# A. Piles With Stones

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

There is a beautiful garden of stones in Innopolis.

Its most beautiful place is the $n$ piles with stones numbered from $1$ to $n$.

EJOI participants have visited this place twice.

When they first visited it, the number of stones in piles was $x_1, x_2, \ldots, x_n$, correspondingly. One of the participants wrote down this sequence in a notebook.

They visited it again the following day, and the number of stones in piles was equal to $y_1, y_2, \ldots, y_n$. One of the participants also wrote it down in a notebook.

It is well known that every member of the EJOI jury during the night either sits in the room $108$ or comes to the place with stones. Each jury member who comes there either takes one stone for himself or moves one stone from one pile to another. We can assume that there is an unlimited number of jury members. No one except the jury goes to the place with stones at night.

Participants want to know whether their notes can be correct or they are sure to have made a mistake.

## Input

The first line of the input file contains a single integer $n$, the number of piles with stones in the garden ($1 \leq n \leq 50$).

The second line contains $n$ integers separated by spaces $x_1, x_2, \ldots, x_n$, the number of stones in piles recorded in the notebook when the participants came to the place with stones for the first time ($0 \leq x_i \leq 1000$).

The third line contains $n$ integers separated by spaces $y_1, y_2, \ldots, y_n$, the number of stones in piles recorded in the notebook when the participants came to the place with stones for the second time ($0 \leq y_i \leq 1000$).

## Output

If the records can be consistent output "Yes", otherwise output "No" (quotes for clarity).

### Examples

**input**
```
5
1 2 3 4 5
2 1 4 3 5
```
**output**
```
Yes
```

**input**
```
5
1 1 1 1 1
1 0 1 0 1
```
**output**
```
Yes
```

**input**
```
3
2 3 9
1 7 9
```
**output**
```
No
```

## Note

In the first example, the following could have happened during the night: one of the jury members moved one stone from the second pile to the first pile, and the other jury member moved one stone from the fourth pile to the third pile.

In the second example, the jury took stones from the second and fourth piles.

It can be proved that it is impossible for the jury members to move and took stones to convert the first array into the second array.

# B. And

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

There is an array with $n$ elements $a_1, a_2, ..., a_n$ and the number $x$.

In one operation you can select some $i$ ($1 \le i \le n$) and replace element $a_i$ with $a_i \& x$, where $\&$ denotes the bitwise and operation.

You want the array to have at least two equal elements after applying some operations (possibly, none). In other words, there should be at least two distinct indices $i \ne j$ such that $a_i = a_j$. Determine whether it is possible to achieve and, if possible, the minimal number of operations to apply.

### Input

The first line contains integers $n$ and $x$ ($2 \le n \le 100\,000$, $1 \le x \le 100\,000$), number of elements in the array and the number to and with.

The second line contains $n$ integers $a_i$ ($1 \le a_i \le 100\,000$), the elements of the array.

### Output

Print a single integer denoting the minimal number of operations to do, or $-1$, if it is impossible.

#### Examples

**input**

```
4 3
1 2 3 7
```

**output**

```
1
```

**input**

```
2 228
1 1
```

**output**

```
0
```

**input**

```
3 7
1 2 3
```

**output**

```
-1
```

### Note

In the first example one can apply the operation to the last element of the array. That replaces 7 with 3, so we achieve the goal in one move.

In the second example the array already has two equal elements.

In the third example applying the operation won't change the array at all, so it is impossible to make some pair of elements equal.

# C. Photo of The Sky

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

Pavel made a photo of his favourite stars in the sky. His camera takes a photo of all points of the sky that belong to some rectangle with sides parallel to the coordinate axes.

Strictly speaking, it makes a photo of all points with coordinates $(x, y)$, such that $x_1 \leq x \leq x_2$ and $y_1 \leq y \leq y_2$, where $(x_1, y_1)$ and $(x_2, y_2)$ are coordinates of the left bottom and the right top corners of the rectangle being photographed. The area of this rectangle can be zero.

After taking the photo, Pavel wrote down coordinates of $n$ of his favourite stars which appeared in the photo. These points are not necessarily distinct, there can be multiple stars in the same point of the sky.

Pavel has lost his camera recently and wants to buy a similar one. Specifically, he wants to know the dimensions of the photo he took earlier. Unfortunately, the photo is also lost. His notes are also of not much help; numbers are written in random order all over his notepad, so it's impossible to tell which numbers specify coordinates of which points.

Pavel asked you to help him to determine what are the possible dimensions of the photo according to his notes. As there are multiple possible answers, find the dimensions with the minimal possible area of the rectangle.

**Input**
The first line of the input contains an only integer $n$ ($1 \leq n \leq 100\,000$), the number of points in Pavel's records.

The second line contains $2 \cdot n$ integers $a_1$, $a_2$, ..., $a_{2 \cdot n}$ ($1 \leq a_i \leq 10^9$), coordinates, written by Pavel in some order.

**Output**
Print the only integer, the minimal area of the rectangle which could have contained all points from Pavel's records.

