

# RACE CONDITION

Vasya has written a program that launches  $n$  threads having  $m_i$  instructions each.

At any point in time the CPU is executing a single instruction from a single thread. The instructions in a thread are always executed in order (switching to other threads is possible).

After all instructions in a thread have been executed, the CPU ignores this thread.

Let us define an *execution path* as an ordered list of actually executed instructions from different threads.

Write a program to calculate the number of different execution paths (accounting for all possible switches between threads) for a multi-threaded program.

## Limitations

$1 \leq n \leq 10$ ;  $1 \leq m_i \leq 20$ ,  $1 \leq i \leq n$ ,  $\sum m_i \leq 20$ .

## Input

The first line of the input file defines the number of threads  $n$ .

The second line contains  $n$  space-delimited integers  $m_i$ , the number of instructions in the threads.

## Output

The number of different execution paths.

## Examples

N°	stdin	stdout
1	2 2 2	6
2	3 1 2 3	60