Langace C - Travaux Dirigés n° 3

— Licence Informatique —

▶ **Exercice 1** Écrire une fonction `map` qui applique une fonction sur un tableau d'entiers. La fonction `map` prend en paramètre une fonction `f`, un tableau d'entiers ainsi que la taille du tableau. La fonction `f` prend un entier en paramètre et retourne un entier. Si le tableau contient les valeurs $x_0, x_1, x_2, \ldots, x_n$, le tableau contient après l'appel les valeurs $f(x_0), f(x_1), f(x_2), \ldots, f(x_n)$.

▶ **Exercice 2** Écrire une fonction `itarray` qui calcule l'itération d'une fonction à deux arguments sur un tableau d'entiers. La fonction `itarray` prend en paramètre une fonction `f`, un tableau d'entiers ainsi que la taille du tableau. La fonction `f` prend deux entiers en paramètre et retourne un entier. Si le tableau contient les valeurs $x_0, x_1, x_2, \ldots, x_n$, la fonction `itarray` retourne la valeur

$$f(f(\ldots f(f(x_0, x_1), x_2), \ldots)x_n).$$

Si la taille du tableau est respectivement 1, 2, 3 ou 4 la fonction `itarray` retourne repsectivement les valeurs x_0, $f(x_0, x_1)$, $f(f(x_0, x_1), x_2)$ et $f(f(f(x_0, x_1), x_2), x_3)$.

▶ **Exercice 3** Écrire une fonction `swapchar` qui échange le contenu de deux variable de type `char`.

▶ **Exercice 4** Utiliser la fonction de l'exercice précédent pour écrire une fonction `sortstring` qui trie les caractères d'une chaîne par ordre alphabétique. On utilisera la méthode de tri par selection.

▶ **Exercice 5** Écrire une fonction `cmpstring` qui compare deux chaînes de caractères pour l'ordre lexicographique. Cette fonction rendra -1, 0 ou 1 comme le fait la fonction `strcmp` de la librairie standard. On utilisera en priorité des pointeurs pour parcourir les chaînes.

▶ **Exercice 6** Écrire une fonction `strsubs` qui substitue une chaîne de caractères dans une autre chaîne. Cette fonction prend les quatre paramètres suivants :

1. une chaîne ***str*** à l'intérieur de laquelle s'effectue la substitution.
2. une chaîne ***motif*** qui remplace une partie de la chaîne `str`.
3. une position ***pos*** dans la chaîne `str` à partir de laquelle on commence à remplacer les caractères de `str` par ceux de `motif`.
4. un nombre ***len*** qui donne le nombre maximal de caractères à substituer. La fonction en substitue moins si les fins des chaînes `str` ou `motif` sont atteintes avant.

La fonction `strsubs` retourne le nombre de caractères qu'elle a effectivement substitués. Le morceau de code

```
char s[] = "bonjour à tous";
strsubs(s, "soirée", 3, 4);
printf("s = \"%s\"\n", s);
```

doit afficher le résultat `s = "bonsoir à tous"`.

Réécrire cette fonction en utilisant des pointeurs.

▶ **Exercice 7** Écrire une fonction `swapchar` qui échange le contenu de deux variables de type `char`. Utiliser cette fonction pour écrire une fonction `strreverse` qui renverse une chaîne (en essayant d'utiliser des pointeurs). Le morceau de code

```
char s[] = "bonjour à tous";
strreverse(s);
printf("s = \"%s\"\n", s);
```

doit afficher le résultat `s = "suot à ruojnob"`

▶ **Exercice 8** Écrire une fonction `sortarray` qui trie un tableau de chaînes de caractères pour l'ordre lexicographique. Écrire une fonction `main` qui applique la fonction `sortstring` sur chacun des arguments de la ligne de commande puis les trie par ordre lexicographique.