A. Soldier and Bananas

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

A soldier wants to buy w bananas in the shop. He has to pay k dollars for the first banana, $2k$ dollars for the second one and so on (in other words, he has to pay $i·k$ dollars for the i-th banana).

He has n dollars. How many dollars does he have to borrow from his friend soldier to buy w bananas?

Input

The first line contains three positive integers k, n, w ($1 \le k, w \le 1000$, $0 \le n \le 10^9$), the cost of the first banana, initial number of dollars the soldier has and number of bananas he wants.

Output

Output one integer — the amount of dollars that the soldier must borrow from his friend. If he doesn't have to borrow money, output 0.

Examples

input

```
3 17 4
```

output

```
13
```

B. Soldier and Badges

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Colonel has n badges. He wants to give one badge to every of his n soldiers. Each badge has a *coolness factor*, which shows how much it's owner reached. Coolness factor can be increased by one for the cost of one coin.

For every pair of soldiers one of them should get a badge with strictly higher factor than the second one. Exact values of their factors aren't important, they just need to have distinct factors.

Colonel knows, which soldier is supposed to get which badge initially, but there is a problem. Some of badges may have the same factor of coolness. Help him and calculate how much money has to be paid for making all badges have different factors of coolness.

Input

First line of input consists of one integer n ($1 \le n \le 3000$).

Next line consists of n integers a_i ($1 \le a_i \le n$), which stand for coolness factor of each badge.

Output

Output single integer — minimum amount of coins the colonel has to pay.

Examples

input

```
4
1 3 1 4
```

output

```
1
```

input

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

output

```
2
```

Note

In first sample test we can increase factor of first badge by 1.

In second sample test we can increase factors of the second and the third badge by 1.

C. Soldier and Cards

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

Two bored soldiers are playing card war. Their card deck consists of exactly n cards, numbered from 1 to n, **all values are different**. They divide cards between them in some manner, it's possible that they have different number of cards. Then they play a "war"-like card game.

The rules are following. On each turn a *fight* happens. Each of them picks card from the top of his stack and puts on the table. The one whose card value is bigger wins this *fight* and takes both cards from the table to the bottom of his stack. More precisely, he first takes his opponent's card and puts to the bottom of his stack, and then he puts his card to the bottom of his stack. If after some turn one of the player's stack becomes empty, he loses and the other one wins.

You have to calculate how many *fights* will happen and who will win the game, or state that game won't end.

Input

First line contains a single integer n ($2 \le n \le 10$), the number of cards.

Second line contains integer k_1 ($1 \le k_1 \le n - 1$), the number of the first soldier's cards. Then follow k_1 integers that are the values on the first soldier's cards, from top to bottom of his stack.

Third line contains integer k_2 ($k_1 + k_2 = n$), the number of the second soldier's cards. Then follow k_2 integers that are the values on the second soldier's cards, from top to bottom of his stack.

All card values are different.

Output

If somebody wins in this game, print 2 integers where the first one stands for the number of *fights* before end of game and the second one is 1 or 2 showing which player has won.

If the game won't end and will continue forever output -1.

Examples

input

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

output

```
6 2
```

input

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

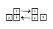
output

```
-1
```

Note

First sample:

Second sample:



D. Soldier and Number Game

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Two soldiers are playing a game. At the beginning first of them chooses a positive integer n and gives it to the second soldier. Then the second one tries to make maximum possible number of rounds. Each round consists of choosing a positive integer $x > 1$, such that n is divisible by x and replacing n with n / x. When n becomes equal to 1 and there is no more possible valid moves the game is over and the score of the second soldier is equal to the number of rounds he performed.

To make the game more interesting, first soldier chooses n of form $a! / b!$ for some positive integer a and b ($a \geq b$). Here by $k!$ we denote the *factorial* of k that is defined as a product of all positive integers not large than k.

What is the maximum possible score of the second soldier?

Input

First line of input consists of single integer t ($1 \leq t \leq 1\,000\,000$) denoting number of games soldiers play.

Then follow t lines, each contains pair of integers a and b ($1 \leq b \leq a \leq 5\,000\,000$) defining the value of n for a game.

Output

For each game output a maximum score that the second soldier can get.

Examples

input

```
2
3 1
6 3
```

output

```
2
5
```

E. Soldier and Traveling

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

In the country there are n cities and m bidirectional roads between them. Each city has an army. Army of the i-th city consists of a_i soldiers. Now soldiers roam. After roaming each soldier has to either stay in his city or to go to the one of neighboring cities by at **moving along at most one road**.

Check if is it possible that after roaming there will be exactly b_i soldiers in the i-th city.

Input

First line of input consists of two integers n and m ($1 \le n \le 100$, $0 \le m \le 200$).

Next line contains n integers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le 100$).

Next line contains n integers $b_1, b_2, \dots, b_n$ ($0 \le b_i \le 100$).

Then m lines follow, each of them consists of two integers p and q ($1 \le p, q \le n$, $p \ne q$) denoting that there is an undirected road between cities p and q.

It is guaranteed that there is at most one road between each pair of cities.

Output

If the conditions can not be met output single word "NO".

Otherwise output word "YES" and then n lines, each of them consisting of n integers. Number in the i-th line in the j-th column should denote how many soldiers should road from city i to city j (if $i \ne j$) or how many soldiers should stay in city i (if $i = j$).

If there are several possible answers you may output any of them.

Examples

input

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

output

```
YES
1 0 0 0
2 0 0 0
0 5 1 0
0 0 2 1
```

input

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

output

```
NO
```