Codeforces Round #773 (Div. 2)

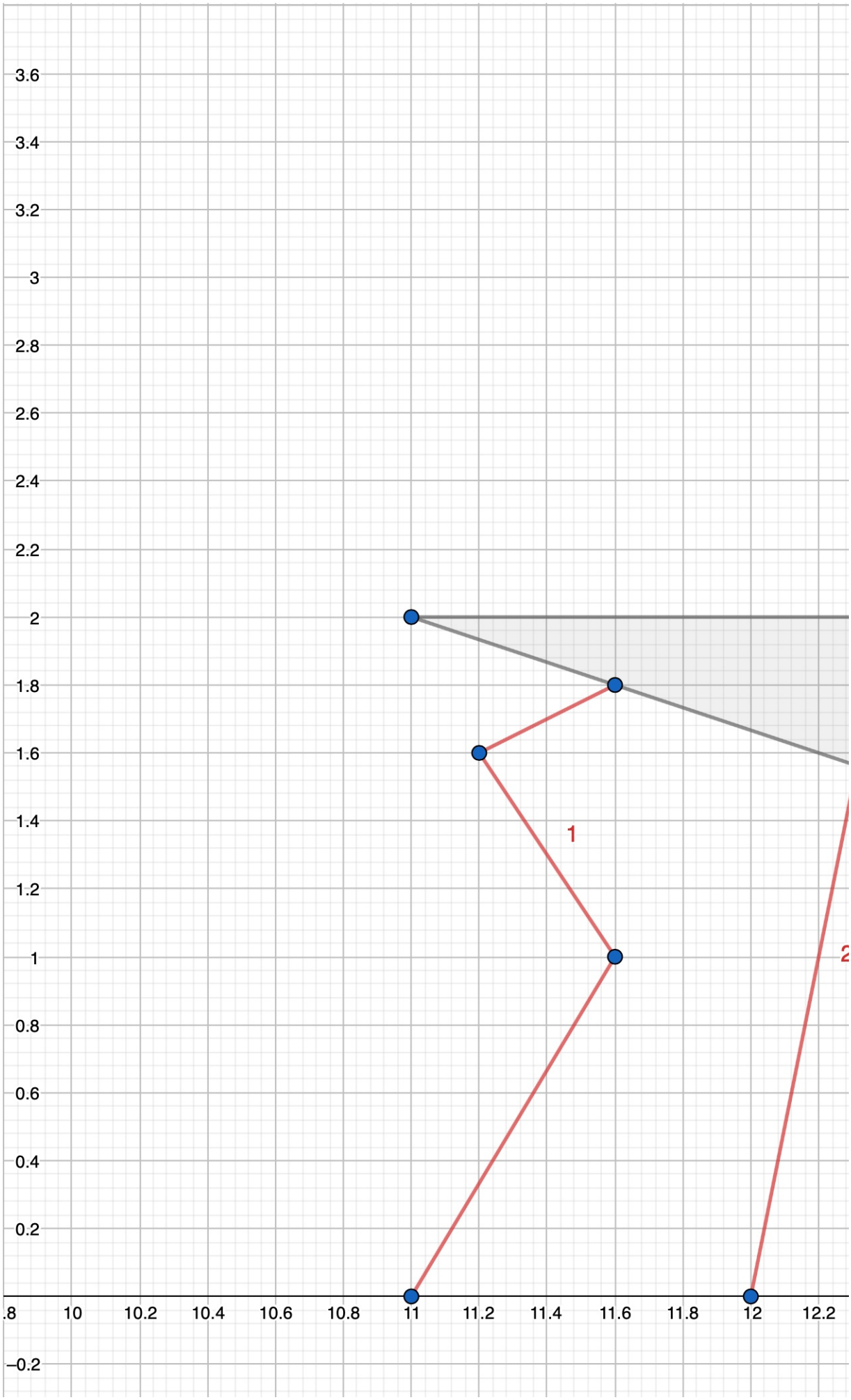
A. Hard Way

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

Sam lives in Awesomeburg, its downtown has a triangular shape. Also, the following is true about the triangle:

- its vertices have integer coordinates,
- the coordinates of vertices are non-negative, and
- its vertices are not on a single line.

