

TP 8 : Les listes simplement chaînées

Exercice1

1- Réécrire les 2 fonctions du cours ci-dessous,

- *InsererEnTete*

- *InsererEnFin*

Les nouvelles fonctions auront deux arguments en entrée: une *liste* et un *entier*.

2- Ecrire la fonction *Afficher* qui affiche le contenu d'une liste chaînée.

3- Ecrire le programme principal qui permet de remplir et afficher la liste chaînée ci-dessous.

NB: Utilisez **une seule boucle for**.

10 9 8 7 6 5 4 3 2 1 1 2 3 4 5 6 7 8 9 10

Exercice2

Créer une *structure produit* identifiée par : une *référence* (int), une *désignation* (tableau de 30 caractères), un *prix d'achat* et un *prix de vente* de type réels et une variable *Suivant* de type pointeur sur la structure produit.

Ecrire des fonctions pour réaliser les opérations suivantes:

- Ajouter un produit en tête de liste si la référence de ce produit n'existe pas déjà dans la liste. **La référence est une clé primaire**. La fonction retourne la liste après l'ajout.

Utiliser le prototype suivant: *produit* ajouter_entete(produit* list)*

- Ajouter un produit en fin de liste. La référence est une clé primaire. La fonction retourne la liste après l'ajout. *produit* ajouter_enfin(produit*list)*

- Afficher tous les produits de la liste. *void afficher_tout(produit*list)*

- Rechercher un produit par référence ou par désignation et afficher toutes ses informations s'il existe. *void rechercher_produit(produit*list)*

- Supprimer un produit de la liste sachant sa référence et quelque soit sa position dans la liste. Cette fonction retourne la liste après la suppression. *produit* supprimer(produit* list)*

- Calculer le nombre d'occurrences de la désignation d'un produit, c'ad, rechercher le nombre de produits qui partagent la même désignation.

*int nb_occurrences(produit*list, char design[])*

- Ecrire le programme principal qui appelle les fonctions précédentes via un *menu*.