

Subarray of length 2

Given an array $A[]$ of N elements.

In one operation, you can select any subarray of **length 2** and multiply all elements of that subarray by **-1**. You can do any number of operations (possibly zero).

Your task is to maximize the sum of all elements of all subarrays of $A[]$.

Example 1:

Input:

$N = 3$

$A[] = \{-1, -2, -3\}$

Output: 14

Explanation:

We can do 1 operation on subarray from index = 2 to index = 3. After performing the operation, modified array $A[] = \{-1, 2, 3\}$. All subarrays of the modified array $A[]$ are $[-1]$, $[2]$, $[3]$, $[-1, 2]$, $[2, 3]$ and $[-1, 2, 3]$. Thus total sum of all subarrays is 14. This is the maximum sum which we can get by performing the operations.

Example 2:

Input:

$N = 2$

$A[] = \{0, 1\}$

Output: 2

Explanation: There is no need to do any operation. All subarrays of the array $A[]$ are $[0]$, $[1]$ and $[0, 1]$ and sum of all subarrays = $0 + 1 + 1 = 2$.

Your Task:

You don't need to read input or print anything. Your task is to complete the function **maxSum()** which takes the array **A[]** and its size **N** as input parameters and returns the maximum sum of all subarrays after performing the above operations zero or more times optimally.

Constraints :

$$1 \leq N \leq 10^5$$

$$-10^3 \leq A[i] \leq 10^3$$