

TD 04

Exercice 01

Écrivez un algorithme qui demande d'abord à l'utilisateur un entier N (combien de nombres il va saisir). L'algorithme doit ensuite lire N nouveaux nombres un par un, saisis par l'utilisateur. Une fois les N nombres lus, l'algorithme doit afficher la somme totale de ces nombres.

Exercice 02

Écrivez un algorithme qui demande à l'utilisateur un entier N (positif). L'algorithme doit calculer et afficher la somme de tous les nombres pairs compris entre 1 et N (inclus). Par exemple, si l'utilisateur saisit 7, l'algorithme doit afficher 12 (car $2 + 4 + 6 = 12$).

Exercice 03

Écrivez un algorithme qui demande un entier positif ou nul N . L'algorithme doit calculer et afficher la factorielle de N (notée $N!$). Rappel : $N! = N \times (N-1) \times \dots \times 2 \times 1$. Par cas particulier, $0! = 1$.

Exercice 04

Écrivez un algorithme qui demande d'abord à l'utilisateur combien de nombres il souhaite saisir (un entier N). L'algorithme doit ensuite lire ces N nombres un par un. Une fois tous les nombres saisis, il doit afficher le plus grand (maximum) et le plus petit (minimum) parmi les nombres entrés.

Exercice 05

Écrivez un algorithme qui demande à l'utilisateur de saisir des notes (réels). La saisie s'arrête lorsque l'utilisateur entre la valeur sentinelle -1. L'algorithme doit calculer et afficher la moyenne des notes saisies, mais en ignorant toute note négative (sauf le -1, qui sert à arrêter la boucle). Si aucune note positive n'a été saisie, il doit l'indiquer.

Exercice 06

Écrivez un algorithme qui demande à l'utilisateur un entier N (supérieur à 1). L'algorithme doit déterminer si N est un nombre premier (c'est-à-dire qu'il n'est divisible que par 1 et lui-même). Il doit afficher un message clair, par exemple "Le nombre est premier" ou "Le nombre n'est pas premier".

Exercice 07

Un nombre est dit "parfait" s'il est égal à la somme de ses diviseurs propres (c'est-à-dire tous ses diviseurs sauf lui-même). Par exemple, 6 est parfait car ses diviseurs propres sont 1, 2 et 3, et $1 + 2 + 3 = 6$. Écrivez un algorithme qui demande un nombre entier N et détermine s'il est parfait ou non.

Exercice 08

Écrivez un algorithme qui demande un entier positif N. L'algorithme doit calculer et afficher deux résultats distincts :

- Le nombre total de chiffres qui composent N.
- La somme de ces mêmes chiffres.

Par exemple, pour $N = 4138$, l'algorithme doit afficher "Nombre de chiffres : 4" et "Somme des chiffres : 16" (car $4+1+3+8=16$).

Exercice 09

Écrivez un algorithme qui demande un nombre entier positif (par exemple, 12345) et qui affiche le nombre "inversé" (par exemple, 54321). Votre algorithme doit construire ce nouveau nombre sans utiliser de chaînes de caractères, mais en utilisant une boucle et les opérateurs MOD et DIV.

Exercice 10

Écrivez un algorithme qui demande un entier N représentant une hauteur. L'algorithme doit dessiner une pyramide de N lignes de hauteur ou chaque ligne i se compose i (*).

Exemple pour $n = 4$:

```
*  
**  
***  
****
```

Exercice 11

Écrivez un algorithme qui demande à l'utilisateur un entier N. L'algorithme doit ensuite afficher la table de multiplication complète pour ce nombre N, de 1 à 10. L'affichage doit être formaté, par exemple : "7 x 1 = 7" "7 x 2 = 14" ... "7 x 10 = 70"

Exercice 12

Écrivez un algorithme qui demande un nombre entier N. L'algorithme doit trouver et afficher tous les nombres premiers qui existent entre 2 et N (inclus). Rappel : un nombre est premier s'il n'a que deux diviseurs, 1 et lui-même.