# 【题目】

给定一个非负整数 **num**。对于 **0 ≤ i ≤ num**范围中的每个数字 **i**，计算其二进制数中的 1 的数目并将它们作为数组返回。

**示例 1:**

**输入:** 2

**输出:** [0,1,1]

**示例 2:**

**输入:** 5

**输出:** [0,1,1,2,1,2]

# 【思路】

该题目主要考察的是移位运算和算法效率问题，首先可以化繁为简，先考虑给定一个数number，计算该数中二进制1的个数，这是该题的算法原型。

例如：十进制数值

一、将number不断右移，判断最低bit位是否为1，为1计数器加1。

0110 0100——>0011 0010 (右移动1位)

&0000 0001

--------------------------------------

0000 0000 计算器不累加

0110 0100——>0001 1001 (右移动2位)

&0000 0001

--------------------------------------

0000 0001 计算器累加

二、可以将一个0x1不断左移，并和number取与，如果不为0表示该位是1，计算器加1。

0110 0100

0000 0001——> 0000 0010 (左移动1位)

--------------------------------------

0000 0000 计算器不累加

0110 0100

0000 0001——> 0000 0100 (左移动2位)

--------------------------------------

0000 0100 计算器累加

三、高效率的算法

例如number = 100 二进制位为：0110 0100

number = 100-1 二进制为：0110 0011

可以看到减1可以使得原数据最低位的1变成0，其后面的0遍变成1，这样只要number&(number-1)不为0，计数器就可以累加。

# 【代码】