the level of interest: $I(t)$. Fortunately, function $I(t)$ is a continuous piecewise linear function. She knows the values of $I(t)$ at integer moments of time: $I(0), I(1), I(2), I(3), \ldots, I(n)$. To get values at non-integer moments, she can draw the graph of the function I with points $(x, I(x))$ for every integer x from 0 to n, and then connect consecutive points by straight line segments. The interest of watching a fragment of the episode is the area under the graph for that fragment.

The video player is very strange. It has only two buttons: fast forward and rewind. The first button will forward the video by k seconds, and the second button will rewind the video by k seconds. So it is not possible to stop the video.

By trial and error, Aizhan noticed that it is not possible to forward the video or rewind it if she ends up outside of the episode's domain; in other words, she must stay in the time interval $[0, n]$ (domain expansion is not possible). The second peculiarity is the fact that the number of uses for both buttons should be equal after she has finished the episode; otherwise, her computer will blow up, which is not fun at all.

Aizhan has finished watching the episode without using the buttons, and the cumulative interest she got is the area under the graph of function I. Now, she wonders what is the maximum possible cumulative interest if she uses the buttons optimally.

Note that the buttons can be used at any moment, even at non-integer moments. The buttons can be used any number of times, as long as, at the end of the episode, the first button is used the same number of times as the second one.

Input

On the first line, you are given two integers n and k ($1 \le k \le n \le 10^5$): the duration of the episode and the characteristic of the video player.

On the second line, you are given $n+1$ integers $I(0), I(1), I(2), I(3), \ldots, I(n)$ ($0 \le I(i) \le 10^5$): the level of interest at all the integer moments.

Output

In the single line, output one number: the answer for the problem.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-6}.

Examples

standard input	*standard output*
2 1 0 1 2	3.000000
2 1 0 5 0	7.500000

Note

In the first example, Aizhan can use the forward button at moment 0, watch the time interval $[1, 2)$ of the video, then rewind it to watch the time interval $[1, 2)$ again.

Note that Aizhan watched time interval $[1, 2)$ twice, and the area under this interval on the graph is 1.5. Thus, the cumulative interest is 3.0.

Problem B. Golden Medals

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The following rule applies at the Interesting Olympiad in Informatics (IOI): gold medals are awarded to k participants who have scored the highest number of points, where k must be at least $n/12$, where n is the total number of participants.

You know the total number of participants who have arrived, determine the minimum number of participants who can receive gold medals. It is assumed that all participants have scored at least one point.

Input

The input contains one integer n ($200 \leq n \leq 10^5$).

Output

Output one integer — the minimum number of gold medalists.

Examples

standard input	standard output
300	25
333	28

Problem C. Nomad Camp

Input file: *standard input*
Output file: *standard output*
Time limit: 5 seconds
Memory limit: 256 mebibytes

On summer vacation, Amir stayed at his grandmother's house, where she told him stories about how nomadic people in ancient times chose pastures for themselves:

There are only n pastures, numbered from 1 to n, and m roads available. Each pasture belongs to one of the four types: қыстау (winter), көктеу (spring), жайлау (summer), and күзеу (autumn).

Each pasture is initially inhabited by people, regardless of the season. When the season changes, from each pasture, people move to the nearest pasture corresponding to the new season. If there are multiple nearest pastures, they choose the pasture with the smallest number. If there is no pasture for the new season, people become sad and stop moving at all.

Now Amir wonders if it would be possible to gather all the people in one place if people could change the season of the year to any other season, as many times as they like.

Input

The first line contains a single integer T ($1 \leq T \leq 10^4$): the number of test cases. For each test case:

The first line contains two integers n and m ($1 \leq n \leq 200$, $1 \leq m \leq \frac{n \cdot (n-1)}{2}$): the number of pastures and the number of roads between them.

The second line contains n integers $c_1, c_2, \ldots, c_n$ ($1 \leq c_i \leq 4$): the types of pastures.

