Problem 3701: Compute Alternating Sum

Problem Information

Difficulty: Easy
Acceptance Rate: 89.02%
Paid Only: No
Tags: Array, Simulation

Problem Description

You are given an integer array `nums`.

The **alternating sum** of `nums` is the value obtained by **adding** elements at even indices and **subtracting** elements at odd indices. That is, `nums[0] - nums[1] + nums[2] - nums[3]...`

Return an integer denoting the alternating sum of `nums`.

Example 1:

Input: nums = [1,3,5,7]

Output: -4

Explanation:

* Elements at even indices are `nums[0] = 1` and `nums[2] = 5` because 0 and 2 are even numbers. * Elements at odd indices are `nums[1] = 3` and `nums[3] = 7` because 1 and 3 are odd numbers. * The alternating sum is `nums[0] - nums[1] + nums[2] - nums[3] = 1 - 3 + 5 - 7 = -4`.

Example 2:

Input: nums = [100]

Output: 100

Explanation:

* The only element at even indices is `nums[0] = 100` because 0 is an even number. * There are no elements on odd indices. * The alternating sum is `nums[0] = 100`.

Constraints:

* `1 <= nums.length <= 100` * `1 <= nums[i] <= 100`

Code Snippets

C++:

```
class Solution {
public:
int alternatingSum(vector<int>& nums) {

}
};
```

Java:

```
class Solution {
public int alternatingSum(int[] nums) {

}
}
```

Python3:

```
class Solution:
def alternatingSum(self, nums: List[int]) -> int:
```