

## C. Array and Operations

time limit per test: 1 second

memory limit per test: 256 megabytes

input: standard input

output: standard output

You have written on a piece of paper an array of  $n$  positive integers  $a[1], a[2], \dots, a[n]$  and  $m$  *good* pairs of integers  $(i_1, j_1), (i_2, j_2), \dots, (i_m, j_m)$ . Each *good* pair  $(i_k, j_k)$  meets the following conditions:  $i_k + j_k$  is an odd number and  $1 \leq i_k < j_k \leq n$ .

In one operation you can perform a sequence of actions:

- take one of the *good* pairs  $(i_k, j_k)$  and some integer  $v$  ( $v > 1$ ), which divides both numbers  $a[i_k]$  and  $a[j_k]$ ;
- divide both numbers by  $v$ , i. e. perform the assignments:  $a[i_k] = a[i_k] / v$  and  $a[j_k] = a[j_k] / v$ .

Determine the maximum number of operations you can sequentially perform on the given array. Note that one pair may be used several times in the described operations.

### Input

The first line contains two space-separated integers  $n, m$  ( $2 \leq n \leq 100, 1 \leq m \leq 100$ ).

The second line contains  $n$  space-separated integers  $a[1], a[2], \dots, a[n]$  ( $1 \leq a[i] \leq 10^9$ ) — the description of the array.

The following  $m$  lines contain the description of *good* pairs. The  $k$ -th line contains two space-separated integers  $i_k, j_k$  ( $1 \leq i_k < j_k \leq n, i_k + j_k$  is an odd number).

It is guaranteed that all the *good* pairs are distinct.

### Output

Output the answer for the problem.

### Examples

input
3 2 8 3 8 1 2 2 3
output
0

  

input
3 2 8 12 8 1 2 2 3
output
2