

Codeforces Round #773 (Div. 1)

A. Great Sequence

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

A sequence of positive integers is called *great* for a positive integer x , if we can split it into pairs in such a way that in each pair the first number multiplied by x is equal to the second number. More formally, a sequence a of size n is great for a positive integer x , if n is even and there exists a permutation p of size n , such that for each i ($1 \leq i \leq \frac{n}{2}$) $a_{p_{2i-1}} \cdot x = a_{p_{2i}}$.

Sam has a sequence a and a positive integer x . Help him to make the sequence great: find the minimum possible number of positive integers that should be added to the sequence a to make it great for the number x .

Input

Each test contains multiple test cases. The first line contains a single integer t ($1 \leq t \leq 20\,000$) — the number of test cases. Description of the test cases follows.

The first line of each test case contains two integers n, x ($1 \leq n \leq 2 \cdot 10^5, 2 \leq x \leq 10^6$).

The next line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$).

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case print a single integer — the minimum number of integers that can be added to the end of a to make it a great sequence for the number x .

Example

input
4 4 4 1 16 4 4 6 2 1 2 2 2 4 7 5 3 5 2 3 5 15 9 10 10 10 10 20 1 100 200 2000 3
output
0 2 3 3

Note

In the first test case, Sam got lucky and the sequence is already great for the number 4 because you can divide it into such pairs: $(1, 4)$, $(4, 16)$. Thus we can add 0 numbers.

In the second test case, you can add numbers 1 and 14 to the sequence, then you can divide all 8 integers into such pairs: $(1, 2)$, $(1, 2)$, $(2, 4)$, $(7, 14)$. It is impossible to add less than 2 integers to fix the sequence.

B. Repetitions Decoding

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

Olya has an array of integers a_1, a_2, \dots, a_n . She wants to split it into tandem repeats. Since it's rarely possible, before that she wants to perform the following operation several (possibly, zero) number of times: insert a pair of equal numbers into an arbitrary position. Help her!

More formally:

- A tandem repeat is a sequence x of even length $2k$ such that for each $1 \leq i \leq k$ the condition $x_i = x_{i+k}$ is satisfied.
- An array a could be split into tandem repeats if you can split it into several parts, each being a subsegment of the array, such that each part is a tandem repeat.

- In one operation you can choose an arbitrary letter c and insert $[c, c]$ to any position in the array (at the beginning, between any two integers, or at the end).
- You are to perform several operations and split the array into tandem repeats or determine that it is impossible. Please note that you do **not** have to minimize the number of operations.

Input

Each test contains multiple test cases. The first line contains a single integer t ($1 \leq t \leq 30\,000$) — the number of test cases. Description of the test cases follows.

The first line of each test case contains a single integer n ($1 \leq n \leq 500$).

The second line contains n integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — the initial array.

It is guaranteed that the sum of n^2 over all test cases does not exceed $250\,000$.

Output

For each test case print answer in the following format.

If you cannot turn the array into a concatenation of tandem repeats, print a single integer -1 .

Otherwise print the number of operations q ($0 \leq q \leq 2 \cdot n^2$) that you want to do. Then print the descriptions of operations.

In each of the following q lines print two integers p and c ($1 \leq c \leq 10^9$), which mean that you insert the integer c twice after p elements of the array. If the length of the array is m before the operation, then the condition $0 \leq p \leq m$ should be satisfied.

Then you should print any way to split the resulting array into tandem repeats. First, print a single integer d , and then print a sequence t_1, t_2, \dots, t_d of even integers of size d ($d, t_i \geq 1$). These numbers are the lengths of the subsegments from left to right.

Note that the size of the resulting array a is $m = n + 2 \cdot q$. The following statements must hold:

- $m = \sum_{i=1}^d t_i$.
- For all integer i such that $1 \leq i \leq d$, the sequence a_l, a_{l+1}, \dots, a_r is a tandem repeat, where $l = \sum_{j=1}^{i-1} t_j + 1, r = l + t_i - 1$.