Each of the next m lines contains three integers, u_i, v_i, and w_i ($1 \leq u_i, v_i \leq n$, $1 \leq w_i \leq 10^5$), which mean there is a bidirectional road between pastures u_i and v_i that has length w_i.

It is guaranteed that the sum of n for all test cases does not exceed 10^5.

It is guaranteed that the sum of m for all test cases does not exceed 10^6.

Output

Output T lines, each of which is the answer to the corresponding test case. As the answer, output "YES" if it is possible to gather everyone in one place, and "NO" otherwise.

You can output the answer in any case (for example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as a positive answer).

Example

standard input	*standard output*
2 4 4 1 2 2 4 1 2 5 2 3 100 3 4 8 1 3 11 7 9 3 1 3 2 4 1 2 3 5 7 7 1 1 1 2 7 1 5 1 4 7 10 4 5 10 5 2 11 2 7 3 3 4 10	YES NO

Problem D. Key Fragments

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

At the CTF competiton, the participants received a memory dump in the form of a sequence of zeros and ones of length l.

It is also known that the key fragment containing the required data contains exactly k zeros (a fragment in this problem is a sequence consisting of all elements of the sequence between the i-th and j-th inclusive, where $0 \leq i \leq j < l$).

Your task is to calculate, for the given dump and the number k, how many fragments of the sequence can be key fragments. Fragments are considered different if the index of the first or last element in the sequence is different (even if these fragments are equal as strings, that is, for example, in the dump 10101, the fragments (0,1) and (2,3) are considered different).

Input

The first line of the input contains a non-empty sequence of zeros and ones, specified without spaces. The length of the sequence l does not exceed 10^6. The second line contains a single integer k: the required number of zeros in the fragment ($1 \leq k \leq l$).

Output

Output a single integer: the number of different fragments containing exactly k zeros.

Example

standard input	standard output
10011001 2	13

Problem E. Bugs and Jars

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

Biologists have n bugs. For an experiment to study the behavior of bugs in a confined space, it is planned to place the bugs in k jars in such a way that each jar contains at least one bug, no two jars have the same number of bugs, and the difference between the largest and smallest number of bugs in one jar is minimal.

For the given n and k, determine the corresponding minimum value of the difference.

Input

The first line of the input contains two integers n and k ($2 \le n \le 10^5$, $2 \le k \le 1000$).

Output

If there is no way to place the bugs in the jars as required, output -1. Otherwise, output the smallest possible value of the difference between the largest and smallest number of bugs in one jar.

Examples

standard input	standard output
`4 3`	`-1`
`7 3`	`3`

Problem F. Geometry Enjoyer

Input file:	*standard input*
Output file:	*standard output*
Time limit:	7 seconds
Memory limit:	512 mebibytes

Altair was playing with the points on the plane (as usual). At some point, he discovered a new game that he will play with you.

He made a convex polygon with k sides on the two-dimensional plane. The polygon had a really nice property: no pair of sides are parallel. Then he extended every side of the polygon to a line, and found the intersection point for every pair of lines.

Now he gives you the points he got. You should find the initial polygon.

Input

The first line contains one integer n ($1 \le n \le 200$): the number of points.

Each of the next n lines contains four integers, p_x, q_x, p_y, and q_y ($-10^6 \le p_x, p_y \le 10^6$, $1 \le q_x, q_y \le 10^6$): the coordinates of the i-th point. The X coordinate equals p_x/q_x, and the Y coordinate equals p_y/q_y. It is guaranteed that the values p_x and q_x are coprime, and the values p_y and q_y are coprime.

It is guaranteed that the polygon can be uniquely determined by the given points.

Output

The first line of the output should contain one integer k: the size of the polygon.

You can output the vertices of the polygon in any order.

Each of the next k lines should contain four integers, p_x, q_x, p_y, and q_y ($-10^6 \le p_x, p_y \le 10^6$, $1 \le q_x, q_y \le 10^6$): the coordinates of the polygon vertices. The X coordinate equals p_x/q_x, and the Y coordinate equals p_y/q_y. The values p_x and q_x should be coprime, and the values p_y and q_y should be coprime.

