6 In Prime Shape

You have designed new cryptography system that is based on using numbers that can be expressed as the products of a particular set of prime numbers. Given a set $P = \{p_1, \ldots, p_n\}$ of primes, a positive integer q is said to be *P-divisible* if q is evenly divisible by no prime number other than the numbers of P. For example, if $P = \{2, 5, 7\}$, the numbers 4 ($= 2^2$), 100 ($= 2^2 \cdot 5^2$), 70 ($= 2 \cdot 5 \cdot 7$), and 1400 ($= 2^4 \cdot 5^3 \cdot 7$) are all P-divisible, whereas 6 (divisible by 3), 26 (divisible by 13), and 53 (divisible by 53) are *not* P-divisible.

Given a set P of primes, your cryptography system relies on the fact that there are lots of P-divisible numbers. To check this, write a program, which given a set P of primes and a positive integer M, computes the number of P-divisible numbers that are less than or equal to M.

You may assume that $n \leq 10^3$ and $M \leq 10^9$.

Input and output files can be found at: `http://challengebox.cs.umd.edu/2019/Prime`

Input:

The first line contains the two numbers n and M, separated by a space. This is followed by n lines, each containing a prime number $p_1, p_2, \ldots, p_n$ of P. You may assume they are distinct.

Output:

A single integer indicating the number of P-divisible numbers not exceeding M.

Example:

Given $P = \{2, 5, 7\}$ and $M = 20$, there are 10 P-divisible numbers that do not exceed 20 (namely, 1, 2, 4, 5, 7, 8, 10, 14, 16, and 20).

Input:	**Output:**
3 20 2 5 7	10