F. Couple Cover

time limit per test: 3 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Couple Cover, a wildly popular luck-based game, is about to begin! Two players must work together to construct a rectangle. A bag with n balls, each with an integer written on it, is placed on the table. The first player reaches in and grabs a ball randomly (all balls have equal probability of being chosen) — the number written on this ball is the rectangle's width in meters. This ball is not returned to the bag, and the second player reaches into the bag and grabs another ball — the number written on this ball is the rectangle's height in meters. If the area of the rectangle is greater than or equal some threshold p square meters, the players win. Otherwise, they lose.

The organizers of the game are trying to select an appropriate value for p so that the probability of a couple winning is not too high and not too low, but they are slow at counting, so they have hired you to answer some questions for them. You are given a list of the numbers written on the balls, the organizers would like to know how many winning pairs of balls exist for different values of p. Note that two pairs are different if either the first or the second ball is different between the two in pair, and two different balls with the same number are considered different.

Input

The input begins with a single positive integer n in its own line ($1 \le n \le 10^6$).

The second line contains n positive integers — the i-th number in this line is equal to a_i ($1 \le a_i \le 3 \cdot 10^6$), the number written on the i-th ball.

The next line contains an integer m ($1 \le m \le 10^6$), the number of questions you are being asked.

Then, the following line contains m positive integers — the j-th number in this line is equal to the value of p ($1 \le p \le 3 \cdot 10^6$) in the j-th question you are being asked.

Output

For each question, print the number of winning pairs of balls that exist for the given value of p in the separate line.

Examples

input
```
5
4 2 6 1 3
4
1 3 5 8
```

output
```
20
18
14
10
```

input
```
2
5 6
2
30 31
```

output
```
2
0
```