

TD N°1

Objectifs : savoir manipuler et utiliser les structures des données listes linéaires (SDLL)

Exercice 1 : Inversion d'une liste chaînée

Écrire une procédure qui permet d'inverser une liste chaînée Ls dans une liste Ls_Inv.

Exercice 2: Tri par bulles d'une liste chaînée

Écrire une procédure qui permet de trier une liste chaînée Ls dans l'ordre croissant des éléments (utiliser la méthode de tri par bulles).

Exercice 3 : Parcours d'une liste chaînée

Etudiant=struct Nom :chaîne Note :réel finStruct	Cellule =Struct Cle :Etudiant Suiv :^Cellule finStruct	Liste=struct Premier :^Cellule Dernier :^Cellule finStruct
---	---	---

On suppose qu'on a en entrée une liste d'étudiants, dont un champ s'appelle note. Calculer la moyenne de la classe (on suppose que la liste est non vide).

Exercice 4 : Insertion dans une liste chaînée

On suppose qu'on a en entrée une liste triée par ordre croissant d'entiers. Ecrivez l'action qui ajoute un entier dans la bonne position de façon que la liste reste encore triée après l'insertion.

Exercice 5 : Tri par fusion d'une liste chaînée

Proposer un algorithme qui fait la concaténation de deux listes (en entrée) retournant une liste (en sortie). Par exemple, si l'on a les listes (3, 7, 2) et (5, 4) on obtient la liste (3, 7, 2, 5, 4).

Proposer ensuite un algorithme séparant une liste en deux. Etant donné une liste en entrée, un élément sur deux va dans la première liste et un élément sur deux va dans la deuxième liste.

Faire une action de fusion de listes supposées triées par ordre croissant en une troisième liste triée.

Faire, en utilisant au choix les trois actions précédentes une action de tri fusion qui étant donné deux listes en entrée (pas forcément triées) retourne une liste triée.

Exercice 6 :

On souhaite gérer les produits d'un magasin en utilisant une liste simplement chaînée permettant l'enregistrement des informations concernant les produits (code, Nom, Date de production, Date d'expiration, prix et Quantité). Pour cela on considère les définitions suivantes :

```
date = Structure
```

```
    mois : entier
```

```
    annee : entier
```

```
Fin Structure
```

```
produit = Structure
```

```
    code : entier
```

```
    nom : chaine
```

```
    d_production : date
```

```
    d_expiration : date
```

```
    prix : réel
```

```
    qte : entier
```

```
Fin Structure
```

```
Cellule = Structure
```

```
    cle : produit
```

```
    suiv : ^ Cellule
```

```
Fin Structure
```

```
Liste_P = Structure
```

```
    premier: ^Cellule
```

```
FinStruct
```

Pour cela on vous demande d'écrire les procédures/fonctions suivantes :

1. Une fonction CREE_LISE_PRODUT qui permet de créer une liste vide de produits.
2. Une fonction SAISIE_PRODUT qui permet de saisir et retourner un produit.
2. Une procédure AJOUT_QUEUE qui permet d'ajouter des produits dans une liste chaînée de produits. **N.B** : Le nombre de produits est inconnu au départ. L'ajout s'arrête lorsque l'utilisateur répond Non à la question « Voulez-vous ajouter un autre produit 1/0 » (1=O pour Oui, 0=N pour Non).
3. Une fonction RECHERCHE permettant de rechercher un produit dans une liste de produits.
4. Une procédure AFFICHE qui permet d'afficher les produits enregistrés dans une liste de produits.
5. Une procédure SUPPRIMER_PRODUT qui permet de supprimer un produit de la liste des produits. N'oublier pas de traiter les cas particuliers (produit en tête de la liste ou en queue)
6. Une fonction COMPARER_DATE qui permet de comparer deux dates, et retourne (-1 si x avant y, 0 si x et y sont identiques , 1 si x après y).
7. Une procédure AFFICHER_EXPIRATIONS qui permet de parcourir une liste de produits et d'afficher les produits expirés par rapport à une date de référence.
8. Une procédure SUPPRIMER_TOUS, cette fonction permet de supprimer les produits expirés par rapport à une date de référence.
9. Écrire le programme principal qui permet de tester les procédures/fonction ci-dessus.