

## C. Pride

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

You have an array  $a$  with length  $n$ , you can perform operations. Each operation is like this: choose two **adjacent** elements from  $a$ , say  $x$  and  $y$ , and replace one of them with  $\gcd(x, y)$ , where  $\gcd$  denotes the [greatest common divisor](#).

What is the minimum number of operations you need to make all of the elements equal to 1?

### Input

The first line of the input contains one integer  $n$  ( $1 \leq n \leq 2000$ ) — the number of elements in the array.

The second line contains  $n$  space separated integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq 10^9$ ) — the elements of the array.

### Output

Print  $-1$ , if it is impossible to turn all numbers to 1. Otherwise, print the minimum number of operations needed to make all numbers equal to 1.

### Examples

<b>input</b>
5 2 2 3 4 6
<b>output</b>
5
<b>input</b>
4 2 4 6 8
<b>output</b>
-1
<b>input</b>
3 2 6 9
<b>output</b>
4

### Note

In the first sample you can turn all numbers to 1 using the following 5 moves:

- [2, 2, 3, 4, 6].
- [2, 1, 3, 4, 6]
- [2, 1, 3, 1, 6]
- [2, 1, 1, 1, 6]
- [1, 1, 1, 1, 6]
- [1, 1, 1, 1, 1]

We can prove that in this case it is not possible to make all numbers one using less than 5 moves.