

数值的整数次方

实现函数 `double Power(double base, int exponent)`，求 `base` 的 `exponent` 次方。不得使用库函数，同时不需要考虑大数问题。

示例 1:

输入: 2.00000, 10 输出: 1024.00000

示例 2:

输入: 2.10000, 3 输出: 9.26100

```
/*class Solution {
public:
    double myPow(double x, int n) {
        if(n==0) return 1;
        double res=x;
        if(n>0)
        {
            for(int i=0;i<n-1;i++)
            {
                res*=x;
            }
            return res;
        }
        if(n<0)
        {
            for(int i=0;i<abs(n)-1;i++)
            {
                res*=x;
            }
            return 1/res;
        }
        return 0;
    }
};*/

class Solution {
public:
    double myPow(double x, int n) {
        if(n == 0 || x == 1.0) return 1;
        if(x == 0.0 && n < 0) return x; // 当 x 是 0 而 n 是负数的时候，特殊情况
        double res = 1.0;
```

```
long exp = n; // 必须是 long, 否则如果 n=INT_MIN, -n 就会越界
if(n < 0)
{
    x = 1 / x;
    exp = - exp;
}
while(exp)
{
    // 快速幂方法 (位运算)
    // 假如 n=9, 9 写成二进制就是 1001
    // 每当 exp & 1 = 1 的时候, 就执行乘方运算
    if(exp & 1) res *= x;
    x *= x;
    exp >>= 1;
}
return res;
}
};
```