

Programmation en C et structures de données

guillaume.revy@univ-perp.fr

Structures et allocations dynamiques

Exercice 1. Allocation dynamique simple

- 1. Écrire une fonction qui alloue dynamiquement un tableau de taille n passée en paramètre, initialise les éléments à 0, puis renvoie ce tableau à la fonction appelante, en utilisant l'instruction **return**.
- 2. Modifier cette fonction pour ne plus utiliser l'instruction **return**.
- 3. Écrire une fonction qui prend en paramètre un tableau alloué dynamiquement, et qui modifie la taille de ce tableau, plus particulièrement qui retourne un nouveau tableau avec une nouvelle taille.

Exercice 2. Entier en précision arbitraire

En langage C, un entier 64 bits non signé permet de manipuler des valeurs dans l'intervalle

$$[0, 2^{64} - 1] = [0, 18446744073709551615],$$

et arithmétique signée, l'intervalle est

$$[-2^{63}, 2^{63} - 1] = [-9223372036854775808, 9223372036854775807].$$

Nous allons proposer une méthode pour représenter des nombres plus grand, voire beaucoup plus grands, et même arbitrairement grands. Plus précisément, un nombre $n \in \mathbb{Z}$ sera représenté par

- un signe,
- et un ensemble de k entiers non signés n_i sur 16 bits, avec $i \in \{0, \dots, k-1\}$, tels que

$$n = \sum_{i=0}^{k-1} n_i \cdot 2^{16 \cdot i}.$$

- 1. Proposer une structure qui permet de manipuler ce genre de nombres, étant donné que la valeur de k peut être différente pour chaque nombre.
- 2. Écrire une fonction qui crée un nombre sur p bits (p pouvant être arbitrairement grand) et l'initialise à 0, avec p multiple de 16.
- 3. Écrire une fonction qui effectue l'addition (ou soustraction) de deux nombres dans cette représentation, et stocke le résultat dans un troisième paramètre.
- 4. Écrire une fonction qui effectue la multiplication de deux nombres dans cette représentation, et stocke le résultat dans un troisième paramètre.
- 5. Écrire une fonction qui affiche un nombre dans cette représentation.
- 6. Écrire enfin une fonction qui détruit un nombre préalablement créé.

Exercice 3. Quelques opérations matricielles

En langage C, une matrice peut être représentée par un tableau 2D.

- 1. Écrire une fonction qui alloue dynamiquement la mémoire pour une matrice de taille $n \times m$ de réels, puis qui renvoie cette matrice à la fonction appelante en utilisant l'instruction **return**.
- 2. Modifier cette fonction pour ne plus utiliser l'instruction **return**.
- 3. Écrire une fonction qui affiche le contenu d'une matrice passée en paramètre.
- 4. Écrire une fonction qui multiplie deux matrices, et stocke le résultat dans un troisième paramètre.
- 5. Écrire une fonction qui calcule la transposée d'une matrice, et modifie la matrice passée en paramètre.
- 6. Écrire enfin une fonction qui détruit une matrice préalablement créée.