

牛客网-华为机试练习题 57

题目描述

在计算机中，由于处理器位宽限制，只能处理有限精度的十进制整数加减法，比如在32位宽处理器计算机中，参与运算的操作数和结果必须在 $-2^{31} \sim 2^{31}-1$ 之间。如果需要更大范围的十进制整数加法，需要使用特殊的方式实现，比如使用字符串保存操作数和结果，采取逐位运算的方式。如下：

9876543210 + 1234567890 = ?

让字符串 num1="9876543210", 字符串 num2="1234567890", 结果保存在字符串 result = "11111111100".

-9876543210 + (-1234567890) = ?

让字符串 num1="-9876543210", 字符串 num2="-1234567890", 结果保存在字符串 result = "-11111111100".

要求编程实现上述高精度的十进制加法。

要求实现方法：

public String add (String num1, String num2)

【输入】num1: 字符串形式操作数1, 如果操作数为负, 则num1的前缀为符号位'-'

num2: 字符串形式操作数2, 如果操作数为负, 则num2的前缀为符号位'-'

【返回】保存加法计算结果字符串, 如果结果为负, 则字符串的前缀为'-'

注：

(1) 当输入为正数时, '+'不会出现在输入字符串中; 当输入为负数时, '-'会出现在输入字符串中, 且一定在输入字符串最左边位置;

(2) 输入字符串所有位均代表有效数字, 即不存在由'0'开始的输入字符串, 比如"0012", "-0012"不会出现;

(3) 要求输出字符串所有位均为有效数字, 结果为正或0时 '+'不出现在输出字符串, 结果为负时输出字符串最左边位置为 '-'。

输入描述:

输入两个字符串

输出描述:

输出给求和后的结果

示例1

输入

9876543210
1234567890

输出

11111111100

解决代码

```
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.math.BigInteger;

public class Main{

    public static void main(String[] args) throws IOException{
        BufferedReader bf = new BufferedReader(new InputStreamReader(System.in));
        String str;
        while((str=bf.readLine()) != null){
```

```
        BigInteger a = new BigInteger(str.trim());
        BigInteger b = new BigInteger(bf.readLine().trim());
        System.out.println(a.add(b).toString());
    }
    bf.close();
}
```