

# Parcours complet [tb07] – Exercice résolu

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 19 mai 2018

## Table des matières

<b>1</b>	<b>Parcours complet / pgcomplet</b>	<b>2</b>
1.1	Fonction saisirTab (saisie d'un tableau) . . . . .	2
1.2	Fonction sommeTab (somme d'un tableau) . . . . .	3
1.3	Programme de test . . . . .	4
<b>2</b>	<b>Références générales</b>	<b>5</b>

## Java - Parcours complet (Solution)



**Mots-Clés** Tableau unidimensionnel ■

**Utilise** Définitions et notations, Tableaux et paramètres, Parcours de tableaux ■

**Difficulté** ●○○ (30 min) ■

# 1 Parcours complet / pgcomplet



## Objectif

Cet exercice calcule la somme des éléments d'un tableau d'entiers.

Exemple :

```
[45, 54, 1, -56, 22, 134, 49, 12, 90, -27]
```

```
==> Somme des éléments est 324
```



## Définitions Java

```
final int TMAX = ...;
```

## 1.1 Fonction saisirTab (saisie d'un tableau)



Écrivez le **profil** d'une fonction `saisirTab(t)` qui effectue la saisie du nombre de valeurs dans un entier, puis saisit les valeurs dans un `ITableau t` et renvoie le nombre de valeurs saisies.

### Solution Paramètres

**Sortants** : Un `ITableau t`

**Résultat de la fonction** : Un entier



### Saisie du nombre de valeurs

Le nombre de valeurs doit être compris entre 1 (inutile d'appeler cette fonction pour saisir un tableau vide) et `TMAX` (taille maximale d'un `ITableau`). On utilise donc une répétitive `TantQue` ou `Répéter` selon que l'on souhaite ou non avertir l'utilisateur d'une erreur de saisie.



Écrivez la saisie contrainte du nombre de valeurs entre 1 et `TMAX`.

Affichez l'invite (où `[x]` désigne le contenu de `x`) :

```
Nombre de valeurs dans [1..[TMAX]]?
```



### Saisie des valeurs

Soit `n` le nombre de valeurs à saisir. Il faut répéter `n` fois la saisie d'une valeur (parcours complet). Par conséquent, le remplissage du tableau sera piloté par une boucle `Pour`.



Écrivez la saisie des valeurs entières dans `t`.

Affichez une invite spécifiant le rang de l'élément à saisir :

```
t[..]?
```



Validez votre fonction avec la solution.

### Solution Java @[UtilsTB.java]

```
/**
 * Saisie de valeurs dans un ITableau
 * @param[out] t - un ITableau
 * @return le nombre de valeurs saisies dans [1..TMAX]
 */
public static int saisirTab(int[] t)
{
    final int TMAX = t.length;
    Scanner input = new Scanner(System.in);
    int n = 0;
    while (!(1 <= n && n <= TMAX))
    {
        System.out.print("Nombre de valeurs dans [1.." + TMAX + "]? ");
        n = input.nextInt();
    }
    for (int j = 0; j < n; ++j)
    {
        System.out.print("t[" + j + "]? ");
        t[j] = input.nextInt();
    }
    return n;
}
```

## 1.2 Fonction sommeTab (somme d'un tableau)



Écrivez le **profil** d'une fonction `sommeTab(t, n)` qui renvoie la somme des `n` premières valeurs d'un `ITableau t`.

### Solution Paramètres

**Entrants** : Un `ITableau t` et un entier `n`

**Résultat de la fonction** : Un entier (somme d'entiers)



Comment initialiser la somme à retourner ?

### Solution simple

A zéro comme dans le cas de la somme d'une séquence de valeurs.



Écrivez le corps de la fonction.

### Solution simple

On déclare une variable `somme` (le résultat) et on l'initialise à zéro. Ensuite on parcourt `t` par la boucle `Pour` (compteur `ix`) et on ajoute la valeur de l'élément `t[ix]` dans `somme`.



Validez votre fonction avec la solution.

**Solution Java** @[UtilsTBOpers.java]

```
/**
 * Somme de valeurs d'un ITableau
 * @param[in] t - un ITableau
 * @param[in] n - nombre de valeurs dans [0..TMAX[
 * @return Somme de n valeurs de t
 */
public static int sommeTab(final int[] t, int n)
{
    int somme = 0;
    for (int j = 0; j < n; ++j)
    {
        somme += t[j];
    }
    return somme;
}
```

### 1.3 Programme de test



Écrivez un programme qui définit la constante `TMAX=20`, demande et stocke des entiers dans un `ITableau` puis en calcule et affiche la somme des valeurs.



Testez. Exemple d'exécution :

```
Nombre d'éléments dans [1..20]? 10
t[0]? 45
t[1]? 54
t[2]? 1
t[3]? -56
t[4]? 22
t[5]? 134
t[6]? 49
t[7]? 12
t[8]? 90
t[9]? -27
==> Somme des éléments est 324
```



Validez votre programme avec la solution.

**Solution Java** @[pgcomplet.java]

```
public static void main(String[] args)
{
    int[] tab = new int[TMAX];
    int nelems = UtilsTB.saisirTab(tab);
}
```

```
System.out.println("=> Somme des elements est " + sommeTab(tab,nelems));  
}
```

## 2 Références générales

Comprend ■