



Problem B. A Classic Problem

TimeLimit: 0.5 seconds
MemoryLimit: 256 megabytes

There is a classic greedy problem:

Given an integer sequence A of length $2 \times N$, and you may perform an arbitrary number of swaps (each swap chooses two positions and exchanges their elements). The goal is to maximize the following sum:
$$A_1 \times A_2 + A_3 \times A_4 + \cdots + A_{2N-1} \times A_{2N}.$$

However, it is too easy to you, so you are given an additional restriction — you are allowed to perform at most N swaps between elements in the sequence.

Can you solve problem with this restriction?

Input

The first line contains a single integer N , representing a sequence of length $2 \times N$.

The second line contains $2 \times N$ integers A_1, A_2, \cdots, A_{2N} .

- $1 \leq N \leq 10^5$
- $-10^6 \leq A_i \leq 10^6$ for all $1 \leq i \leq 2N$

Output

Print an integer — the maximum sum with at most N swaps.

Examples

standard input	standard output
2 3 1 3 7	24
3 0 -10 2 1 1 1	3