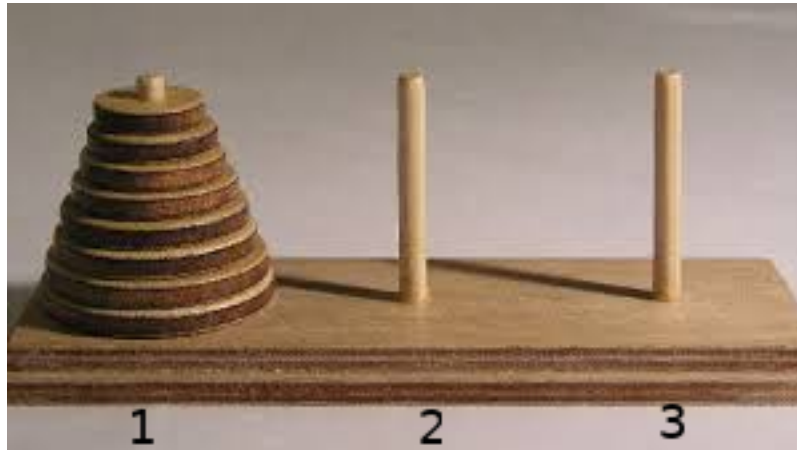


# 1 Descriptif du jeu et convention

Le jeu des Tours d'**Hanoï** est constitué de trois piquets verticaux, notés 1, 2 et 3 et de  $n$  disques superposés de tailles strictement décroissantes avec un trou au centre et enfilés autour du piquet 1.

Ces disques forment les tours, comme sur la figure ci-dessous :



Le but du jeu consiste à déplacer l'ensemble des disques du piquet 1 au piquet 3 en respectant les règles suivantes :

- les disques sont déplacés un par un ;
- un disque ne doit jamais être posé sur un disque de diamètre inférieur ;

## convention :

Le programme commencera par demander le nombre  $n$  de disques. Les disques seront représentés par un nombre pair du caractère « '-' » :

Par exemple, le premier disque sera représenté par la chaîne « '-' », le deuxième par « '--' », le troisième par « '---' », ainsi de suite.

Un emplacement libre sera représenté par le caractère « 'x' ». On utilisera un dictionnaire nommé **dico\_des\_coups** illustré par la figure ci-dessous, que l'on remplira au fur et à mesure du déroulement du jeu.

Par exemple, l'état du dictionnaire avec  $n = 3$  disques et au troisième coups est le suivant :

```
dico_des_coups = {
0: [['--', '----', '-----'], ['x', 'x', 'x'], ['x', 'x', 'x']],
1: [['x', '----', '-----'], ['x', 'x', 'x'], ['x', 'x', '--']],
2: [['x', 'x', '-----'], ['x', 'x', '----'], ['x', 'x', '--']],
3: [['x', 'x', '-----'], ['x', '--', '----'], ['x', 'x', 'x']],
}
```

Toujours avec  $n = 3$  disques et à la fin de la partie, l'état du dictionnaire est le suivant :

```
dico_des_coups = {
0: [['--', '----', '-----'], ['x', 'x', 'x'], ['x', 'x', 'x']],
1: [['x', '----', '-----'], ['x', 'x', 'x'], ['x', 'x', '--']],
2: [['x', 'x', '-----'], ['x', 'x', '----'], ['x', 'x', '--']],
3: [['x', 'x', '-----'], ['x', '--', '----'], ['x', 'x', 'x']],
4: [['x', 'x', 'x'], ['x', '--', '----'], ['x', 'x', '-----']],
5: [['x', 'x', '--'], ['x', 'x', '----'], ['x', 'x', '-----']],
6: [['x', 'x', '--'], ['x', 'x', 'x'], ['x', '----', '-----']],
7: [['x', 'x', 'x'], ['x', 'x', 'x'], ['--', '----', '-----']]
}
```

La valeur de chaque **clé** est une liste de trois listes qui correspondent chacune aux trois piquets :

- La clé 0 représente l'état initial avec tous les disques situés sur le premier piquet.

0: [['--', '----', '-----'], ['x', 'x', 'x'], ['x', 'x', 'x']]

Les deux autres piquets sont libres et sont représentés par les deux listes suivantes

['x', 'x', 'x'], ['x', 'x', 'x']

Le caractère « 'x' » représente un emplacement libre. On remarquera que l'on empile les disques dans l'ordre décroissant des indices des listes.

En effet, dans le cas de trois disques, le plus petit disque, représenté par la chaîne de caractères « '- -' » a pour indice 0, le plus grand disque « '- - - -' » a pour indice 2.

