

République du Cameroun
Paix-Travail-Patrie

Ministère de l'Enseignement Supérieur

UNIVERSITE DE MAROUA

FACULTE DES SCIENCES

DEPARTEMENT DE MATHÉMATIQUES-
INFORMATIQUE



Republic of Cameroon
Peace-Work-Fatherland

Ministry of Higher Education

THE UNIVERSITY OF MAROUA

FACULTY OF SCIENCE

DEPARTMENT OF MATHEMATICS AND
COMPUTER SCIENCE

Structures des données

Code UE : INF/IGE233

Année Académique : 2023/2024

Fiche de TPE

Consigne :

- 1- C'est un travail à faire en groupe de cinq étudiants par ordre alphabétique et les thèmes pris de façon cyclique (groupe 1 => sujet 1 ; groupe 2 => sujet 2 ; ... ; groupe 5 => sujet 1 ; groupe 6 => sujet 2 ...).
- 2- Chaque groupe réalise **l'ensemble** des exercices de son sujet (algorithme et programme C correspondant)
- 3- Pour les groupes ayant le même sujet, la reproduction totale ou partielle de tout ou partie du contenu de l'un ou de l'autre **entraîne la note de 0/20** pour les groupes concernés.
- 4- Les **codes sources** sont rendus en même temps que le TPE.
- 5- Les codes portent le nom de l'exercice, le numéro du sujet et le nom du chef de groupe.
Exemple : « *exercice1_sujet4_Aminatou* », « *exercice 3_sujet4_Aminatou* », etc...
- 6- Chaque groupe doit déposer son travail (une version physique et une version numérique du travail (rapport ; codes sources)) chez le délégué au plus tard le 19 Janvier 2024 à 12 heures.
NB : Aucun travail ne sera accepté après la date indiquée !!

SUJET 1 :

Exercice 1 :

- a) Ecrire une fonction qui renvoie le nombre d'éléments d'une liste chaînée.
- b) Ecrire une fonction qui renvoie le nombre d'éléments d'une liste chaînée ayant une valeur donnée (champ Valeur)
- c) Ecrire une fonction qui vérifie si une liste chaînée est triée par valeurs croissantes du champ valeur

Exercice 2 :

En utilisant la structure suivante.

Type Tnœud = enregistrement

 info : entier ;

 gauche : Pointeur(Tnœud) ;

 droit : Pointeur(Tnœud) ;

Fin

Var Arbre : Pointeur(Tnœud) ;

Ecrire les fonctions suivantes.

- 1) Fonction qui détermine si un ABR est vide ou non.
- 2) Fonction qui retourne l'élément à la racine de l'arbre si l'arbre est non nul.
- 3) Fonction qui retourne le fils gauche de l'arbre s'il n'est pas vide
- 4) Fonction qui retourne le fils droit de l'arbre s'il n'est pas vide
- 5) Fonction qui détermine si un nœud est une feuille.
- 6) Fonction qui teste si un élément 'e' est dans l'arbre
- 7) Fonction qui retourne le minimum des valeurs de l'arbre
- 8) Fonction qui retourne le maximum des valeurs de l'arbre
- 9) Fonction qui teste si deux arbres sont égaux
- 10) Fonction qui teste si arbre est un sous-arbre d'un autre

Exercice3 :

Un établissement d'enseignement souhaite informatiser son service de scolarité. Son système d'information comprend entre autres des étudiants, des matières et une fiche de scolarité. Un étudiant se caractérise par son nom, son matricule et sa date de naissance. Une matière a un code, un intitulé, une note et un coefficient. Une fiche comprend un étudiant et ses matières. On limitera à cinq matières (algèbre, algorithmique, analyse, langageC, anglais).

