

Feuille 5 : tableaux

Exercice 1 :

Écrire un programme qui lit une suite de 10 nombres réels et qui *affiche* ces 10 nombres dans l'ordre inverse de leur lecture. Il n'est pas demandé de modifier le tableau.

Exercice 2 :

1. Écrire une fonction, nommée `lectureEntier`, qui saisit (avec contrôle) un entier `n` tel que $1 \leq n \leq 20$.
2. Écrire une fonction, nommée `lectureSuite`, qui saisit dans un tableau `n` nombres entiers.
3. Écrire une fonction, nommée `moyenne`, qui retourne la moyenne des `n` éléments d'un tableau d'entiers.
4. Écrire une fonction, nommée `minimum`, qui retourne la valeur minimum des `n` éléments d'un tableau d'entiers.
5. Écrire une fonction, nommée `indMinPair`, qui retourne la position du premier entier pair d'un tableau d'entiers à `n` éléments ou -1 si tous les éléments du tableau sont impairs.
6. Écrire une fonction, nommée `variance`, qui retourne la variance des `n` éléments d'un tableau d'entiers de moyenne `moy`. La variance permet de mesurer la dispersion des valeurs par rapport à la moyenne. Soit x_1, x_2, \dots, x_n les différentes valeurs. Soit \bar{x} la moyenne des n valeurs. La variance V se calcule de la manière suivante :

$$V = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n} \quad (1)$$

7. Écrire une fonction, nommée `nbPositifsIndCons`, qui retourne le nombre maximum de valeurs strictement positives, ayant des indices consécutifs dans un tableau d'entiers à `n` éléments. Par exemple, pour $(2, 4, -1, 1, 5, 2, -1, 4, -1)$ et $n = 9$, la fonction retourne 3.
8. Écrire un programme qui successivement
 - a) saisit un entier `n`, $1 \leq n \leq 20$,
 - b) saisit, dans un tableau, `n` nombres entiers,
 - c) calcule la moyenne des éléments du tableau, puis affiche la moyenne et les éléments du tableau, avec leur indice, qui sont supérieurs à la moyenne,
 - d) calcule la valeur minimum du tableau puis affiche le minimum et tous les rangs dans le tableau où il est atteint,
 - e) affiche le premier entier pair du tableau avec sa position,
 - f) calcule et affiche la variance des éléments du tableau,
 - g) affiche le nombre maximum de valeurs consécutives du tableau qui sont strictement positives.

Exercice 3 :

1. Donner l'affichage provoqué par le programme suivant :

```
#include <stdio.h>
void echange(int, int);
int main() {
```

```

        int n=10, p=20, a=30;
        printf("avant_appele: %d %d\n", n, p);
        echange(n, p);
        printf("apres_appele: %d %d\n", n, p);
        return 0;
    }
    void echange(int a, int b) {
        int n;
        printf("debut_echange: %d %d\n", a, b);
        n = a; a = b; b = n;
        printf("fin_echange: %d %d\n", a, b);
    }
}

```

2. Écrivez une fonction `lireT`, qui lit 10 nombres réels et les stocke dans un tableau.
3. Écrivez une fonction `afficheT`, qui prend en paramètre un tableau de 10 nombres réels, et qui affiche le tableau.
4. Écrire une fonction `inverseT`, de paramètre un tableau de 10 nombres réels, qui modifie le tableau en inversant l'ordre de ses éléments.
5. Si vous exécutez `lireT`, `inverseT` puis `afficheT`, dans quel ordre les valeurs du tableau seront-elles affichées ? Est-ce qu'elles seront dans l'ordre de la saisie ou dans l'ordre inverse ?

Exercice 4 :

```

#include <stdio.h>
int main() {
    int i, j, n;
    int tab[10];
    int boites[100];
    printf("Entrez les valeurs : ");
    for (i=0; i<10; i++) {
        scanf("%d", &tab[i]);
    }
    for (i=0; i<100; i++) {
        boites[i] = 0;
    }
    for (i=0; i<10; i++) {
        boites[tab[i]] = boites[tab[i]]+1;
    }
    for (i=0; i<100; i++) {
        for (j=0; j<boites[i]; j++) {
            printf("%d ", i);
        }
    }
    printf("\n");
    return 0;
}

```

1. Exécuter le programme sur un tableau rempli avec les valeurs :

22	14	56	5	35	69	8	15	12	1
----	----	----	---	----	----	---	----	----	---
2. À quoi sert ce programme ?

3. Est-ce que ce programme peut traiter un tableau contenant des valeurs arbitrairement grandes ?
4. Est-ce que ce programme peut traiter un tableau contenant plusieurs fois la même valeur ?

Exercice 5 :

1. Écrire une fonction qui remplit une matrice d'ordre n ($n \geq 3$, passé en paramètre), la valeur de chaque coefficient devant être entière, comprise entre -9 et 9 et obtenue aléatoirement.
2. Écrire une fonction qui retourne la somme des coefficients diagonaux d'une matrice d'ordre n .
3. Écrire une fonction qui retourne la somme de tous les coefficients d'une matrice d'ordre n .
4. Écrire une fonction qui retourne la plus grande somme des coefficients d'une colonne d'une matrice d'ordre n .
5. Écrire une fonction qui retourne 1 ou 0 selon que deux lignes l_1 et l_2 (passées en paramètre) d'une matrice d'ordre n sont égales ou non.
6. Écrire une fonction qui, à partir d'une matrice A d'ordre n , construit une matrice B égale à la transposée de A .
7. Écrire une fonction qui, à partir d'une matrice A d'ordre n , construit une matrice B dont chaque coefficient b_{ij} est la moyenne du coefficient a_{ij} et des coefficients de A qui sont voisins de a_{ij} (c'est-à-dire les a_{kl} où $|k - i| \leq 1$ et $|l - j| \leq 1$). Attention "aux bords".

Exercice 6 :

On représente un échiquier par un tableau de 8 lignes et 8 colonnes. La case située à la ligne L et à la colonne C contient une dame, les autres cases sont vides. Aux échecs, une dame peut se déplacer sur la ligne horizontale, la ligne verticale, ou l'une des deux diagonales passant par sa position actuelle.

1. Écrire une fonction, nommée `lectureEntier`, qui saisit (avec contrôle) un entier n tel que $1 \leq n \leq 8$.
2. Écrire une fonction qui prend en paramètre les coordonnées x et y d'une case, ainsi que la position l et c de la dame, et qui retourne 1 si la dame peut se déplacer en (x, y) , 0 sinon.
3. Écrire une fonction, nommée `casesAccess`, qui, à partir de la position de la dame (ligne et colonne) sur l'échiquier, construit le tableau représentant l'échiquier de la façon suivante : le coefficient du tableau correspondant à la position de la dame vaut 2, ceux correspondant à des cases accessibles par la dame au coup suivant valent 1 et les autres valent 0.
4. Écrire le programme permettant de saisir la position de la dame (ligne et colonne) sur l'échiquier et d'afficher (en respectant les lignes et les colonnes) l'échiquier en plaçant un D dans la case contenant la dame, le symbole * dans les cases accessibles par la dame au coup suivant et le symbole - dans les autres cases. Compléter le programme pour que l'échiquier soit encadré par des +

Exemple : pour L=2 et C=5

```

+ + + + + + + + +
+ - - - * * * - - +
+ * * * * D * * * +
+ - - - * * * - - +
+ - - * - * - * - +
+ - * - - * - - * +
+ * - - - * - - - +
+ - - - - * - - - +
+ - - - - * - - - +
+ + + + + + + + +

```