**Examples**

| input |
| --- |
| 4<br>4 1 3 2 3 2 1 3 |
| **output** |
| 1 |

| input |
| --- |
| 3<br>5 8 5 5 7 5 |
| **output** |
| 0 |

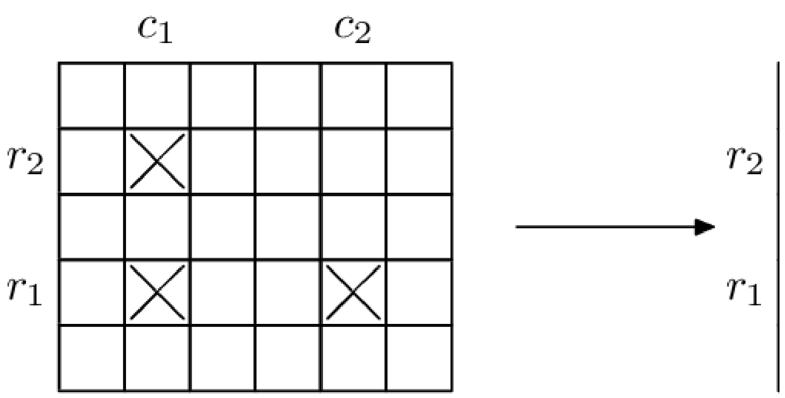
**Note**
In the first sample stars in Pavel's records can be $(1,3)$, $(1,3)$, $(2,3)$, $(2,4)$. In this case, the minimal area of the rectangle, which contains all these points is $1$ (rectangle with corners at $(1,3)$ and $(2,4)$).

## D. Chemical table

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Innopolis University scientists continue to investigate the periodic table. There are $n \cdot m$ known elements and they form a periodic table: a rectangle with $n$ rows and $m$ columns. Each element can be described by its coordinates $(r, c)$ ($1 \leq r \leq n$, $1 \leq c \leq m$) in the table.

Recently scientists discovered that for every four different elements in this table that form a rectangle with sides parallel to the sides of the table, if they have samples of three of the four elements, they can produce a sample of the fourth element using nuclear fusion. So if we have elements in positions $(r_1, c_1)$, $(r_1, c_2)$, $(r_2, c_1)$, where $r_1 \neq r_2$ and $c_1 \neq c_2$, then we can produce element $(r_2, c_2)$.




Samples used in fusion are not wasted and can be used again in future fusions. Newly crafted elements also can be used in future fusions.

Innopolis University scientists already have samples of $q$ elements. They want to obtain samples of all $n \cdot m$ elements. To achieve that, they will purchase some samples from other laboratories and then produce all remaining elements using an arbitrary number of nuclear fusions in some order. Help them to find the minimal number of elements they need to purchase.

**Input**
The first line contains three integers $n$, $m$, $q$ ($1 \leq n, m \leq 200\,000$; $0 \leq q \leq min(n \cdot m, 200\,000)$), the chemical table dimensions and the number of elements scientists already have.

The following $q$ lines contain two integers $r_i$, $c_i$ ($1 \leq r_i \leq n$, $1 \leq c_i \leq m$), each describes an element that scientists already have. All elements in the input are different.

**Output**
Print the minimal number of elements to be purchased.

**Examples**

**input**

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

**output**

```
0
```

**input**

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

**output**

```
2
```

**input**

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

**output**

```
1
```

## Note

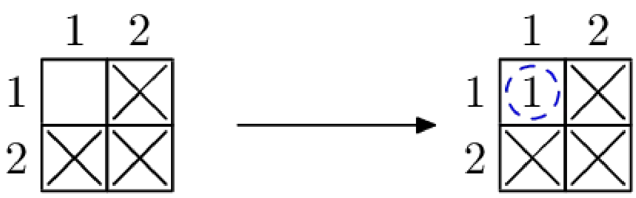
For each example you have a picture which illustrates it.

The first picture for each example describes the initial set of element samples available. Black crosses represent elements available in the lab initially.

The second picture describes how remaining samples can be obtained. Red dashed circles denote elements that should be purchased from other labs (the optimal solution should minimize the number of red circles). Blue dashed circles are elements that can be produced with nuclear fusion. They are numbered in order in which they can be produced.

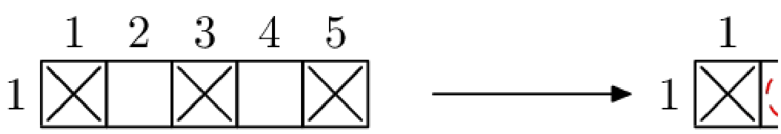
**Test 1**

We can use nuclear fusion and get the element from three other samples, so we don't need to purchase anything.



