# 题目

给你一个由正整数组成的整数数组 nums ，返回其中可被 3 整除的所有偶数的平均值。

注意：n 个元素的平均值等于 n 个元素 求和 再除以 n ，结果 向下取整 到最接近的整数。

示例 1：

输入：nums = [1,3,6,10,12,15]

输出：9

解释：6 和 12 是可以被 3 整除的偶数。(6 + 12) / 2 = 9 。

示例 2：

输入：nums = [1,2,4,7,10]

输出：0

解释：不存在满足题目要求的整数，所以返回 0 。

提示：

1 <= nums.length <= 1000

1 <= nums[i] <= 1000

# 分析

## 方法一：遍历

思路：

题目给定正整数组成的整数数组 nums，返回其中可被 3整除的所有偶数的平均值。被3整除的所有偶数，等价于6的倍数。

遍历nums中的每一个数，判断是否除以6没有余数，没有余数即是6的倍数，累加到总和中。

最后，返回求和后的平均数，k个元素的平均值等于求和后除以k，结果向下取整。如果不存在6的倍数，返回0。

代码：

class Solution {

public:

int averageValue(vector<int>& nums) {

int total = 0, k = 0;

for (int a : nums) {

if (a % 6 == 0) {

total += a;

k++;

}

}

return k > 0 ? total / k : 0;

}

};

复杂度分析

时间复杂度：O(n)，其中n是数组nums的长度。

空间复杂度：O(1)。