

## Devoir Surveillé 3 : Listes, Dictionnaires, Algorithmes et Renards

L'évaluation porte sur 3 exercices indépendants.

Les exercices sont notés sur 18 et la rigueur, rédaction, justifications et en-têtes sont notés sur 2 points.

### Exercice 1 : Analyse du régime alimentaire des renards (6 points)

Des chercheurs étudient le régime alimentaire des renards en analysant le contenu de leurs estomacs. Les aliments trouvés sont enregistrés dans une liste `aliments_trouves`. Chaque élément de la liste est une chaîne de caractères représentant un type d'aliment (ex: "souris", "lapin", "baies", "insectes").

1. Écrire une fonction `compter_proies` qui prend en paramètre la liste `aliments_trouves` et un type de proie (chaîne de caractères), et renvoie le nombre de fois où cette proie a été trouvée.

```
# Exemple d'utilisation
>>> aliments = ["souris", "lapin", "souris", "baies", "souris"]
>>> compter_proies(aliments, "souris")
3
>>> compter_proies(aliments, "insectes")
0
```

2. Écrire une fonction `diversite_alimentaire` qui prend en paramètre la liste `aliments_trouves` et renvoie le nombre de types d'aliments *différents* consommés par les renards.

```
# Exemple d'utilisation
>>> aliments = ["souris", "lapin", "souris", "baies", "souris", "lapin"]
>>> diversite_alimentaire(aliments)
3
```

3. Écrire une fonction `aliments_rares` qui prend en paramètre la liste `aliments_trouves` et un seuil (entier), et renvoie la liste des types d'aliments qui apparaissent moins de `seuil` fois.

```
# Exemple d'utilisation
>>> aliments = ["souris", "lapin", "souris", "baies", "souris", "lapin", "insectes"]
>>> aliments_rares(aliments, 2)
['baies', 'insectes']
```

4. Écrire une fonction `frequence_aliment_periode` qui prend en paramètre la liste `aliments_trouves`, un type d'aliment et un entier `n`, et renvoie une liste indiquant, pour chaque période de `n` jours (ou analyses), combien de fois cet aliment spécifique a été trouvé.

```
# Exemple d'utilisation

>>> aliments = ["souris", "lapin", "souris", "baies", "souris", "lapin", "souris", "souris"]
>>> frequence_aliment_periode(aliments, "souris", 3)
[2, 1, 2] # Période 1: 2 souris, Période 2: 1 souris, Période 3: 2 souris
```

## Exercice 2 : Suivi des terriers de renards (6 points)

Vous suivez les terriers de renards dans une région. Chaque terrier est représenté par un dictionnaire avec les clés suivantes : `"id_terrier"` (entier unique), `"localisation"` (chaîne), `"espece_renard"` (chaîne, ex: "Renard roux", "Renard polaire"), `"nombre_renardeaux"` (entier), et `"actif"` (booléen indiquant si le terrier est actuellement utilisé).

1. Écrire une fonction `ajouter_terrier` qui prend en paramètre une liste de terriers et les informations d'un nouveau terrier (localisation, espèce, nombre de renardeaux), génère un nouvel ID unique, et ajoute ce terrier (actif par défaut) à la liste.

```
# Exemple d'utilisation

>>> terriers = [{'id_terrier': 1, 'localisation': 'Forêt Nord', 'espece_renard': 'Renard roux',
                 'nombre_renardeaux': 3, 'actif': True}]
>>> ajouter_terrier(terriers, 'Plaine Est', 'Renard polaire', 2)
>>> terriers # L'ID peut varier
[{'id_terrier': 1, 'localisation': 'Forêt Nord', 'espece_renard': 'Renard roux',
  'nombre_renardeaux': 3, 'actif': True},
 {'id_terrier': 2, 'localisation': 'Plaine Est', 'espece_renard': 'Renard polaire',
  'nombre_renardeaux': 2, 'actif': True}]
```

