

## 2026 sujet 11

**Question 1**

```
def distance(hab_1, hab_2):
    """
    Calcule la distance euclidienne entre deux habitats.
    entrée :
        - habitat_1 : dictionnaire représentant un habitat.
        - habitat_2 : dictionnaire représentant un autre habitat.
    sortie :
        - float : distance euclidienne entre habitat_1 et habitat_2.
    """
    # On calcule d'abord les carrés
    v = (hab_1['vegetation'] - hab_2['vegetation'])**2
    p = (hab_1['proximite_eau'] - hab_2['proximite_eau'])**2
    u = (hab_1['densite_urbaine'] - hab_2['densite_urbaine'])**2
    d = (hab_1['disponibilite_proies'] - hab_2['disponibilite_proies'])**2
    return sqrt(v + p + u + d)
```

**Question 2**

On suppose qu'il faut donner le dictionnaire de l'habitat de la liste.

```
def distance_d_un_habitat(habitat, habitats):
    """
    Calcule la distance entre un habitat et chaque habitat de la liste.
    entrée :
        - habitat : dictionnaire représentant un habitat.
        - habitats : liste de dictionnaires représentant des habitats.
    sortie :
        - list[tuple] : liste de tuples (distance, habitat) où distance est la distance
    """
    liste = []
    for habitat_2 in habitats:
        liste.append((distance(habitat, habitat_2), habitat_2))
    return liste
```

**Question 3**

On teste la fonction distance\_d\_un\_habitat :

```
>>> distance_d_un_habitat(nouveau, zones_connues)[:3]
[(7.211102550927978, {'vegetation': 9, 'proximite_eau': 6,
'densite_urbaine': 0, 'disponibilite_proies': 4, 'presence_renard': True}),
(8.660254037844387, {'vegetation': 10, 'proximite_eau': 5,
'densite_urbaine': 9, 'disponibilite_proies': 10, 'presence_renard': False}),
(5.196152422706632, {'vegetation': 8, 'proximite_eau': 5,
'densite_urbaine': 1, 'disponibilite_proies': 6, 'presence_renard': False})]
```

#### Question 4

Il faut changer le distance en caracteristiques dans le `if`:

```
def presence_renard(k, habitat, habitats):
    """
    Vérifie si l'habitat donné a plus de k/2 voisins avec des renards.
    entrée :
        - k : entier représentant le nombre d'habitats à considérer.
        - habitat : dictionnaire représentant un habitat.
        - habitats : liste de dictionnaires représentant des habitats.
    sortie :
        - bool : True si l'habitat a plus de k/2 voisins avec des renards, False sinon.
    """
    habitats = k_plus_proches(k, habitat, habitats)
    n_renards = 0
    for habitat in habitats:
        distance = habitat[0]
        caracteristiques = habitat[1]
        if caracteristiques['presence_renard']: # ici !
            n_renards += 1
    return n_renards > k/2
```

#### Question 5

Pour k égal à 1 ou 2, on trouve une majorité de renards. Un renard pourrait alors s'y installer. Mais pour k plus grand que 2, il n'y a pas de renards dans la majorité des habitats. Ainsi un renard ne pourrait pas s'y installer.

```
>>> presence_renard(1, nouveau, zones_connues)
False
>>> presence_renard(2, nouveau, zones_connues)
False
>>> presence_renard(3, nouveau, zones_connues)
True
>>> presence_renard(5, nouveau, zones_connues)
True
>>> presence_renard(10, nouveau, zones_connues)
True
```