B. Naughty Stone Piles

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There are n piles of stones of sizes $a_1, a_2, ..., a_n$ lying on the table in front of you.

During one move you can take one pile and add it to the other. As you add pile i to pile j, the size of pile j increases by the current size of pile i, and pile i stops existing. The cost of the adding operation equals the size of the added pile.

Your task is to determine the minimum cost at which you can gather all stones in one pile.

To add some challenge, the stone piles built up conspiracy and decided that each pile will let you add to it not more than k times (after that it can only be added to another pile).

Moreover, the piles decided to puzzle you completely and told you q variants (not necessarily distinct) of what k might equal.

Your task is to find the minimum cost for each of q variants.

Input

The first line contains integer n $(1 \le n \le 10^5)$ — the number of stone piles. The second line contains n space-separated integers: $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$ — the initial sizes of the stone piles.

The third line contains integer q $(1 \le q \le 10^5)$ — the number of queries. The last line contains q space-separated integers $k_1, k_2, ..., k_q$ $(1 \le k_i \le 10^5)$ — the values of number k for distinct queries. Note that numbers k_i can repeat.

Output

Print q whitespace-separated integers — the answers to the queries in the order, in which the queries are given in the input.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the `cin`, `cout` streams or the `%I64d` specifier.

Examples

input
5 2 3 4 1 1 2 2 3
output
9 8

Note

In the first sample one way to get the optimal answer goes like this: we add in turns the 4-th and the 5-th piles to the 2-nd one; then we add the 1-st pile to the 3-rd one; we add the 2-nd pile to the 3-rd one. The first two operations cost 1 each; the third one costs 2, the fourth one costs 5 (the size of the 2-nd pile after the first two operations is not 3, it already is 5).

In the second sample you can add the 2-nd pile to the 3-rd one (the operations costs 3); then the 1-st one to the 3-th one (the cost is 2); then the 5-th one to the 4-th one (the costs is 1); and at last, the 4-th one to the 3-rd one (the cost is 2).