TP 2 : tableaux, pointeurs, références

Exercice 1 :
Écrire une fonction qui échange les valeurs de deux variables de type `int`.

Exercice 2 :
Écrire une fonction qui renvoie la valeur maximale d'un tableau de variables de type `double`.

Exercice 3 :
Écrire une fonction qui remplit un tableau avec des valeurs aléatoires tirées uniformément dans $]0,1[$. Écrire une deuxième fonction qui calcule un histogramme à 10 classes du tableau ainsi créé (autrement dit, qui compte le nombre de valeurs se trouvant dans l'intervalle $]i/10,(i+1)/10[$ pour $i \in \{0,\dots,9\}$), puis qui affiche le résultat.

Exercice 4 :
Écrire une fonction qui trie un tableau de variables de type `double`[1].

Exercice 5 :
Écrire une fonction qui calcule le produit de deux matrices à coefficients de type `double`. Remarque : le type naturel pour manipuler les matrices est le type `double**`.

Exercice 6 :
Écrire un programme qui calcule une solution approchée de l'équation

$$\begin{cases} \partial_t u(t,x) = c\partial_x^2 u(t,x) & t>0, x\in]0,1[, \\ u(0,x) = \cos(3\pi x) & x \in]0,1[, \\ u(t,0) = 1 & t \geq 0, \\ u(t,1) = -1 & t \geq 0. \end{cases}$$

Pour cela, on choisit un pas de temps δ_t et un pas d'espace $\delta_x = 1/N$, et on va approcher la valeur de $u(n\delta_t, k\delta_x)$ par la valeur $\bar{u}_{n,k}$, définie par $\bar{u}_{0,k} = \cos(3\pi k/N)$, et par

$$\bar{u}_{n+1,k} = \begin{cases} 1 & \text{si } k = 0, \\ -1 & \text{si } k = N, \\ \bar{u}_{n,k} + c\frac{\delta_t}{\delta_x^2}(\bar{u}_{n,k+1} + \bar{u}_{n,k-1} - 2\bar{u}_{n,k}) & \text{sinon.} \end{cases}$$

On prendra $c = 0.1$, et on calculera la solution jusqu'au temps $t = 1$. Attention, pour que le schéma soit stable, la condition $2c\delta_t < \delta_x^2$ doit être vérifiée.

1. La complexité optimale pour trier un tableau de taille n est $\mathcal{O}(n\log(n))$. L'algorithme "naïf" (celui auquel on penserait pour trier un jeu de cartes) est en $\mathcal{O}(n^2)$.