

Le Binaire est un casse-tête logique qui se joue sur une grille de jeu carrée de taille paire pré-remplie. L'objectif est de remplir chaque case avec le symbole 0 ou le symbole 1, en respectant les règles suivantes :

- chaque ligne et chaque colonne contiennent autant de 0 que de 1 (**règle de comptage**) ;
- on ne peut pas placer strictement plus de deux symboles 0 ou deux symboles 1 l'un à côté de l'autre ou l'un en dessous de l'autre (**règle de positionnement**) ;
- on ne peut pas avoir deux lignes ou deux colonnes identiques.

Voici un exemple de grille de taille 10 résolue :

1	1	0	1	0	1	0	1	0	0
0	1	0	0	1	1	0	0	1	1
0	0	1	0	1	0	1	0	1	1
1	0	1	1	0	1	0	1	0	0
1	1	0	0	1	1	0	1	0	0
0	0	1	1	0	0	1	0	1	1
0	1	1	0	1	0	1	0	1	0
1	0	0	1	0	1	0	1	0	1
0	0	1	0	1	0	1	1	0	1
1	1	0	1	0	0	1	0	1	0

Figure 1: Grille de Binaire résolue

NB Les grilles de taille 10 se trouvent couramment dans les jeux en ligne mais sont bien trop grosses pour être utilisées comme grilles de test ! Pour les tests vous pourrez utiliser des grilles de taille 4 (et pour les fonctions auxiliaires des listes de taille 4).

Vous êtes libre de programmer le jeu comme bon vous souhaite, la base étant :

- charger une grille initiale
- remplir la grille tant qu'il reste des cases vides
- valider la grille en appliquant les règles ci-dessus

Pour commencer on ne distinguera pas les cases pré-remplies des cases remplies par le joueur et on optera pour un affichage basique, pas exemple :

```
+---+---+---+---+
| 1 | 1 |   | 0 |
+---+---+---+---+
|   | 0 |   |   |
+---+---+---+---+
|   |   | 0 |   |
+---+---+---+---+
|   | 1 |   | 0 |
+---+---+---+---+
```

Vous pouvez décider de valider la grille une fois qu'elle est complètement remplie ou interdire de placer un symbole si le voisinage sur la ligne et la colonne invalide les règles de comptage ou de positionnement...

La configuration initiale doit contenir la taille de la grille, ainsi que le contenu des cases pré-remplies. Vous pouvez dans un premier temps écrire un module `config_binairo.py` qui contient les configurations sous la forme de structures de donnée Python. Voilà un exemple qui produira l'affichage ci-dessus.

```
grille1 = (4, [(0, 0, 1), (0, 1, 1), (0, 3, 0), (1, 1, 0), (2, 2, 0), (3, 1, 1), (3, 3, 0)])
```

ou encore (selon vos préférences de typage des symboles) :

```
grille1 = (4, [(0, 0, '1'), (0, 1, '1'), (0, 3, '0'), (1, 1, '0'), (2, 2, '0'), (3, 1, '1'), (3, 3, '0')])
```

puis écrire à la main dans la fonction principale la grille avec laquelle on souhaite jouer :

```
from config_binairo import grille1
...
def jouer():
    grille = init_grille(grille1)
    ...
```

N'oubliez pas de nommer vos symboles via des constantes, par ex

```
ZERO = ...
UN = ...
VIDE = ...
```