Lista AA 2022 #4

A. Going Home

2 seconds, 256 megabytes

It was the third month of remote learning, Nastya got sick of staying at dormitory, so she decided to return to her hometown. In order to make her trip more entertaining, one of Nastya's friend presented her an integer array a.

Several hours after starting her journey home Nastya remembered about the present. To entertain herself she decided to check, are there four **different** indices x, y, z, w such that $a_x + a_y = a_z + a_w$.

Her train has already arrived the destination, but she still hasn't found the answer. Can you help her unravel the mystery?

Input

The first line contains the single integer n ($4 \le n \le 200\,000$) — the size of the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 2.5 \cdot 10^6$).

Output

Print "YES" if there are such four indices, and "NO" otherwise.

If such indices exist, print these indices x, y, z and w ($1 \le x, y, z, w \le n$).

If there are multiple answers, print any of them.

input
6 2 1 5 2 7 4
output
YES 2 3 1 6

input
5 1 3 1 9 20
output
NO

In the first example $a_2 + a_3 = 1 + 5 = 2 + 4 = a_1 + a_6$. Note that there are other answer, for example, 2 3 4 6.

In the second example, we can't choose four indices. The answer 1 2 2 3 is wrong, because indices should be different, despite that $a_1 + a_2 = 1 + 3 = 3 + 1 = a_2 + a_3$

B. Orac and Medians

2 seconds, 256 megabytes

Slime has a sequence of positive integers $a_1, a_2, \ldots, a_n$.

In one operation Orac can choose an arbitrary subsegment $[l \ldots r]$ of this sequence and replace all values $a_l, a_{l+1}, \ldots, a_r$ to the value of median of $\{a_l, a_{l+1}, \ldots, a_r\}$.

In this problem, for the integer multiset s, the median of s is equal to the $\lfloor \frac{|s|+1}{2} \rfloor$-th smallest number in it. For example, the median of $\{1, 4, 4, 6, 5\}$ is 4, and the median of $\{1, 7, 5, 8\}$ is 5.

Slime wants Orac to make $a_1 = a_2 = \ldots = a_n = k$ using these operations.

Orac thinks that it is impossible, and he does not want to waste his time, so he decided to ask you if it is possible to satisfy the Slime's requirement, he may ask you these questions several times.

Input

The first line of the input is a single integer t: the number of queries.

The first line of each query contains two integers n ($1 \le n \le 100\,000$) and k ($1 \le k \le 10^9$), the second line contains n positive integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$)

The total sum of n is at most $100\,000$.

Output

The output should contain t lines. The i-th line should be equal to 'yes' if it is possible to make all integers k in some number of operations or 'no', otherwise. You can print each letter in lowercase or uppercase.

input
5 5 3 1 5 2 6 1 1 6 6 3 2 1 2 3 4 3 3 1 2 3 10 3 1 2 3 4 5 6 7 8 9 10
output
no yes yes no yes

In the first query, Orac can't turn all elements into 3.

In the second query, $a_1 = 6$ is already satisfied.

In the third query, Orac can select the complete array and turn all elements into 2.

In the fourth query, Orac can't turn all elements into 3.

In the fifth query, Orac can select $[1, 6]$ at first and then select $[2, 10]$.

C. Painting Fence

1 second, 512 megabytes

Bizon the Champion isn't just attentive, he also is very hardworking.

Bizon the Champion decided to paint his old fence his favorite color, orange. The fence is represented as n vertical planks, put in a row. Adjacent planks have no gap between them. The planks are numbered from the left to the right starting from one, the i-th plank has the width of 1 meter and the height of a_i meters.

Bizon the Champion bought a brush in the shop, the brush's width is 1 meter. He can make vertical and horizontal strokes with the brush. During a stroke the brush's full surface must touch the fence at all the time (see the samples for the better understanding). What minimum number of strokes should Bizon the Champion do to fully paint the fence? Note that you are allowed to paint the same area of the fence multiple times.

