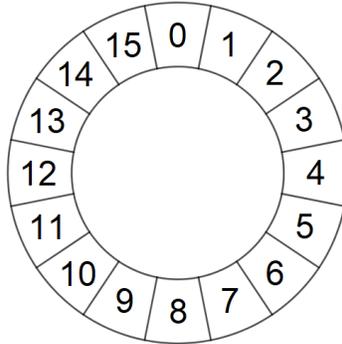


# Projet NSI : *Tourner numériquement en rond*

Dans ce projet, on s'intéresse au "jeu" suivant :

On considère un plateau de jeu circulaire qui est constitué de 16 cases numérotées de 0 inclus à 16 exclu comme ci-dessous :



Au départ, le joueur (c'est un jeu qui peut se jouer seul) place son pion sur la case 0. A chaque "coup", il lance un dé à six faces et avance son pion d'autant de cases que le nombre indiqué par le dé (entre 1 et 6 inclus) dans le sens des aiguilles d'une montre.

Par exemple, s'il obtient 3 au premier lancer, il pose son pion sur la case 3, puis s'il obtient 6 au deuxième lancer, il le pose sur la case 9, puis s'il obtient 5, il pose le pion sur la case 14, puis s'il obtient à nouveau 5, il pose le pion sur la case 3 (sur laquelle il a déjà précédemment posé son pion), ... et ainsi de suite.

Le jeu se termine lorsque le joueur a posé son pion au moins une fois **sur chacune des cases** de la piste.

## Réflexion et outils :

- Pour simuler le lancer de dé, on pourra utiliser la fonction `randint` du module `random`) qu'il faudra commencer par importer :

```
from random import randint
```

- La `position` (`int`) du pion sur la piste se comportera comme un *accumulateur* initialement nul et qui augmentera à chaque coup de la valeur du lancer de dé. Si la `position` dépasse 15, il faudra diminuer la valeur de `position`... (faire quelques exemples sur papier pour réfléchir à cette situation)
- Il sera nécessaire de mémoriser les différentes positions des cases sur lesquelles le joueur aura posé son pion : quelle structure de données vous pourrait pertinente pour cela ?
- Il ne faudra mémoriser que les *nouvelles* positions occupées :

```
if ... not in nom_de_la_liste :  
if ... not in nom_du_dictionnaire.keys() :  
if ... not in nom_du_dictionnaire.values() :
```
- Un *test* (dans la structure de répétition) sera nécessaire pour mettre fin au jeu.

**Objectif 1 :** Ecrire une fonction `nb_de_coups()` qui simule le jeu et renvoie le nombre de lancers aléatoires de dé nécessaires pour terminer le jeu.

**Objectif 2 :** Modifier cette fonction `nb_de_coups(N)` pour qu'elle prenne en paramètre le nombre `N` de cases du plateau de jeu (dans l'exemple, `N` vaut 16)

**Objectif 3 :** Ecrire une fonction `joueur_gagnant(N)` qui simule l'affrontement de 2 joueurs jouant successivement, l'un après l'autre, avec un pion distinct et qui renvoie le joueur (1 ou 2) qui est le premier à avoir atteint l'objectif d'avoir posé son pion au moins une fois sur chacune des `N` cases.

Remarque : il sera nécessaire de dupliquer les différentes variables précédemment utilisées.

Toute tentative (même infructueuse) sera valorisée.

**L'utilisation d'affichages intermédiaires (avec concaténation du texte et des valeurs) sera très appréciée pour vos tests et sera valorisée.**

Exemple:

```
>>> nb_de_coups(16)
```

```
Le jeu commence : le pion est posé sur la case 0.  
Le dé tombe sur 5.  
Le joueur avance son pion et le pose sur la case 5.  
C'est la première fois que le joueur pose son pion sur cette case.  
Le dé tombe sur 3.  
Le joueur avance son pion et le pose sur la case 8.  
C'est la première fois que le joueur pose son pion sur cette case.  
...  
Le dé tombe sur 2.  
Le joueur avance son pion et le pose sur la case 12.  
Le joueur avait déjà posé son pion sur cette case.  
Le dé tombe sur 6.  
Le joueur avance son pion et le pose sur la case 2.  
C'est la première fois que le joueur pose son dé sur cette case.  
Le joueur a (enfin) posé son pion au moins une fois sur chacune  
des cases !  
Le jeu est fini après 52 lancers de dés.  
52
```