D. Array GCD

You are given array a_i of length n. You may consecutively apply two operations to this array:

- remove some subsegment (continuous subsequence) of length $m < n$ and pay for it $m \cdot a$ coins;
- change some elements of the array by at most 1, and pay b coins for each change.

Please note that each of operations may be applied at most once (and may be not applied at all) so you can remove only one segment and each number may be changed (increased or decreased) by at most 1. Also note, that you are not allowed to delete the whole array.

Your goal is to calculate the minimum number of coins that you need to spend in order to make the greatest common divisor of the elements of the resulting array be greater than 1.

Input

The first line of the input contains integers n, a and b ($1 \le n \le 1\,000\,000$, $0 \le a, b \le 10^9$) — the length of the array, the cost of removing a single element in the first operation and the cost of changing an element, respectively.

The second line contains n integers a_i ($2 \le a_i \le 10^9$) — elements of the array.

Output

Print a single number — the minimum cost of changes needed to obtain an array, such that the greatest common divisor of all its elements is greater than 1.

Examples

input
```
3 1 4
4 2 3
```
output
```
1
```

input
```
5 3 2
5 17 13 5 6
```
output
```
8
```

input
```
8 3 4
3 7 5 4 3 12 9 4
```
output
```
13
```

Note

In the first sample the optimal way is to remove number 3 and pay 1 coin for it.

In the second sample you need to remove a segment $[17, 13]$ and then decrease number 6. The cost of these changes is equal to $2 \cdot 3 + 2 = 8$ coins.