2. Écrire une fonction `recherche_par_espece` qui prend en paramètre une liste de terriers et le nom d'une espèce de renard, et renvoie la liste des terriers occupés par cette espèce.

```
# Exemple d'utilisation
>>> terriers = [
{'id_terrier': 1, 'localisation': 'Forêt Nord', 'espece_renard': 'Renard roux',
'nombre_renardeaux': 3, 'actif': True},
{'id_terrier': 2, 'localisation': 'Plaine Est', 'espece_renard': 'Renard polaire',
'nombre_renardeaux': 2, 'actif': True},
{'id_terrier': 3, 'localisation': 'Montagne Sud', 'espece_renard': 'Renard roux',
'nombre_renardeaux': 4, 'actif': False} ...]

>>> recherche_par_espece(terriers, 'Renard roux')

[{'id_terrier': 1, 'localisation': 'Forêt Nord', 'espece_renard': 'Renard roux',
'nombre_renardeaux': 3, 'actif': True}]
```

3. Écrire une fonction `desactiver_terrier` qui prend en paramètre une liste de terriers et l'ID d'un terrier, et change son statut à inactif (`actif = False`).

La fonction renvoie `True` si l'opération a réussi, `False` si le terrier est déjà inactif ou n'existe pas.

```
# Exemple d'utilisation
>>> terriers = [
{'id_terrier': 1, 'localisation': 'Forêt Nord', 'espece_renard': 'Renard roux',
'nombre_renardeaux': 3, 'actif': True},
{'id_terrier': 2, 'localisation': 'Plaine Est', 'espece_renard': 'Renard polaire',
'nombre_renardeaux': 2, 'actif': True}
... ]

>>> desactiver_terrier(terriers, 1)

True

>>> terriers[0]['actif']

False

>>> desactiver_terrier(terriers, 3) # ID inexistant

False
```

4. Écrire une fonction `renardeaux_par_espece` qui prend en paramètre une liste de terriers actifs et renvoie un dictionnaire où les clés sont les espèces de renard et les valeurs sont le nombre total de renardeaux pour chaque espèce.

```
# Exemple d'utilisation

>>> terriers_actifs = [
    {'id_terrier': 1, 'localisation': 'Forêt Nord', 'espece_renard': 'Renard roux',
     'nombre_renardeaux': 3, 'actif': True},
    {'id_terrier': 2, 'localisation': 'Plaine Est', 'espece_renard': 'Renard polaire',
     'nombre_renardeaux': 2, 'actif': True},
    {'id_terrier': 4, 'localisation': 'Colline Ouest', 'espece_renard': 'Renard roux',
     'nombre_renardeaux': 5, 'actif': True}
    ... ]

>>> renardeaux_par_espece(terriers_actifs)

{'Renard roux': 8, 'Renard polaire': 2}
```

### Exercice 3 : Refuge pour renards (6 points)

Vous gérez un refuge qui accueille différentes espèces de renards. Chaque renard est représenté par un dictionnaire avec les clés : `"id"`, `"nom"`, `"age"`, `"espece"`, `"poids"` (en kg) et `"date_arrivee"` (au format "JJ/MM/AAAA").

1. Écrire une fonction `tri_renards_age` qui prend en paramètre une liste de renards et renvoie une nouvelle liste triée par âge croissant.

Utilisez l'algorithme de tri par sélection.

```
# Exemple d'utilisation

>>> renards = [
    {'id': 1, 'nom': 'Foxy', 'age': 5, 'espece': 'Renard roux', 'poids': 6.5, 'date_arrivee': '10/01/2023'},
    {'id': 2, 'nom': 'Snowy', 'age': 2, 'espece': 'Renard polaire', 'poids': 8.0, 'date_arrivee': '15/03/2023'},
    {'id': 3, 'nom': 'Rusty', 'age': 7, 'espece': 'Renard roux', 'poids': 5.8, 'date_arrivee': '01/02/2023'}
    ... ]

>>> tri_renards_age(renards)

[
    {'id': 2, 'nom': 'Snowy', 'age': 2, 'espece': 'Renard polaire', 'poids': 8.0, 'date_arrivee': '15/03/2023'},
    {'id': 1, 'nom': 'Foxy', 'age': 5, 'espece': 'Renard roux', 'poids': 6.5, 'date_arrivee': '10/01/2023'},
    {'id': 3, 'nom': 'Rusty', 'age': 7, 'espece': 'Renard roux', 'poids': 5.8, 'date_arrivee': '01/02/2023'}
    ... ]
```

2. Écrire une fonction `recherche_renard_dichotomique` qui prend en paramètre une liste de renards triée par ID et un ID recherché, et renvoie le renard correspondant ou `None` si aucun renard n'a cet ID.

