Codeforces Round #709 (Div. 2, based on Technocup 2021 Final Round)

A. Prison Break

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

Michael is accused of violating the social distancing rules and creating a risk of spreading coronavirus. He is now sent to prison. Luckily, Michael knows exactly what the prison looks like from the inside, especially since it's very simple.

The prison can be represented as a rectangle $a \times b$ which is divided into ab cells, each representing a prison cell, common sides being the walls between cells, and sides on the perimeter being the walls leading to freedom. Before sentencing, Michael can ask his friends among the prison employees to make (very well hidden) holes in some of the walls (including walls between cells and the outermost walls). Michael wants to be able to get out of the prison after this, no matter which cell he is placed in. However, he also wants to break as few walls as possible.

Your task is to find out the smallest number of walls to be broken so that there is a path to the outside from every cell after this.

Input
The first line contains a single integer t ($1 \leq t \leq 100$) — the number of test cases.

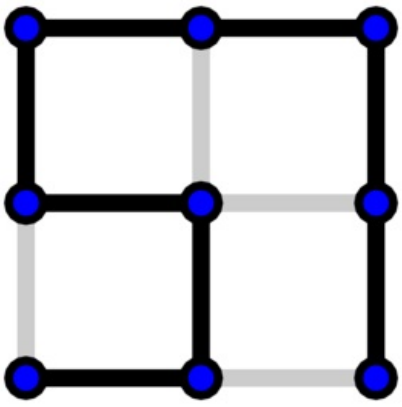
Each of the following t lines contains two integers a and b ($1 \leq a, b \leq 100$), representing a corresponding test case.

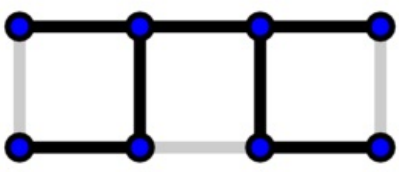
Output
For each test case print the single integer on a separate line — the answer to the problem.

Example

input
2 2 2 1 3
output
4 3

Note
Some possible escape plans for the example test cases are shown below. Broken walls are shown in gray, not broken walls are shown in black.





B. Restore Modulo

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

output: standard output

For the first place at the competition, Alex won many arrays of integers and was assured that these arrays are very expensive. After the award ceremony Alex decided to sell them. There is a rule in arrays pawnshop: you can sell array only if it can be compressed to a generator.

This generator takes four non-negative numbers n, m, c, s. n and m must be positive, s non-negative and for c it must be true that $0 \le c < m$. The array a of length n is created according to the following rules:

- $a_1 = s \bmod m$, here $x \bmod y$ denotes remainder of the division of x by y;
- $a_i = (a_{i-1} + c) \bmod m$ for all i such that $1 < i \le n$.

For example, if $n = 5$, $m = 7$, $c = 4$, and $s = 10$, then $a = [3, 0, 4, 1, 5]$.

Price of such an array is the value of m in this generator.

Alex has a question: how much money he can get for each of the arrays. Please, help him to understand for every array whether there exist four numbers n, m, c, s that generate this array. If yes, then maximize m.

Input
The first line contains a single integer t ($1 \le t \le 10^5$) — the number of arrays.

The first line of array description contains a single integer n ($1 \le n \le 10^5$) — the size of this array.

The second line of array description contains n integers $a_1, a_2, \ldots, a_n$ ($0 \le a_i \le 10^9$) — elements of the array.

It is guaranteed that the sum of array sizes does not exceed 10^5.

Output
For every array print:

- -1, if there are no such four numbers that generate this array;
- 0, if m can be arbitrary large;
- the maximum value m and any appropriate c ($0 \le c < m$) in other cases.

Example

input
6 6 1 9 17 6 14 3 3 4 2 2 3 7 3 4 3 2 2 4 5 0 1000000000 0 1000000000 0 2 1 1
output
19 8 -1 -1 -1 2000000000 1000000000 0

C. Basic Diplomacy

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Aleksey has n friends. He is also on a vacation right now, so he has m days to play this new viral cooperative game! But since it's cooperative, Aleksey will need one teammate in each of these m days.

On each of these days some friends will be available for playing, and all others will not. On each day Aleksey must choose one of his available friends to offer him playing the game (and they, of course, always agree). However, if any of them happens to be chosen strictly more than $\left\lceil \frac{m}{2} \right\rceil$ times, then all other friends are offended. Of course, Aleksey doesn't want to offend anyone.

