CODEFORCES β
Sponsored by Telegram

# Codeforces Round #376 (Div. 2)

## A. Night at the Museum

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

Grigoriy, like the hero of one famous comedy film, found a job as a night security guard at the museum. At first night he received *embosser* and was to take stock of the whole exposition.

*Embosser* is a special devise that allows to "print" the text of a plastic tape. Text is printed sequentially, character by character. The device consists of a wheel with a lowercase English letters written in a circle, static pointer to the current letter and a button that print the chosen letter. At one move it's allowed to rotate the alphabetic wheel one step clockwise or counterclockwise. Initially, static pointer points to letter 'a'. Other letters are located as shown on the picture:

After Grigoriy add new item to the base he has to print its name on the plastic tape and attach it to the corresponding exhibit. It's not required to return the wheel to its initial position with pointer on the letter 'a'.

Our hero is afraid that some exhibits may become alive and start to attack him, so he wants to print the names as fast as possible. Help him, for the given string find the minimum number of rotations of the wheel required to print it.

### Input

The only line of input contains the name of some exhibit — the non-empty string consisting of no more than $100$ characters. It's guaranteed that the string consists of only lowercase English letters.

### Output

Print one integer — the minimum number of rotations of the wheel, required to print the name given in the input.

### Examples

input
```
zeus
```
output
```
18
```

input
```
map
```
output
```
35
```

input
```
ares
```
output
```
34
```

### Note

To print the string from the first sample it would be optimal to perform the following sequence of rotations:

1. from 'a' to 'z' ($1$ rotation counterclockwise),
2. from 'z' to 'e' ($5$ clockwise rotations),
3. from 'e' to 'u' ($10$ rotations counterclockwise),
4. from 'u' to 's' ($2$ counterclockwise rotations).

In total, $1 + 5 + 10 + 2 = 18$ rotations are required.

# B. Coupons and Discounts

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

The programming competition season has already started and it's time to train for ICPC. Sereja coaches his teams for a number of year and he knows that to get ready for the training session it's not enough to prepare only problems and editorial. As the training sessions lasts for several hours, teams become hungry. Thus, Sereja orders a number of pizzas so they can eat right after the end of the competition.

Teams plan to train for $n$ times during $n$ consecutive days. During the training session Sereja orders exactly one pizza for each team that is present this day. He already knows that there will be $a_i$ teams on the $i$-th day.

There are two types of discounts in Sereja's favourite pizzeria. The first discount works if one buys two pizzas at one day, while the second is a coupon that allows to buy one pizza during two **consecutive** days (two pizzas in total).

As Sereja orders really a lot of pizza at this place, he is the golden client and can use the unlimited number of discounts and coupons of any type at any days.

Sereja wants to order exactly $a_i$ pizzas on the $i$-th day while using only discounts and coupons. Note, that he will never buy more pizzas than he need for this particular day. Help him determine, whether he can buy the proper amount of pizzas each day if he is allowed to use only coupons and discounts. Note, that it's also prohibited to have any active coupons after the end of the day $n$.

### Input

The first line of input contains a single integer $n$ ($1 \leq n \leq 200\,000$) — the number of training sessions.

The second line contains $n$ integers $a_1, a_2, ..., a_n$ ($0 \leq a_i \leq 10\,000$) — the number of teams that will be present on each of the days.

### Output

If there is a way to order pizzas using only coupons and discounts and do not buy any extra pizzas on any of the days, then print "`YES`" (without quotes) in the only line of output. Otherwise, print "`NO`" (without quotes).

### Examples

input

```
4
1 2 1 2
```

output

```
YES
```

input

```
3
1 0 1
```

output

```
NO
```

### Note

In the first sample, Sereja can use one coupon to buy one pizza on the first and the second days, one coupon to buy pizza on the second and the third days and one discount to buy pizzas on the fourth days. This is the only way to order pizzas for this sample.

In the second sample, Sereja can't use neither the coupon nor the discount without ordering an extra pizza. Note, that it's possible that there will be no teams attending the training sessions on some days.

# C. Socks

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

Arseniy is already grown-up and independent. His mother decided to leave him alone for $m$ days and left on a vacation. She have prepared a lot of food, left some money and washed all Arseniy's clothes.

Ten minutes before her leave she realized that it would be also useful to prepare instruction of which particular clothes to wear on each of the days she will be absent. Arseniy's family is a bit weird so all the clothes is enumerated. For example, each of Arseniy's $n$ socks is assigned a unique integer from $1$ to $n$. Thus, the only thing his mother had to do was to write down two integers $l_i$ and $r_i$ for each of the days — the indices of socks to wear on the day $i$ (obviously, $l_i$ stands for the left foot and $r_i$ for the right). Each sock is painted in one of $k$ colors.

