

```

import Foundation
/// **剑指 Offer 16. 数值的整数次方**
/// 实现 pow(x, n) , 即计算 x 的 n 次幂函数 (即,  $x^n$ ) 。
/// 不得使用库函数, 同时不需要考虑大数问题。
class Solution {
    func myPow(_ x: Double, _ n: Int) -> Double {
        if x == 0 { return 0 }
        var res: Double = 1.0
        var _n: Int = (n < 0) ? abs(n) : n
        var _x = (n < 0) ? 1/x : x

        while (_n > 0) {
            if (_n & 1 == 1) { res *= _x }
            _n >>= 1
            _x *= _x
        }
        return res
    }
}

let solution = Solution()
print(solution.myPow(2, 10))
print(solution.myPow(2.1, 3))
print(solution.myPow(2.0, -2))
print(solution.myPow(0, -2))

```