

23 NSI 04

```
In [ ]: bombes = [(1, 1), (2, 4), (3, 1), (3, 3), (4, 4)]
grille_test = [[1, 1, 1, 0, 0], [1, -1, 1, 1, 1], [2, 2, 3, 2, -1], [1, -1, 2, -1, 3], [1, 1, 2, 2, -1]]
```

```
In [ ]: 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
```

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

```
In [ ]: 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] = ... # place la bombe
        ... # incrémente ses voisins

    return grille
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js