

Algorithmique

Partie 1 : Généralités

Exercice 1 : Écrire un programme qui saisit deux entiers a et b, calcule et affiche le quotient entier, le reste de la division ,le ratio (quotient réel) et modulo.

Exercice 2 Écrire un programme qui demande à l'utilisateur de donner une mesure en dm puis détermine et affiche la correspondance en m,mm,cm,hm.

Exercice 3 Écrire un programme qui demande à l'utilisateur de donner une température en celsius puis détermine et affiche la correspondance en Fahrenheit.

Exercice 4 Écrire un programme qui demande à l'utilisateur de donner le rayon d'un cercle et lui retourne sa surface et son périmètre.

Exercice 5 Écrire un programme qui demande à l'utilisateur de donner valeur d'une monnaie en CFA, puis détermine et affiche sa correspondance en DOLLAR \$ et en LIVRE STERLING £.

Exercice 6 Écrire un programme qui demande à l'utilisateur de donner la longueur et la largeur d'un rectangle et lui retourne son périmètre , sa surface et la longueur d'un des diagonales. ,

Exercice 7 : Faire un programme qui saisit 3 résistances : R1, R2 et R3.

Calculer et afficher la résistance en série : $R1 + R2 + R3$

Calculer et afficher la résistance en parallèle : $(R1 * R2 * R3) / (R1 * R2 + R2 * R3 + R1 * R3)$

Exercice 8 :

Faire un programme qui saisit les coordonnées de 2 points A (x1, y1) et b(x2, y2) et qui affiche la distance entre les 2 points.

Formule : distante = racine carrée de $((x1 - x2)^2 + (y1 - y2)^2)$

Exercice 9 : Écrire un algorithme qui donne la durée de vol en heure minute connaissant l'heure de départ et l'heure d'arrivée.

a. On considère que le départ et l'arrivé ont lieu le même jour

b. On suppose que la durée de vol est inférieure à 24 heures mais peut avoir lieu le lendemain

Exercice 10 : Écrire un algorithme qui lit un nombre de secondes puis le convertit en heure , minutes et secondes.

Exercice 11 : Écrire un algorithme qui lit les données d'un produit. Un produit est caractérisé par son libelle(chaine), sa quantité en stock(réel),le prix unitaire (réel).L'algorithme affiche les informations du produit,puis calcule et affiche :

- Le montant en stock de chaque produit ,**MStock=prix unitaire * stock**
- Le montant TTC ,**MTTC=MStock + MStock* TVA**
- **TVA=18%**

Exercice 12 : Écrire un algorithme qui lit les données d'un étudiant .Un étudiant est caractérisé par son nom(chaine),prénom (réel),date de naissance (j/mois/annee).L'algorithme affiche les informations de l'étudiant ,puis calcule et affiche l'âge.L'année en cours est considérée comme une constante.

Exercice 13: Ecrire un programme intitulé CAPACITE, qui convertit en Octets, Kilo octets, Méga octets et Giga octets un nombre donné en bits.

Partie 2 : Structures Conditionnelles

Exercice 1 :

Demander à l'utilisateur d'indiquer son choix.

S'il entre la valeur 1, calculer et afficher la résistance en série.

S'il entre la valeur 2, calculer et afficher la résistance en parallèle.

Résistance en série : $R1 + R2 + R3$

Résistance en parallèle : $(R1 * R2 * R3) / (R1 * R2 + R2 * R3 + R1 * R3)$

Exercice 2 : Écrire un algorithme calculatrice permettant la saisie du premier entier (a) de l'opération (+ ou – ou * ou / : sont des caractères) et du deuxième entier (b) et qui affiche le résultat.

Exercice 3 : Écrire un algorithme permettant de résoudre une équation du premier degré $ax + b = 0$

Exercice 4 : Écrire un algorithme permettant de résoudre une équation du second degré.
 $ax^2 + bx + c = 0$

Exercice 5 : On désire écrire un algorithme qui permet d'afficher le jour correspondant à un chiffre allant de 1 à 7 , entré au clavier. Résoudre ce problème avec deux méthodes : (si imbriquée , primitive cas).

Exercice 6 : Décomposition d'un montant en euros Écrire un algorithme permettant de décomposer un montant entré au clavier en billets de 20, 10, 5 euros et pièces de 2, 1 euros, de façon à minimiser le nombre de billets et de pièces.

