2025 sujet 28

Exercice 1

```
def a_doublon(tab):
    for i in range(len(tab)-1):
        if tab[i] == tab[i+1]:
            return True
    return False
```

Exercice 2

```
def voisinage(n, ligne, colonne):
    """ Renvoie la liste des coordonnées des voisins de la case
    (ligne, colonne) en gérant les cases sur les bords. """
    voisins = []
    for l in range(max(0,ligne-1), min(n, ligne+2)):
        for c in range(max(0, colonne-1), min(n, colonne+2)):
            if (l, c) != (ligne, colonne):
                voisins.append((l,c))
    return voisins

def incremente_voisins(grille, ligne, colonne):
    """ Incrémente de 1 toutes les cases voisines d'une bombe."""
    voisins = voisinage(len(grille), ligne, colonne)
    for l, c in voisins:
        if grille[l][c] != -1: # si ce n'est pas une bombe
            grille[l][c] = grille[l][c] + 1  # on ajoute 1 à sa valeur

def genere_grille(bombes):
    """ Renvoie une grille de démineur de taille nxn où n est
    le nombre de bombes, en plaçant les bombes à l'aide de
    la liste bombes de coordonnées (tuples) passée en
    paramètre. """
    n = len(bombes)
    # Initialisation d'une grille nxn remplie de 0
    grille = [[0 for colonne in range(n)] for ligne in range(n)]
    # Place les bombes et calcule les valeurs des autres cases
    for ligne, colonne in bombes:
        grille[ligne][colonne] = -1 # place la bombe
        incremente_voisins(grille, ligne, colonne)  # incrémente ses voisins
    return grille
```