Cuarta Competencia

A. Unraveling Monty Hall

0.5 s, 1024 MB

On the stage of an auditorium program there are three closed doors: door 1, door 2 and door 3. Behind one of these doors there is a car and behind the other two doors there is a goat. The production of the program randomly chooses the door where the car is without cheating. Only the host of the program knows where the car is. He asks the player to choose one of the doors. We can see that, because there is only one car, and at at least one of the two doors that the player did not choose, there has to be a goat!

Therefore, the presenter can always do the following: between the two doors that the Player did not choose, he opens one that has a goat, so that the player and the spectators can see the goat. The presenter now asks: "Do you want to change your door to the other door that is still closed?". Is it a beneficial change or not? The player wants to stay with the door that has the car, of course!

Paulinho saw a rigorous demonstration that the odds of the car being behind door the player chose initially is $\frac{1}{3}$ and the odds of the car being behind the other door which is still closed and the player did not choose initially is $\frac{2}{3}$ and therefore the exchange is advantageous. Paulinho doesn't conform, his intuition tells him that either way, the probability is $\frac{1}{2}$ for both doors still closed.

To bring this matter to an end, let's simulate this game thousands of times and count how many times the player won the car. We will assume that:

- The player picks door 1 initially;
- The player always changes doors after the presenter reveals a goat by opening one of the two doors that were not initially selected.

In these conditions, in a game, given the number of the door that contains the car, we can know exactly whether the player will win the car or not.

Input

The first line of the input contains an integer N $(1 \le N \le 10^4)$, indicating the number of games in the simulation. Each of the N following rows contains an integer: 1, 2 or 3; representing the door number containing the car in that game.

Output

Your program must produce a single line containing an integer representing the number of times the player won the car in this simulation, assuming that he always chooses the door 1 and always switch doors after the host reveals a goat by opening one of the two doors that were not initially selected.

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

input
```
1
1
```
output
```
0
```

input
```
15
3
2
3
1
1
3
3
2
2
1
2
3
2
1
1
```
output
```
10
```

B. Enigma

0.5 s, 1024 MB

Given an initial configuration, the World War II German encryption machine Enigma replaces each letter typed on the keyboard with some other letter. The replacement strategy was quite complex, but the machine had a vulnerability: a letter would never be replaced by itself! This vulnerability was exploited by Alan Turing, who worked in the cryptanalysis of Enigma during the war. His goal was to find the initial configuration of the machine using the assumption that the message contained a certain usual expression of communication, such as the word `ARMADA`, for example. These expressions were called cribs. If the message `FDMLCRDMRALF` was encrypted, for example, the process of testing the possible configurations of the machine would be simpler because the word `ARMADA`, if present in the encrypted message, could only be in two positions, illustrated in the table below with an arrow. The remaining five positions could not match the crib `ARMADA` because at least one letter in the crib, underlined in the table below, would match its correspondent in the encrypted message; as the machine would never replace a letter by itself, these five positions could be ignored in the tests.

F	D	M	L	C	R	D	M	R	A	L	F
A	R	$\underline{M}$	A	D	A						
	A	R	M	A	D	A	←				
		A	R	M	A	$\underline{D}$	A				
			A	R	M	A	D	A	←		
				A	$\underline{R}$	M	A	D	$\underline{A}$		
					A	R	$\underline{M}$	A	D	A	
						A	R	M	$\underline{A}$	D	A

In this problem your program should compute, given a ciphertext and a crib, the number of possible positions for the crib in the encrypted message.

Input

The first line of the input contains the encrypted message, which is a sequence of at least one letter and a maximum of 10^4 letters. The second line of the input contains the crib, which is a sequence of at least one letter and at most the same number of letters of the message. Only the 26 uppercase English letters appear on message and in the crib.

Output

Print a line containing an integer, indicating the number of possible starting positions for the crib in the encrypted message.

input

```
FDMLCRDMRALF
ARMADA
```

output

```
2
```

input

```
AAAAABABABABABABABABA
ABA
```

output

```
7
```

C. Switches

0.5 s, 1024 MB

In the control panel of an enormous amphitheatre, there are N switches, numbered from 1 to N, that control the M light bulbs of the place, which are numbered from 1 to M. Note that the number of switches and light bulbs is not necessarily the same and this happens because each switch is associated not to a single light bulb, but to a set of light bulbs. When a switch is activated, each one of the bulbs associated to it is toggled. If the bulb is lit, then it will be switched off, otherwise it will be switched on.

Some bulbs are lit initially and the janitor in charge of the amphitheatre has to switch off all them. He started trying to press the switches randomly, but as soon as he figured out that it wouldn't necessarily work, he decided to follow a fixed strategy. He will trigger the switches $1, 2, 3, \ldots, N, 1, 2, 3, \ldots$ in other words, every time after triggering the switch number N, he resumes the procedure from the switch number 1. He intends to keep pressing switches by the given strategy until all bulbs are switched off at the same time (in that moment he stops pressing the switches). Will his strategy work?