Exercice 7 : Écrire un algorithme qui lit deux valeurs entières (A et B) puis les affiche dans l'ordre croissant et décroissant

Exercice 8: Écrire un algorithme qui lit trois valeurs entières (A, B et C) puis les affiche dans l'ordre croissant et décroissant.

Exercice 9: Écrire un algorithme qui lit quatre valeurs entières (A, B ,C et D) puis les affiche dans l'ordre croissant et décroissant.

Exercice 10: Faire un programme qui saisit une année puis indique si l'année est bissextile ou pas.

Exercice 11: Faire un programme qui saisit une année et un mois puis détermine et affiche le nombre de jours de ce mois dans cette année .

Exercice 12 : Faire un programme qui saisit une date (jour, mois et année) puis indique si la date est valide ou pas.

Exercice 13: Faire un programme qui saisit une date (jour, mois et année) puis détermine et affiche la date suivante.

Exercice 14: Faire un programme qui saisit une date (jour, mois et année) puis détermine et affiche la date précédente.

Exercice 15: Faire un programme qui saisit une date (jour, mois et année) puis détermine et affiche la date qu'il fera dans 5 jours.

Exercice 16: Faire un programme qui saisit une date (jour, mois et année) puis détermine et affiche la date qu'il faisait il y a N jours. N est saisi au clavier et est positif.

Exercice 17: Un journal décide de faire des remises sur les prix d'abonnement à une revue scientifique selon le menu suivant :

Ancien abonné : -15%

Etudiant : -20%

Nouvel abonné : 00%

Etranger : + 25%

Le calcul du prix d'abonnement de fait en fonction du tarif normal d'abonnement (TN) et de la qualité de l'abonné (Q). (une seule qualité est acceptée par abonné).

Écrire un algorithme permettant de calculer le prix à payer.

Exercice 18 : Écrire un algorithme qui compare deux dates représentées chacune par trois entiers

Exercice 19: Écrire un programme qui permet de saisir une valeur entière et qui convertit le temps saisi en secondes en heures, minutes et secondes. Exemple : Si temps = 3674 alors le programme affichera : 3674 s = 1 h : 1 min : 14 s

Exercice 20: Ecrire un programme qui permet de saisir le sexe (M/F), la taille (cm), et le poids (kg) d'une personne et d'afficher :

1. 1. PI, le poids idéal d'une personne, sachant que ce poids théorique est donné par la formule de Lorenz comme suit :

- a. Pour un homme : $PI = (taille - 100) - (taille - 150) / 4$

- b. Pour une femme : $PI = (taille - 100) - (taille - 120) / 4$

NB : BMI, l'indicateur d'obésité (Body Mass Index) où $BMI = \text{poids} / \text{taille}^2$ avec taille en mètre.

Une personne est considérée comme :

- Normale ($BMI \leq 27$)
- obèse ($BMI > 27$)
- Malade ($BMI \geq 32$).

Exercice 21: Écrire une analyse et un algorithme qui permet de saisir un entier puis de vérifier s'il est divisible par son chiffre d'unité. Exemple : 24 est divisible par 4 127 n'est pas divisible par 7

Remarque: prévoir le cas d'erreur (unité=0)

Exercice 22: Écrire un programme qui permet de saisir le n° de mois puis affiche la saison correspondante.

Exemple : n°mois=7 affiche été

12,1,2 saison hiver

3,4,5 saison printemps

6,7,8 saison été

9,10,11 saison automne

Exercice 23:

Écrire un programme qui permet de saisir deux dates (jour, mois et années) puis affiche la date la plus récente.

Partie 3 : Structures itératives

Exercice 1 : Un nombre est parfait s'il est égal à la somme de ses diviseurs stricts (différents de lui-même). Ainsi par exemple, l'entier 6 est parfait car $6 = 1 + 2 + 3$. Écrire un algorithme permettant de déterminer si un entier naturel est un nombre parfait.

Exercice 2 : Écrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on tape 4, l'algorithme doit calculer: $1 + 2 + 3 + 4 = 10$ Réécrire l'algorithme qui calcule cette fois la moyenne !

Exercice 3: Faire un programme qui calcule et affiche la division de a par b par soustractions successives

Exercice 4: Faire un programme qui calcule le PGCD de deux nombres saisis au clavier en utilisant l'astuce suivante: soustrait le plus petit des deux entiers du plus grand jusqu'à ce qu'ils soient égaux.

