Supercomputer Pipeline Optimization

Problem Description

A research lab runs a weekly data-crunching ritual on its beloved supercomputer. The pipeline applies a fixed chain of linear transformations $A_1, A_2, \ldots, A_{n-1}$ to a dataset. Transformation A_i maps a vector space of dimension p_{i-1} to dimension p_i, so A_i can be viewed as a matrix of size $p_{i-1} \times p_i$.

Matrix multiplication is associative, so the entire pipeline $(\cdots((A_1A_2)A_3)\cdots A_{n-1})$ can be parenthesized in many ways. However, different parenthesizations can require vastly different numbers of scalar multiplications. Your job is to choose the order of multiplications that minimizes the total number of scalar multiplications.

Formally, given the dimension array $p_0, p_1, \ldots, p_n$ (so there are $n-1$ matrices: A_1 is $p_0 \times p_1$, A_2 is $p_1 \times p_2$, …, A_{n-1} is $p_{n-2} \times p_{n-1}$), compute the minimum number of scalar multiplications needed to evaluate $A_1A_2\cdots A_{n-1}$.

Input

There is exactly *one* test case per input file.

The first line contains a single integer n ($2 \le n \le 100$). The second line contains n integers $p_0, p_1, \ldots, p_{n-1}$ ($1 \le p_i \le 1000$), where consecutive matrices are conformable: matrix A_i has dimensions $p_{i-1} \times p_i$ for $i = 1, \ldots, n-1$.

Note. The input describes $n-1$ matrices via n dimensions.

Output

Output a single integer: the minimum number of scalar multiplications required to compute the product $A_1A_2\cdots A_{n-1}$.

Samples

Sample Input
4 10 30 5 60

Sample Output
4500