

```

import Foundation
/// 剑指 Offer 56 - II. 数组中数字出现的次数 II
/// 在一个数组 nums 中除一个数字只出现一次之外，其他数字都出现了三次。
/// 请找出那个只出现一次的数字。
///
/// 示例 1:
/// 输入: nums = [3,4,3,3]
/// 输出: 4
///
/// 示例 2:
/// 输入: nums = [9,1,7,9,7,9,7]
/// 输出: 1
class Solution {
    /// 考虑数字的二进制形式，对于出现三次的数字，各二进制位出现的次数都是 3 的倍数。
    /// 因此，统计所有数字的各二进制位中 1 的出现次数，并对 3 求余，结果则为只出现一次的数字。
    /// 3 = 0 0 1 1
    /// 3 = 0 0 1 1
    /// 3 = 0 0 1 1
    /// 4 = 0 1 0 0
    /// 1的个数 = 0 1 3 3 --> 对 3 取余数 = 0 1 0 0 = 4
    func singleNumber(_ nums: [Int]) -> Int {
        /// 由于题目给定**Int**的范围为32位，因此我们需要初始化一个32位的数组
        /// 来存储统计所有数字的各二进制位中 1 的出现次数
        var numsOfOne: [Int] = Array(repeating: 0, count: 32)
        for var num in nums {
            for j in 0..<32 {
                numsOfOne[j] += (num & 1)
                num >>= 1
            }
        }
        var res: Int = 0
        let numRepeatTimes: Int = 3
        for j in 0..<32 {
            /// res左移动一位
            res <<= 1
            /// 注意到`numsOfOne`是从左到右计算的结果
            /// 因此使用按位或运算符**|**修改的是 res 的最后一位
            res |= (numsOfOne[31-j] % numRepeatTimes)
        }
        return res
    }
}

let solution = Solution()
print(solution.singleNumber([3,4,3,3]))
print(solution.singleNumber([9,1,7,9,7,9,7]))

```