

1. Structures de données	2. Les Listes chaînées	3. Les Files	4. Les Piles	5. Les Arbres	6. Les graphes
--------------------------	------------------------	--------------	--------------	---------------	----------------

#### Atelier 4 : Listes chaînées : Structure – Mémoire – Pointeurs – Fonctions

##### Rappel

- 1) Dans le cours, nous avons implémenté une liste chaînée qui contient des entiers.

Les structures nécessaires pour gérer une liste chaînée d'entiers est la suivante :

```
typedef struct Element{  
    int data ;  
    struct Element *next ;  
}Element ;
```

Pour manipuler cette liste, nous avons réalisé plusieurs fonctions dont les plus importantes sont les suivantes :

- Créer un élément de la liste (nœud) :

**Element \*createElement(int c)**

- Ajouter un élément à la liste chaînée (vide ou pleine) :

**void insertElementLast(Element \*\*List, int c)**

**void insertElementBeginning(Element \*\*List, int c)**

**void insertElementAtPosition(Element \*\*List, int c)**

- Afficher le contenu de la liste chaînée

**void displayList(Element \*List)**

- Supprimer un élément de la liste chaînée

**void deleteElement (Element \*\*List, int c)**

- 2) Pour le projet de gestion de magasin, le module de gestion des produit a été réalisé et dont la démonstration se trouve en ligne.

**Exercice 1 : Liste chaînée de points géométriques**

On souhaite mettre en place une liste chaînée de points géométriques. Chaque point est défini par son abscisse x et par son ordonnée y.

- 1) Modéliser le problème en définissant les données à gérer
- 2) Proposer deux modèles pour manipuler la liste chaînées des points. Pour chaque modèle, donner les structures de données à mettre en place
- 3) Quel est le meilleur modèle à choisir, justifier votre choix
- 4) En se basant sur les fonctions déjà réalisées pour manipuler la liste chaînée des entiers, reprendre le code pour l'adapter à la liste chaînée des points géométriques avec le minimum d'effort.
  - Ajouter une fonction createPoint(int x,int y) pour créer un point
  - Ajouter une fonction displayPoint( ? ) pour afficher le détail d'un Point
- 5) Pour tester vos fonctions, ajouter dans le main les éléments suivants :
  - a. Créer la liste chaînée \*LISTPOINTS
  - b. Ajouter quatre points dans la liste \* LISTPOINTS: (1,1) – (2,2) – (3,3) – (4,4)
  - c. Afficher le contenu de la liste sous la forme : (1,1) – (2,2) – (3,3) – (4,4)

**Exercice 2 : Liste chaînée de Segments (travailler dans le même projet de l'exercice précédent)**

On souhaite mettre en place une liste chaînée pour gérer des segments. Chaque segment est défini par deux extrémités, une extrémité est un point défini par son abscisse x et par son ordonnée y.

- 1) Proposer un modèle de données accompagné des structures à mettre en place
- 2) En se basant sur le code source de l'exercice précédent, ajouter les fonctions nécessaires pour manipuler les segments.
- 3) Pour tester vos fonctions, ajouter dans le main les éléments suivants :
  - a. Créer la liste chaînée \*LISTSEGMENTS
  - b. Ajouter quatre de votre choix à la liste\* LISTSEGMENTS en utilisant les points déjà créés
- 4) Afficher le contenu de la liste sous la forme : [ (1,1), (2,2) ] – [(3,3),(4,4)]...

**Exercice 3 : Les k-plus proches voisins avec les listes chaînées (voir version avec les tableaux)**

À partir d'une liste chaînée de points géométrique, l'objectif est de trouver les k-plus proches voisins d'un point géométrique dont les coordonnées sont lues au clavier par l'utilisateur.

**Exercice 4 : liste chaînée de lieux géographiques (étape 1 de l'application de jeu)****Réutiliser le code source de l'exercice 2**

L'objectif est de réaliser une application qui permet à des personnes de se déplacer dans un espace géographique. L'espace géographique est défini par un ensemble de lieux. Chaque lieu est défini par un point (x,y), un nom et contient un objet. L'objet a un code, une unité de cout, un signe + ou -. (les autres champs seront définis par la suite).

Pour le moment, les personnes sont définies par un numéro et une énergie.

- 1) Créer un espace géographique (une liste chaînée de lieux) ayant NOMBRELIEUX lieux. Ou NOMBRELIEUX est une constante. Initialiser les lieux par les données qu'il faut.
- 2) Créer une population (une liste chaînée de personnes) chaque personne a un numéro et une énergie initiale
- 3) Écrit un programme qui permet à une personne de consulter l'ensemble de lieux
  - a. À son passage par chaque lieu, l'énergie de la personne peut augmenter ou diminuer selon la nature du lieu (utiliser l'unité de cout et le signe pour mettre à jour l'énergie de la personne)
  - b. À la fin du parcours, afficher l'énergie au départ, l'énergie à l'arrivée et la différence.