```
/*
给定两个表示复数的字符串。

返回表示它们乘积的字符串。注意，根据定义 i2 = -1 。

示例 1:

输入: "1+1i", "1+1i"
输出: "0+2i"
解释: (1 + i) * (1 + i) = 1 + i2 + 2 * i = 2i ，你需要将它转换为 0+2i 的形式。

示例 2:

输入: "1+-1i", "1+-1i"
输出: "0+-2i"
解释: (1 - i) * (1 - i) = 1 + i2 - 2 * i = -2i ，你需要将它转换为 0+-2i 的形式。

注意:

    输入字符串不包含额外的空格。
    输入字符串将以 a+bi 的形式给出，其中整数 a 和 b 的范围均在 [-100, 100] 之间。输出也应当符合这种形式。

来源：力扣（LeetCode）
链接：https://leetcode-cn.com/problems/complex-number-multiplication
著作权归领扣网络所有。商业转载请联系官方授权，非商业转载请注明出处。
*/
```

分析:

- 简单题:学会$sscanf$和$sprintf$这两个函数即可.

方法一:C++

```
class Solution
{
    public:
        string complexNumberMultiply(string a, string b)
        {
            int a1;
            int b1;
            int a2;
            int b2;
            sscanf(a.c_str(),"%d+%di",&a1,&b1);
            sscanf(b.c_str(),"%d+%di",&a2,&b2);
            return string( to_string(a1*a2-b1*b2) + "+" +
to_string(a1*b2+a2*b1) + "i");
        }
};
/*
执行结果:
```

```
通过
显示详情
执行用时 :8 ms, 在所有 C++ 提交中击败了25.00% 的用户
内存消耗 :8.5 MB, 在所有 C++ 提交中击败了53.09%的用户
*/
```

方法一:C_malloc

```c
char * complexNumberMultiply(char * a, char * b)
{
    char*   ret_val =   (char*)malloc(20*sizeof(char))  ;
    int     a1      =   0                               ;
    int     b1      =   0                               ;
    int     a2      =   0                               ;
    int     b2      =   0                               ;
    int     a3      =   0                               ;
    int     b3      =   0                               ;

    sscanf(a,"%d+%di",&a1,&b1);
    sscanf(b,"%d+%di",&a2,&b2);
    a3 = a1*a2-b1*b2;
    b3 = a1*b2+a2*b1;
    memset(ret_val,0,20*sizeof(char));
    sprintf(ret_val,"%d+%di",a3,b3);
    return ret_val;
}
/*
执行结果:
通过
显示详情
执行用时 :8 ms, 在所有 C 提交中击败了5.77% 的用户
内存消耗 :6.8 MB, 在所有 C 提交中击败了77.78%的用户
*/
```

方法一:C_全局变量

```c
char    ret_val[20];
char*   complexNumberMultiply(char * a, char * b)
{
    int a1  =   0;
    int b1  =   0;
    int a2  =   0;
    int b2  =   0;
    int a3  =   0;
    int b3  =   0;

    sscanf(a,"%d+%di",&a1,&b1);
    sscanf(b,"%d+%di",&a2,&b2);
    a3 = a1*a2-b1*b2;
    b3 = a1*b2+a2*b1;
    memset(ret_val,0,20*sizeof(char));
    sprintf(ret_val,"%d+%di",a3,b3);
    return ret_val;
}
```

```
/*
执行结果：
通过
显示详情
执行用时 :0 ms, 在所有 C 提交中击败了100.00% 的用户
内存消耗 :6.8 MB, 在所有 C 提交中击败了77.78%的用户
*/
```

AlimyBreak
2019.09.28