

# Problem 1217: Minimum Cost to Move Chips to The Same Position

## Problem Information

**Difficulty:** Easy

**Acceptance Rate:** 72.61%

**Paid Only:** No

**Tags:** Array, Math, Greedy

## Problem Description

We have `n` chips, where the position of the `ith` chip is `position[i]`.

We need to move all the chips to \*\*the same position\*\*. In one step, we can change the position of the `ith` chip from `position[i]` to:

\* `position[i] + 2` or `position[i] - 2` with `cost = 0`.  
\* `position[i] + 1` or `position[i] - 1` with `cost = 1`.

Return \_the minimum cost\_ needed to move all the chips to the same position.

**Example 1:**



**Input:** position = [1,2,3] **Output:** 1 **Explanation:** First step: Move the chip at position 3 to position 1 with cost = 0. Second step: Move the chip at position 2 to position 1 with cost = 1. Total cost is 1.

**Example 2:**



**Input:** position = [2,2,2,3,3] **Output:** 2 **Explanation:** We can move the two chips at position 3 to position 2. Each move has cost = 1. The total cost = 2.

**\*\*Example 3:\*\***

**\*\*Input:\*\*** position = [1,1000000000] **\*\*Output:\*\*** 1

**\*\*Constraints:\*\***

\* `1 <= position.length <= 100` \* `1 <= position[i] <= 10^9`

## Code Snippets

**C++:**

```
class Solution {  
public:  
    int minCostToMoveChips(vector<int>& position) {  
  
    }  
};
```

**Java:**

```
class Solution {  
public int minCostToMoveChips(int[] position) {  
  
}  
}
```

**Python3:**

```
class Solution:  
    def minCostToMoveChips(self, position: List[int]) -> int:
```