

# TD6 - Structures de données

## 1. Vérification syntaxique

En utilisant la classe `deque` de la bibliothèque `collections`, écrire un programme qui vérifie si les éléments suivants sont correctement utilisés: `() [] {}`.

Le programme affichera `syntaxe OK` si toutes les parenthèses, accolades et crochets ouverts sont bien fermés.

Par exemple :

```
Le (petit) loup est entré dans la bergerie. -> OK
La 3ème lettre d'une chaîne (de caractères (ou d'un tableau)) est accessible en écrivant tab[3]. -> OK
Ceci est un texte entre parenthèses : (et un groupe d'accolades {})). -> KO
```

### Pour aller plus loin...

Améliorer le programme pour qu'il vérifie la syntaxe et supprime tous les commentaires d'un code en utilisant la pile.

Un commentaire commence par `/*` et termine par `*/`.

## 2. Patate chaude

N joueurs sont placés en cercle, ils se passent une pomme de terre tout droit sortie du four.

Au début de chaque tour, on détermine aléatoirement un nombre de passes (entre 1 et 10).

Lorsque ce nombre tombe à 0 le joueur qui tient la patate est éliminé. La pomme de terre est toujours passée au voisin de droite.

Ecrire une fonction `elimination` qui prend en entrée une liste de joueurs (noms) et affiche l'évolution de la liste à chaque élimination.

On affichera également le nom du gagnant.

### Pour aller plus loin...

Le compteur représente désormais un nombre de secondes (5 à 60). Chaque joueur passera la patate à son voisin au bout de X secondes (1 à 4).

## 3. Catalogue des cours

On souhaite gérer la liste des UV de l'UTC. Chaque UV dispose d'un code unique au format court (<5 caractères).

On veut gérer le catalogue des UVs de manière à pouvoir récupérer rapidement la description, l'effectif et le responsable.

Définir une classe `Catalogue` qui aura pour méthodes :

- `ajouter_cours` prenant en paramètres le code court, la description, l'effectif et le responsable. Attention si le code court existe déjà, on affichera une erreur.
- `supprimer_cours` prenant en paramètre le code court de l'UV
- `recuperer_cours` qui prend en paramètre le code court et renvoie le détail du cours
- `modifier_cours` qui prend en paramètre obligatoire le code court et en optionnels la description, l'effectif et le responsable. Un paramètre non renseigné est ignoré.
- `changement_code_cours` qui prend en paramètres l'ancien code court et le nouveau
- `lister_cours` qui affiche la liste des cours avec leur description

Attention au choix de la structure de données pour le stockage des cours.

### Pour aller plus loin...

- Améliorer la méthode `lister_cours` en ajoutant un paramètre optionnel `asc`.  
Si `asc` vaut 1, la liste devra être affichée par code d'UV croissant.  
S'il vaut -1, elle sera affichée dans l'ordre décroissant.  
Dans tous les autres cas, l'ordre ne comptera pas.