Après avoir proposé des structures de données adéquates pour modéliser cette situation, écrire un programme qui permet de :

- Enregistrer la fiche de scolarité de 06 étudiants.
- Calculer et afficher la moyenne de chacun des étudiants.
- Afficher la mention de chaque étudiant.
- Afficher les étudiants ayant eu moins de 10 en algorithmique.
- L'étudiant ayant eu la plus grande moyenne

NB : [0-9] Médiocre ; [9-10] Nul ; [10-12] Passable ; [12-14] Assez-bien ; [14-16] Bien ; [16---] Très Bien

SUJET2 :

Exercice1 :

Soient L1 et L2 deux listes linéaires chaînées unidirectionnelle. Ecrire la procédure qui permet de construire la liste $L=L1-L2$ contenant tous les éléments appartenant à L1 et n'appartenant pas à L2

Exercice 2 :

1. Dessiner l'arbre binaire de recherche obtenu en insérant les éléments de la liste suivante dans leur ordre d'arrivée : 30, 40, 23, 58, 48, 26, 11, 13, 20
2. Donner le résultat des parcours préfixe, infixe et postfixe de l'arbre précédent
3. Donner l'arbre précédent après la suppression des clés suivantes : 48 puis 23.
4. Donner l'arbre précédent (avant la suppression) après l'ajout des clés suivantes : 60 puis 17.
5. Ecrire la fonction ***NB_Sup (R :Nœud, x :clé)*** qui retourne le nombre de clés supérieures à x dans l'arbre binaire de recherche de racine R.
6. Ecrire la fonction ***Val_Sup (R :Nœud, x :clé)*** qui retourne la valeur de la plus petite clé supérieure à x dans l'arbre binaire de recherche de racine R.
7. Ecrire la fonction ***Fusion(R1,R2 : Nœud)*** qui prend deux arbres binaires de recherches de racines R1, R2 et retourne la racine d'un arbre binaire de recherche contenant les deux. Utilisez les deux procédures ***Suppr(R, clé)*** et ***Inser(R, clé)*** qui permettent respectivement la suppression et l'insertion de clé dans l'arbre de racine R

Exercice3 :

Une banque souhaite informatiser son service de client. Son système d'information comprend entre autres des clients, des offres bancaires et un relevé de souscription. Un client est connu par son nom, son rib (un numéro) et son lieu de résidence. Une offre a un matricule, une durée (en nombre de jours), un nom, et une valeur. Un relevé comprend au plus 04 offres nommée offre1, offre2, etc...

Après avoir proposé des structures de données adéquates pour modéliser cette situation, écrire un programme qui permet de :

- Enregistrer un relevé de souscription pour de 10 clients.
- Affiche les clients dont la somme des valeurs des offres est supérieure a 50 000.
- Afficher tous les clients ayants souscris a l'offre1.
- Afficher les clients ayant souscrit à l'offre dont la durée est la plus courte.
- Affiche le plus gros client en termes de valeurs des offres

SUJET3 :

Exercice1 :

Ecrire les algorithmes de base suivants sur les listes linéaires chaînées unidirectionnelles.

- a) Ecrire une procédure qui insère un nouvel élément en tête d'une liste chaînée.
- b) Ecrire une procédure qui insère un nouvel élément en queue d'une liste chaînée.
- c) Ecrire une procédure qui insère un nouvel élément de sorte qu'il se trouve à une position donnée dans la liste. La position est un entier et correspond au numéro du futur élément dans la liste. Le premier élément porte le numéro 1.
- d) Ecrire une procédure qui supprime un élément d'une liste chaînée à une position donnée.
- e) Ecrire une procédure d'inversion d'une liste :
 - i) En créant une nouvelle liste
 - ii) En inversant le chainage dans la même liste.

Exercice2 :

Nous considérons la construction d'une liste des nombres premiers inférieurs ou égaux à un entier n donnée. Pour construire cette liste, on commence, dans une première phase par y ajouter tous les entiers de 2 à n en commençant par le plus grand et en terminant par le plus petit qui se trouvera à la tête de la liste. On considère ensuite successivement les éléments de la liste dans l'ordre croissant en supprimant tous leurs multiples stricts. Ecrire cet algorithme.

Exercice3 :

Un apiculteur souhaite informatiser son service de production. Son système d'information comprend entre autres des abeilles, des ruches et une banque de croisement. Une abeille se caractérise par son sexe (masculin ou féminin), son code et son patrimoine. Un patrimoine comprend le lieu d'origine de l'abeille et sa capacité de travail (entier). Une ruche a une référence, et un nom. Une banque comprend une abeille et ses potentielles ruches. Une abeille ne peut être liée à plus de 02 ruches.

