Weird Subarray

locked

Problem Submissions Discussions

Time Limit: $C/C{+}{+}\ (1s)\ ,\ Java\ (2s)$
Memory Limit: $256MB$

You're given an array of length N. We define the *compositeness* of an index i as the length of the **largest** subarray $[L_i, R_i]$ such that the below conditions meet:

1. $L_i \le i \le R_i$. In other words, the subarray contains the ith index.
2. The greatest common divisor of the elements of the subarray is greater than 1. That is,

$$\underset{j=L_i}{\overset{R_i}{\gcd}}(A_j) > 1$$

If no such subarray exists for index i, the compositeness of i is considered to be 0.

For each index i $(1 \le i \le N)$ of the given array, find its compositeness.

Note: A subarray is a contiguous part of an array. For example, $[2]$, $[2, 3]$ and $[5, 7]$ are subarrays of the array $[2, 3, 5, 7]$ but $[2, 5]$ is not.

Input Format

The first line contains one integer N — the length of the array.

Next line contains N integers $A_1, A_2, \ldots, A_N$ — the elements of the array

Constraints

$1 \le N \le 3 \times 10^5$
$1 \le A_i \le 10^{18}$

Output Format

Print N integers — the i^{th} of which denotes the compositeness of index i.

Sample Input 0

```
7
2 4 6 24 15 3 3
```

Sample Output 0

```
4 4 5 5 5 5 5
```

Explanation 0

For index 1 and 2, we can choose the subarray $[1, 4]$.

For the rest of the indices, we can choose the subarray $[3, 7]$.

Submissions: 195

Max Score: 100

Rate This Challenge:

☆☆☆☆☆

More

C

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}
```

Line: 1 Col: 1

Upload Code as File   □ Test against custom input   Run Code   Submit Code