Input

The first line contains integer n ($1 \le n \le 5000$) — the number of fence planks. The second line contains n space-separated integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 10^9$).

Output

Print a single integer — the minimum number of strokes needed to paint the whole fence.

input
5 2 2 1 2 1
output
3

input
2 2 2
output
2

input
1 5
output
1

In the first sample you need to paint the fence in three strokes with the brush: the first stroke goes on height 1 horizontally along all the planks. The second stroke goes on height 2 horizontally and paints the first and second planks and the third stroke (it can be horizontal and vertical) finishes painting the fourth plank.

In the second sample you can paint the fence with two strokes, either two horizontal or two vertical strokes.

In the third sample there is only one plank that can be painted using a single vertical stroke.

D. Not Adding

2 seconds, 256 megabytes

You have an array $a_1, a_2, \ldots, a_n$ consisting of n **distinct** integers. You are allowed to perform the following operation on it:

- Choose two elements from the array a_i and a_j $(i \neq j)$ such that $\gcd(a_i, a_j)$ is not present in the array, and add $\gcd(a_i, a_j)$ to the end of the array. Here $\gcd(x, y)$ denotes greatest common divisor (GCD) of integers x and y.

Note that the array changes after each operation, and the subsequent operations are performed on the new array.

What is the **maximum** number of times you can perform the operation on the array?

Input

The first line consists of a single integer n $(2 \leq n \leq 10^6)$.

The second line consists of n integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq 10^6)$. All a_i are **distinct**.

Output

Output a single line containing one integer — the maximum number of times the operation can be performed on the given array.

input
5 4 20 1 25 30
output
3

input
3 6 10 15
output
4

In the first example, one of the ways to perform maximum number of operations on the array is:

- Pick $i = 1, j = 5$ and add $\gcd(a_1, a_5) = \gcd(4, 30) = 2$ to the array.
- Pick $i = 2, j = 4$ and add $\gcd(a_2, a_4) = \gcd(20, 25) = 5$ to the array.
- Pick $i = 2, j = 5$ and add $\gcd(a_2, a_5) = \gcd(20, 30) = 10$ to the array.

It can be proved that there is no way to perform more than 3 operations on the original array.

In the second example one can add 3, then 1, then 5, and 2.

E. Gargari and Permutations

2 seconds, 256 megabytes

Gargari got bored to play with the bishops and now, after solving the problem about them, he is trying to do math homework. In a math book he have found k permutations. Each of them consists of numbers $1, 2, ..., n$ in some order. Now he should find the length of the longest common subsequence of these permutations. Can you help Gargari?

You can read about longest common subsequence there: `https://en.wikipedia.org/wiki/Longest_common_subsequence_problem`

Input

The first line contains two integers n and k $(1 \leq n \leq 1000; 2 \leq k \leq 5)$. Each of the next k lines contains integers $1, 2, ..., n$ in some order — description of the current permutation.

Output

Print the length of the longest common subsequence.

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

The answer for the first test sample is subsequence [1, 2, 3].

F. Lost Tree

3 seconds, 256 megabytes

This is an interactive problem.

Little Dormi was faced with an awkward problem at the carnival: he has to guess the edges of an unweighted tree of n nodes! The nodes of the tree are numbered from 1 to n.

The game master only allows him to ask one type of question:

- Little Dormi picks a node r $(1 \leq r \leq n)$, and the game master will reply with an array $d_1, d_2, \ldots, d_n$, where d_i is the length of the shortest path from node r to i, for all $1 \leq i \leq n$.

Additionally, to ~~make the game unfair~~ challenge Little Dormi the game master will allow at most $\lceil \frac{n}{2} \rceil$ questions, where $\lceil x \rceil$ denotes the smallest integer greater than or equal to x.

Faced with the stomach-churning possibility of not being able to guess the tree, Little Dormi needs your help to devise a winning strategy!

Note that the game master creates the tree before the game starts, and does not change it during the game.