Exercice 5: Faire un programme qui calcule et affiche le PPCM de deux entiers saisis au clavier.

Exercice 6 : Écrire l'algorithme qui affiche la somme des prix d'une suite d'articles en CFA (entiers) saisies par l'utilisateur et se terminant par zéro.

Exercice 7 : Écrire un algorithme qui demande successivement 10 nombres à l'utilisateur, et qui affiche à la fin le plus grand de ces 10 nombres Et affiche aussi son rang dans la liste saisie

Exercice 8 : Écrire un algorithme mettant en œuvre le jeu suivant entre deux joueurs : Le premier utilisateur saisit un entier que le second doit deviner. Pour cela, il a le droit à autant de tentatives qu'il le souhaite. A chaque échec, le programme lui indique si l'entier est plus grand ou plus petit que sa proposition. Un score est affiché lorsque l'entier est trouvé.

Exercice 9 : On se propose de saisir N entiers différents entre 1 et 100 (N étant un entier naturel compris entre 10 et 50) puis afficher la plus longue séquence croissante tout en précisant la position du premier nombre de cette séquence.

Exercice 10: Écrire un programme qui permet de tester si deux nombres sont AMIS ou pas. Deux nombres M et N sont amis si la somme des diviseurs de N excepté 1 et lui-même est égale à M et la somme des diviseurs de M excepté 1 et lui-même est égale à N.

Exemple: les nombres 48 et 75 sont deux nombres amis puisque :

Les diviseurs de 48 sont: 2, 3, 4, 6, 8, 12, 16, 24 $\Rightarrow 2 + 3 + 4 + 6 + 8 + 12 + 16 + 24 = 75$

Les diviseurs de 75 sont : 3, 5, 15, 25 $\Rightarrow 3 + 5 + 15 + 25 = 48$

Exercice 11: Écrire un programme qui permet de saisir un nombre entier positif, le programme détermine et affiche si le nombre est premier ou pas.

Exercice 12: Écrire un programme qui permet de saisir un nombre entier positif, le programme détermine et affiche si le nombre est parfait ou pas.

Exercice 13: Écrire un programme qui permet de saisir une suite de nombres premiers, le programme s'arrête lorsque l'utilisateur tape la valeur zéro (0). Le programme affiche le nombre d'entiers saisi, le nombre de nombre premiers et affiche tous les nombre premiers.

Exercice 14: Écrire un programme qui permet de saisir un entier positif N, le programme affiche tous les nombres premiers compris entre 1 et N.

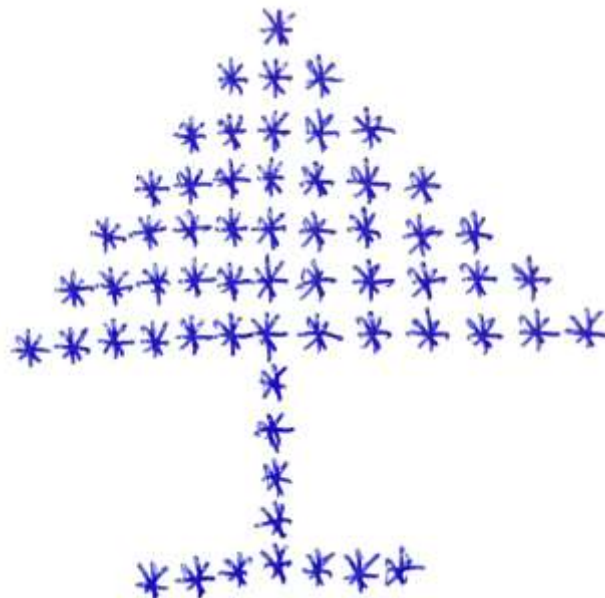
Exercice 15: Écrire un programme qui permet de saisir un entier positif N, le programme affiche les n premiers nombre premiers

Exercice 16: Écrire un programme qui permet de dessiner un sapin. le programme demande les informations suivantes avant d’afficher le sapin :

- le symbole à utiliser pour dessiner le sapin : \$,*,+ ou 0.
- la hauteur des feuilles.
- la hauteur du tronc.
- la largeur des racines.

Example:

symbole : * hauteur feuilles : 7 hauteur tronc: 4 largeur
 racine : 7



Partie 4 : Fonctions et Procédures

Reprendre les exercices ci dessous sous avec les fonctions

Exercice 1: Faire un sous programme qui reçoit une année puis indique si l'année est bissextile ou pas.

