

2026 sujet 7

Question 1

```
c1 = Coccinelle("femelle", 10, 2)
c2 = Coccinelle("femelle", 10, 2)
c3 = Coccinelle("male", 10, 2)

population = [c1, c2, c3]
nb_proies = 200

for i in range(5):
    population, nb_proies = evolution(population, nb_proies)
    print("Jour ", i + 1, " : ", len(population), " coccinelles et ", nb_proies, " pucerons")
```

La sortie est :

```
Jour 1 : 7 coccinelles et 187 pucerons
Jour 2 : 7 coccinelles et 105 pucerons
Jour 3 : 11 coccinelles et 46 pucerons
Jour 4 : 11 coccinelles et 0 pucerons
Jour 5 : 11 coccinelles et 0 pucerons
```

Question 2

```
population = [c1, c2, c3]
nb_proies = 2000

def simulation_simple(population, nb_proies):
    jour = 0
    while len(population) > 0 and nb_proies > 0 and jour < 30:
        population, nb_proies = evolution(population, nb_proies)
        jour = jour + 1
    return len(population), nb_proies, jour

print(simulation_simple(population, nb_proies))
```

La fonction renvoie :

```
(239, 0, 14)
```

Question 3

```

def chasser(self, nb_proies, nb_coccinelles):
    """
    Simule la consommation des pucerons par la coccinelle
    Calcule le nombre de pucerons en fonction de la consommation
    Ajuste le niveau de nutrition de la coccinelle en fonction de sa
    consommation

    nb_proies est le nombre de pucerons
    nb_coccinelles est le nombre de coccinelles

    Cette méthode renvoie le nombre de pucerons après la consommation
    de la coccinelle.
    """
    # S'il n'y a pas de coccinelle, il n'y a pas de chasse
    if nb_coccinelles == 0:
        return nb_proies

    # Évaluation de la consommation de la coccinelle en fonction
    # du nombre de pucerons disponibles par coccinelle
    proies_par_cocci = nb_proies / nb_coccinelles

    if proies_par_cocci > 20:
        consomme = random.randint(12, 20)
    elif proies_par_cocci > 10:
        consomme = random.randint(8, 15)
    else:
        consomme = random.randint(3, 8)

    consomme = min(consomme, nb_proies)

    # Si la coccinelle a suffisamment mangé son niveau de
    # nutrition augmente, sinon il diminue.
    if consomme >= 10:
        self.niv_nutrition += 1
    else:
        self.niv_nutrition = max(0, self.niv_nutrition - 1)

    return nb_proies - consomme

```

Question 4

On ajoute la condition sur l'âge dans reproduction et la chance de mourir dans a_survecu.

```

def reproduction(self):
    """
    Une femelle avec un niveau de nutrition >= 2 engendre exactement
    deux descendants : un mâle et une femelle.
    """
    descendants = []
    if self.sexe == "femelle" and self.niv_nutrition >= 2 and self.age >= 20:
        descendants.append(Coccinelle("male", 0, 0))

```

```

    descendants.append(Coccinelle("femelle", 0, 0))
    self.niv_nutrition = 0

    return descendants

def a_survecu(self):
    """
    Met à jour l'âge de la coccinelle et indique si elle est encore en vie.
    """
    self.age = self.age + 1
    # 1 chance sur 3 de mourrir de faim
    if self.niv_nutrition == 0:
        if random.randint(1,3) == 1:
            return False
    return self.age < self.esperance_de_vie

```

La sortie est alors :

```
(79, 294408, 30)
```

On voit que les pucerons prennent complètement le dessus et la population de coccinelle est moindre que dans la simulation précédente.