

TD ALSD-1

SÉRIE DE TD N°03 (INSTRUCTIONS ITÉRATIVES : POUR, TANT-QUE & RÉPÉTER)

Exercice 01

Écrire un algorithme pour chaque question suivante :

1. Calculer et afficher la somme : $1 + 2 + 3 + 4 + \dots + n$
2. Calculer et afficher le x^n via la formule suivante : $x^n = x * x * x * \dots$ (n fois)
3. Calculer et afficher le factoriel de n ($n \geq 3$)
4. Calculer et Afficher le produit $A \times B$ sans utiliser l'opération de multiplication (A et B deux entier positifs)

Exercice N°02

Soit N une valeur entière supérieure ou égale à 10. Écrire un algorithme qui permet d'afficher toutes les valeurs positives (≥ 0) paires qui sont inférieures ou égale à N :

- Dans l'ordre croissant
- Dans l'ordre décroissant

Exercice N°03

- Écrire une algorithme qui permet de calculer la valeur approximative de π telle que

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \pm \frac{1}{n} \quad ?$$

- Modifier l'algorithme en s'arrêtant lorsque le terme $\frac{1}{i}$ soit inférieur à ε (epsilon).

Exercice N°04

1- Écrire un algorithme qui indique si un nombre entier positif est premier ou non (un nombre premier est un nombre possédant exactement deux diviseurs) ?

2- Écrire un algorithme qui affiche toutes les nombres premiers entre 1 et N ($N \geq 100$) ?

Exercice N°05

- Écrire un algorithme qui permet d'afficher tous les nombres parfaits qui sont entre 1 et N ? tel-que N est nombre entier supérieur ou égale à 10.

- Modifier l'algorithme pour qu'il affiche les nombres parfait en commençant la recherche à partir de N jusqu'à 1 ?

Exercices supplémentaires

1) Écrire un algorithme (avec l'organigramme) qui permet de trouver le PGCD (Plus Grand Commun Diviseur) de deux nombres entiers strictement positifs A et B, en utilisant cette méthode :

- i.** soit X et Y deux nombres tel-que : $X = A$ et $Y = B$;
- ii.** Si $X = Y$ alors aller à (v);
- iii.** Si $X > Y$ alors X devient $X - Y$; et Aller à (ii) ;
- iv.** Si $X < Y$ alors Y devient $Y - X$ et Aller à (ii);
- v.** PGCD de A et B = X (Fin)

Compléter l'algorithme pour calculer aussi PPCM (Plus Petit Commun Multiplicateur) de A et B ?

2) Écrire un algorithme avec le Programme C ainsi que son organigramme (Représentation graphique de l'algorithme) qui permet de rechercher tous les nombre cubiques entre 100 et 999. Un nombre cubique est un nombre est égale à la somme des ces chiffres élevé à la puissance 3. Par exemple : $153 = 1^3 + 5^3 + 3^3$

3) Soit M, valeur entière supérieure ou égale à 25, qui représente une somme d'argent (Montant). Soit a , b (aussi deux valeurs entières strictement positives) qui représentent les valeurs de deux pièces de monnaie. Écrire un algorithme qui permet de chercher et d'afficher Toutes les possibilités pour rendre la monnaie de la somme M en utilisant uniquement les deux pièces a et b (Afficher aussi, à la fin, le nombre de possibilités trouvés).

4) Écrire l'algorithme d'une fiche de paie journalière de N ouvriers ($N \geq 50$), d'une entreprise, rémunérés à la tâche. Pour cela, on donne :

- La valeur de cette rémunération par pièces réalisées VP,
- Le salaire brut (SB) est calculé selon le nombre de pièces correctes réalisées pendant la journée (NPC) comme suit :

Si NPC ≤ 100 , l'ouvrier touche $NPC * VP$

Si NPC > 100 , l'ouvrier touche $150 * VP$

- On enlève à la fin 10% du salaire pour les charges sociales (CS).

Calculer et afficher le salaire journalier brut (SB), les charges sociales (CS) et salaire journalier net (SN).

NB : Salaire brut=salaire totale ; Salaire net = salaire sans les charges sociales.