Example

standard input	*standard output*
6 1 1 2 1 12 5 24 5 0 1 0 1 3 1 3 1 -3 1 0 1 4 1 0 1	4 0 1 0 1 1 1 2 1 3 1 3 1 4 1 0 1

Problem G. Kangaroo

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

There are kangaroos living in the $1 \times n$ grid. Each cell of the grid contains either kangaroo's habitat or the food. One cell with food can provide the food for no more than one kangaroo.

Kangaroo can move k or less cells left or right from its cell to find the free food cell for meal. During the travel, it can pass cells with other kangaroos as well as the cells with food (free or occupied by the kangaroo).

Given k and distributions of cells, find the maximum number of kangaroos that can have the meal simultaneously.

Input

The first line of the input contains two integers n and k ($1 \leq n \leq 2 \cdot 10^4$, $1 \leq k \leq 10$) — the number of the cells and the maximum travel distance by kangaroo, in cells (if $k = 1$, a kangaroo can move only to the neighboring cells). The second line contains n characters '`K`' and '`F`', denoting the cells with the kangaroos and the food, respectively: if the i-th letter is '`K`', there is a kangaroo's habitat, otherwise there is a food.

Output

Print one integer — maximal number of the kangaroos that can get a meal simultaneously if we will distribute kangaroos on the food cells optimally.

Examples

standard input	standard output
12 1 FKFKFKFFKKFK	5
12 2 FKFKFKFFKKFK	6

Problem H. Lost Table

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 256 mebibytes

Er-Tostik had a table of size $n \times m$ with positive integers. Aldar-Kose decided to prank Er-Tostik and stole the table, but told Er-Tostik the maximum value in each row and column. Aldar-Kose will only return the table if Er-Tostik can tell how many different tables can have these maximum values. As their number can be very large, Aldar-Kose only asks to find this value modulo $10^9 + 7$. Help Er-Tostik to get his table back.

Input

The first line of input contains two integers n and m ($1 \le n, m \le 2 \cdot 10^5$): the dimensions of the table.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$): the maximum values in each row.

The third line contains m integers $b_1, b_2, \ldots, b_m$ ($1 \le b_j \le 10^9$): the maximum values in each column.

Output

Output a line with a single integer: the number of different tables satisfying the conditions. Since the answer can be very large, output it modulo $10^9 + 7$.

Note that, as Aldar-Kose is mischievous, the input might not be consistent with any table at all. In such case, naturally, the correct answer is 0.

Examples

standard input	*standard output*
3 3 2 2 3 2 3 3	89
1 1 1 2	0
5 5 2 2 3 3 3 2 2 2 3 3	49049891
12 13 2 2 2 3 3 4 4 4 4 5 5 5 2 3 3 3 3 4 5 5 5 5 5 5 5	808346164
2 3 2 3 3 1 5	0

Problem I. Stations

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

The monorail road in the capital of Byteland consists of a single line with n stations. The i-th station from the beginning of the road generates a revenue of s_i (revenue can be negative — some stations are unprofitable).

The city hall plans to divide the stations among four management companies in such a way that each station is controlled by exactly one company, all stations controlled by one company are consecutive, and the total revenue from the stations of each company is the same. Formally, the city hall wants to choose three integers i, j, k, such that $1 \le i < j < k < n$ and $s_1 + \ldots + s_i = s_{i+1} + \ldots + s_j = s_{j+1} + \ldots + s_k = s_{k+1} + \ldots + s_n$.

In how many ways can the city hall divide the stations as required?

Input

The first line of the input contains a single integer n, the number of stations ($4 \le n \le 10^5$). The second line contains n integers. The i-th of these numbers, s_i ($-1000 \le s_i \le 1000$), represents the revenue generated by the i-th station from the beginning.

Output

Output a single integer, the number of ways to choose i, j, and k and divide the stations according to the city hall's plans.

