

## 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 is 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  $x$  ( $1 \leq x \leq c - 1$ ) 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  $c$  ( $2 \leq n \leq 500\,000$ ,  $1 \leq c \leq 10^6$ ) — 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  $l_i$  ( $1 \leq l_i \leq 500\,000$ ), that denotes the length of the  $i$ -th word, followed by  $l_i$  integers  $w_{i,1}, w_{i,2}, \dots, w_{i,l_i}$  ( $1 \leq w_{i,j} \leq c$ ) — 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  $x$  ( $0 \leq x \leq c - 1$ ) 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
<pre>4 3 2 3 2 1 1 3 2 3 1 4 2 3 1 2</pre>
output
<pre>1</pre>

input
<pre>2 5 2 4 2 2 4 2</pre>

### UTRGV Practice Contest #7 - Based on Round #376

**Contest is running**

30:10:33

Contestant



### → Submit?

Language: GNU G++14 6.2.0

Choose file:

Choose File

No file chosen

Submit

<b>output</b>
0
<b>input</b>
4 4 1 2 1 3 1 4 1 2
<b>output</b>
-1

**Note**

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

- at first position  $i$ , such that  $a_i \neq b_i$ , 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  $m \leq k$ , 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.

---

[Codeforces](#) (c) Copyright 2010-2017 Mike Mirzayanov  
The only programming contests Web 2.0 platform  
Server time: Jul/28/2017 23:16:24<sup>UTC-5</sup> (c4).  
Desktop version, switch to [mobile version](#).  
[Privacy Policy](#)