Travaux dirigés

*Langage C*

## Partie I : les structures

Un point est définit par ses coordonnées : abscisse et ordonné. Dans cet exercice on vous demande de :

1. Définir la structure permettant de définir un point.
2. Ecrire la fonction qui permet d’afficher les coordonnées d’un point.
3. Ecrire la fonction qui permet de retourner le point milieu entre deux points.
4. En utilisant les pointeurs, écrire une fonction permettant de saisir les coordonnées d’un point.
5. Définir un tableau de 5 points. Remplir et afficher ce tableau

## Partie II : les piles

**Définition :**

La pile est une structure très utilisée en informatique. La pile est une liste ordonnée sans fin d’éléments dans laquelle on ne peut introduire ou enlever un élément qu'à une extrémité appelée tête de pile ou sommet de pile. Exemple une pile de livres, une pile d’assiettes, une pile à un jeu de cartes si l’on se refuse le droit de manipuler plus d’un de leurs éléments à la fois. Noter que dans une pile, le dernier élément inséré sera le premier à être supprimé, retiré : on parle de liste LIFO ‘Last In First Out’.

**Opérations sur les piles :**

1. Une pile est une liste sur laquelle on autorise seulement 4 opérations:
2. Initialiser la pile
3. consulter le dernier élément de la pile (le sommet de la pile)
4. empiler un élément, le mettre au sommet de la pile
5. dépiler un élément (par le sommet)

**Questions**

1. Ecrire le code permettant d’implémenter une pile avec ses différentes fonctions.
2. Tester votre code en remplissant et affichant la pile.

## Partie III : Les listes chainées

**Définition :**

Une liste est un ensemble d’objets de même type constituant les éléments de la liste. Les éléments sont chaînés entre eux et on peut facilement ajouter ou extraire un ou plusieurs éléments.

Une liste est une structure de donnée telle que chaque élément contient :

* Des informations sur l’objet
* Un pointeur sur un autre élément de la liste, ou un pointeur NULL s’il n’y a pas d’élément suivant.

L’exemple ci-dessous, montre la représentation en mémoire d’une liste des mots classés en ordre alphabétique

@500

@10

@200

|  |  |
| --- | --- |
| *Arbre* | *@10* |

|  |  |
| --- | --- |
| *Bureau* | *@500* |

|  |  |
| --- | --- |
| *Ciel* | *NULL* |

En allocation dynamique, les emplacements mémoires sont dispersés en mémoire centrale. L’espace est réservée au fur et à mesure des créations des éléments de la liste. La seule limite étant la taille de la mémoire centrale.

Une liste chainée peut être représentée comme le montre la figure suivante :

**

**Opérations sur les listes :**

1. Initialiser une liste
2. Insérer un élément en début de la liste.
3. Insérer un élément en fin de la liste.
4. Afficher les différents éléments de la liste.

**Questions**

1. *Ecrire en C les structures de donnée qui permettent de :*
2. *définir un élément de la liste. On suppose qu’un élément contient une donnée de type int, et un pointeur vers un autre élément.*
3. *Définir une liste chainée*
4. *Coder les différentes fonctions d’une liste chainée.*
5. Tester votre code en remplissant et affichant la liste.