

## LANGAGE C AVANCÉ

### TD 3 : Exercices sur les pointeur Encore des listes chaînées

---

**Objectif :** le présent TD a pour but de manipuler les listes chaînées sous plusieurs formes. Il vient en complément du TD précédent. Il aborde aussi les pointeurs de fonctions et les fonctions aux arguments variés.

#### Listes

Pour les exercices de cette section, on considère que la structure d'un élément de liste liée suivante

```
struct Tlist
{
    int nb;
    struct Tlist * next;
}
```

Les éléments dans la liste sont des entiers signés, les algorithmes ne dépendent cependant pas du type des éléments.

1. Écrire une fonction `erase_0` prenant comme argument une liste d'entiers et retournant cette liste privée de toutes les occurrences de 0.
2. Définir la fonction `list_count` à deux arguments qui renvoie le nombre de fois que l'on a rencontré le premier argument dans la liste en second argument.
3. Définir la fonction `list_last` qui renvoie le dernier élément de la liste donnée en argument. On suppose que celle-ci contient au minimum un élément.
4. Écrire une fonction `list_big` qui prend comme arguments un nombre entier `n` et une liste `l` de nombres entiers, telle que `list_big(n, l)` est la liste des éléments de `l` plus grands que `n`.
5. Construire une fonction prenant une liste liée `l` triée et un nombre `n` quelconques en arguments, et qui renvoie la liste triée contenant tous les éléments de `l` ainsi que `n`.
6. Définir la fonction `list_reverse` qui rend une liste contenant les éléments, de la liste donnée en argument, dans l'ordre inverse.

#### Fonctions à nombre d'arguments variable

7. Écrire une fonction prenant un nombre `n`, et `n` entiers (`n` quelconque) et renvoie un tableau contenant les `n` entiers
8. Soit la structure d'un élément de liste liée

```
struct Tlist
{
    int nb;
    struct Tlist * next;
}
```

Construire une fonction prenant un nombre  $n$  quelconque comme premier argument, et  $n$  arguments entiers, et qui renvoie une liste contenant ces arguments dans l'ordre.

### Fonctions en argument

**9.** Utiliser `qsort` pour trier un tableau de nombres.

**10.** A l'image de `qsort` créer une fonction appliquant une fonction donnée en argument (par un pointeur) à tous les éléments d'un tableau.

**11.** Comme précédemment, mais pour une liste liée d'entiers au lieu d'un tableau.

**12.** Utiliser les deux fonctions précédentes pour sortir à l'écran les éléments d'un tableau (d'une liste) d'entiers.

**13.** Écrire une fonction  $f$  qui prend comme argument une fonction  $g$  ( $g$  prend un entier en argument et renvoie un flottant) et deux entiers  $a$  et  $b$  ( $a < b$ ), et qui renvoie le flottant  $\sum_{a \leq i \leq b} g(i)$ .

Utiliser cette fonction pour calculer  $\sum_{1 \leq i \leq 20} \sqrt{i}$ .

**14.** Écrire une fonction `list_filter` prenant deux arguments, une liste `l` d'objets et un prédicat unaire `p`, renvoyant la liste des éléments de `l` pour lesquels `p` renvoie un entier non null.