

## DS 1 - Informatique - 1h30

**Exercice 1** On dispose d'un dictionnaire contenant la hauteur (en cm) et le poids (en kg) d'un ensemble d'étudiants :

```
dico={'Cierra_Vega': [175, 70], 'Alden_Cantrell': [180, 65], 'Kierra_Gentry': [165, 68]}
```

1. Donner le type des clés et le type des valeurs de ce dictionnaire.
2. Écrire l'instruction permettant d'ajouter l'étudiant Pierre Cox (taille : 1.90m, poids : 76kg).
3. Écrire une fonction qui renvoie la taille et le poids d'un étudiant donné en argument si cet étudiant est dans le dictionnaire et qui renvoie une liste vide sinon.
4. Écrire une fonction qui renvoie la taille moyenne et le poids moyen des étudiants (le dictionnaire ne contenant pas nécessairement 4 étudiants).

### Exercice 2

1. Écrire une fonction qui crée un dictionnaire à partir d'une liste de couples (clé, valeurs) donnée en argument et renvoie ce dictionnaire.

Par exemple, si la liste donnée en argument est

```
[('yellow', 1), ('blue', 2), ('yellow', 3), ('blue', 4), ('red', 1)]
```

La fonction renvoie le dictionnaire :

```
{'yellow': [1, 3], 'blue': [2, 4], 'red': [1]}
```

2. Écrire une fonction ayant en argument un dictionnaire dont les valeurs sont des listes (comme dans la question précédente) et qui renvoie la liste des clés de ce dictionnaire dont la valeur est une liste ayant un nombre maximal d'éléments.

Dans l'exemple précédent, la fonction doit renvoyer la liste ['yellow', 'blue'].

### Exercice 3 (hachage)

On dispose des variables globales :

```
alpha='abcdefghijklmnopqrstuvwxyz'  
Alpha='ABCDEFGHIJKLMNOPQRSTUVWXYZ'
```

1. À l'aide d'une seule boucle, créer un dictionnaire `dico` dont les clés sont les lettres minuscules et majuscules et les valeurs, 0 si le rang de la lettre dans l'alphabet est pair, 1 s'il est impair.

Par exemple, `dico['a']` et `dico['A']` valent 1, `dico['b']` et `dico['B']` valent 0, etc.

2. On définit la fonction de hachage suivante. Si `nom` est une chaîne de caractères de plus de 4 lettres,  $h(nom) = 8i_0 + 4i_1 + 2i_2 + i_3$  avec  $i_j = 1$  si la lettre `nom[j]` est en position impaire dans l'alphabet, et 0 sinon.

(a) Vérifier que  $h('Allier') = 9$  et  $h('Ariege') = 11$ .

(b) Déterminer l'ensemble des valeurs possibles de la fonction de hachage. On note  $N$  la plus grande de ces valeurs.

3. Écrire une fonction `hachage` ayant pour argument une chaîne de caractères `nom` et qui renvoie  $h(nom)$ . On suppose dans cette question que la variable `nom` est une chaîne de longueur au moins 4 et qui ne contient que des lettres non accentuées.

4. On dispose d'une liste `Dept` contenant des départements français de taille inférieure ou égale à  $N$ . On veut créer une liste `L` (initialement vide) de  $N$  chaînes de caractères contenant les éléments de la liste `Dept` telle que si `nom` est un département de la liste, on doit avoir  $L[n] = nom$  où  $n = h(nom)$ .

(a) Quel problème peut se poser pour réaliser une telle liste ?

(b) Écrire un script qui réalise cette liste (si c'est possible). Le script doit signaler un problème s'il y en a un.

**Exercice 4 (Diagramme de Voronoï)**

En mathématiques, un diagramme de Voronoï est un pavage (découpage) du plan en cellules (régions adjacentes) à partir d'un ensemble discret de points appelés « germes ». Chaque cellule enferme un seul germe, et forme l'ensemble des points du plan plus proches de ce germe que d'aucun autre. La cellule représente en quelque sorte la « zone d'influence » du germe.

Le diagramme doit son nom au mathématicien russe Gueorgui Voronoï (1868-1908). Le découpage est aussi appelé décomposition de Voronoï, partition de Voronoï ou tessellation de Dirichlet

On dispose d'une liste de points et d'une liste de couleurs de même taille :

```
Germes =[[95,30],[60,50],[50,10],[20,60],[10,20]]
Couleurs=['red','yellow','green','blue','orange']
```

1. Écrire script (avec une boucle) qui affiche les points de la liste `Germes` de la couleur correspondante (rouge pour le premier, jaune pour le deuxième, etc.)
2. Écrire une fonction `distance` qui calcule la distance entre deux points du plan.
3. Écrire une fonction `plusproche(x,y)` qui détermine et renvoie la couleur du point le plus proche (parmi les points de la liste `Germes`) du point de coordonnées  $(x, y)$ .
4. Afficher tous les points de coordonnées entières et comprises entre 0 et 100 de la couleur du « germe » le plus proche.