Input

The first line of input contains the integer n ($2 \leq n \leq 2\,000$), the number of nodes in the tree.

You will then begin interaction.

Output

When your program has found the tree, first output a line consisting of a single "`!`" followed by $n-1$ lines each with two space separated integers a and b, denoting an edge connecting nodes a and b ($1 \leq a, b \leq n$). Once you are done, terminate your program normally immediately after flushing the output stream.

You may output the edges in any order and an edge (a, b) is considered the same as an edge (b, a). Answering is not considered as a query.

Interaction

After taking input, you may make at most $\lceil \frac{n}{2} \rceil$ queries. Each query is made in the format "`? r`", where r is an integer $1 \leq r \leq n$ that denotes the node you want to pick for that query.

You will then receive n space separated integers $d_1, d_2, \ldots, d_n$, where d_i is the length of the shortest path from node r to i, followed by a newline.

After printing a query do not forget to output end of line and flush the output. Otherwise, you will get `Idleness limit exceeded`. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

If at any point you make an invalid query or try to make more than $\lceil \frac{n}{2} \rceil$ queries, the interaction will terminate immediately and you will receive a Wrong Answer verdict.

Hacks

To hack a solution, use the following format.

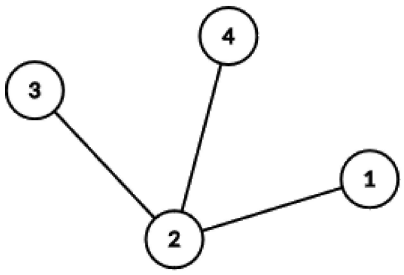
The first line contains the integer n ($2 \leq n \leq 2\,000$).

The next $n-1$ lines contain two integers u and v ($1 \leq u, v \leq n$) denoting an edge between u and v ($u \neq v$). These $n-1$ edges must form a tree.

input
4 0 1 2 2 1 0 1 1
output
? 1 ? 2 ! 4 2 1 2 2 3

input
5 2 2 1 1 0
output
? 5 ! 4 5 3 5 2 4 1 3

Here is the tree from the first example.

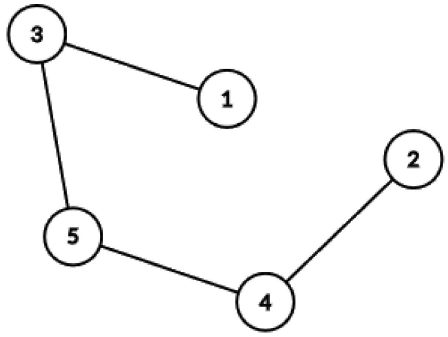


Notice that the edges can be output in any order.

Additionally, here are the answers for querying every single node in example 1:

- 1: $[0, 1, 2, 2]$
- 2: $[1, 0, 1, 1]$
- 3: $[2, 1, 0, 2]$
- 4: $[2, 1, 2, 0]$

Below is the tree from the second example interaction.



Lastly, here are the answers for querying every single node in example 2:

- 1: $[0, 4, 1, 3, 2]$
- 2: $[4, 0, 3, 1, 2]$
- 3: $[1, 3, 0, 2, 1]$
- 4: $[3, 1, 2, 0, 1]$
- 5: $[2, 2, 1, 1, 0]$

G. Need for Pink Slips

1 second, 256 megabytes

After defeating a Blacklist Rival, you get a chance to draw 1 reward slip out of x hidden valid slips. Initially, $x = 3$ and these hidden valid slips are Cash Slip, Impound Strike Release Marker and Pink Slip of Rival's Car. Initially, the probability of drawing these in a random guess are c, m, and p, respectively. There is also a volatility factor v. You can play any number of Rival Races as long as you don't draw a Pink Slip. Assume that you win each race and get a chance to draw a reward slip. In each draw, you draw one of the x valid items with their respective probabilities. Suppose you draw a particular item and its probability of drawing before the draw was a. Then,

- If the item was a Pink Slip, the quest is over, and you will not play any more races.
- Otherwise,
  1. If $a \leq v$, the probability of the item drawn becomes 0 and the item is no longer a valid item for all the further draws, reducing x by 1. Moreover, the reduced probability a is distributed equally among the other remaining valid items.
  2. If $a > v$, the probability of the item drawn reduces by v and the reduced probability is distributed equally among the other valid items.