Après avoir proposé des structures de données adéquates pour modéliser cette situation, écrire un programme qui permet de :

- Enregistrer une banque de croisement de 05 abeilles.
- Afficher les abeilles de sexe féminin.
- Afficher les abeilles dont le patrimoine présente une capacité de travail supérieur à 10.
- Afficher les étudiants ayant eu moins de 10 en algorithmique.
- Afficher toutes les abeilles avec le détail de leur patrimoine

SUJET4 :

Exercice 1 :

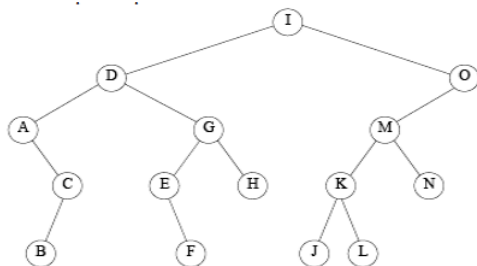
Soit les valeurs des clés triés selon l'ordre d'arrivée : 14, 23, 4, 9, 17, 11, 28, 16, 3, 7

a) Dessiner l'arbre binaire de recherche obtenu en insérant ces clés dans l'ordre d'arrivée

b) Dessiner l'arbre après la suppression des clés : 11, 17, 23, 14

Exercice 2 :

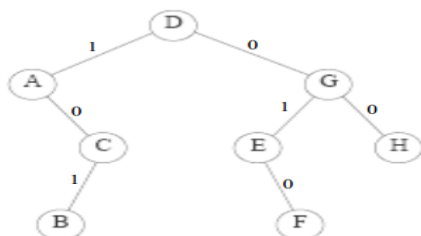
Soit l'arbre binaire de recherche suivant composé de caractères et basé sur l'ordre alphabétique :



1. Donner les chaînes de caractères obtenus en parcourant l'arbre en profondeur dans l'ordre préfixe, infixe puis postfixe.

2. Donner l'arbre après la suppression de « D » puis de « I »

On associe à chaque caractère un code binaire représentant sa position dans l'arbre binaire comme suit :



A chaque fois qu'on passe à un fils gauche, on considère un 1 et chaque fois qu'on passe à un fils droit, on considère un 0.

Il est demandé d'écrire les deux fonctions suivantes :

3. Fonction **Code(Racine: Pointeur(TNoeud), C: caractère): chaîne** ;

qui retourne le code binaire du caractère C dans une chaîne de caractères contenant des '0' et des '1'.

Exemple : Code('C') = '10', Code('F') = '010'

4. Fonction **Caract(Racine:Pointeur(TNoeud), B:chaîne):caractère**; qui retourne le caractère correspondant au code binaire donné dans la chaîne B.

Exemple : Caract('00') = 'H' ; Caract('101') = 'B'

N.B : On utilise la relation d'ordre entre les caractères alphabétiques, si C1 et C2 sont deux caractères, C1>C2 si C2 se trouve après C1 dans l'ordre alphabétique.

Exercice3 :

Un agriculteur-éleveur souhaite informatiser son activité. Son système d'information comprend entre autres des engrais, des composants et des bancs de dosages. Un engrais se caractérise par son nom, son matricule et sa date de fabrication. Un composant a un code, une teneur (entier) et une date de péremption. Un dosage comprend un engrais et des composants qui lui sont associés. Un dosage contient 04 composants nommés C1, C2, C3 et C4 pour un engrais ;

Après avoir proposé des structures de données adéquates pour modéliser cette situation, écrire un programme qui permet de :

- Enregistrer un banc de dosage de 08 engrais.
- Afficher les engrais pour lesquels le total des teneurs des composant est inférieur a 70.
- Afficher l'engrais ayant la plus petite teneur pour le composant C3.
- Afficher l'engrais ayant la plus grande teneur pour le composant C1.
- Afficher les engrais dont la teneur en composant C2 se situe entre 05 et 60