

# Bases de C

## TP 2 - Tableaux et Chaînes

Ce TP a pour objectif de vous faire manipuler les tableaux et chaînes de caractères (appelées *strings* en anglais).

### 1 Programmes sur tableaux

- 1) Écrivez une fonction comparant deux tableaux d'entiers.

Pour vous aider, voici le prototype attendu pour la fonction :

```
int CompareTab(int tab1[], int tab2[], int len1, int len2)
```

- 2) Écrivez maintenant une fonction palindrome fonctionnant sur un tableau dont la longueur est donnée en paramètre.

Pour vous aider, voici le prototype attendu pour la fonction :

```
int PalindromeTab(int tab[], int len)
```

- 3) Écrivez une fonction qui renvoie la valeur la plus grande/petite d'un tableau.
- 4) Écrivez une fonction qui calcule la somme de tous les éléments d'un tableau.
- 5) Écrivez une fonction qui recherche un élément dans un tableau. Si l'élément est trouvé, la fonction doit renvoyer l'index de la case où l'élément se trouve. Si l'élément n'est pas trouvé, la fonction doit renvoyer  $-1$ .
- 6) Écrivez une fonction qui teste si les éléments d'un tableau sont dans l'ordre croissant/décroissant.
- 7) Écrivez une fonction qui insère un élément dans un tableau à une position précise, et renvoie l'élément s'y trouvant. Si la position est plus grande que la taille du tableau, alors vous effectuerez le modulo.
- 8) Écrivez une fonction qui insère un élément dans un tableau à une position précise tout en poussant tous les éléments du tableaux vers la fin, et qui renvoie l'élément qui sera hors du tableau. Si la position est plus grande que la taille du tableau, alors vous effectuerez le modulo.
- 9) Écrivez une fonction qui supprime un élément dans un tableau à une position précise, puis qui décale tous les éléments suivants d'un cran en arrière (pour boucher le trou). Si la position est plus grande que la taille du tableau, alors vous effectuerez le modulo.
- 10) Écrivez une fonction qui supprime un élément dans un tableau à une position précise, puis qui décale tous les éléments suivants d'un cran en arrière (pour boucher le trou). Enfin, la fonction prendra un paramètre supplémentaire qui sera l'élément ajouté en fin de tableau. Si la position est plus grande que la taille du tableau, alors vous effectuerez le modulo.

## 2 Programmes sur chaînes de caractères

- 11) Écrivez une fonction palindrome fonctionnant sur une chaîne de caractères (le '\0' final ne sera bien entendu pas pris en compte dans le palindrome).
- 12) Écrivez une fonction qui vérifie si une chaîne est le préfixe d'une autre chaîne, et renvoie la position du début de la sous-chaîne, ou  $-1$  si elle n'est pas trouvée.
- 13) Écrivez une fonction qui vérifie si une chaîne est le suffixe d'une autre chaîne, et renvoie la position du début de la sous-chaîne, ou  $-1$  si elle n'est pas trouvée.
- 14) Écrivez une fonction qui remplace un préfixe de chaîne par un autre préfixe. Si le préfixe n'est pas trouvé, vous retournerez  $-1$ , sinon vous renverrez le nombre de caractères modifiés.
- 15) Écrivez une fonction qui remplace un suffixe de chaîne par un autre suffixe. Si le suffixe n'est pas trouvé, vous retournerez  $-1$ , sinon vous renverrez le nombre de caractères modifiés.
- 16) Écrivez une fonction qui remplace tous les caractères minuscules en majuscules d'une chaîne de caractères.
- 17) Écrivez une fonction qui remplace tous les caractères majuscules en minuscules d'une chaîne de caractères.
- 18) Écrivez une fonction qui standardise tous les noms de fichiers d'une liste en renvoyant un pointeur vers une nouvelle liste.

Pour cet exercice, considérons les noms standards comme étant : **IMG\_XXXX.jpg** (où **XXXX** sont des chiffres de 0 à 9).

Les noms donnés seront contenus dans un tableau (il s'agira donc d'un tableau contenant des **char\***). Les noms de la liste peuvent contenir des majuscules ou minuscules n'importe où, et ne pas avoir suffisamment de chiffres. Vous devez tous les mettre au format standard en les recopiant dans une liste de sortie.

Si des éléments sont déjà au bon format, vous devez les recopier vers la nouvelle liste quoiqu'il advienne (ceci permettra de libérer l'ancienne liste de noms sans perdre les nouveaux noms).

Pour réaliser cet exercice, vous devrez à la fois lire les noms, les mettre au bon format en ajoutant éventuellement des 0, et construire un tableau de pointeurs. Les noms non standards démarreront toujours par un **img** dans n'importe quelle casse, suivi d'un underscore ( **\_** ), suivi d'*au moins* un chiffre et *au plus* quatre chiffres, et enfin, terminé par une extension **jpg** dans n'importe quelle casse.

Exemples de noms standards :

- **IMG\_0000.jpg**
- **IMG\_0042.jpg**
- **IMG\_1234.jpg**
- **IMG\_9000.jpg**

Exemples de noms non standards :

- **IMG\_0.jpg**
- **ImG\_042.JPG**
- **img\_1234.JpG**
- **iMg\_9.jpg**