Examples

standard input	standard output
10 1 7 3 1 2 8 2 9 5 6	1
10 9 -4 -5 0 -1 1 -4 4 -5 5	4
4 -417 379 797 -43	0

Problem J. Reachability in a Matrix

Input file:	*standard input*
Output file:	*standard output*
Time limit:	3 seconds
Memory limit:	512 mebibytes

You are given a matrix A of size $n \times m$ consisting of **distinct** integers from 1 to $n \cdot m$. The rows of the matrix are numbered from 1 to n, and the columns are numbered from 1 to m. Also, a **positive** integer k is given.

Let us construct a graph consisting of $n \cdot m$ vertices, where the vertices will be the cells of the matrix, labeled as (a, b) ($1 \le a \le n$, $1 \le b \le m$). We will draw a **directed** edge from cell (a, b) to cell (c, d) if both of the following conditions are met:

- The cells are in the same row or column of the matrix. More formally, $a = c$ or $b = d$.
- $A_{a,b} \ge A_{c,d} + k$.

You are given q queries of the form (a, b, c, d). You need to determine whether there exists a path in this graph along the directed edges, starting at vertex (a, b) and ending at vertex (c, d).

Input

The first line of the input file contains three integers, n, m, and k ($1 \le n, m \le 250$, $1 \le k \le n \cdot m$).

Each of the next n lines contains m integers separated by spaces: the values $A_{i,j}$ ($1 \le A_{i,j} \le n \cdot m$). It is guaranteed that all numbers in the matrix are distinct.

The next line contains a single integer q: the number of queries ($1 \le q \le 250\,000$).

Each of the next q lines contains four integers, a_i, b_i, c_i, and d_i: the vertices in the i-th query ($1 \le a_i, c_i \le n$, $1 \le b_i, d_i \le m$, $(a_i, b_i) \ne (c_i, d_i)$).

Output

For each of the q queries, output a line with the word “`Ia`” if a path exists. Otherwise, output a line with the word “`Joq`”.

Example

standard input	*standard output*
3 3 2 2 4 6 1 8 3 5 9 7 6 3 2 1 3 3 1 1 1 3 2 1 1 1 3 2 1 3 2 2 3 2 2 3 3	Joq Ia Ia Ia Ia Joq

Note

In the third query, there exist paths $(3, 2) \to (3, 1) \to (1, 1)$ and $(3, 2) \to (1, 2) \to (1, 1)$.

In the fourth query, there exists a path $(1, 3) \to (2, 3) \to (2, 1)$.

Problem K. Bitvzhuh

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Daniyar recently learned a new spell called “Bitvzhuh”. Although it is a very high level spell, Daniyar was able to master it completely and unlock its deepest secrets.

“Bitvzhuh”, when cast on a set of integers, transforms the set into a new set which contains the XORs of all pairs in the initial set.

Formally, say you have a set $A = \{a_1, a_2, \ldots, a_n\}$ of size n. After one “Bitvzhuh”, A turns into the set $\{a_i \oplus a_j \mid 1 \le i < j \le n\}$, where $\oplus$ denotes the bitwise XOR operation.

Given the initial set and the number k, find out if Daniyar can apply “Bitvzhuh” a certain **non-zero** number of times so that the resulting set will contain each integer in the range $[1, 2^k - 1]$.

Input

The first line contains two integers n and k ($3 \le n \le 10^6$, $2 \le k \le 62$): the size of the initial set and the parameter.

The second line contains n distinct integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i < 2^k$): the elements of the initial set.

Output

Print a single line with the word “`Yes`” if the set will contain each integer in the range $[1, 2^k - 1]$ after a certain **non-zero** number of casts of “Bitvzhuh”. Otherwise, print a single line with the word “`No`”.

Examples

standard input	*standard output*
4 3 1 2 3 4	Yes
4 3 1 2 4 7	No

Note

In the first example, the answer is achieved after two casts:

$\{1, 2, 3, 4\} \to \{1, 2, 3, 5, 6, 7\} \to \{1, 2, 3, 4, 5, 6, 7\}$.

In the second example, the first cast turns the set $\{1, 2, 4, 7\}$ into $\{3, 5, 6\}$, and it never changes after.