When mother already left Arseniy noticed that according to instruction he would wear the socks of different colors on some days. Of course, that is a terrible mistake cause by a rush. Arseniy is a smart boy, and, by some magical coincidence, he posses $k$ jars with the paint — one for each of $k$ colors.

Arseniy wants to repaint some of the socks in such a way, that for each of $m$ days he can follow the mother's instructions and wear the socks of the same color. As he is going to be very busy these days he will have no time to change the colors of any socks so he has to finalize the colors now.

The new computer game Bota-3 was just realised and Arseniy can't wait to play it. What is the minimum number of socks that need their color to be changed in order to make it possible to follow mother's instructions and wear the socks of the same color during each of $m$ days.

### Input

The first line of input contains three integers $n$, $m$ and $k$ ($2 \le n \le 200\,000$, $0 \le m \le 200\,000$, $1 \le k \le 200\,000$) — the number of socks, the number of days and the number of available colors respectively.

The second line contain $n$ integers $c_1, c_2, ..., c_n$ ($1 \le c_i \le k$) — current colors of Arseniy's socks.

Each of the following $m$ lines contains two integers $l_i$ and $r_i$ ($1 \le l_i, r_i \le n$, $l_i \ne r_i$) — indices of socks which Arseniy should wear during the $i$-th day.

### Output

Print one integer — the minimum number of socks that should have their colors changed in order to be able to obey the instructions and not make people laugh from watching the socks of different colors.

### Examples

input

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

output

```
2
```

input

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

output

```
0
```

### Note

In the first sample, Arseniy can repaint the first and the third socks to the second color.

In the second sample, there is no need to change any colors.

# D. 80-th Level Archeology

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

Archeologists have found a secret pass in the dungeon of one of the pyramids of Cycleland. To enter the treasury they have to open an unusual lock on the door. The lock consists of $n$ words, each consisting of some hieroglyphs. The wall near the lock has a round switch. Each rotation of this switch changes the hieroglyphs according to some rules. The instruction nearby says that the door will open only if words written on the lock would be sorted in *lexicographical order* (the definition of lexicographical comparison in given in notes section).

The rule that changes hieroglyphs is the following. One clockwise rotation of the round switch replaces each hieroglyph with the next hieroglyph in alphabet, i.e. hieroglyph $x$ ($1 \le x \le c - 1$) is replaced with hieroglyph $(x + 1)$, and hieroglyph $c$ is replaced with hieroglyph $1$.

Help archeologist determine, how many clockwise rotations they should perform in order to open the door, or determine that this is impossible, i.e. no cyclic shift of the alphabet will make the sequence of words sorted lexicographically.

### Input

The first line of the input contains two integers $n$ and $c$ ($2 \le n \le 500\,000$, $1 \le c \le 10^6$) — the number of words, written on the lock, and the number of different hieroglyphs.

Each of the following $n$ lines contains the description of one word. The $i$-th of these lines starts with integer $l_i$ ($1 \le l_i \le 500\,000$), that denotes the length of the $i$-th word, followed by $l_i$ integers $w_{i,1}$, $w_{i,2}$, ..., $w_{i,l_i}$ ($1 \le w_{i,j} \le c$) — the indices of hieroglyphs that make up the $i$-th word. Hieroglyph with index $1$ is the smallest in the alphabet and with index $c$ — the biggest.

It's guaranteed, that the total length of all words doesn't exceed $10^6$.

### Output

If it is possible to open the door by rotating the round switch, print integer $x$ ($0 \le x \le c - 1$) that defines the required number of clockwise rotations. If there are several valid $x$, print any of them.

If it is impossible to open the door by this method, print $-1$.

### Examples

input

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

output

```
1
```

input

```
2 5
2 4 2
2 4 2
```

output

```
0
```

input

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

output

```
-1
```

### Note

Word $a_1, a_2, ..., a_m$ of length $m$ is *lexicographically not greater* than word $b_1, b_2, ..., b_k$ of length $k$, if one of two conditions hold:

- at first position $i$, such that $a_i \ne b_i$, the character $a_i$ goes earlier in the alphabet than character $b_i$, i.e. $a$ has smaller character in the first position where they differ;
- if there is no such position $i$ and $m \le k$, i.e. the first word is a prefix of the second or two words are equal.

