TP 3 - Parallelisation

1 - Mettre les cellules dans une Dish

Créer une classe `Dish`, qui remplace totalement la fonction `main`

Ajouter à la fonction `__init__` tout le code de l'initialisation de l'algorithme (création des cellules, etc ...)

Ajouter aux arguments de `__init__` tous les paramètres de l'algorithme (nombre de cellules, de dimensions, etc ...)

Créer une méthode `run` contenant une boucle infinie appelant en boucle la fonction `new_generation`, contenant le code pour chaque génération

Supprimer l'ancienne fonction `main`

2 - Héritage de Thread

Faire hériter `Dish` de `Thread`

Appeler `self.start` à la fin de `__init__` (démarrer automatiquement le thread)

Ajouter un `Event` en argument de `__init__`

Dans `run`, attendre que l'event soit `True` avant de commencer la boucle

Modifier la boucle infinie pour qu'elle s'arrête si l'event devient `False`

3 - Gestion du processus

En bas du code, instancier 2 objets `Dish`

Attendre 2 secondes (`time.sleep(2)`) et autoriser leur démarrage

Attendre 10 secondes après leur démarrage, et demander leur fermeture

Gérer les KeyboardInterrupt

```
try:
  # Mon code
  pass
except KeyboardInterrupt:
  # Ce qu'on doit faire en cas d'interrupt
  # Ici, on veut demander l'extinction propre de nos threads
  pass
```

4 - Se passer des messages

Créer une `Queue`, l'ajouter aux arguments de `__init__`

À chaque fin de génération, envoyer un dictionnaire contenant les données de cette génération

```
data = {
  "nb_generation": nbgen,
  "best_output": best_cell.output,
  "best_genome": best_cell.genome,
}
```

Supprimer le code qui affiche la ligne de résultats dans `Dish`

5 - Algorithme

Créer un algorithme principal suivant ce principe:

```
from queue import Queue, Empty

# Initialisation

tstart = time.time()   # Nombre de secondes depuis 1 Janvier 1970
while (time.time() - tstart) > 10: # Si ça fait moins de 10 secs qu'on a commencé

  try:
    # Récupérer données depuis la Queue
  except Empty: # Si aucune donnée dans la queue
    pass # On continue comme si de rien n'était
  except KeyboardInterrupt:
    # Code demandant la fermeture propre des threads
    pass

  # Afficher les données reçues

print("Fini !")
```