Exercice 2: Faire un sous programme qui reçoit une année et un mois puis détermine et affiche le nombre de jours de ce mois dans cette année .

Exercice 3 : Faire un sous programme qui reçoit une date (jour, mois et année) puis indique si la date est valide ou pas.

Exercice 4: Faire un sous programme qui reçoit qui saisit une date (jour, mois et année) puis détermine et affiche la date suivante.

Exercice 5: Faire un sous programme qui reçoit une date (jour, mois et année) puis détermine et affiche la date précédente.

Exercice 6: Faire un sous programme qui reçoit une date (jour, mois et année) puis détermine et affiche la date qu'il fera dans 5 jours.

Exercice 7: Faire un sous programme qui reçoit une date (jour, mois et année) puis détermine et affiche la date qu'il faisait il y a N jours. N est saisi au clavier et est positif.

Exercice 8: Faire un sous programme qui reçoit un nombre positif puis détermine la somme des diviseurs de ce nombre .

Exercice 9: Écrire un sous programme qui permet de tester si deux nombres sont AMIS ou pas. Deux nombres M et N sont amis si la somme des diviseurs de N excepté 1 et lui-même est égale à M et la somme des diviseurs de M excepté 1 et lui-même est égale à N.

Exemple: les nombres 48 et 75 sont deux nombres amis puisque :

Les diviseurs de 48 sont: 2, 3, 4, 6, 8, 12, 16, 24 => $2 + 3 + 4 + 6 + 8 + 12 + 16 + 24 = 75$

Les diviseurs de 75 sont : 3, 5, 15, 25 => $3 + 5 + 15 + 25 = 48$

Exercice 10: Faire un sous programme qui reçoit un nombre positif puis détermine si nombre est parfait ou pas.

Exercice 11: Faire un sous programme qui reçoit un nombre positif puis détermine si nombre est premier ou pas.

Exercice 12: Écrire un programme qui contient les sous programme suivants :

- Un sous programme qui permet de saisir un nombre positifs
- Un sous programme qui saisit une série de nombre positifs, la saisie s'arrête lorsque l'utilisateur tape la valeur zéro (0). Ce sous programme détermine et affiche :
 - le nombre d'entiers saisi
 - le nombre de nombre premiers
 - tous les nombres premiers.
 - produit des nombres parfaits

Exercice 13: Écrire un programme qui permet de saisir un entier positif N, le programme affiche tous les nombres premiers compris entre 1 et N. Ce programme doit être subdivisé en sous programme.

Exercice 14: Écrire un programme qui permet de saisir un entier positif N, le programme affiche les n premiers nombre premiers. Ce programme doit être subdivisé en sous programme.

Exercice 15:

On se propose de saisir N entiers différents entre 1 et 100 (N étant un entier naturel compris entre 10 et 50) puis afficher la plus longue séquence croissante tout en précisant la position du premier nombre de cette séquence. Ce programme doit être subdivisé en sous programme.

Partie 5: Tableaux

EXERCICE 1 :

Écrire un sous-programme qui, à partir d'un tableau d'entier, retourne VRAI si les éléments du tableau sont consécutifs sinon FAUX.

EXERCICE 2 :

Écrire un sous-programme qui permet de faire la fusion de deux tableaux entiers.

EXERCICE 3 :

Écrire un sous-programme qui permet d'inverser un tableau entier.

EXERCICE 4 :

Écrire un sous-programme qui, à partir de deux tableaux de même taille $N = 250$, crée un autre tableau résultant de la somme des deux premiers tableaux.

EXERCICE 5 :

Écrire un sous-programme qui, à partir de deux tableaux de même taille respective $N1 = 150$ et $N2 = 30$, retourne le résultat SOM. Pour calculer SOM, il faut multiplier chaque élément du premier tableau par chaque élément du deuxième tableau, et additionner le tout.

Exercice 6

Écrire un programme permettant de saisir, dans un tableau, des nombres compris entre -100 et +100 ensuite calcule et affiche le minimum, le maximum et la moyenne à partir du tableau.

a - le nombre d'élément du tableau est 10

b - l'utilisateur donne le nombre de valeur qu'il veut saisir

EXERCICE 7 :

Soit T un tableau qui contient n valeurs réelles triés dans l'ordre croissant. Écrire une procédure qui prend comme paramètre le Tableau T, l'entier n (la taille de T) et un nombre réel x, et elle effectue l'insertion de x dans le tableau T, de telle manière que le tableau T reste trié.

