Correction sujet épreuve pratique bac NSI

Exercice 1 :

```
def a_doublon(tab):
    """Renvoie True si tab contient au moins deux éléments identiques, False sinon."""
    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 coordonnÃ©es des voisins de la case
    (ligne, colonne) en gÃ©rant 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):
    """ IncrÃ©mente 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] += 1  # on ajoute 1 Ã  sa valeur


def genere_grille(bombes):
    """ Renvoie une grille de dÃ©mineur de taille nxn oÃ¹ n
est
    le nombre de bombes, en plaÃ§ant les bombes Ã  l'aide
de
    la liste bombes de coordonnÃ©es (tuples) passÃ©e en
    paramÃ¨tre. """
    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)  # incrÃ©mente ses voisins
    return grille
```