

Algorithmique et structures de données 1

TP 3

Exercice 1. Ecrire un programme C qui calcul la somme de tous les nombres naturels entre 1 et n en utilisant la boucle **for**. Réécrire le programme en utilisant la boucle **do...while** et la boucle **while**. Par exemple : Entrée : 10. Sortie : Somme des nombres naturels 1-10 : 55.

Exercice 2. Ecrire un programme C qui calcul la puissance en utilisant uniquement la multiplication (utiliser la boucle **while** et la boucle **for**).

Exercice 3. Ecrire un programme C qui calcul la factorielle d'un nombre entier en utilisant la boucle **while** et la boucle **for**. Par exemple, la factorielle de 7 : $7! = 7 * 6 * 5 * 4 * 3 * 2 * 1 = 5040$.

Exercice 4. Ecrire un programme C pour afficher la table de multiplication d'un entier donné. Par exemple pour 9 :

9 x 1 = 9
9 x 2 = 18
9 x 3 = 27
9 x 4 = 36
9 x 5 = 45
9 x 6 = 54
9 x 7 = 63
9 x 8 = 72
9 x 9 = 81
9 x 10 = 90

Exercice 5. Ecrire un programme C qui permet de vérifier si un nombre est un nombre premier ou non. Utiliser les deux boucles **while** et **for**.

Exercice 6. Ecrire un programme C qui permet de trouver le PGCD (Plus Grand Commun Diviseur) de deux nombres entiers. Le PGCD est le plus grand nombre qui divise exactement les deux nombres entiers (sans reste). Par exemple, le PGCD de 54 et 24 est de 6.

Exercice 7. Ecrire un programme C qui permet d'afficher la suite de Fibonacci jusqu'à n termes, sachant que :

$$u(n) = \begin{cases} 0 & \text{si } n = 0 \\ 1 & \text{si } n = 1 \\ u(n-2) + u(n-1), & \text{si } n > 1 \end{cases}$$

Par exemple : le nombre de termes : 10. Suite de Fibonacci : 0, 1, 1, 2, 3, 5, 8, 13, 21, 34.

Exercice 8. Pour chaque suite, écrire un programme C qui permet d'afficher la somme.

- | | |
|--|---|
| 1) $1 + 1/2 + 1/3 + 1/4 + 1/5 \dots + 1/n$ | 5) $1 - 1/3 + 1/5 - \dots \pm 1/n$ |
| 2) $1/1! + 1/2! + 1/3! + \dots + 1/n!$ | 6) $x + x^2/2 + x^3/3 + \dots + x^n/n$ |
| 3) $1/1! + 2/2! + 3/3! + \dots + n/n!$ | 7) $1 + x + x^2/2! + x^3/3! + \dots + x^n/n!$ |
| 4) $1 + 1/3 + 1/5 + \dots + 1/n$ | |