

Série de TD N° 1
Rappel

Exercice 1:

Écrire un algorithme qui affiche la moyenne d'une suite d'entiers positifs entrés au clavier. On arrêtera la saisie quand le nombre -1 est entré, comme dans l'exemple suivant :

Entrez un entier positif : 5

Entrez un entier positif : 2

Entrez un entier positif : 3

Entrez un entier positif : -1

La moyenne de ces 3 entiers vaut 3.333333

Exercice 2:

1. Écrire un algorithme qui affiche la factorielle d'un nombre entier entré par l'utilisateur (en utilisant la boucle Pour).
2. Refaire la question 1 en utilisant la boucle TantQue et en considérant l'entier comme une entrée de l'algorithme.

Exercice 3:

- Écrire un programme qui lit un entier naturel et affiche le nombre de chiffre de cet entier.
- Écrire un programme qui a un entier m en entrée et qui affiche la somme de ses chiffres.
- Proposer un algorithme qui demande un entier n et affiche les chiffres de n en ordre inverse.

Ex : si 1593 est entré alors on affiche 3951 sera affiché.

NB. Utiliser la fonction $\text{mod}(z, n)$ qui renvoie le modulo de z par n et la fonction $\text{div}(a, b)$ qui renvoie le résultat de la division entière de a par b.

Exercice 4 :

Écrire un algorithme qui demande un entier n et affiche si cet entier est premier ou pas (**Attention : essayer optimiser le nombre d'itérations**).

Exercice 5:

Écrire un programme qui lit un entier non nul n puis affiche le résultat de la somme des n premiers carrés.

Exemple : si n = 3, l'algorithme doit afficher 14 ($14 = 1^2 + 2^2 + 3^2$).

Exercice 6: (nombre parfait)

Un nombre parfait est un entier positif supérieur à 1, égal à la somme de ses diviseurs ;

On compte 1 comme diviseur, mais on ne compte pas le nombre lui-même comme diviseur.

Exemple : 6 est un nombre parfait puisque : $6 = 3 + 2 + 1$.

Écrire un algorithme qui prend en entrée un nombre entier n et affiche si n est un entier parfait ou non.

Exercice 7 : (Tableaux)

1. Écrire un programme qui lit un tableau d'entiers naturels Tab de taille 10 et affiche si le tableau est trié dans un ordre croissant ou pas.
2. Suggérer un algorithme qui initialise deux vecteurs appartenant à R de taille maximale N et affiche leur produit scalaire.

Ex : $a = [2 \ 5 \ 3]$ et $b = [7 \ 1 \ 0]$ alors le produit scalaire de a et b est $a.b = 2*7 + 5*1 + 3*0$

3. Écrire un programme qui lit un tableau d'entier T de taille maximale N et range ensuite les éléments du tableau T dans l'ordre inverse sans utiliser de tableau d'aide. Afficher le tableau résultant.

Exercice 8: Chaînes de caractères.

1. Ecrire un programme qui lit une chaîne de caractère **ch** et un caractère **car** et affiche le nombre d'occurrence du caractère **car** dans la chaîne **ch**.
2. Ecrire un programme qui lit une chaîne de caractère **ch** et affiche si elle est palindrome ou pas.
N.B. une chaîne est palindrome si elle se lit de la même façon de gauche et de droite Ex : rotor
3. Ecrire un algorithme qui calcule le nombre de mots dans une chaîne de caractères **ch**.
Exemple: si **ch** = "A l' examen de la section SMI3" alors la fonction **NB_Mot** doit retourner 6 ("l'examen" est considéré comme un seul mot).
4. Proposer un programme qui lit une chaîne de caractères et qui supprime les espaces, s'ils existent, au début de ce cette chaîne.
Exemple 1 : « SMI3 – 2019/2020» devient «SMI3 – 2019/2020»

Exercice 9: Dans cet exercice les matrices sont supposées être des données d'entrée des algorithmes.

1. Ecrire un programme qui transforme une matrice carrée **M** en sa transposée.
2. Une matrice **Antisymétrique** est une matrice carrée opposée à sa transposée, càd $A^T = -A$

Exemples : $\begin{pmatrix} 0 & 4 \\ -4 & 0 \end{pmatrix}$ ou $\begin{pmatrix} 0 & 2 & -5 \\ -2 & 0 & 9 \\ 5 & -9 & 0 \end{pmatrix}$

Ecrire un programme qui vérifie si une matrice carrée **A** est anti-symétrique.

3. Une matrice triangulaire supérieure est une matrice carrée dont toutes les valeurs au-dessous de la diagonale principale sont nulles.

Ex : $\begin{pmatrix} 5 & 3 & 2 & 8 & 10 \\ 0 & -9 & 12 & 1 & 6 \\ 0 & 0 & 7 & 3 & 7 \\ 0 & 0 & 0 & 2 & 4 \\ 0 & 0 & 0 & 0 & 3 \end{pmatrix}$

Ecrire un programme qui vérifie si une matrice carrée **A** est triangulaire supérieure.

4. Ecrire un programme qui vérifie si la matrice **A(m,n)** est la transposée de la matrice **B(n,m)**

Exercice 8: (Expression bien parenthésée)

Une expression est dite bien parenthésée si :

- Pour toute parenthèse fermante, il existe une parenthèse ouvrante qui la précède, et qui n'a pas encore été fermée.
- Toute parenthèse ouvrante est fermée par une parenthèse fermante.

Ecrire un algorithme ayant en entrée une chaîne de caractère **ch** (tableau de caractère) de longueur **N**, qui affiche si la chaîne **ch** est bien parenthésée ou non.

Exemples :

Les expressions suivantes sont bien parenthésées :

- (3 - (2 - 9))
- 1 + 4

Les expressions suivantes ne sont pas bien parenthésées :

- ((3) : une des deux parenthèses n'a pas été fermée.
- (4) : on ferme une parenthèse qui n'a pas été ouverte.