For example,

- If $(c, m, p) = (0.2, 0.1, 0.7)$ and $v = 0.1$, after drawing Cash, the new probabilities will be $(0.1, 0.15, 0.75)$.
- If $(c, m, p) = (0.1, 0.2, 0.7)$ and $v = 0.2$, after drawing Cash, the new probabilities will be $(Invalid, 0.25, 0.75)$.
- If $(c, m, p) = (0.2, Invalid, 0.8)$ and $v = 0.1$, after drawing Cash, the new probabilities will be $(0.1, Invalid, 0.9)$.
- If $(c, m, p) = (0.1, Invalid, 0.9)$ and $v = 0.2$, after drawing Cash, the new probabilities will be $(Invalid, Invalid, 1.0)$.

You need the cars of Rivals. So, you need to find the expected number of races that you must play in order to draw a pink slip.

Input

The first line of input contains a single integer t ($1 \le t \le 10$) — the number of test cases.

The first and the only line of each test case contains four real numbers c, m, p and v ($0 < c, m, p < 1$, $c + m + p = 1$, $0.1 \le v \le 0.9$).

Additionally, it is guaranteed that each of c, m, p and v have at most 4 decimal places.

Output

For each test case, output a single line containing a single real number — the expected number of races that you must play in order to draw a Pink Slip.

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

Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-6}$.

input
4 0.2 0.2 0.6 0.2 0.4 0.2 0.4 0.8 0.4998 0.4998 0.0004 0.1666 0.3125 0.6561 0.0314 0.2048
output
1.532000000000 1.860000000000 5.005050776521 4.260163673896

For the first test case, the possible drawing sequences are:

- P with a probability of 0.6;
- CP with a probability of $0.2 \cdot 0.7 = 0.14$;
- CMP with a probability of $0.2 \cdot 0.3 \cdot 0.9 = 0.054$;
- CMMP with a probability of $0.2 \cdot 0.3 \cdot 0.1 \cdot 1 = 0.006$;
- MP with a probability of $0.2 \cdot 0.7 = 0.14$;
- MCP with a probability of $0.2 \cdot 0.3 \cdot 0.9 = 0.054$;
- MCCP with a probability of $0.2 \cdot 0.3 \cdot 0.1 \cdot 1 = 0.006$.

So, the expected number of races is equal to $1 \cdot 0.6 + 2 \cdot 0.14 + 3 \cdot 0.054 + 4 \cdot 0.006 + 2 \cdot 0.14 + 3 \cdot 0.054 + 4 \cdot 0.006 = 1.532$.

For the second test case, the possible drawing sequences are:

- P with a probability of 0.4;
- CP with a probability of $0.4 \cdot 0.6 = 0.24$;
- CMP with a probability of $0.4 \cdot 0.4 \cdot 1 = 0.16$;
- MP with a probability of $0.2 \cdot 0.5 = 0.1$;
- MCP with a probability of $0.2 \cdot 0.5 \cdot 1 = 0.1$.

So, the expected number of races is equal to $1 \cdot 0.4 + 2 \cdot 0.24 + 3 \cdot 0.16 + 2 \cdot 0.1 + 3 \cdot 0.1 = 1.86$.

H. Nauuo and Cards

1.5 seconds, 256 megabytes

Nauuo is a girl who loves playing cards.

One day she was playing cards but found that the cards were mixed with some empty ones.

There are n cards numbered from 1 to n, and they were mixed with another n empty cards. She piled up the $2n$ cards and drew n of them. The n cards in Nauuo's hands are given. The remaining n cards in the pile are also given in the order from top to bottom.