It can be shown that if the array can be turned into a concatenation of tandem repeats, then there exists a solution satisfying all constraints. If there are multiple answers, you can print any.

Example

input
<div>4</div> <div>2</div> <div>5 7</div> <div>2</div> <div>5 5</div> <div>6</div> <div>1 3 1 2 2 3</div> <div>6</div> <div>3 2 1 1 2 3</div>
output
<div>-1</div> <div>0</div> <div>1</div> <div>2</div> <div>4</div> <div>1 3</div> <div>5 3</div> <div>5 3</div> <div>10 3</div> <div>2</div> <div>8 6</div> <div>5</div> <div>0 3</div> <div>8 3</div> <div>5 3</div> <div>6 2</div> <div>7 1</div> <div>4</div> <div>2 6 6 2</div>

Note

In the first test case, you cannot apply operations to the array to make it possible to split it into tandem repeats.

In the second test case the array is already a tandem repeat $[5, 5] = \underbrace{([5] + [5])}_{t_1=2}$, thus we can do no operations at all.

In the third test case, initially, we have the following array:

$$[1, 3, 1, 2, 2, 3].$$

After the first insertion with $p = 1, c = 3$:

$[1, \mathbf{3}, \mathbf{3}, 3, 1, 2, 2, 3]$.

After the second insertion with $p = 5, c = 3$:

$[1, 3, 3, 3, 1, \mathbf{3}, \mathbf{3}, 2, 2, 3]$.

After the third insertion with $p = 5, c = 3$:

$[1, 3, 3, 3, 1, \mathbf{3}, \mathbf{3}, 3, 2, 2, 3]$.

After the fourth insertion with $p = 10, c = 3$:

$[1, 3, 3, 3, 1, 3, 3, 3, 2, \mathbf{3}, \mathbf{3}, 2, 3]$.

The resulting array can be represented as a concatenation of tandem repeats:

$$\underbrace{([1, 3, 3, 3] + [1, 3, 3, 3])}_{t_1=8} + \underbrace{([3, 2, 3] + [3, 2, 3])}_{t_2=6}.$$

In the fourth test case, initially, we have the following array:

$[3, 2, 1, 1, 2, 3]$.

After the first insertion with $p = 0, c = 3$:

$[\mathbf{3}, \mathbf{3}, 3, 2, 1, 1, 2, 3]$.

After the second insertion with $p = 8, c = 3$:

$[3, 3, 3, 2, 1, 1, 2, 3, \mathbf{3}, \mathbf{3}]$.

After the third insertion with $p = 5, c = 3$:

$[3, 3, 3, 2, 1, \mathbf{3}, \mathbf{3}, 1, 2, 3, 3, 3]$.

After the fourth insertion with $p = 6, c = 2$:

$[3, 3, 3, 2, 1, 3, \mathbf{2}, \mathbf{2}, 3, 1, 2, 3, 3, 3]$.

After the fifth insertion with $p = 7, c = 1$:

$[3, 3, 3, 2, 1, 3, 2, \mathbf{1}, \mathbf{1}, 2, 3, 1, 2, 3, 3, 3]$.

The resulting array can be represented as a concatenation of tandem repeats:

$$\underbrace{([3] + [3])}_{t_1=2} + \underbrace{([3, 2, 1] + [3, 2, 1])}_{t_2=6} + \underbrace{([1, 2, 3] + [1, 2, 3])}_{t_3=6} + \underbrace{([3] + [3])}_{t_4=2}.$$

C. Anonymity Is Important

time limit per test: 1.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

In the work of a doctor, it is important to maintain the anonymity of clients and the results of tests. The test results are sent to everyone personally by email, but people are very impatient and they want to know the results right away.