He calls a point on the downtown's border (that is the border of the triangle) *safe* if he can reach this point from **at least one point** of the line $y = 0$ walking along some **straight line**, without crossing the interior of the triangle.



In the picture the downtown is marked with grey color. The first path is invalid because it does not go along a straight line. The second path is invalid because it intersects with the interior of the downtown. The third and fourth paths are correct.

Find the total length of the unsafe parts of the downtown border. It can be proven that these parts are segments and their number is finite.

Input

Each test contains multiple test cases. The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases. Description of the test cases follows.

Each test case contains three lines, each of them contains two integers x_i, y_i ($0 \le x_i, y_i \le 10^9$) — coordinates of the vertices of the

downtown's border.

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

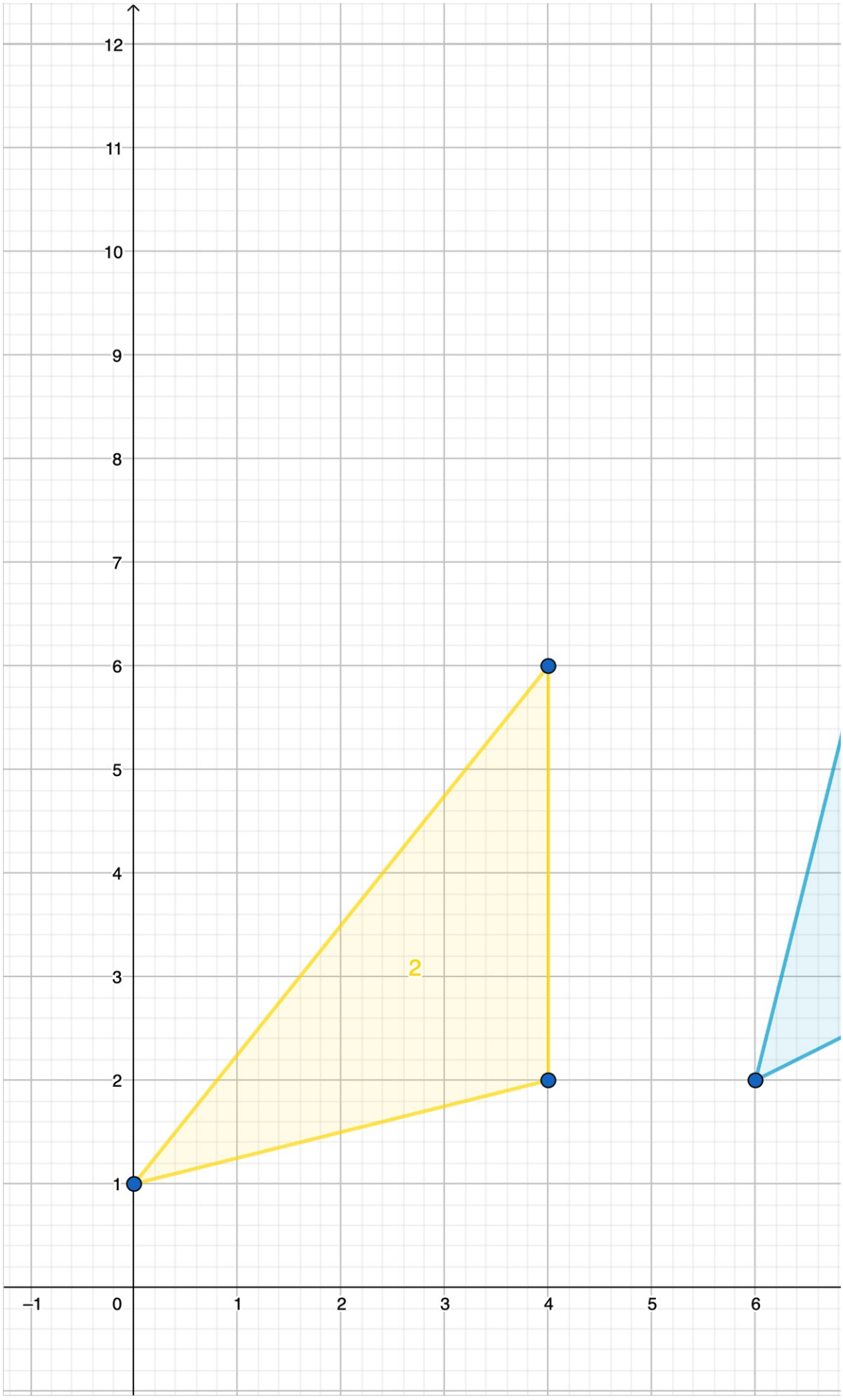
Your answer will be considered correct if its absolute or relative error does not exceed 10^{-9}. Formally let your answer be a, jury answer be b. Your answer will be considered correct if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-9}$.

