

# La meilleure somme

Un ami vient de vous envoyer une opération par email :  $1234=46$

Ca n'a aucun sens ! Mais après une analyse approfondie, vous comprenez que c'est que la touche "+" de son clavier est cassée. En effet, votre ami voulait vous envoyer  $12+34=46$

Sachant que votre ami doit vous transmettre prochainement d'autres opérations, vous décidez de réaliser un programme permettant de déterminer où les opérateurs d'addition doivent être placés pour rendre l'opération correcte.

Seule la partie gauche de l'opération peut nécessiter l'ajout d'opérateurs. La partie de droite sera toujours le nombre entier fourni initialement.

## Entrée de la fonction

Votre fonction prendra en entrée :

- Une première ligne contenant la partie gauche de l'équation, représentée sous la forme d'un nombre entier comprenant entre 1 et 19 chiffres
- Une deuxième ligne contenant la partie droite de l'opération, représentée sous la forme d'un nombre entier comprenant entre 1 et 19 chiffres

## Sortie de la fonction

Votre fonction doit retourner l'opération corrigée, en ajoutant si nécessaire des opérateurs "+" dans la partie gauche de l'équation. L'opération affichée ne doit pas contenir d'espaces.

Les chiffres de la partie gauche doivent être affichés dans le même ordre que l'ordre initial.

ex.  $1+2+3+4=10$

## Exemple

### Entrée

```
1234
46
```

### Sortie

```
12+34
```

Langage : **Java**

Les données d'entrée de votre fonction vous sont transmises automatiquement . Vous n'avez pas à les saisir.

Pour lire les données en entrée : `r.readLine();`

Pour écrire les données en sortie : **System.out.println();**

Vous pouvez également utiliser votre **IDE** en local. Pensez ensuite à copier/coller votre code ci-dessous et à vérifier qu'il fonctionne.

```
import java.io.BufferedReader;

import java.io.BufferedReader;

import java.io.InputStreamReader;

class kskills{

    public static void main (String[] args) throws java.lang.Exception {

        BufferedReader r = new BufferedReader (new InputStreamReader (System.in));

        //Write your code here ...

    }

}
```

- ☒ Valeurs de l'exemple
- ☒ Aucune somme à réaliser
- ☒ 10 chiffres
- ☒ Somme de chaque chiffre

*Votre code doit passer tous les cas de test avec succès.*

Valeurs de l'exemple

## Input

```
1234
46
```

## Expected output

```
12+34
```

Aucune somme à réaliser

## Input

483  
483

### Expected output

483

10 chiffres

### Input

8465027591  
353

### Expected output

$8+4+6+50+275+9+1$

Somme de chaque chiffre

### Input

8395817418973258  
88

### Expected output

$8+3+9+5+8+1+7+4+1+8+9+7+3+2+5+8$