That's why in the testing lab "De-vitro" doctors came up with an experimental way to report the results. Let's assume that n people took the tests in the order of the queue. Then the chief doctor Sam can make several statements, in each telling if there is a sick person among the people in the queue from l -th to r -th (inclusive), for some values l and r .

During the process, Sam will check how well this scheme works and will be interested in whether it is possible to find out the test result of i -th person from the information he announced. And if it can be done, then is that patient sick or not.

Help Sam to test his scheme.

Input

The first line contains two integers n, q ($1 \leq n, q \leq 2 \cdot 10^5$) — the number of people and the number of queries.

In each of the next q lines, the description of the query is given. The first number in the line is t ($t = 0$ or $t = 1$) — the type of the query.

If $t = 0$, the line contains three more integers l, r, x ($1 \leq l \leq r \leq n, x = 0$ or $x = 1$). This query means that Sam tells that among the people in the queue from l -th to r -th (inclusive):

- there was at least one sick person, if $x = 1$,
- there is no sick people, if $x = 0$.

If $t = 1$, the line contains one more integer j ($1 \leq j \leq n$) — the position of the patient in the queue, for which Sam wants to know the status.

All queries are correct, that means that there always exists an example of the queue of length n for which all reported results (statements from queries with $t = 0$) are true.

Output

After each Sam question (query with $t = 1$) print:

- "NO", if the patient is definitely not sick,
- "YES", if the patient is definitely sick.
- "N/A", if it is impossible to definitely identify the status of patient having the given information.

Example

input
6 9 0 4 5 0 1 5 1 6 0 4 6 1 1 6 0 2 5 1 0 2 2 0 1 3 1 2
output
NO N/A YES YES NO

Note

In the first test for the five first queries:

1. Initially Sam tells that people 4, 5 are not sick.
2. In the next query Sam asks the status of the patient 5. From the previous query, we know that the patient is definitely not sick.
3. In the next query Sam asks the status of the patient 6. We don't know any information about that patient now.
4. After that Sam tells that there exists a sick patient among 4, 5, 6.
5. In the next query Sam asks the status of the patient 6. Now we can tell that this patient is definitely sick.

D. Two Arrays

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Sam changed his school and on the first biology lesson he got a very interesting task about genes.

You are given n arrays, the i -th of them contains m different integers — $a_{i,1}, a_{i,2}, \dots, a_{i,m}$. Also you are given an array of integers w of length n .

Find the minimum value of $w_i + w_j$ among all pairs of integers (i, j) ($1 \leq i, j \leq n$), such that the numbers $a_{i,1}, a_{i,2}, \dots, a_{i,m}, a_{j,1}, a_{j,2}, \dots, a_{j,m}$ are distinct.

Input

The first line contains two integers n, m ($2 \leq n \leq 10^5, 1 \leq m \leq 5$).

The i -th of the next n lines starts with m distinct integers $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ and then w_i follows ($1 \leq a_{i,j} \leq 10^9, 1 \leq w_i \leq 10^9$).

Output

Print a single number — the answer to the problem.

If there are no suitable pairs (i, j) , print -1 .

Examples

input
4 2 1 2 5 4 3 1 2 3 2 4 5 3
output
5

input
4 3 1 2 3 5 2 3 4 2 3 4 5 3 1 3 10 10
output
-1

Note

In the first test the minimum value is $5 = w_3 + w_4$, because numbers $\{2, 3, 4, 5\}$ are distinct.

In the second test case, there are no suitable pair (i, j) .

E. Special Positions

time limit per test: 5 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

You are given an array a of length n . Also you are given m distinct positions p_1, p_2, \dots, p_m ($1 \leq p_i \leq n$).

A **non-empty** subset of these positions T is randomly selected with equal probability and the following value is calculated:

$$\sum_{i=1}^n (a_i \cdot \min_{j \in T} |i - j|).$$

In other word, for each index of the array, a_i and the distance to the closest chosen position are multiplied, and then these values are summed up.

