```swift
import Foundation
/// 剑指 Offer 56 - I. 数组中数字出现的次数
/// 一个整型数组 nums
    里除两个数字之外，其他数字都出现了两次。请写程序找出这两个只出现一次的数字。
/// 要求时间复杂度是O(n)，空间复杂度是O(1)。
///
/// 示例 1:
/// 输入: nums = [4,1,4,6]
/// 输出: [1,6] 或 [6,1]
///
/// 示例 2:
/// 输入: nums = [1,2,10,4,1,4,3,3]
/// 输出: [2,10] 或 [10,2]
class Solution {
    /// 按位异或运算符，或称“排外的或运算符”（^），可以对两个数的比特位进行比较。
    /// 它返回一个新的数，当两个数的对应位不相同时，新数的对应位就为 1，并且对应位相同时则为 0
    func singleNumbers(_ nums: [Int]) -> [Int] {
        var xyXOR: Int = 0
        /// 计算所有数字的异或 XOR 最终得到非重复两个数字 X 与 Y 的 XOR 结果
        for num in nums {
            xyXOR ^= num
        }
        /// 寻找 X 与 Y 二进制位中不相同的位置
        var position: Int = 1
        while (xyXOR & position == 0) { position <<= 1 }

        /// 遍历寻找 X 与 Y
        var (x, y): (Int, Int) = (0, 0)
        for num in nums {
            /// 通过判断 num position 位上对应的数与 posotion 的与来划分数组
            /// 由于其他数字是成对出现的因此不论其与position的与是否为0都
            /// 一定保证能被消除掉
            if (num & position == 0) { x ^= num }
            else { y ^= num }
        }
        return [x, y]
    }
}

let solution = Solution()
print(solution.singleNumbers([4,1,4,6]))
print(solution.singleNumbers([1,2,10,4,1,4,3,3]))
```