

TP sur le chapitre 3 (fonctions et tableaux)

Exercice 1 : Dans un terminal, consultez l'aide de la commande Linux `ls` (avec `man ls`), de la commande Linux `gcc` (avec la commande `man gcc`) en vous concentrant sur les paramètres permettant d'inclure des bibliothèques (= *libraries*), de la fonction `abs` du C (avec `man abs`), de la fonction `sqrt` du C (avec `man sqrt`), de la fonction `cos` du C (avec `man cos`).

Exercice 2 : Écrivez un programme qui calcule et affiche la racine carrée d'un réel x .

Exercice 3 : Écrivez un programme qui :

- déclare un tableau de 10 valeurs,
- remplit ces valeurs par des entiers choisis entre 1 et 20 aléatoirement (cf fonctions `srand` et `rand`),
- affiche ces valeurs,
- calcule et affiche la moyenne des valeurs.

Remarque : vous aurez à définir au moins les fonctions suivantes : `choisirEntier`, `initialiseTab`, `afficheTab` et `calculeMoyenne`.

Exercice 4 : Écrivez un programme qui :

- déclare, initialise aléatoirement et affiche un tableau de 10 valeurs,
- trie le tableau en utilisant la méthode du tri par sélection du maximum,
- affiche le tableau trié.

Remarques :

- Le tri par sélection du maximum consiste à choisir à chaque étape le plus grand élément d'un tableau, à l'échanger avec le dernier élément du tableau, et à considérer le tableau sans son dernier élément. Au bout de $n-1$ étapes (avec n la taille du tableau), le tableau est trié.
- Vous aurez à définir au moins les fonctions suivantes : `initialiseTab`, `afficheTab`, `rechercheMaxTab`, `etapeTriTab` et `triTab`.

Exercice 5 : Écrivez un programme qui :

- déclare, initialise aléatoirement et affiche un tableau de 10 valeurs,
- trie le tableau en utilisant la méthode du tri rapide,
- affiche le tableau trié.

Remarques :

- Le tri rapide consiste à choisir à chaque étape un élément pivot (arbitrairement), à ranger dans le tableau les éléments en fonction du pivot (les éléments strictement inférieurs au pivot sont avant lui, et les éléments supérieurs ou égaux sont après lui), puis à procéder récursivement sur les deux sous-tableaux se trouvant avant le pivot et après le pivot.
- Pour bien comprendre la méthode du tri rapide, il est conseillé de dessiner un exemple complet sur un tableau de 10 valeurs.
- Pour gérer des sous-tableaux, vous utiliserez le tableau principal, et vous passerez en paramètre le début et la fin du sous-tableau.
- Vous aurez à définir au moins les fonctions suivantes : `initialiseTab`, `afficheTab`, `choisirPivot`, `rangerTab`, `triSousTab` (qui prend trois arguments), `triTab` (qui ne prend qu'un argument).

Exercice 6 : Écrivez un programme qui :

- déclare quatre tableaux à deux dimensions (3×5), nommés A, B, C et D, et représentant des matrices,
- initialise A et B avec des valeurs aléatoires entre 0 et 3,
- calcule dans C la somme de A et de B,
- calcule dans D le produit de A et de B,
- affiche C et D.

Remarque : Vous aurez à définir au moins les fonctions suivantes : `initialiseMat`, `afficheMat`, `produitMat`, `sommeMat`.

Exercice 7 : Consultez rapidement les règles du jeu de dame sur Internet. Écrivez ensuite un programme qui :

- déclare un tableau de 10×10 entiers représentant un damier,
- initialise toutes les cases du tableau à 0 (qu'elles soient blanches ou noires),
- sauvegarde dans le tableau la position initiale des pièces du jeu de dames, en représentant les pions blancs par 1, les dames blanches par 2, les pions noirs par 3 et les dames noires par 4.
- affiche le damier de la manière suivante :
 - les pions blancs par 'p' et les pions noirs par 'P',
 - les dames blanches par 'd' et les dames noires par 'D',
 - les cases blanches vides par ' ' et les cases noires (qui sont forcément vides) par '# '.