In one operation she can choose a card in her hands and play it — put it at the bottom of the pile, then draw the top card from the pile.

Nauuo wants to make the n numbered cards piled up in increasing order (the i-th card in the pile from top to bottom is the card i) as quickly as possible. Can you tell her the minimum number of operations?

Input

The first line contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of numbered cards.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le n$) — the initial cards in Nauuo's hands. 0 represents an empty card.

The third line contains n integers $b_1, b_2, \ldots, b_n$ ($0 \le b_i \le n$) — the initial cards in the pile, given in order from top to bottom. 0 represents an empty card.

It is guaranteed that each number from 1 to n appears exactly once, either in $a_{1..n}$ or $b_{1..n}$.

Output

The output contains a single integer — the minimum number of operations to make the n numbered cards piled up in increasing order.

input
3 0 2 0 3 0 1
output
2

input
3 0 2 0 1 0 3
output
4

input
11 0 0 0 5 0 0 0 4 0 0 11 9 2 6 0 8 1 7 0 3 0 10
output
18

Example 1

We can play the card 2 and draw the card 3 in the first operation. After that, we have $[0, 3, 0]$ in hands and the cards in the pile are $[0, 1, 2]$ from top to bottom.

Then, we play the card 3 in the second operation. The cards in the pile are $[1, 2, 3]$, in which the cards are piled up in increasing order.

Example 2

Play an empty card and draw the card 1, then play 1, 2, 3 in order.

I. Maximum Subsequence

1 second, 256 megabytes

You are given an array a consisting of n integers, and additionally an integer m. You have to choose some sequence of indices $b_1, b_2, ..., b_k$ ($1 \le b_1 < b_2 < ... < b_k \le n$) in such a way that the value of $\sum_{i=1}^{k} a_{b_i} \bmod m$ is maximized. Chosen sequence can be empty.

Print the maximum possible value of $\sum_{i=1}^{k} a_{b_i} \bmod m$.

Input

The first line contains two integers n and m ($1 \le n \le 35$, $1 \le m \le 10^9$).

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

Output

Print the maximum possible value of $\sum_{i=1}^{k} a_{b_i} \bmod m$.

input
4 4 5 2 4 1
output
3

input
3 20 199 41 299
output
19

In the first example you can choose a sequence $b = \{1, 2\}$, so the sum $\sum_{i=1}^{k} a_{b_i}$ is equal to 7 (and that's 3 after taking it modulo 4).

In the second example you can choose a sequence $b = \{3\}$.

J. Engineer Artem

1 second, 256 megabytes

Artem is building a new robot. He has a matrix a consisting of n rows and m columns. The cell located on the i-th row from the top and the j-th column from the left has a value $a_{i,j}$ written in it.

If two adjacent cells contain the same value, the robot will break. A matrix is called **good** if no two adjacent cells contain the same value, where two cells are called adjacent if they share a side.

Artem wants to **increment the values in some cells by one** to make a good.

More formally, find a good matrix b that satisfies the following condition —

- For all valid (i, j), either $b_{i,j} = a_{i,j}$ or $b_{i,j} = a_{i,j} + 1$.

For the constraints of this problem, it can be shown that such a matrix b always exists. If there are several such tables, you can output any of them. Please note that you do not have to minimize the number of increments.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10$). Description of the test cases follows.

The first line of each test case contains two integers n, m ($1 \le n \le 100$, $1 \le m \le 100$) — the number of rows and columns, respectively.

The following n lines each contain m integers. The j-th integer in the i-th line is $a_{i,j}$ ($1 \le a_{i,j} \le 10^9$).

Output

For each case, output n lines each containing m integers. The j-th integer in the i-th line is $b_{i,j}$.

input
3 3 2 1 2 4 5 7 8 2 2 1 1 3 3 2 2 1 3 2 2
output
1 2 5 6 7 8 2 1 4 3 2 4 3 2

In all the cases, you can verify that no two adjacent cells have the same value and that b is the same as a with some values incremented by one.