

NEO

Given array with n integer elements. We divide it into several part (may be 1), each part is a consecutive of elements. The **NEO value** in that case is computed by: Sum of value of each **part**. Value of a **part** is sum all elements in this part multiple by its length.

Example: We have array: [2 3 -2 1]. If we divide it look likes: [2 3] [-2 1]. Then $NEO = (2 + 3) * 2 + (-2 + 1) * 2 = 10 - 2 = 8$.

Because there are many ways to divide an array into several part, so we can get many of NEO value. Your task is find the NEO with maximum value.

Input

First line: T (number of test case, $T \leq 10$)

For each of testcase:

+ First line: n ($1 \leq n \leq 10^5$)

+ Second line: $a[1], a[2], \dots, a[n]$ ($-10^6 \leq a[i] \leq 10^6$)

Output

Each testcase print the

Example

Input:

```
1
4
1 2 -4 1
```

Output:

```
3
```