

371. Sum of Two Integers

题目描述: <https://leetcode.com/problems/sum-of-two-integers/>

给定两个整数（可能为负数）要求不能使用‘+’和‘-’求这两个数的和

解题思路:

1. 转换为二进制数求和
2. 首先不考虑进位的话则就是 $a \text{ xor } b$ ，将其赋值为新的a，用来进行下次加上下步得到的进位。
3. 考虑进位在这个的基础上加上进位，判断每一位上有无进位就是 $a \& b$ ，如果往上一步结果上加进位，需要在这个的基础上向左移位因为进位是加在高一位上的。
4. 新的b用来进位，如此循环运算直到进位为0。

不能按位拆开去算，因为如果有负数不会处理

代码:

```
class Solution {
public:
    int getSum(int a, int b) {
        int res = a;
        while(b != 0){
            res = a ^ b;
            b = (a & b) << 1;
            a = res;
        }
        return res;
    }
};
```