

# Parcours partiel [tb07] – Exercice résolu

Karine Zampieri, Stéphane Rivière

Unisciel  algoprogram  Version 19 mai 2018

## Table des matières

<b>1</b>	<b>Parcours partiel / pgpartiel</b>	<b>2</b>
1.1	Fonction tabPositifs (test d'entiers positifs) . . . . .	2
1.2	Programme de test . . . . .	3
1.3	Fonction tabNegatifs (test d'entiers négatifs) . . . . .	4
<b>2</b>	<b>Références générales</b>	<b>4</b>

## C - Parcours partiel (Solution)



**Mots-Clés** Tableau unidimensionnel ■

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

**Difficulté** ●○○ (30 min) ■

# 1 Parcours partiel / pgpartiel



## Objectif

Cet exercice indique si un tableau est composé uniquement d'entiers positifs.



## Définitions C

```
enum { TMAX = ... };  
typedef int ITableau[TMAX];
```

## 1.1 Fonction tabPositifs (test d'entiers positifs)



Écrivez le **profil** d'une fonction `tabPositifs(t,n)` qui renvoie l'**indice du premier élément négatif** parmi les `n` premières valeurs d'un `ITableau t`, et `-1` sinon (c.-à-d qu'il ne contient que des éléments positifs ou nuls).

### Solution Paramètres

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

**Résultat de la fonction** : Un entier



Quel est le type de parcours : complet, partiel, imbriqué ? Justifiez.

### Solution simple

C'est un parcours partiel car dès qu'un élément négatif est détecté on peut s'arrêter. On utilise donc une boucle `TantQue`.



Écrivez la boucle de test ainsi que le résultat de la fonction.



Validez votre fonction avec la solution.

### Solution C @[pgpartiel.c]

```
int main(void)
```

### Solution commentée

La boucle traverse le tableau tant que les `n` éléments non pas été visités **et** que l'élément courant est positif ou nul. A la sortie, on teste la valeur de `pos` par rapport à `n` pour savoir si tous les éléments ont été visités ou bien si on est sur un élément négatif.

## 1.2 Programme de test



Téléchargez le fichier suivant et mettez-le dans votre dossier.

C @[UtilsTB.c]



Copiez/collez ensuite les lignes suivantes :

C Au début de votre programme :

```
#include "UtilsTB.c"
```



Soit la fonction `saisirTab(t)` qui effectue la saisie contrôlée du nombre de valeurs (entier compris entre 1 et `TMAX`), saisit les valeurs entières dans un `ITableau t` puis renvoie l'entier du nombre de valeurs saisies.

C @[saisirTab] (dans UtilsTB)



Écrivez un programme qui demande et stocke des entiers dans un `ITableau` puis indique s'il est composé uniquement d'entiers positifs. Dans la négative, il indique la valeur et la position du premier élément négatif.



Testez. Exemples d'exécution :

```
Nombre d'éléments dans [1..30]? 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
==> Tableau d'entiers relatifs
==> Premier entier -56 en position 3
```

```
Nombre d'éléments dans [1..30]? 5
t[0]? 12
t[1]? 35
t[2]? 8
t[3]? 6
t[4]? 11
==> Tableau d'entiers positifs
```



Validez votre programme avec la solution.

**Solution C** @[pgpartiel.c]

```
int main(void)
{
    Tableau tab;
    int nelems = saisirTab(tab);
    int pos = tabPositifs(tab,nelems);
    if (pos == -1)
    {
        printf("=> Tableau d'entiers positifs\n");
    }
    else
    {
        printf("=> Tableau d'entiers relatifs\n");
        printf("=> Premier entier %d en position %d\n",tab[pos],pos);
    }
}
```

### 1.3 Fonction tabNegatifs (test d'entiers négatifs)

En partant de la fonction écrite ci-dessus,



Écrivez une fonction `tabNegatifs(t,n)` qui renvoie l'indice du premier élément positif parmi les `n` premières valeurs d'un `ITableau t`, `-1` sinon.



Validez votre fonction avec la solution.

**Solution C** @[pgpartiel.c]

```
int tabNegatifs(const Tableau t,int n)
{
    int pos = 0;
    while (pos < n && t[pos] <= 0)
    {
        pos += 1;
    }
    return (pos < n ? pos : -1);
}
```

## 2 Références générales

Comprend ■