*mercredi 24 septembre 2014*
*Sébastien Combéfis*

# *Exercices en salle 3*
# *Les tableaux à une dimension*

## 1 Exercices rapides d'appropriation

1. Définissez une constante `ALPHA` qui vaut 132
2. Quelle est la portée d'un `#define` ?
3. Créez un nouveau type "caractere" sur base du `char` existant et déclarez une variable de ce type
4. Si vous créez un type `bool` pour des booléens, comme quel type agit-il ?
5. Si un tableau a une taille de 10, quel sera son premier et son dernier indice ?
6. Déclarez un tableau d'`int` de taille 2 et attribuez la valeur 1 à chaque case
7. Affichez la valeur de la case d'indice `1` du tableau précédent

## 2 Définition de type

Voici quatre variables déjà déclarées mais seulement trois sont initialisées avec des valeurs inconnues :

```
1 int a = ....;
2 int b = ....;
3 int c = ....;
4
5 bool isTrue;
```

Vous devez créer un code qui :

1. Prenne en charge le type `bool`, sur base du type `int` existant.
2. Stocke dans la variable `isTrue` la valeur `TRUE` si `a` est différent de `b` et que `b` est aussi plus petit ou égal à `c`. Sinon, la valeur `FALSE` y sera stockée.

## 3 Tableaux à une dimension

Pour chaque exercice, il faut définir le problème en donnant les entrées, les sorties et les effets de bord. Écrivez ensuite l'algorithme qui résout le problème. Les différents problèmes agiront sur un tableau de taille $N$ déclaré comme :

```
int tab[N]
```

1. Le problème consiste à compter le nombre de zéros dans le tableau. Voici un tableau d'exemple :

```
int N = 5;
int tab[N] = {0, -5, -7, 0, 1};

// La réponse doit être 2
```

2. Le problème consiste à incrémenter de 2 toutes les valeurs stockées dans le tableau. Voici un tableau d'exemple :

```
int N = 3;
int tab[N] = {0, 6, -3};

// Après exécution le tableau contient {2, 8, -1}
```

3. Le problème consiste à calculer la somme des éléments d'un tableau. Voici un tableau d'exemple :

```
int N = 4;
int tab[N] = {0, 6, -3, 4};

// La réponse doit être 7 pour cet exemple
```

4. Le problème consiste à remplacer chaque valeur du tableau par sa valeur absolue. Voici un tableau d'exemple :

```
int N = 6;
int tab[N] = {0, 3, 4, -2, -5, -3};

// Après exécution du code, le tableau contient {0, 3, 4, 2, 5, 3}
```

5. Le problème consiste à calculer un nouveau tableau qui est la somme de deux autres. La case à l'indice `i` du nouveau tableau vaudra la somme des cases aux indices `i` des deux autres tableaux. Voici deux tableaux d'exemple :

```
1 int tab[5] = {1, -2, 3, 0, 5};
2 int tab2[5] = {5, 4, 0, 2, 3};
3
4 // Le nouveau tableau contiendra {6, 2, 3, 2, 8}
```

6. Le problème consiste à compter le nombre d'occurrences de chaque lettre d'un mot.

```
1 int N = ....;
2 char tab_lettres[N] = ....;
3 int tab[26] = {0, 0, 0, ..., 0};
```

`tab_lettres` contient des caractères correspondant aux 26 lettres minuscules.

`tab[0]` contient le nombre de fois que la lettre `'a'` apparait, `tab[1]` le nombre de `'b'`...