

TD/TP listes chaînées

Supposons que les éléments sont des étudiants caractérisés par un numéro, nom, prénom et date et lieu de naissance que peut être aussi une structure.

Exercice 1 : Comptage dans une liste chaînée

Ecrire une fonction qui renvoie le nombre d'éléments d'une liste chaînée.

Exercice 2 : Comptage d'occurrences dans une liste chaînée

Ecrire une fonction qui renvoie le nombre d'éléments d'une liste chaînée ayant une valeur donnée (champ Info).

Exercice 3 : Vérification d'une liste chaînée triée

Ecrire une fonction qui vérifie si une liste chaînée est triée par valeurs croissantes du champ Info.

Exercice 4 : Insertion en tête de liste chaînée

Ecrire une procédure qui insère un nouvel élément en tête d'une liste chaînée.

Exercice 5 : Insertion en queue de liste chaînée

Ecrire une procédure qui insère un nouvel élément en queue d'une liste chaînée.

Exercice 6 : Insertion à une position donnée

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.

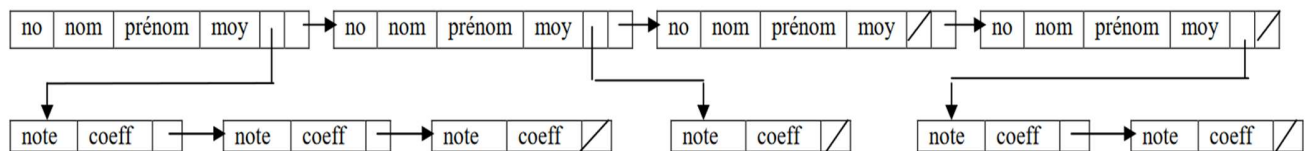
Exercice 7 : Suppression d'un élément d'une liste chaînée à une position donnée

Ecrire une procédure qui supprime un élément d'une liste chaînée à une position donnée.

Exercice 7 : Calcul de moyenne des étudiants

Le département dans lequel vous êtes inscrit souhaite gérer les notes de ses étudiants. Les étudiants ont pour identifiant leur numéro d'étudiant. Ils ont un nom et un prénom. Ces informations sont stockées dans une liste chaînée dont chaque élément comporte aussi un champ moy pour la moyenne de l'étudiant et un champ eval qui est un pointeur sur sa liste de notes. La liste de notes de chaque étudiant est aussi une liste chaînée dont la tête est le champ eval de la cellule de l'étudiant.

On suppose que tous les champs de la liste des étudiants sont remplis sauf le champ moy. On suppose que toutes les notes des étudiants et tous les coefficients sont remplis. Écrire une procédure moyennesEtudiants qui parcourt la liste des étudiants, et qui calcule et met à jour le champ moy de chaque étudiant à l'aide de la liste des notes sur laquelle pointe le champ eval. La procédure moyennesEtudiants prend en paramètre la tête de la liste des étudiants. On peut représenter cette structure par la figure ci-dessous :



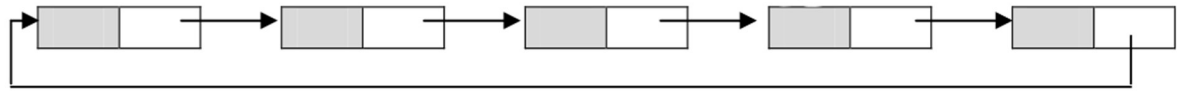
Remarques :



Cette notation équivaut à Nil

Exercice 8 : Parcours d'une liste circulaire ou anneau

Les listes circulaires ou anneaux sont des listes linéaires dans lesquelles le dernier élément pointe sur le premier. Il n'y a donc ni premier, ni dernier. Il suffit de connaître l'adresse d'un élément pour parcourir tous les éléments de la liste



Ecrire une procédure `traite_liste` qui « traite » chaque élément de la liste en appelant une procédure `traiter` qui aura comme paramètre un pointeur sur l'élément courant à traiter. La procédure `traite_liste` prend en paramètre un pointeur sur un élément quelconque de la liste. On considère que la liste contient au moins un élément (liste non vide).

Exercice 9 : Insertion d'un élément dans une liste doublement chaînée

Ecrire une procédure insérant un nouvel élément dans une liste doublement chaînée, avant l'élément de la liste ayant une valeur donnée (dans sa zone info). On dispose d'une fonction `PtV(Tête,Val)` qui renvoie l'adresse du premier élément de la liste qui porte la valeur "val", ou Nil si cette Valeur n'existe pas.

Exercice 10 : Suppression d'un élément dans une liste doublement chaînée

Ecrire une procédure supprimant, dans une liste doublement chaînée, un élément ayant une Valeur donnée (dans sa zone info). Dans les paramètres de la procédure, il doit y avoir un parameter booléen qui aura comme valeur vrai si la suppression a pu avoir lieu, faux sinon.

Exercice 11 : Suppression d'un étudiant dans une liste doublement chaînée

Ecrire une procédure supprimant, dans une liste doublement chaînée, un étudiant ayant un numéro donné. Dans les paramètres de la procédure, il doit y avoir un paramètre booléen qui aura comme valeur vrai si la suppression a pu avoir lieu, faux sinon.