Find the expected value of this sum.

This value must be found modulo 998 244 353. More formally, let $M = 998\,244\,353$. It can be shown that the answer can be represented as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \pmod{M}$. In other words, output such integer x that $0 \leq x < M$ and $x \cdot q = p \pmod{M}$.

Input

The first line contains two integers n and m ($1 \leq m \leq n \leq 10^5$).

The second line contains n integers a_1, a_2, \dots, a_n ($0 \leq a_i < 998\,244\,353$).

The third line contains m distinct integers p_1, p_2, \dots, p_m ($1 \leq p_i \leq n$).

For every $1 \leq i < m$ it is guaranteed that $p_i < p_{i+1}$.

Output

Print a single integer — the answer to the problem.

Examples

input
4 2 1 2 3 4 1 4
output
665496247

input
6 6 4 2 4 2 4 2 1 2 3 4 5 6
output
855638030

Note

In the first test:

- If only 1 is choosen, than the value equals to $1 \cdot 0 + 2 \cdot 1 + 3 \cdot 2 + 4 \cdot 3 = 20$.
- If only 4 is choosen, than the value equals to $1 \cdot 3 + 2 \cdot 2 + 3 \cdot 1 + 4 \cdot 0 = 10$.
- If both positions are chosen, than the value equals to $1 \cdot 0 + 2 \cdot 1 + 3 \cdot 1 + 4 \cdot 0 = 5$.

The answer to the problem is $\frac{20+10+5}{3} = \frac{35}{3} = 665\,496\,247$ (modulo 998 244 353).

F. Covering Circle

time limit per test: 6 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Sam started playing with round buckets in the sandbox, while also scattering pebbles. His mom decided to buy him a new bucket, so she needs to solve the following task.

You are given n distinct points with integer coordinates A_1, A_2, \dots, A_n . All points were generated from the square $[-10^8, 10^8] \times [-10^8, 10^8]$ uniformly and independently.

You are given positive integers k, l , such that $k \leq l \leq n$. You want to select a subsegment $A_i, A_{i+1}, \dots, A_{i+l-1}$ of the points array (for some $1 \leq i \leq n + 1 - l$), and some circle on the plane, containing $\geq k$ points of the selected subsegment (inside or on the border).

What is the smallest possible radius of that circle?

Input

Each test contains multiple test cases. The first line contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases. Descriptions of test cases follow.

The first line of each test case contains three integers n, l, k ($2 \leq k \leq l \leq n \leq 50\,000, k \leq 20$).

Each of the next n lines contains two integers x_i, y_i ($-10^8 \leq x_i, y_i \leq 10^8$) — the coordinates of the point A_i . It is guaranteed that all points are distinct and were generated independently from uniform distribution on $[-10^8, 10^8] \times [-10^8, 10^8]$.

It is guaranteed that the sum of n for all test cases does not exceed 50 000.

In the first test, points were not generated from the uniform distribution on $[-10^8, 10^8] \times [-10^8, 10^8]$ for simplicity. It is **the only** such test and your solution **must** pass it.

Hacks are disabled in this problem.

Output

For each test case print a single real number — the answer to the problem.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-9} . Formally let your answer be a , jury answer be b . Your answer will be considered correct if $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-9}$.

Example

input
4 3 2 2 0 0 0 4 3 0 5 4 3 1 1 0 0 2 2 0 2 2 0 8 3 2 0 3 1 0 0 2 1 1 0 1 1 2 0 0 1 3 5 4 4 1 1 -3 3 2 2 5 3 5 5
output
2.000000000000000000 1.000000000000000000 0.500000000000000000 4.000000000000000000

Note

In the first test case, we can select subsegment A_1, A_2 and a circle with center $(0, 2)$ and radius 2.

In the second test case, we can select subsegment A_1, A_2, A_3, A_4 and a circle with center $(1, 2)$ and radius 1.