Utilisez l'algorithme de recherche dichotomique.

```
# Exemple d'utilisation
>>> renards_tries = [
{'id': 1, 'nom': 'Foxy', 'age': 5, 'espece': 'Renard roux', 'poids': 6.5, 'date_arrivee': '10/01/2023'},
{'id': 2, 'nom': 'Snowy', 'age': 2, 'espece': 'Renard polaire', 'poids': 8.0, 'date_arrivee': '15/03/2023'},
{'id': 3, 'nom': 'Rusty', 'age': 7, 'espece': 'Renard roux', 'poids': 5.8, 'date_arrivee': '01/02/2023'}
... ]
>>> recherche_renard_dichotomique(renards_tries, 2)
{'id': 2, 'nom': 'Snowy', 'age': 2, 'espece': 'Renard polaire', 'poids': 8.0, 'date_arrivee': '15/03/2023'}
>>> recherche_renard_dichotomique(renards_tries, 4)
None
```

3. Écrire une fonction `statistiques_especes` qui prend en paramètre une liste de renards et renvoie un dictionnaire contenant pour chaque espèce : le nombre de renards, l'âge moyen et le poids moyen.

```
# Exemple d'utilisation
>>> renards = [
{'id': 1, 'nom': 'Foxy', 'age': 5, 'espece': 'Renard roux', 'poids': 6.5, 'date_arrivee': '10/01/2023'},
{'id': 2, 'nom': 'Snowy', 'age': 2, 'espece': 'Renard polaire', 'poids': 8.0, 'date_arrivee': '15/03/2023'},
{'id': 3, 'nom': 'Rusty', 'age': 7, 'espece': 'Renard roux', 'poids': 5.8, 'date_arrivee': '01/02/2023'}
... ]
>>> statistiques_especes(renards)
{'Renard roux': {'nombre': 2, 'age_moyen': 6.0, 'poids_moyen': 6.15},
'Renard polaire': {'nombre': 1, 'age_moyen': 2.0, 'poids_moyen': 8.0}
... }
```

4. Écrire une fonction `renards_a_surveiller` qui prend en paramètre une liste de renards et renvoie la liste des renards dont le poids est inférieur à 3 kg ou supérieur à 10 kg.

```
# Exemple d'utilisation
>>> renards = [
{'id': 1, 'nom': 'Foxy', 'age': 5, 'espece': 'Renard roux', 'poids': 6.5, 'date_arrivee': '10/01/2023'},
{'id': 2, 'nom': 'Snowy', 'age': 2, 'espece': 'Renard polaire', 'poids': 11.0, 'date_arrivee': '15/03/2023'},
{'id': 3, 'nom': 'Rusty', 'age': 7, 'espece': 'Renard roux', 'poids': 5.8, 'date_arrivee': '01/02/2023'},
{'id': 4, 'nom': 'Tiny', 'age': 1, 'espece': 'Fennec', 'poids': 2.5, 'date_arrivee': '20/04/2023'}
... ]
>>> renards_a_surveiller(renards)
[
{'id': 4, 'nom': 'Tiny', 'age': 1, 'espece': 'Fennec', 'poids': 2.5, 'date_arrivee': '20/04/2023'},
{'id': 2, 'nom': 'Snowy', 'age': 2, 'espece': 'Renard polaire', 'poids': 11.0, 'date_arrivee': '15/03/2023'}
... ]
```

Question bonus : Modifier la fonction `renards_a_surveiller` pour qu'elle renvoie une liste triée par date d'arrivée.