

**DEVOIR SEMESTRE 1****Matière : INFORMATIQUE****Classes : 2<sup>ème</sup> Année****Durée : 1 h****Préparation : MP, PC et PT****Exercice 1**

Le but de l'exercice est de concevoir une application pour gérer un carnet d'adresses. Celui-ci se compose de *fiches*, chacune formée d'un certain nombre de champs, nommés, comme ceci :

<u>nom du champ</u>	<u>valeur du champ</u>
nom	Salah Tounsi
âge	38
fonction	responsable du magasin
telephone	68 02 03 04
e-mail	Salah_T@gmail.com
adresse	Avenue Hédi Chaker

Chaque fiche est un dictionnaire associant les noms des champs à leurs valeurs et le carnet d'adresse sera lui-même un dictionnaire associant chaque nom à une fiche (le nom, supposé unique, représente la clé).

1. Ecrire une fonction **nouvelleFiche** permettant la création des fiches qui reçoit (vNom, vAge, vFonction, vTel, vMail, vAdresse) comme argument et retourne un dictionnaire correspondant : { 'nom': vNom, 'age': vAge, 'fonction': vFonction, 'tel': vTel, 'mail': vMail, 'adresse': vAdresse }
2. En supposant qu'un dictionnaire *carnet\_adr* représentant le carnet d'adresses est déjà prédéfini, Ecrire une fonction **saisie\_ins()** qui saisit au clavier les composants d'une fiche, crée la fiche et la rajoute au *carnet\_adr*,
3. Ecrire une fonction **cherch\_aff()** qui retrouve et affiche la fiche d'une personne à partir du nom de celle-ci,
4. Ecrire une fonction **cherch\_aff\_plus()** permettant de retrouver et afficher une information à partir de la fiche d'une personne. La fonction demandera le nom et le nom du champ désiré,
5. Ecrire un « programme principal » qui propose un menu pour choisir et exécuter une opération parmi les précédentes, genre :

Menu :

C	Créer une nouvelle fiche
A	Afficher la fiche correspondant à un nom donné
V	affiche la Valeur d'un champ donné d'un non donnée
Q	Quitter ce programme
Votre commande ?	

## Exercice 2

On désire écrire une fonction **Permut(P,n,s)** qui réalise la permutation circulaire d'une pile en indiquant le sens de la permutation ainsi que le nombre de pas à réaliser.

### **Exemple :**

Si on suppose que la pile est non bornée et qu'elle est définie par une liste L dont le premier élément désigne la taille de P et le dernier son sommet, alors :

- l'appel de la fonction **Permut(P,3,g)** transforme la liste L = [6,4,2,5,8,3,1] en [6,8,3,1,4,2,5]
- l'appel **Permut(P,2,d)** transforme la liste L = [6,4,2,5,8,3,1] en [6,3,1,4,2,5,8]

1. Ecrire une fonction **Permut\_g(P)** qui reçoit une pile P et renvoi une pile décalée d'un pas vers la gauche sans modifier la pile P.
2. Ecrire une fonction **Permut\_d(P)** qui reçoit une pile P et renvoi une pile décalée d'un pas vers la gauche sans modifier la pile P.
3. Ecrire une fonction **Permut(P,n,s)** en utilisant les deux fonctions précédentes.

**N.B. :** Pour le traitement des piles on demande d'utiliser les fonctions :

creer\_pile\_vide(), compiler(p,e) ; depiler(p) ; taille(p), est\_vide(p) ; sommet(p), pour effectuer respectivement les opérations de création et renvoi de pile vide, d'empiler un élément *e* dans la pile *p*, de dépiler une pile *p* en retournant l'élément dépilé, de retourner la taille de *p*, de vérifier si une pile *p* est vide et de retourner le sommet d'une pile *p*.