The sequence of words is said to be sorted in lexicographical order if each word (except the last one) is lexicographically not greater than the next word.

In the first sample, after the round switch is rotated $1$ position clockwise the words look as follows:

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

In the second sample, words are already sorted in lexicographical order.

In the last sample, one can check that no shift of the alphabet will work.

# E. Funny Game

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

Once upon a time Petya and Gena gathered after another programming competition and decided to play some game. As they consider most modern games to be boring, they always try to invent their own games. They have only stickers and markers, but that won't stop them.

The game they came up with has the following rules. Initially, there are $n$ stickers on the wall arranged in a row. Each sticker has some number written on it. Now they alternate turn, Petya moves first.

One move happens as follows. Lets say there are $m \geq 2$ stickers on the wall. The player, who makes the current move, picks some integer $k$ from $2$ to $m$ and takes $k$ leftmost stickers (removes them from the wall). After that he makes the new sticker, puts it to the left end of the row, and writes on it the new integer, equal to the sum of all stickers he took on this move.

Game ends when there is only one sticker left on the wall. The score of the player is equal to the sum of integers written on all stickers he took during all his moves. The goal of each player is to maximize the difference between his score and the score of his opponent.

Given the integer $n$ and the initial sequence of stickers on the wall, define the result of the game, i.e. the difference between the Petya's and Gena's score if both players play optimally.

### Input

The first line of input contains a single integer $n$ ($2 \leq n \leq 200\,000$) — the number of stickers, initially located on the wall.

The second line contains $n$ integers $a_1, a_2, ..., a_n$ ($-10\,000 \leq a_i \leq 10\,000$) — the numbers on stickers in order from left to right.

### Output

Print one integer — the difference between the Petya's score and Gena's score at the end of the game if both players play optimally.

### Examples

input

```
3
2 4 8
```

output

```
14
```

input

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

output

```
-3
```

### Note

In the first sample, the optimal move for Petya is to take all the stickers. As a result, his score will be equal to $14$ and Gena's score will be equal to $0$.

In the second sample, the optimal sequence of moves is the following. On the first move Petya will take first three sticker and will put the new sticker with value $-8$. On the second move Gena will take the remaining two stickers. The Petya's score is $1 + (-7) + (-2) = -8$, Gena's score is $(-8) + 3 = -5$, i.e. the score difference will be $-3$.

# F. Video Cards

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

Little Vlad is fond of popular computer game Bota-2. Recently, the developers announced the new add-on named Bota-3. Of course, Vlad immediately bought only to find out his computer is too old for the new game and needs to be updated.

There are $n$ video cards in the shop, the power of the $i$-th video card is equal to integer value $a_i$. As Vlad wants to be sure the new game will work he wants to buy not one, but several video cards and unite their powers using the cutting-edge technology. To use this technology one of the cards is chosen as the leading one and other video cards are attached to it as secondary. For this new technology to work it's required that the power of each of the secondary video cards is divisible by the power of the leading video card. In order to achieve that the power of any secondary video card can be reduced to any integer value less or equal than the current power. However, the power of the leading video card should remain unchanged, i.e. it **can't** be reduced.

Vlad has an infinite amount of money so he can buy any set of video cards. Help him determine which video cards he should buy such that after picking the leading video card and may be reducing some powers of others to make them work together he will get the maximum total value of video power.

### Input

The first line of the input contains a single integer $n$ $(1 \le n \le 200\,000)$ — the number of video cards in the shop.

The second line contains $n$ integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 200\,000)$ — powers of video cards.

### Output

The only line of the output should contain one integer value — the maximum possible total power of video cards working together.

### Examples

input

```
4
3 2 15 9
```

output

```
27
```

input

```
4
8 2 2 7
```

output

```
18
```

### Note

In the first sample, it would be optimal to buy video cards with powers $3$, $15$ and $9$. The video card with power $3$ should be chosen as the leading one and all other video cards will be compatible with it. Thus, the total power would be $3 + 15 + 9 = 27$. If he buys all the video cards and pick the one with the power $2$ as the leading, the powers of all other video cards should be reduced by $1$, thus the total power would be $2 + 2 + 14 + 8 = 26$, that is less than $27$. Please note, that it's not allowed to reduce the power of the leading video card, i.e. one can't get the total power $3 + 1 + 15 + 9 = 28$.

In the second sample, the optimal answer is to buy all video cards and pick the one with the power $2$ as the leading. The video card with the power $7$ needs it power to be reduced down to $6$. The total power would be $8 + 2 + 2 + 6 = 18$.