Given the bulbs which are initially lit and the sets of lamps associated to each switch, your program should compute the number of times the janitor will press switches. If by following the given strategy the janitor is never able to switch off all the lamps at the same time, your program should print `-1`.

Input

The first line contains two integers N and M ($1 \leq N, M \leq 1000$) representing, respectively, the number of switches and the number of light bulbs. The second line contains an integer L ($1 \leq L \leq M$) followed by distinct integers X_i ($1 \leq X_i \leq M$), representing the bulbs that are lit in the first place. Each of the following N rows contains an integer K_i ($1 \leq K_i \leq M$) followed by K_i distinct integers Y_i ($1 \leq Y_i \leq M$), representing the bulbs attached to switch i ($1 \leq i \leq N$).

Output

Your program must print a single line containing an integer representing the number of times the janitor will press the switches by following the strategy described, until all the lights were off at the same time. If that never happens, print `-1`.

input

```
6 3
2 1 3
3 1 2 3
2 1 3
2 1 2
2 2 3
1 2
3 1 2 3
```

output

```
5
```

input

```
3 3
2 2 3
1 3
2 1 2
1 2
```

output

```
-1
```

D. Deja vu of … Go Players

1.0 s, 256 MB

An ordinary sequence rather than magic codes? Rikka's opinion is much more important than the "truth" itself. She soon felt sleepy and fell asleep to take a trip to dreamlands in passing.

Rikka found two elders playing Go game when her sanity got back. The ethereal clouds, hardy vigorous pines, and rugged rocks shocked her. A fiction *isekai*! Excited Rikka's eyes ran around and around and focused on the Go chessboard in the end.

She found that the two players, wearing white and red respectively, were not playing Go game — they divided the black and white chess pieces into some piles, and were taking turns removing them. They kept silent for her questions, so Rikka had to stand there and keep her eyes on the chessboard.

They seemed to be playing an unexpectedly simple game. The red player has n black piles and its opponent has m white piles at the beginning. They take turns removing any positive number of chess pieces from arbitrary one of their assigned piles. Red goes first, and the player who first removes all chess pieces assigned to oneself "wins", and the other player has to drink.

Drinking is illegal for minors in Japan, so Rikka wonders if she can ensure to win when she is the red player.

Input

The first line contains an integer T $(1 \leq T \leq 100)$, the number of test cases. Then T test cases follow.

The input format of each test case is as follows:

The first line contains two integers n, m $(1 \leq n, m \leq 100)$, the numbers of piles of the red and the white player, respectively.

The following line contains n integers a_i $(1 \leq a_i \leq 10^9)$, in which each integer indicates the number of pieces in a black pile of the red player.

The following line contains m integers b_i $(1 \leq b_i \leq 10^9)$, in which each integer indicates the number of pieces in a white pile of the white player.

Output

Output a string in the only line, "Yes" if the red player who moves first can ensure to win, or "No" otherwise, without quotation marks.

input

```
2
3 2
1 1 1
2 2
1 1
4
3
```

output

```
No
Yes
```

E. Eventual … Journey

1.0 s, 256 MB

LCR is really an incredible being.

Thinking so, sitting on the plane and looking at the sea, Rikka savours the fantastic journey. A fire never happened. It must be interesting to share it with her mates, and another travel is also fascinating.

But before all, home is the best.

She travels by the JR lines. There are n stations in total, and m public bidirectional railway lines are built among them. Each station belongs to either JR West or JR East. Both JR West and JR East have their own private railways connecting all stations owned by themselves.

Rikka has some through tickets and two types of special passes: ICOCA for JR West and Suica for JR East. She pays a through ticket each time and does one of the following:

1. Travel from one terminal to another via a public railway line.
2. Travel to any station which has the same owner as the current one, using one of her special passes. A pass can be used for multiple times.

Rikka wonders, for each start station, the sum of the minimal numbers of tickets she needs to pay to reach every possible one of the other stations.

Input

The first line contains two integers $n, m\ (1 \le n \le 10^5, 0 \le m \le 10^5)$, the numbers of stations and public railways.

The next line contains n integers $A_i\ (A_i \in \{0, 1\}, i = 1, 2, \ldots, n)$, separated by spaces, describing the owner of each station. $A_i = 0$ if the station i belongs to JR west, and vice versa.

The following m lines describe all the public railways, each of which contains two integers $u, v\ (1 \le u, v \le n, u \ne v)$, representing a bidirectional railway connecting u and v. It is guaranteed that no two public railways connect the same pair of stations, and Rikka is able to travel between every pair of stations. Notice that the private railways are not given directly in the input, and Rikka may have to use her passes rather than traveling only through the public railways.