Example

input
5 8 10 10 4 6 2 4 6 0 1 4 2 14 1 11 2 13 2 0 0 4 0 2 4 0 1 1 1 0 0

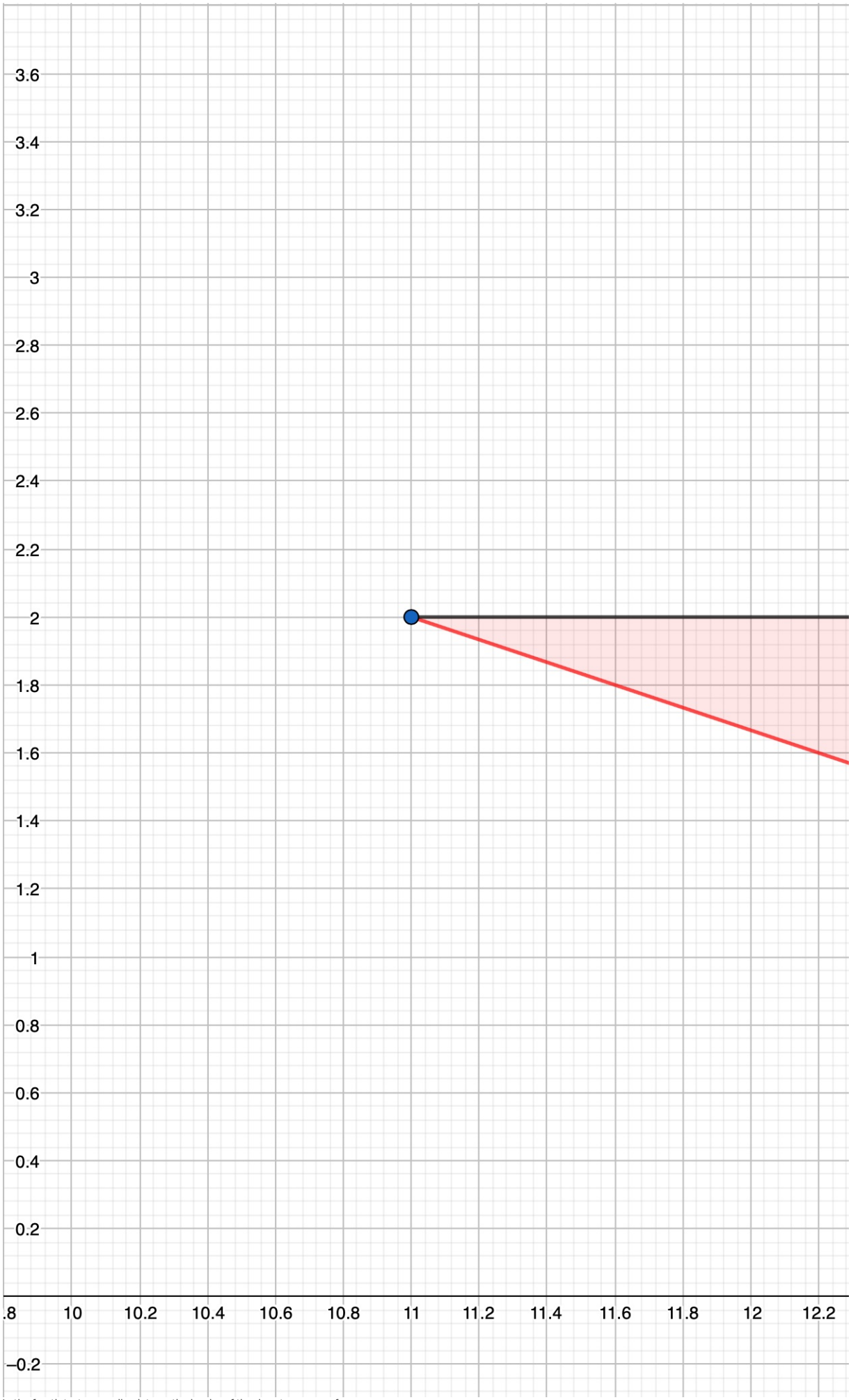
output
0.0000000 0 2.0000 0.00 1

Note
In the picture, the downtowns of the first three test cases are illustrated. Triangles are enumerated according to the indices of test cases they belong to.



