

---

**TD 6 (programmation) - Ensembles et dictionnaires.**

---

**Exercice 1 (set et dict)**

(a) Qu'affiche le code suivant ?

```
1 dico = {0: 4, 3: 5, 5: 8, 7: 7, 8: 9}
2 s1 = set(dico)
3 print(s1)
```

(b) Créez l'ensemble `s2` des valeurs du dictionnaire `dico`.

(c) Qu'affiche le code suivant ?

```
1 s1 | s2
2 s1 & s2
3 s1 <= s2
```

**Exercice 2 (Dictionnaires)**

Que fait le programme ci-dessous ?

```
1 if __name__ == '__main__':
2     dico = dict()
3     for i in range(100):
4         dico[i] = list(range(i))
5     nb = int(input('Votre nombre:'))
6     if nb in dico:
7         print(dico[nb])
```

**Exercice 3 (Erreurs)**

Donnez la ou les erreurs dans le code suivant :

```
1 if __name__ == '__main__':
2     dico = dict()
3     dico['aaaa'] = 2
4     dico[(1, 2, 3)] = [1, 2, 3]
5     dico[[1, 2, 3]] = 5
6     s = (1, 2, 3)
7     print(s[0])
8     s[0] = 4
```

**Exercice 4 (Fréquences)**

- (a) Écrivez une fonction `freq_chaine` qui prend une chaîne de caractères en paramètre et renvoie un dictionnaire contenant pour chaque lettre de la chaîne son nombre d'occurrences. Par exemple, `freq_chaine('abbabba!')` renverra `{'a' : 3 , 'b' : 4, '!' : 1}`.
- (b) Écrivez une fonction `max_dic` qui prend en paramètre un dictionnaire (dont les valeurs sont des naturels) et renvoie la clé correspondant à la plus grande valeur. Par exemple, `max_dic({'a': 3, 'b' : 15 , 'c' : 0, 'd' : 10})` renverra `'b'`.
- (c) Écrivez une fonction `tri_cles` qui prend un dictionnaire (dont les valeurs sont des naturels) et renvoie la liste des clés du dictionnaire par ordre de valeurs décroissantes dans le dictionnaire. Par exemple, `tri_cles({'a': 3, 'b' : 5 , 'c' : 0, 'd' : 10})` renverra `['d', 'b', 'a', 'c']`. Vous ne pouvez pas utiliser `sorted` ni `list.sort`.

**Exercice 5 (Lancement de dés)**

- (a) Écrivez un programme qui demande à l'utilisateur un entier  $n$  et réalise  $n$  lancers de deux dés, que l'on simulera à l'aide de la fonction `randrange`. Le programme devra ensuite afficher chaque combinaison de chiffres obtenue avec le nombre de fois qu'elle s'est produite. On considère pour cette sous-question que deux lancers de dés donnant les mêmes chiffres mais dans un ordre différent ne sont **pas** équivalents.
- (b) Modifiez votre programme pour que les lancers équivalents soient fusionnés : on considère maintenant qu'obtenir  $a$  puis  $b$  ou  $b$  puis  $a$  au cours de deux lancers correspond au même résultat.
- (c) Vos solutions aux deux sous-questions précédentes fonctionnent-elles encore si le joueur a le droit de lancer plus de deux dés (mais toujours le même nombre) à chaque essai ? Justifiez et proposez des corrections le cas échéant.

**Exercice 6**

Écrivez `compression(dico)`, qui renvoie une représentation plus compacte d'un dictionnaire en regroupant dans des ensembles les clés ayant la même valeur et en échangeant les rôles des clés et des valeurs. Par exemple :

```
>>> compression({'t': 3, 'e': 2, 'd': 2, 'b': 1, 's': 0, 'c': 1, 'a': 1})
{0: {'s'}, 1: {'a', 'b', 'c'}, 2: {'d', 'e'}, 3: {'t'}}
```