

Série 3

1(*) Contrôle de saisie : écrire un algorithme qui demande à l'utilisateur un nombre dans l'intervalle $[1;3]$, qui lit cette valeur et qui recommence tant que la réponse ne convient pas.

2(*) Ecrire un compte à rebours à partir d'un entier positif saisi par l'utilisateur.

3(*) Ecrire un programme qui affiche les nombres entiers pairs de l'intervalle $[0 ;N]$, N étant saisi par l'utilisateur.

4(*) A partir de la donnée d'un capital de départ **C**, d'un taux d'intérêt **I** et d'un nombre de périodes **P**, calculer et afficher le capital obtenu en appliquant le principe des intérêts composés à **C** pendant **P** périodes.

5(*) Calculer et afficher le plus petit et le plus grand nombre d'une liste de 10 nombres entiers.

6(*) Compter combien il y a de nombres inférieurs ou égaux à 15 dans une liste de 10 nombres entiers.

7(*) Ecrire un algorithme qui demande un nombre entier de départ, et qui calcule la somme des entiers à partir de 1 jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer : $1 + 2 + 3 + 4 + 5 = 15$

8(*) Ecrire un algorithme qui demande un nombre de départ (entier positif), et qui calcule sa factorielle.
Exemple : la factorielle de 5, notée $5!$, vaut $1 \times 2 \times 3 \times 4 \times 5 = 120$.

9()** L'utilisateur choisit un nombre N de notes et saisi ces N notes. A la fin de la saisie, le programme affiche la somme, la moyenne et la plus grande de ces notes. De plus, il faut assurer le contrôle de la validité de notes saisies $[1.0;6.0]$

10()** Idem mais cette fois-ci, on ne demande pas à l'utilisateur combien de notes il veut saisir, mais c'est lui qui stoppe la saisie en entrant une note invalide (hors de l'intervalle de validité).

11()** Calculer et afficher le **PGCD** de deux nombres donnés **a** et **b**.

12()** Afficher les premiers termes de la suite de Fibonacci jusqu'à l'ordre **N**. La donnée **N** doit être positive ou nulle.

La suite de Fibonacci est définie par:

$$F_0=0$$

$$F_1=1$$

$$F_k=F_{k-2}+F_{k-1} \quad \text{pour } k>1$$

13()** Afficher un échiquier 8×8 : case noire = caractère #, case blanche = caractère o.

14(*)** Un nombre premier est un entier naturel strictement supérieur à 1, admettant exactement deux diviseurs distincts: 1 et lui-même. En fonction d'un nombre entier positif saisi par l'utilisateur, trouver si ce nombre est premier ou pas et informer l'utilisateur. Vous devez aussi valider la donnée saisie par l'utilisateur. Un des objectifs de cet exercice consiste à réfléchir à des solutions d'optimisation.