Problem A. Discounts

Time limit 2500 ms
Mem limit 262144 kB

You came to a local shop and want to buy some chocolate bars. There are n bars in the shop, i-th of them costs a_i coins (and you want to buy all of them).

You have m different coupons that allow you to buy chocolate bars. i-th coupon allows you to buy q_i chocolate bars while you have to pay only for the $q_i - 1$ most expensive ones (so, the cheapest bar of those q_i bars is for free).

You can use only one coupon; if you use coupon i, you have to choose q_i bars and buy them using the coupon, and buy all the remaining $n - q_i$ bars without any discounts.

To decide which coupon to choose, you want to know what will be the minimum total amount of money you have to pay if you use one of the coupons optimally.

Input

The first line contains one integer n ($2 \leq n \leq 3 \cdot 10^5$) — the number of chocolate bars in the shop.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$), where a_i is the cost of i-th chocolate bar.

The third line contains one integer m ($1 \leq m \leq n - 1$) — the number of coupons you have.

The fourth line contains m integers $q_1, q_2, \ldots, q_m$ ($2 \leq q_i \leq n$), where q_i is the number of chocolate bars you have to buy using i-th coupon so that the least expensive of them will be for free. **All values of q_i are pairwise distinct**.

Output

Print m integers, i-th of them should be the minimum amount of money you have to pay if you buy q_i bars with i-th coupon, and all the remaining bars one by one for their full price.

Sample 1

Input	Output

7
7 1 3 1 4 10 8
2
3 4
	27
	30

Note

Consider the first example.

If we use the first coupon, we may choose chocolate bars having indices 1, 6 and 7, and we pay 18 coins for them and 9 coins for all other bars.

If we use the second coupon, we may choose chocolate bars having indices 1, 5, 6 and 7, and we pay 25 coins for them and 5 coins for all other bars.