Output

Output n integers in one line, separated by spaces. The i-th integer is $\sum_{j=1}^{n} D(i, j)$, where $D(u, v)$ is the minimal number of tickets needed to travel from u to v.

input

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

output

```
2 2 2
```

input

```
5 3
1 0 1 0 1
1 2
2 3
4 5
```

output

```
5 5 5 6 5
```

F. Effective network

4.0 s, 256 MB

Quite recently, in a very close galaxy cluster, there was a bunch of planets. As the galaxy expands to a vast space, the only effective way of traveling between them is to use the network of teleporters. Due to technological problems, one can only travel between certain pairs of teleporters, using so-called links. This means that in order to travel from planet A to planet B, one may need to first teleport from A to some intermediate planet before reaching B. The distance between planets A and B is the minimum number of links required to travel between them. It is guaranteed that one can travel between each pair of planets using a finite number of links, but this number may be quite large for some pairs.

You are an employee of Teleport GmbH and were tasked to set up terms and conditions for the GA Travel Card such that the customers are happy. In order to do this, you need to select a non-empty subset of planets, called promoted planets, and designate all the links between them (and only those) to be included in the GA. We will call these free links. Thanks to customer survey you know that there are three conditions you need to fulfill:

- There should be at least two promoted planets.
- It should be possible to travel between any two promoted planets A and B using only free links
- The distance between any two promoted planets A and B, using only free links, cannot be larger than R - K, where R is the number of promoted planets, and K is a fixed constant representing the demands of the customers.

Determine whether it is possible to find a set of promoted planets, and if so, return one set that satisfies the conditions. If there are multiple solutions, return any of them.

Input

First line contains integers $1 \le N \le 5000$, $N-1 \le M \le \min(\frac{N(N-1)}{2}, 30000)$, $1 \le K \le N$ – the number of planets, the number of links, and the height of the demands, respectively.

Next M lines contain description of links. Each link is represented by two integers, $1 \le u_i, v_i \le N$, meaning that there is a link between planets u_i and v_i. It is guaranteed that no pair occurs more than once, and also that $u_i \ne v_i$.

It is guaranteed that one can travel between each pair of planets using a finite number of links.

Output

If there is a solution, output two lines: the first containing a single integer R – the number of promoted planets. On the second line, print R space separated integers, specifying the indices of promoted planets.

If there is no set of planets that would satisfy the conditions, output a single line containing the integer 0.

input

```
7 9 3
1 3
1 4
1 5
2 5
3 4
6 3
6 1
4 6
1 7
```

output

```
4
1 3 4 6
```

input

```
5 4 3
1 3
1 2
3 4
3 5
```

output

```
0
```

input

```
5 6 1
1 3
1 2
3 4
3 5
2 3
5 4
```

output

```
3
1 3 5
```

In the first sample case, we select a set of four planets that are all pairwise connected. As $R = 4$ and $K = 3$, the conditions are satisfied.

In the second test case, there is no subset of planets satisfying the conditions.

In the third case note that the pair of planets 1 and 5 would not be a solution, as one cannot travel between them without interchange at a non-promoted planet.

G. Collection

2.0 s, 64 MB

Alan recently started collecting the cards of the well known "Famous Computer Scientists" card game by the ACM. As buying cards is starting to get expensive, he would like to start trading. To do this, he needs to know the number of duplicate cards in his collection. Write a program to help him calculate this number.

Input

The first line contains a single integer n, the number of cards in Alan's collection. The second line contains n integers c_i, where c_i is the id of the i-th card. It holds that $1 \le n \le 200'000$ and $0 \le c_i \le 10^9$.

Output

Print a single integer denoting the number of duplicates, that is the number of cards he can safely trade whilst still having the same number of unique cards.

input

```
4
127 0 0 1
```

output

```
1
```

input

```
6
1 1 1000000 1 1 1
```

output

```
4
```

H. Horsemeet

1.0 s, 256 MB

Traditional games such as chess or checkers with slight modifications, are also played in Binary Casino. However, not many people play them, as these games are often referred as boring. The visitors are more attracted to more dynamic games which cause adrenaline rushes. To attract players to traditional games, your boss wants to introduce a chess-based game called Horsemeet. The rules of the game are:

The game is played by two players on a 8×8 chessboard. One player plays a white knight and the other player plays a black knight. The players alternate in moves, the white knight moves first. In each move a knight is moved from its current position to a random valid position. Valid position within the chessboard is a position, which is two tiles away in one coordinate and one tile away in other coordinate from the original position. All moves to a valid position are equally probable. The first knight to move to a tile already occupied by the other knight wins.

In order to check whether this game could be at least partially interesting to visitors you have to determine the probability of victory for knights at given start positions. If the probabilities of victory for both knights differs by less than 10^{-6}, the outcome of such configuration is a draw.

Input

The first line of input contains two integers A and B ($1 \leq A, B \leq 8$), the start position of the white knight. The second line of each input consists of two integers C and D ($1 \leq C, D \leq 8$), the start position of the black knight. You can assume both positions are distinct.

Output

Output the knight with a higher probability of victory: `"white"` or `"black"`. In case of equal probabilities output `"draw"`.

input

```
1 1
4 7
```

output

```
white
```

input	output
1 1 8 8	black