

## Chaînes de Caractères

### A `libchaine.c`

Les fonctions de cette partie sont à intégrer dans une bibliothèque `libchaine.c`. Les tests de validité sont effectués dans `testlibchaine.c`.

A.1 Ecrire une fonction `char* newChaine(int n)` qui renvoie une chaîne de caractères permettant de stocker au plus  $n$  caractères.

A.2 Ecrire l'exemple `miroir.c` du cours sous la forme d'une fonction `miroir`.

La spécification précédente est en fait incomplète du point de vue gestion de la mémoire. Quelles sont les différents prototypes possibles et leurs conséquences ?

A.3 Ecrire une fonction `int longueur(char * chaine)` qui renvoie la longueur de la chaîne de caractères `chaine`. On supposera que cette chaîne respecte la convention usuelle de terminaison par `'\0'`.

Quelle est la longueur de la chaîne vide ? Quelle est sa représentation en mémoire ?

A.4 Améliorer, si ce n'est déjà fait, la fonction `miroir` précédente en utilisant la fonction `longueur`.

A.5 Ecrire une fonction `char * souschaine(char * chaine, char carac)` qui renvoie la sous-chaîne de `chaine` qui commence par la lettre `carac`, si elle existe, et le pointeur `NULL` sinon.

A.6 Ecrire une fonction `signed int compare(char *, char *)` qui compare deux chaînes de caractères pour l'ordre lexicographique. Cette fonction renvoie un entier strictement négatif, nul, ou strictement positif selon le résultat de la comparaison.

### B Manipulation de pointeurs

B.7 Ecrire une fonction `echange` qui échange deux chaînes de caractères, passées par référence, si elles ne sont pas dans l'ordre alphabétique croissant.

B.8 Ecrire un programme `trialpha` qui classe ses arguments par ordre alphabétique croissant avant de les afficher.

On utilisera un tri à bulles.

### C Saisie

C.9 Ecrire une fonction `char * lire_chaine_au_clavier(void)` qui lit une chaîne au clavier lettre par lettre jusqu'au retour à la ligne. Les chaînes feront moins de 64 caractères ou seront tronquées.

C.10 Ecrire une fonction `char * saisie()` qui renvoie une chaîne de caractères saisie au clavier par l'utilisateur. On utilisera `int getchar()`.

Comment faire pour qu'il n'y ait pas de limite a priori sur la longueur de la chaîne saisie ? (*Indication* Implémenter une technique de réallocation de mémoire.)