Maximum Sum of the Array

You are given an array of ***N*** integers.You tried to sum all the elements of the array, but you see the *summation* does not reach the maximum.

So you decide to divide the array into two arrays. You can choose any index $(1 \le i \le N)$ from array **Ar** and remove the value from the **Ar** array and add it to another array *A*. Also, you can choose any index $(1 \le j \le N)$ and remove the value from **Ar** and add it to another array *B*

Suppose an array *ar = [1, 2, 3, 0, 5]*, so you can choose index *1, 3, 5* and remove it from array *ar* and add it to array *A*. So *ar = [., 2, ., 0, .]* and *A=[1, 3, 5]*. You can choose index *2, 4* and remove it from array *ar* and add it to array *B*. So *ar = [., ., ., ., .]* and *B = [2, 0]*.

You can do any operation until array *ar* becomes empty.

Here is the main problem. You need to divide ar in such way that ***SUM(A) - SUM(B)*** gets *maximized*.
Here ***SUM(X)*** means summetion of the array *x*. Example *x = [1, 5, 8]* so *SUM(x) = 1 + 5 + 8 = 14*.
Now Print the *maximum* ***SUM(A) - SUM(B)*** you can get.
Both *A* and *B* array needs to contain atleast **1** element from array *ar*.

Input Format

- First Line will contain ***N*** $(2 \le N \le 10^6)$, *the number of elements present in the array.*
- Second line will contain array **ar** of ***N*** elements. $(-10^9 \le ar_i \le 10^9)$.

Output Format

Print the maximum value you can get of *SUM(A) - SUM(B)* after doing those operations.

Example

Input 01:
5
1 2 3 4 5
Output 01:
13

Input 02:
6
-1 1 2 -2 3 -3
Output 02:
12