Problem A. Teleports

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In SUA, there are n cities which are conveniently labeled with $1, 2, \ldots, n$. There are also m bidirectional roads: the i-th road connects cities a_i and b_i.

Recently, some teleport gates are built to make the traffic more easier. For example, if city A and city B are connected by a teleport gate, then after arriving on the road to city A, you will be immediately teleported to city B. Similarly, if you arrived on the road to city B, then you will be immediately teleported to city A.

You need to find a travel plan to visited every road exactly once.

Input

The input contains zero or more test cases. For each test case:

The first line contains three integers n, m and k ($1 \le n \le 10^5, 1 \le m \le 10^5, 0 \le k \le 10^5$) – the number of cities, the number of roads and the number of teleport gates.

Each of the next m lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n, a_i \ne b_i$) – the indices of the cities connected by the i-th road.

Each of the next k lines contains two integers x_i and y_i ($1 \le x_i, y_i \le n, x_i \ne y_i$) – the indices of the cities connected by the i-th teleport gate.

Two cities may be connected by multiple roads. No city is connected by road with itself. No city is connected by a teleport gate to itself. Any city is connected by teleport to no more than one other city.

It is guaranteed that the total number of cities in all tests does not exceed 10^5. Similarly, the total number of roads and the total number of teleport gates also do not exceed 10^5.

The last line of input contains three zeros. They do not need to be processed.

Output

For each test case, if you cannot find a travel plan, output “No” (without the quotes). Otherwise, output “Yes” (without the quotes), and in the second line output m numbers, denoting the indices of roads. The roads should be displayed in the order in which they should be visited. Roads are numbered from one in the order in which they are given in the input data. For each test, roads are numbered independently.

Example

standard input	standard output

2 1 1
1 2
1 2
4 2 1
1 2
3 4
3 4
4 3 1
1 2
2 3
3 4
1 4
8 10 1
1 3
1 2
3 4
2 4
4 5
2 5
5 7
5 6
6 8
7 8
1 8
0 0 0
	Yes
	1
	No
	Yes
	1 2 3
	Yes
	4 3 1 10 7 6 2 9 8 5

Problem B. Birds

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

As you know, birds like to sit on the wires. In SUA, there is a long wire. At first, several birds were sitting on it. Everything would have been fine, but only a pig running under the wire picked up a cloud of dust that made the birds very angry.

Becoming wicked, the birds began to run along this wire in different directions - someone to the left, someone to the right. All the birds began to run at the same speed, which is one meter per minute. When two birds meet, moving towards each other, they immediately turn around and begin to run at the same speed in the opposite direction.

Since the wire is finite, as soon as one of the birds reaches the end of the wire, it immediately takes off, and all the other birds turn around and start running in the opposite direction. If two birds reach the ends simultaneously, then these two take off, nothing happens else.

You need to find the time when each bird takes off.

Input

The first line contains an integer L ($1 \leq L \leq 10^9$) – the length of the wire in meters.

The second line contains an integer n ($0 \leq n \leq 10^5$) – the number of birds running to the right. The third line contains n different integers $a_1, a_2, \ldots, a_n$ ($0 < a_i < L$) – distances in meters from the left end of the wire to the birds.

The fourth line contains an integer m ($0 \leq m \leq 10^5$) – the number of birds running to the left. The fifth line contains m different integers $b_1, b_2, \ldots, b_m$ ($0 < b_i < L$) – distances in meters from the left end of the wire to the birds.

No two birds are initially in the same place. It is guaranteed that at least one bird is sitting on the wire.

Output

In the first line, output n integers $t_1, t_2, \ldots, t_n$, where t_i means that after t_i minutes the i-th bird facing right will take off.

In the second line, output m integers $u_1, u_2, \ldots, u_m$, where u_i means that after u_i minutes the i-th bird facing left will take off.

Examples

standard input	standard output

10
2
8 9
3
2 5 7
	5 1
	10 13 10

10 0 1 9	9

Problem C. Siege

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

For the glorious city of SUA, hard times have come. From minute to minute, countless enemies will go to the assault. Only magical barriers can help keep the city.

In SUA, there are n artifacts that allow to put a barrier. To activate the i-th artifact, a_i mana (units of magical energy) is required. After that, the enemy can destroy the artifact using b_i mana.

Defenders of SUA have A mana, while the enemies have B mana. The defenders decided to activate artifacts so that after the destruction by the enemy's magicians, the maximum number of artifacts remained active.

Help the city's defenders choose which artifacts to activate.

Input

The first line contains three integers A, B and n ($0 \le A, B \le 10^5, 0 \le n \le 1000$).

Each of the following n lines contains two integers a_i and b_i ($1 \le a_i, b_i \le 10^5$).

Output

In the first line output the number of artifacts that will remain active with optimal actions of both sides.

In the second line output the number of artifacts that the defenders should activate, and the indices of these artifacts. Numbers in one line should be separated by spaces. Artifacts are numbered, beginning with one, in the order they are specified in the input.

Example

standard input	standard output

1 1 2
1 1
1 2
	1
	1 2

Problem D. Shuffle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Chiaki has n integers $a_1, a_2, \ldots, a_n$. Initially, the integers are sorted in non-decreasing order. Chiaki would like to shuffle the integers such that no two consecutive integers have the same value.

To shuffle the integers, a device is used and it can perform the following type of operation: for some i and j, take integers from position i to position j inclusively, and move them to the end in the same order.

You need to find the minimum number of operations need to shuffle the integers.

Input

There are multiple test cases. The first line of input contains an integer T denoting the number of test cases. For each test case:

The first line contains an integer n – the number integers. The next line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_1 \le a_2 \le \cdots \le a_n \le n)$.

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

Output

For each test case, output "-1" (without the quotes) if you can not shuffle the cards using the operations described above. Otherwise, output an integer k in the first line – the minimum number of operation needed. Each of the next k lines output two integers i and j – denoting an operation.

Example

standard input	standard output

2
8
1 1 2 2 3 3 4 4
6
1 1 1 1 2 3
	2
	6 7
	2 3
	-1

Problem E. Timer

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

Chiaki has recently found a big electronic display. The computer that manages the display stores some integer number. The number has n decimal digits, the display shows the encoded version of the number, where each digit is shown using some lowercase letter of the English alphabet.

There is a legend near the display, that describes how the number is encoded. For each digit position i and each digit j the character c is known, that encodes this digit at this position. Different digits can have the same code characters.

Each second the number is increased by 1. And one second after a moment when the number reaches the value that is represented as n 9-s in decimal notation, the loud beep sounds.

Andrew knows the number that is stored in the computer. Now she wants to know how many seconds must pass until Chiaki can definitely tell what was the original number encoded by the display. Assume that Chiaki can precisely measure time, and that the encoded number will first be increased exactly one second after Chiaki started watching at the display.

Input

Input data contains multiple test cases. The first line of input contains t $(1 \leq t \leq 100)$ – the number of test cases. For each test case:

The first line contains an integer n $(1 \leq n \leq 18)$ – the number of digits in the number. The second line contains n decimal digits without spaces (but possibly with leading zeroes) – the number initially stored in the display computer. The following n lines contain 10 characters each. The j-th character of the i-th of these lines is the code character for a digit $j-1$ in position i, most significant digit positions are described first.

Output

For each test case print an integer: the number of seconds until Chiaki definitely knows what was the initial number stored on the display of the computer. Do not print leading zeroes.

Example

standard input	standard output
3 2 42 abcdefghij jihgfedcba 2 42 aaaaaaaaaa aaaaaaaaaa 1 2 abcdabcdff	0 58 2