Problem

Chef has an array A of length N. He defines the *alternating sum* of the array as:

- $S = |A_1| - |A_2| + |A_3| - |A_4| + \ldots (-1)^{N-1} \cdot |A_N|$

Chef is allowed to perform the following operation on the array **at most once**:

- Choose two indices i and j $(1 \leq i < j \leq N)$ and swap the elements A_i and A_j.

Find the **maximum** *alternating sum* Chef can achieve by performing the operation **at most once**.

Note: $|X|$ denotes the absolute value of X. For example, $|-4| = 4$ and $|7| = 7$.

Input Format

- The first line will contain T - the number of test cases. Then the test cases follow.
- First line of each test case contains a single integer N - size of the array A.
- Second line of each test case contains N space separated integers - denoting the elements of array A.

Output Format

For each testcase, output in a single line, the **maximum** *alternating sum* Chef can obtain by performing the operation **at most once**.

Constraints

- $1 \leq T \leq 10^5$
- $2 \leq N \leq 10^5$
- $-10^9 \leq A_i \leq 10^9$
- Sum of N over all test cases does not exceed $2 \cdot 10^5$.

Sample 1:

Input	Output
2 2 10 -10 7 -3 -2 -1 0 1 2 3	0 6

Explanation:

Test Case 1: One optimal way is to perform no operations. Thus the alternating sum is $|10| - |-10| = 10 - 10 = 0$.

Test Case 2: One optimal way is to choose $i = 2$ and $j = 5$. After swapping, the array is $[-3, 1, -1, 0, -2, 2, 3]$. The alternating sum in this case is $|-3| - |1| + |-1| - |0| + |-2| - |2| + |3| = 6$.