Help him to choose teammates so that nobody is chosen strictly more than $\left\lceil \frac{m}{2} \right\rceil$ times.

Input
Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10\,000$). Description of the test cases follows.

The first line of each test case contains two integers n and m ($1 \le n, m \le 100\,000$) standing for the number of friends and the number of days to play, respectively.

The i-th of the following m lines contains an integer k_i ($1 \le k_i \le n$), followed by k_i distinct integers f_{i1}, ..., f_{ik_i} ($1 \le f_{ij} \le n$), separated by spaces — indices of available friends on the day i.

It is guaranteed that the sums of n and m over all test cases do not exceed $100\,000$. It's guaranteed that the sum of all k_i over all days of all test cases doesn't exceed $200\,000$.

Output
Print an answer for each test case. If there is no way to achieve the goal, print "NO".

Otherwise, in the first line print "YES", and in the second line print m space separated integers c_1, ..., c_m. Each c_i must denote the chosen friend on day i (and therefore must be one of f_{ij}).

No value must occur more than $\left\lceil \frac{m}{2} \right\rceil$ times. If there is more than one possible answer, print any of them.

Example

input
2 4 6 1 1 2 1 2 3 1 2 3 4 1 2 3 4 2 2 3 1 3 2 2 1 1 1 1
output
YES 1 2 1 1 2 3 NO

D. Playlist

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

Arkady has a playlist that initially consists of n songs, numerated from 1 to n in the order they appear in the playlist. Arkady starts listening to the songs in the playlist one by one, starting from song 1. The playlist is cycled, i. e. after listening to the last song, Arkady will continue listening from the beginning.

Each song has a genre a_i, which is a positive integer. Let Arkady finish listening to a song with genre y, and the genre of the next-to-last listened song be x. If $\gcd(x, y) = 1$, he deletes the last listened song (with genre y) from the playlist. After that he continues listening normally, skipping the deleted songs, and **forgetting** about songs he listened to before. In other words, after he deletes a song, he can't delete the next song immediately.

Here $\gcd(x, y)$ denotes the greatest common divisor (GCD) of integers x and y.

For example, if the initial songs' genres were $[5, 9, 2, 10, 15]$, then the playlist is converted as follows: [**5**, 9, 2, 10, 15] $\to$ [**5**, **9**, 2, 10, 15] $\to$ [5, 2, 10, 15] (because $\gcd(5, 9) = 1$) $\to$ [5, **2**, 10, 15] $\to$ [5, **2**, **10**, 15] $\to$ [5, 2, **10**, **15**] $\to$ [**5**, 2, 10, **15**] $\to$ [**5**, **2**, 10, 15] $\to$ [5, 10, 15] (because $\gcd(5, 2) = 1$) $\to$ [5, **10**, 15] $\to$ [5, **10**, **15**] $\to$ ... The bold numbers represent the two last played songs. Note that after a song is deleted, Arkady forgets that he listened to that and the previous songs.

Given the initial playlist, please determine which songs are eventually deleted and the order these songs are deleted.

Input
Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10\,000$). Description of the test cases follows.

The first line of each test case contains a single integer n ($1 \le n \le 10^5$) — the number of songs.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the genres of the songs.

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

Output
For each test case, print a single line. First, print a single integer k — the number of deleted songs. After that print k distinct integers: deleted songs in the order of their deletion.

Example

input

5
5
5 9 2 10 15
6
1 2 4 2 4 2
2
1 2
1
1
1
2

output

2 2 3
2 2 1
2 2 1
1 1
0

Note

Explanation of the first test case is given in the statement.

In the second test case, the playlist is converted as follows: [**1**, 2, 4, 2, 4, 2] → [**1**, **2**, 4, 2, 4, 2] → [1, 4, 2, 4, 2] (because $\gcd(1, 2) = 1$) → [1, **4**, 2, 4, 2] → [1, **4**, **2**, 4, 2] → [1, 4, **2**, **4**, 2] → [1, 4, 2, **4**, **2**] → [**1**, 4, 2, 4, **2**] → [4, 2, 4, 2] (because $\gcd(2, 1) = 1$) → [**4**, 2, 4, 2] → ...

In the third test case, the playlist is converted as follows: [**1**, 2] → [**1**, **2**] → [1] (because $\gcd(1, 2) = 1$) → [**1**] → [**1**] (Arkady listened to the same song twice in a row) → [] (because $\gcd(1, 1) = 1$).

