Input: An array $A = [A_1, A_2, \ldots, A_n]$ of $n \geq 2$ positive integers.

Definitions: Denote the sum of all the integers in the array by $S$:

$$S = A_1 + \cdots + A_n$$

and for $1 \leq i \leq n$, denote the sum of all the integers in the array except $A_i$ by $S_i$:

$$S_i = S - A_i = A_1 + \cdots + A_{i-1} + A_{i+1} + \cdots + A_n$$

Example: Let $A = [16, 2, 128, 64, 1, 8, 32, 4]$. Then $S = 255$ and

$S_1 = 239; S_2 = 253; S_3 = 127; S_4 = 191; S_5 = 254; S_6 = 247; S_7 = 223; S_8 = 251$

Task: Design an algorithm that computes $S_1, S_2, \ldots, S_n$ only with plus operations (you are not allowed to use the minus operation).

Complexity: What is the number of plus operations used by your algorithm?

Optimization goal: Minimize the number of plus operations performed by the algorithm.

Efficiency: Full credit will be given to a correct algorithm that performs $\Theta(n)$ plus operations. Partial credit will be given to any other correct algorithm.

Bonus: Extra credit will be given for answers that state and prove the exact number of plus operations of an efficient correct algorithm.

Hint: Apply the Prefix-Sum technique to get the $\Theta(n)$ algorithm.

Remark: You must justify the correctness and complexity of your algorithm.

Grading: The grade is always 100. The percentage of the final grade varies: 5% for efficient correct algorithms and at most 3% for non-efficient correct algorithms. The bonus may add 2%.

Submission: Email a copy of your answer on one page to amotz@sci.brooklyn.cuny.edu.