

EXAMEN : STRUCTURES DE DONNÉES EN C
(Durée : 01h30)

Questions de cours : (6 points)

1. Qu'est-ce qu'un pointeur ?
2. Quelle est la loi d'évolution avec laquelle fonctionne une file ?
3. Quelles sont les différences entre la fonction `malloc()` et la fonction `calloc()` ?
4. Peut-on changer la taille d'un tableau statique en cours d'exécution du programme ?
5. Que se passe-t-il en mémoire avec l'instruction suivante : `malloc(sizeof(int) * 10);` ?
6. Classer les structures de données suivantes dans la colonne convenable du tableau ci-dessous : Tableau statique ; Tableau dynamique ; Liste simplement chaînée ; Liste doublement chaînée ; File ; Pile.

Structure de données linéaire	Structure de données non linéaire

Exercice 1 : (14 points)

On souhaite gérer un magasin d'allocation de voitures à l'aide d'une liste simplement chaînée. Chaque maillon (Element) de la liste contient la référence d'une voiture (RefVoiture, de type entier) sa marque (MrqVoiture, de type chaîne de caractères), son kilométrage (KlmVoiture, de type entier) et éventuellement un pointeur (Suivant) vers le prochain élément. Voici des exemples :

- RefVoiture : 10001, 10002, ...
- MrqVoiture : Toyota, Peugeot, ...
- KlmVoiture : 50098, 50099, ...

1. Définir la structure d'un élément et la structure de contrôle de la liste.
2. Définir les fonctions **Element* CreerElement()** et **Liste* CreerListe()** qui créent respectivement, un élément vide et une liste vide.
3. Écrire une fonction **void ConstruireListe(Liste* Li)** qui permet de créer une liste de voitures en rangeant ses éléments par ordre croissant du champ référence (RefVoiture).
4. Écrire une fonction **int SupprimerListeFin(Liste* Li)** qui supprime une voiture à la fin de la liste Li et retourne sa référence.
5. Écrire une fonction **void AfficherListe(Liste* Li)** qui affiche les voitures de la liste.
6. Écrire une fonction **int NbVoiture(Liste* Li)** qui retourne le nombre de voitures présentes dans le magasin.

Bonne chance