The fourth test case is same as the third after deletion of the second song.

In the fifth test case, the same song is listened to over and over again, but since $\gcd(2, 2) \neq 1$, it is not deleted.

E. Skyline Photo

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Alice is visiting New York City. To make the trip fun, Alice will take photos of the city skyline and give the set of photos as a present to Bob. However, she wants to find the set of photos with maximum beauty and she needs your help.

There are n buildings in the city, the i-th of them has positive height h_i. All n building heights in the city are different. In addition, each building has a beauty value b_i. Note that beauty can be positive or negative, as there are ugly buildings in the city too.

A set of photos consists of one or more photos of the buildings in the skyline. Each photo includes one or more buildings in the skyline that form a contiguous segment of indices. Each building needs to be in **exactly one** photo. This means that if a building does not appear in any photo, or if a building appears in more than one photo, the set of pictures is not valid.

The beauty of a photo is equivalent to the beauty b_i of the shortest building in it. The total beauty of a set of photos is the sum of the beauty of all photos in it. Help Alice to find the maximum beauty a valid set of photos can have.

Input

The first line contains an integer n ($1 \leq n \leq 3 \cdot 10^5$), the number of buildings on the skyline.

The second line contains n distinct integers $h_1, h_2, \ldots, h_n$ ($1 \leq h_i \leq n$). The i-th number represents the height of building i.

The third line contains n integers $b_1, b_2, \ldots, b_n$ ($-10^9 \leq b_i \leq 10^9$). The i-th number represents the beauty of building i.

Output

Print one number representing the maximum beauty Alice can achieve for a valid set of photos of the skyline.

Examples

input

5
1 2 3 5 4
1 5 3 2 4

output

15

input

5
1 4 3 2 5
-3 4 -10 2 7

output

10

input
2 2 1 -2 -3
output
-3

input
10 4 7 3 2 5 1 9 10 6 8 -4 40 -46 -8 -16 4 -10 41 12 3
output
96

Note

In the first example, Alice can achieve maximum beauty by taking five photos, each one containing one building.

In the second example, Alice can achieve a maximum beauty of 10 by taking four pictures: three just containing one building, on buildings 1, 2 and 5, each photo with beauty -3, 4 and 7 respectively, and another photo containing building 3 and 4, with beauty 2.

In the third example, Alice will just take one picture of the whole city.

In the fourth example, Alice can take the following pictures to achieve maximum beauty: photos with just one building on buildings 1, 2, 8, 9, and 10, and a single photo of buildings 3, 4, 5, 6, and 7.

F. Useful Edges

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

You are given a weighted undirected graph on n vertices along with q triples (u, v, l), where in each triple u and v are vertices and l is a positive integer. An edge e is called *useful* if there is at least one triple (u, v, l) and a path (not necessarily simple) with the following properties:

- u and v are the endpoints of this path,
- e is one of the edges of this path,
- the sum of weights of all edges on this path doesn't exceed l.

Please print the number of useful edges in this graph.

Input

The first line contains two integers n and m ($2 \le n \le 600$, $0 \le m \le \frac{n(n-1)}{2}$).

Each of the following m lines contains three integers u, v and w ($1 \le u, v \le n$, $u \ne v$, $1 \le w \le 10^9$), denoting an edge connecting vertices u and v and having a weight w.

The following line contains the only integer q ($1 \le q \le \frac{n(n-1)}{2}$) denoting the number of triples.

Each of the following q lines contains three integers u, v and l ($1 \le u, v \le n$, $u \ne v$, $1 \le l \le 10^9$) denoting a triple (u, v, l).

It's guaranteed that:

- the graph doesn't contain loops or multiple edges;
- all pairs (u, v) in the triples are also different.

Output

Print a single integer denoting the number of useful edges in the graph.

Examples

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

input
4 2 1 2 10 3 4 10 6 1 2 11 1 3 11 1 4 11 2 3 11 2 4 11 3 4 9
output
1

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

Note

In the first example each edge is useful, except the one of weight 5.

In the second example only edge between 1 and 2 is useful, because it belongs to the path $1 - 2$, and $10 \leq 11$. The edge between 3 and 4, on the other hand, is not useful.

In the third example both edges are useful, for there is a path $1 - 2 - 3 - 2$ of length exactly 5. Please note that the path may pass through a vertex more than once.