**Test 2**

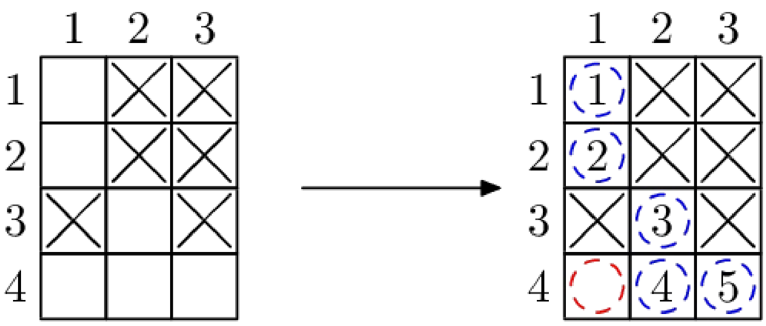
We cannot use any nuclear fusion at all as there is only one row, so we have to purchase all missing elements.



**Test 3**

There are several possible solutions. One of them is illustrated below.

Note that after purchasing one element marked as red it's still not possible to immidiately produce the middle element in the bottom row (marked as 4). So we produce the element in the left-top corner first (marked as 1), and then use it in future fusions.



## E. Hills

time limit per test: 1 second
memory limit per test: 512 megabytes
input: standard input
output: standard output

Welcome to Innopolis city. Throughout the whole year, Innopolis citizens suffer from everlasting city construction.

From the window in your room, you see the sequence of $n$ hills, where $i$-th of them has height $a_i$. The Innopolis administration wants to build some houses on the hills. However, for the sake of city appearance, a house can be only built on the hill, which is strictly higher than neighbouring hills (if they are present). For example, if the sequence of heights is $5, 4, 6, 2$, then houses could be built on hills with heights $5$ and $6$ only.

The Innopolis administration has an excavator, that can decrease the height of an arbitrary hill by one in one hour. The excavator can only work on one hill at a time. It is allowed to decrease hills up to zero height, or even to negative values. Increasing height of any hill is impossible. The city administration wants to build $k$ houses, so there must be at least $k$ hills that satisfy the condition above. What is the minimum time required to adjust the hills to achieve the administration's plan?

However, the exact value of $k$ is not yet determined, so could you please calculate answers for all $k$ in range $1 \leq k \leq \lceil \frac{n}{2} \rceil$? Here $\lceil \frac{n}{2} \rceil$ denotes $n$ divided by two, rounded up.

### Input

The first line of input contains the only integer $n$ ($1 \leq n \leq 5000$)—the number of the hills in the sequence.

Second line contains $n$ integers $a_i$ ($1 \leq a_i \leq 100\,000$)—the heights of the hills in the sequence.

### Output

Print exactly $\lceil \frac{n}{2} \rceil$ numbers separated by spaces. The $i$-th printed number should be equal to the minimum number of hours required to level hills so it becomes possible to build $i$ houses.

#### Examples

**input**

```
5
1 1 1 1 1
```

**output**

```
1 2 2
```

**input**

```
3
1 2 3
```

**output**

```
0 2
```

**input**

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

**output**

```
0 1 3
```

### Note

In the first example, to get at least one hill suitable for construction, one can decrease the second hill by one in one hour, then the sequence of heights becomes $1, 0, 1, 1, 1$ and the first hill becomes suitable for construction.

In the first example, to get at least two or at least three suitable hills, one can decrease the second and the fourth hills, then the sequence of heights becomes $1, 0, 1, 0, 1$, and hills $1, 3, 5$ become suitable for construction.

## F. AB-Strings

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

There are two strings $s$ and $t$, consisting only of letters a and b. You can make the following operation several times: choose a prefix of $s$, a prefix of $t$ and swap them. Prefixes can be empty, also a prefix can coincide with a whole string.

Your task is to find a sequence of operations after which one of the strings consists only of a letters and the other consists only of b letters. The number of operations should be minimized.

### Input

The first line contains a string $s$ ($1 \leq |s| \leq 2 \cdot 10^5$).

The second line contains a string $t$ ($1 \leq |t| \leq 2 \cdot 10^5$).

Here $|s|$ and $|t|$ denote the lengths of $s$ and $t$, respectively. It is guaranteed that at least one of the strings contains at least one a letter and at least one of the strings contains at least one b letter.

### Output

The first line should contain a single integer $n$ ($0 \le n \le 5 \cdot 10^5$) — the number of operations.

Each of the next $n$ lines should contain two space-separated integers $a_i$, $b_i$ — the lengths of prefixes of $s$ and $t$ to swap, respectively.

If there are multiple possible solutions, you can print any of them. It's guaranteed that a solution with given constraints exists.

#### Examples

**input**

```
bab
bb
```

**output**

```
2
1 0
1 3
```

**input**

```
bbbb
aaa
```

**output**

```
0
```

### Note

In the first example, you can solve the problem in two operations:

1. Swap the prefix of the first string with length $1$ and the prefix of the second string with length $0$. After this swap, you'll have strings ab and bbb.
2. Swap the prefix of the first string with length $1$ and the prefix of the second string with length $3$. After this swap, you'll have strings bbbb and a.

In the second example, the strings are already appropriate, so no operations are needed.