

# Codeforces Round 511 Div.1 A

## Description

Given an array  $a[]$  with length of  $n$ , and  $a_i \leq 1.5 \times 10^7$ .

Try to erase minimum numbers of elements to enlarge the GCD of the rest of the array.

## Solution

We can think some method on prime-factors, but at first I didn't get the correct solution.

First we can divide all elements by their GCD.

Then if we count the number of elements that contain prime  $p$  as factor, the maximum is what we want, because we can erase  $n - \max$  elements to get all remaining elements with at least co-factor  $p \times GCD$ . And that's what we want.

Use *Sieve of Euler* to deal it.

```
#include <iostream>
#include <algorithm>
#include <cstring>
#include <cstdio>
#include <cctype>

using namespace std;

const int maxn = 3e5 + 5;
const int mx = 15e6 + 1;
const int maxp = 1e7;

int n, a[maxn], G;
int mindiv[mx], pr[maxp], ptr;
int cnt[mx];

int gcd(int a, int b) {
    return b ? gcd(b, a % b) : a;
}

void euler() {
    for (int i = 2; i < mx; ++i) {
        if (!mindiv[i]) mindiv[i] = pr[++ptr] = i;
        for (int j = 1; j <= ptr && i * pr[j] < mx; ++j) {
            mindiv[i * pr[j]] = pr[j];
            if (!(i % pr[j])) break;
        }
    }
}
```

```

int main() {
    euler();
    scanf("%d", &n);
    for (int i = 1; i <= n; ++i) {
        scanf("%d", &a[i]);
        if (i > 1) G = gcd(G, a[i]);
        else G = a[i];
    }
    for (int i = 1; i <= n; ++i) a[i] /= G;
    for (int i = 1; i <= n; ++i) {
        while (a[i] > 1) {
            cnt[mindiv[a[i]]]++;
            int k = mindiv[a[i]];
            while (mindiv[a[i]] == k) a[i] /= mindiv[a[i]];
        }
    }
    int siz = 0;
    for (int i = 2; i < mx; ++i) siz = max(siz, cnt[i]);
    printf("%d\n", siz ? n - siz : -1);
    return 0;
}

```