

Enonce

On souhaite générer des grilles du jeu de démineur à partir de la position des bombes à placer.

On se limite à la génération de grilles carrées de taille $n \times n$ où n est le nombre de bombes du jeu.

Dans le jeu du démineur, chaque case de la grille contient soit une bombe, soit une valeur qui correspond aux nombres de bombes situées dans le voisinage direct de la case (au-dessus, en dessous, à droite, à gauche ou en diagonale : chaque case a donc 8 voisins si elle n'est pas située au bord de la grille).

Voici un exemple de grille 5×5 de démineur dans laquelle la bombe est représentée par une étoile :



On utilise une liste de listes pour représenter la grille et on choisit de coder une bombe par la valeur -1.

L'exemple ci-contre sera donc codé par la liste :

Script Python

```
[[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]]
```

Compléter le code suivant afin de générer des grilles de démineur, on pourra vérifier que l'instruction

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

produit bien la liste donnée en exemple.

Script Python

```
1 def voisinage(n, ligne, colonne):
2     """ Renvoie la liste des coordonnées des voisins de la case
3     (ligne, 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
12 def incremente_voisins(grille, ligne, colonne):
13     """ Incrémente de 1 toutes les cases voisines d'une bombe."""
14     voisins = ...
15     for l, c in voisins:
16         if grille[l][c] != ...: # si ce n'est pas une bombe
17             ... # on ajoute 1 à sa valeur
18
19
20 def genere_grille(bombes):
21     """ Renvoie une grille de démineur de taille nxn où n est
```

```
23     le nombre de bombes, en plaçant les bombes à l'aide de
24     la liste bombes de coordonnées (tuples) passée en
25     paramètre. """
26     n = len(bombes)
27     # Initialisation d'une grille nxn remplie de 0
28     grille = [[0 for colonne in range(n)] for ligne in range(n)]
29     # Place les bombes et calcule les valeurs des autres cases
30     for ligne, colonne in bombes:
31         grille[ligne][colonne] = ... # place la bombe
32         ... # incrémente ses voisins
33
34     return grille
```