In the first two test cases, all points on the borders of the downtowns are safe, thus the answers are 0.

In the following picture unsafe points for the third test case are marked with black color:



In the fourth test case, all points on the border of the downtown are safe.

B. Power Walking

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

Sam is a kindergartener, and there are n children in his group. He decided to create a team with some of his children to play

"brawl:go 2".

Sam has n power-ups, the i-th has type a_i. A child's strength is equal to the number of **different** types among power-ups he has.

For a team of size k, Sam will distribute all n power-ups to k children in such a way that each of the k children receives at least one power-up, and each power-up is given to someone.

For each integer k from 1 to n, find the **minimum** sum of strengths of a team of k children Sam can get.

Input
Each test contains multiple test cases. The first line contains a single integer t ($1 \le t \le 3 \cdot 10^5$) — the number of test cases. Description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 3 \cdot 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — types of Sam's power-ups.

It is guaranteed that the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output
For every test case print n integers.

The k-th integer should be equal to the minimum sum of strengths of children in the team of size k that Sam can get.

Example

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

Note
One of the ways to give power-ups to minimise the sum of strengths in the first test case:

- $k = 1 : \{1, 1, 2\}$
- $k = 2 : \{1, 1\}, \{2\}$
- $k = 3 : \{1\}, \{1\}, \{2\}$

One of the ways to give power-ups to minimise the sum of strengths in the second test case:

- $k = 1 : \{1, 2, 2, 2, 4, 5\}$
- $k = 2 : \{2, 2, 2, 4, 5\}, \{1\}$
- $k = 3 : \{2, 2, 2, 5\}, \{1\}, \{4\}$
- $k = 4 : \{2, 2, 2\}, \{1\}, \{4\}, \{5\}$
- $k = 5 : \{2, 2\}, \{1\}, \{2\}, \{4\}, \{5\}$
- $k = 6 : \{1\}, \{2\}, \{2\}, \{2\}, \{4\}, \{5\}$

C. 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 ($1 \le i \le \frac{n}{2}$) $a_{p_{2i-1}} \cdot x = a_{p_{2i}}$.

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 ($1 \le t \le 20\,000$) — the number of test cases. Description of the test cases follows.

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

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

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.

D. 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 $a_1, a_2, \ldots, a_n$. 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 $1 \le i \le k$ 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 ($1 \le t \le 30\,000$) — the number of test cases. Description of the test cases follows.

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

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — 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 ($0 \le q \le 2 \cdot n^2$) that you want to do. Then print the descriptions of operations.

In each of the following q lines print two integers p and c ($1 \le c \le 10^9$), 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 $0 \le p \le m$ 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 $t_1, t_2, \ldots, t_d$ of even integers of size d ($d, t_i \ge 1$). 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:

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

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
4 2 5 7 2 5 5 6 1 3 1 2 2 3 6 3 2 1 1 2 3

output
-1 0 1 2 4 1 3 5 3 5 3 10 3 2 8 6 5 0 3 8 3 5 3 6 2 7 1 4 2 6 6 2

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 $p = 1, c = 3$:

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

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

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

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

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

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

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

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 $p = 0, c = 3$:

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

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

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

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

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

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

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

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

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

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

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

E. 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 ($1 \le n, q \le 2 \cdot 10^5$) — 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 ($1 \le l \le r \le n$, $x = 0$ 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 ($1 \le j \le n$) — 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.

F. 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 — $a_{i,1}, a_{i,2}, \ldots, a_{i,m}$. 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) ($1 \le i, j \le n$), such that the numbers $a_{i,1}, a_{i,2}, \ldots, a_{i,m}, a_{j,1}, a_{j,2}, \ldots, a_{j,m}$ are distinct.

Input

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

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

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).