EXERCICE 8 :

Écrire un sous-programme qui permet de tester si deux nombres sont AMIS ou pas.

Deux nombres M et N sont amis si la somme des diviseurs de N excepté 1 et lui-même est égale à M et la somme des diviseurs de M excepté 1 et lui-même est égale à N.

Exemple : les nombres 48 et 75 sont deux nombres amis puisque :

Les diviseurs de 48 sont : 2, 3, 4, 6, 8, 12, 16, 24 $2 + 3 + 4 + 6 +$

$8 + 12 + 16 + 24 = 75$ Les diviseurs de 75 sont : 3, 5, 15, 25 $3 +$

$5 + 15 + 25 = 48$.

EXERCICE 9 :

Écrire un sous-programme qui détermine la position et le nombre de valeur de la sous-suite de valeurs la plus longue.

EXERCICE 10 :

Soient deux tableaux triés dans l'ordre croissant, écrire un sous-programme permettant de les fusionner. Le tableau de fusion devrait être trié aussi.

EXERCICE 11 :

Soit une matrice binaire (composée que de 0 et/ou 1). Écrire un sous-programme permettant de déterminer :

- Le numéro de la colonne contenant le plus de 0 ;

Le numéro de la ligne contenant le plus de 1 ; - Le pourcentage de 0 dans le tableau ; - Le

pourcentage de 1 dans le tableau.

EXERCICE 12 :

Soit une matrice d'entier de $N = 75$ lignes et $M = 50$ colonnes.

Écrire un sous-programme qui permet de calculer la moyenne des valeurs de chaque colonne de la matrice.

EXERCICE 13 :

Écrire un sous-programme qui permet de transférer dans un tableau, les nombre de la diagonale principale qui ne sont pas dans la diagonale secondaire.

EXERCICE 14 :

Écrire une fonction qui permet de saisir une matrice

EXERCICE 15 :

Écrire une fonction qui permet de saisir deux matrices (en utilisant l'exerce 6)et fait les opérations suivantes :

a - Somme

b - Produit

Pour chaque opération on vérifie sa faisabilité

EXERCICE 16 :

Écrire une fonction qui permet de saisir et calculer la transposé d'une matrice

Partie 6: Enregistrements

EXERCICE 1

Créer une structure « Point » contenant une abscisse (x) et une ordonnée (y)

Définissez un type « Point » pour cette structure.

Créer une fonction prenant en paramètre deux « Point » puis calcule et affiche la distance entre ces deux points.

Soit $A(X_a, Y_a)$, $B(X_b, Y_b)$, $Dist(A,B) = \sqrt{[(X_b - X_a)]^2 + [(Y_b - Y_a)]^2}$

On suppose avoir une fonction nommée *racine* qui permet de calculer la racine carré.

Partie 7: Tableaux Enregistrement

EXERCICE 1 :

Soit deux tableaux d'animaux de taille respective $N_1 = 450$ et $N_2 = 300$.

Écrire un sous-programme qui crée un nouveau tableau contenant les chevaux du premier tableau qui ne sont pas dans le second. Un animal est caractérisé par sa catégorie, son poids et sa date de naissance (jj/mm/aaaa).

EXERCICE 2 :

Soit un tableau de tableau d'artistes de taille $N = 200$.

Écrire un sous-programme qui détermine et affiche le pourcentage de présence de chaque type d'artiste dans le tableau. Un artiste est caractérisé par son nom, son prénom, sa date de naissance et son type.

EXERCICE 3 :

Écrire un sous-programme qui permet de supprimer une valeur d'un tableau des données des produits périmés dans un tableau de produits.

Un produit est caractérisé par son code, son libellé, son prix, sa quantité et sa date de péremption.

EXERCICE 4 :

Écrire un sous-programme qui permet de retourner la plus vieille femme d'un tableau de personne. Une personne est caractérisée par son nom, son prénom, son âge et son sexe.

EXERCICE 5 :

Écrire un sous-programme qui permet de faire l'union de deux tableaux d'Étudiants. Un étudiant est caractérisé par son matricule, son nom, son prénom et sa classe.