

二进制求和

给你两个二进制字符串，返回它们的和（用二进制表示）。

输入为 **非空** 字符串且只包含数字 1 和 0。

示例 1:

输入: a = "11", b = "1"

输出: "100"

示例 2:

输入: a = "1010", b = "1011"

输出: "10101"

```
class Solution {
public:
    string addBinary(string a, string b) {
        int a1 = a.size();
        int b1 = b.size();
        while(a1 < b1) //让两个字符串等长，若不等长，在短的字符串前补零，否则之
后的操作会超出索引
        {
            a = '0' + a;
            ++ a1;
        }
        while(a1 > b1)
        {
            b = '0' + b;
            ++ b1;
        }
        for(int j = a.size() - 1; j > 0; -- j) //从后到前遍历所有的位数，同位
相加
        {
            a[j] = a[j] - '0' + b[j];
            if(a[j] >= '2') //若大于等于字符'2'，需要进一
            {
                a[j] = (a[j] - '0') % 2 + '0';
                a[j-1] = a[j-1] + 1;
            }
        }
        a[0] = a[0] - '0' + b[0]; //将 ab 的第 0 位相加
```

```
    if(a[0] >= '2') //若大于等于 2，需要进一
    {
        a[0] = (a[0] - '0') % 2 + '0';
        a = '1' + a;
    }
    return a;
}
};
```