

Corrigé sujet **04** - Année : 2023

Sujet 04 - 2023 ↓

Exercice 1

```
1 def a_doublon(liste_triee):
2     for i in range(len(liste_triee)-1):
3         if liste_triee[i] == liste_triee[i+1]:
4             return True
5     return False
```

Note

- La liste est supposée **triée**, donc si un doublon est présent c'est que deux nombres consécutifs sont égaux.
- On parcourt la liste jusqu'à l'*avant-dernier* élément (le dernier n'a pas de successeur)

Exercice 2

```
1 def voisinage(n, ligne, colonne):
2     """ Renvoie la liste des coordonnées des voisins de la case (ligne,
3     colonne) en gérant les cases sur les bords. """
4     voisins = []
5     for l in range(max(0, ligne-1), min(n, ligne+2)):
6         for c in range(max(0, colonne-1), min(n, colonne+2)):
7             if (l, c) != (ligne, colonne):
8                 voisins.append((l,c))
9     return voisins
10
11 def incremente_voisins(grille, ligne, colonne):
12     """ Incrémente de 1 toutes les cases voisines d'une bombe. """
13     voisins = voisinage(len(grille), ligne, colonne)
14     for l, c in voisins:
15         if grille[l][c] != -1: # si ce n'est pas une bombe
16             grille[l][c] = grille[l][c] + 1 # on ajoute 1 à sa valeur
17
```

```

18
19 def genere_grille(bombes):
20     """ Renvoie une grille de démineur de taille nxn où n est le nombre
21     de bombes, en plaçant les bombes à l'aide de la liste bombes de
22     coordonnées (tuples) passée en paramètre. """
23     n = len(bombes)
24     # Initialisation d'une grille nxn remplie de 0
25     grille = [[0 for colonne in range(n)] for ligne in range(n)]
26     # Place les bombes et calcule les valeurs des autres cases
27     for ligne, colonne in bombes:
28         grille[ligne][colonne] = -1# place la bombe
           incremente_voisins(grille,ligne,colonne)# incrémente ses voisins
       return grille

print(genere_grille([(1, 1), (2, 4), (3, 1), (3, 3), (4, 4)]))

```

Attention

Dans le pdf, La fonction `genere_grille` s'écrit parfois avec des accents (`gènère_grille`) et parfois sans.