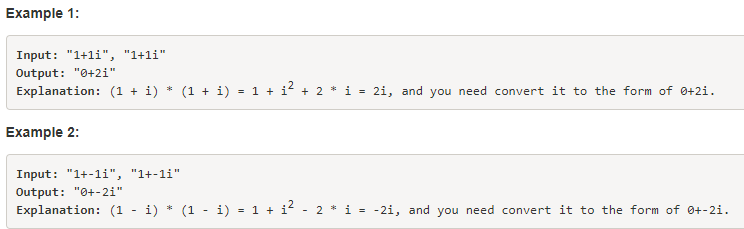
# 537. Complex Number Multiplication复数相乘

Given two strings representing two complex numbers.

You need to return a string representing their multiplication. Note i2 = -1 according to the definition.

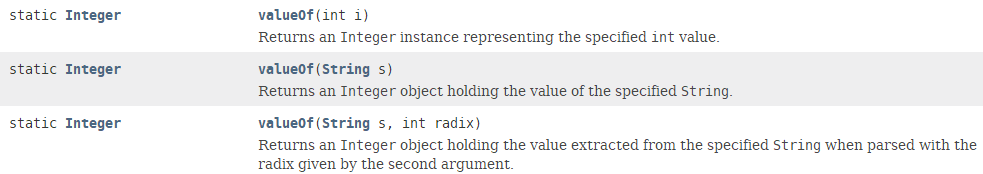


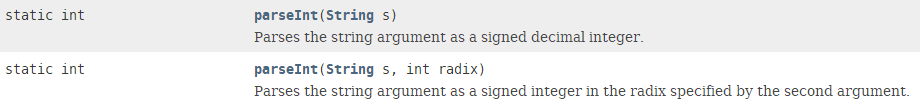
Note:

* The input strings will not have extra blank.
* The input strings will be given in the form of **a+bi**, where the integer a and b will both belong to the range of [-100, 100]. And the output should be also in this form.

**字符串转成int**：利用Integer的valueOf方法或parseInt方法。

int之间相乘直接使用\*即可，也可以使用**Math. multiplyExact(a,b)**函数。





## 算法1：

思路：首先获取+的索引位置pluseA和长度len，然后0-(pluseA-1)是实部,(pluseA+1)-(len-1)是虚部。然后根据公式计算实部和虚部，最后拼接成字符串。

class Solution {

public String complexNumberMultiply(String a, String b) {

int plusA = a.indexOf("+");

int plusB = b.indexOf("+");

int nA = a.length();

int nB = b.length();

int aReal = Integer.valueOf(a.substring(0, plusA));

int bReal = Integer.valueOf(b.substring(0, plusB));

int aImg = Integer.valueOf(a.substring(plusA + 1, nA - 1));

int bImg = Integer.valueOf(b.substring(plusB + 1, nB - 1));

int real = aReal \* bReal + -1 \* aImg \* bImg;

int img = aReal \* bImg + bReal \* aImg;

return real + "+" + img + "i";

}

}

## 算法2：

思路：首先利用正则表达式，将字符串分割成两部分，然后分别转换成int。再进行计算返回字符串结果即可。

class Solution {

public String complexNumberMultiply(String a, String b)

{

String x[] = a.split("\\+");

String y[] = b.split("\\+");

int a\_real = Integer.parseInt(x[0]);

int a\_img = Integer.parseInt(x[1].substring(0, x[1].length()-1));

int b\_real = Integer.parseInt(y[0]);

int b\_img = Integer.parseInt(y[1].substring(0, y[1].length()-1));

return (a\_real \* b\_real - a\_img \* b\_img) + "+" + (a\_real \* b\_img + a\_img \* b\_real) + "i";

}

}

## 算法3：

**仅用三次实数乘法完成复数相乘的运算**

有两个复数m=a+b\*i,n=c+d\*i,现在计算它们的乘法，设计一个只使用三次实数乘法的算法来完成m和n的计算。

根据复数乘法的计算公式：m\*n=（a\*c-b\*d）+(a\*d+b\*c)i，可以使用加法或者减法来削减乘法运算的数量。

计算C1=a\*d,C2=b\*c,C3=(a+b)(c-d)，虚部为C1+C2，实部为C3+C1-C2，计算完毕。

思路：

class Solution {

public String complexNumberMultiply(String a, String b) {

String[] ab = a.split("\\+");

int a1 = Integer.parseInt(ab[0]);

int b1 = Integer.parseInt(ab[1].substring(0,ab[1].length()-1));

String[] ab2 = b.split("\\+");

int a2 = Integer.parseInt(ab2[0]);

int b2 = Integer.parseInt(ab2[1].substring(0,ab2[1].length()-1));

**//三次实数乘法实现复数相乘**

**int C1 = Math.multiplyExact(a1,b2);**

**int C2 = Math.multiplyExact(b1,a2);**

**int C3 = Math.multiplyExact(a1+b1,a2-b2);**

return C3+C1-C2+"+"+(C1+C2)+"i";

}

}

## 注意：数值与字符串的拼接问题。

public static void main(String[] args) {

int a = 12;

int b = 16;

int c = 11;

int d = 66;

System.out.println(a+b+"+"+c+d+"i");//28+1166i

System.out.println(a+b+"+"+(c+d)+"i");//28+77i

}

字符串拼接按照从左向右，当多个数值与字符串进行拼接的过程中，遇到字符串之前，按照数值进行相加，遇到字符串之后，转变为字符串拼接。如果需要字符串之后的需要先进行数值计算，此时必须添加小括号。