Problem A. Convenient Coins

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	1024 mebibytes

In Berland, the main currency is burle, and purchases are usually made using coins of 1 or 2 burles.

For a multiset of coins S, we denote its *convenience* as the maximum non-negative integer x such that for every integer $i \in [0, x]$, it is possible to choose some coins from S (possibly all of them, possibly none of them) so that their total value equals to x burles.

You are given the array $a_1, a_2, \ldots, a_n$, where each element is either 1 or 2. You have to calculate $\sum\limits_{l=1}^{n}\sum\limits_{r=l}^{n} F(l, r)$, where $F(l, r)$ is the convenience of the multiset consisting of coins with values $a_l, a_{l+1}, \ldots, a_r$.

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of test cases.

Each test case consists of two lines:

- the first line contains a single integer n ($2 \le n \le 3 \cdot 10^5$).
- the second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 2$).

Additional constraint on the input: the sum of n over all test cases doesn't exceed $3 \cdot 10^5$.

Output

For each test case, print one integer — the value of $\sum\limits_{l=1}^{n}\sum\limits_{r=l}^{n} F(l, r)$.

Example

standard input	standard output
4 3 1 2 1 2 2 2 4 2 1 2 1 7 1 2 2 1 1 2 1	12 0 26 113