- Par exemple pour la clé 1 la liste représente la situation après le déplacement du disque « - - » du piquet 1 vers le piquet 3 :

['x', '----', '-----'], ['x', 'x', 'x'], ['x', 'x', '--']

pour représenter les déplacements de chaque tour de la manière suivante.

## 2 Une première version en console

On organisera le code de la manière suivante :

# création des fonctions :

# Programme principal :

```
if __name__ == "__main__":
    n = int(input('saisir le nombre de tours : '))
    dico_coups = initialisation()
    hanoi(n)
```

Pour réaliser le script, nous aurons besoin de quatre fonctions :

- une fonction **initialisation()** qui permet de construire le dictionnaire **dico\_coups** avec la clé 0 et sa valeur initiale.
- Une fonction **last\_elt(liste, caractere)** ayant en argument une liste et un caractere. Cette fonction permet de renvoyer l'**indice** du premier emplacement libre dans une liste (une des trois listes du dictionnaire) où le caractere est celui de l'emplacement vide, à savoir ici, « X » .

Cette fonction nous servira dans la fonction **deplacer(a, b, c)**.

- une procédure **deplacer(a, b, c)** qui permet de déplacer le **disque a** situé sur une **tige b** vers la **tige c**. les arguments **b** et **c** peuvent prendre les valeurs 1, 2 ou 3. Dans cette fonction nous aurons besoin de connaître la première case vide du piquet c, la fonction **last\_elt** permet d'obtenir cet indice.
- une fonction **réursive hanoi(n, i=1, j=2, k=3)** ayant en argument le nombre de disques *n* et les piquets *i*, *j* et *k*. Le principe de récurrence est le suivant :  
pour déplacer *n* disques de la tige 1 vers la tige 3 il suffit de déplacer *n* – 1 disques de la tige 1 vers la tige 2 puis de la tige 2 vers la tige 3. Autrement dit, il suffit de généraliser le problème de manière à décrire le déplacement de *n* disques de la tige *i* à la tige *k* en utilisant la tige *j* comme pivot.

Cette fonction ne renvoie rien (None).

On pourra placer au bon endroit dans cette fonction **hanoi()** la ligne de code `print('déplacer le disque {} de la tige {}` permet d'avoir un affichage des déplacements à chaque étape.

```
saisir le nombre de tours : 3
{0: [['--', '-----', '-----'], ['x', 'x', 'x'], ['x', 'x', 'x']],
1: [['x', '-----', '-----'], ['x', 'x', 'x'], ['x', 'x', '-----']],
2: [['x', 'x', '-----'], ['x', 'x', '-----'], ['x', 'x', '-----']],
3: [['x', 'x', '-----'], ['x', '-----', '-----'], ['x', 'x', 'x']],
4: [['x', 'x', 'x'], ['x', '-----', '-----'], ['x', 'x', '-----']],
5: [['x', 'x', '-----'], ['x', 'x', '-----'], ['x', 'x', '-----']],
6: [['x', 'x', '-----'], ['x', 'x', 'x'], ['x', '-----', '-----']],
7: [['x', 'x', 'x'], ['x', 'x', 'x'], ['-----', '-----', '-----']]}
```

In [2]:

### 3 Une version graphique du jeu

Dans ce paragraphe l'idée est de proposer une version graphique de ce jeu, en utilisant le module **Tkinter**. On importe le module **tkinter** par :

```
from tkinter import *
```

**Création de la fenêtre** : On crée une fenêtre « **tkinter** » à l'aide de la commande suivante, la deuxième ligne permettant d'ajouter un titre à cette instance :

```
ma_fenetre = Tk()
ma_fenetre.title("Tours d'Hanoi")
```

Tous les autres objets vont hériter de cette **fenêtre tkinter**. Ils sont décrits dans le tableau ci-dessous :

Objets	description	implémentation
Frame		frame_1=Frame(ma_fenetre)
Canvas	zone graphique	Canvas(ma_fenetre, height = 500,width = 1100,bd=10,bg='ivory')
Label	un label	label_1 = label_nbre_disques = Label(frame_2, text = 'saisir le nombre de disques')
Button	Un « widget » bouton	bouton_1 = Button(frame_1, text = 'Quitter', command = quitter)
Entry	Une entrée pour saisir un nombre, un texte	saisie = Entry(frame_2,width=5,justify='center',bd=3)

On termine toujours une instance de « **tkinter** » par une boucle infinie avec la commande

```
ma_fenetre.mainloop()
```

Écrire un script permettant de créer une **fenêtre tkinter** comportant :



FIGURE 1 – une fenêtre tkinter

- Deux « Frame (s) », un pour les boutons

#### Création des fonctions :

Nous avons besoin de créer les fonctions suivantes :

- Une fonction **quitter()** qui permet de quitter la fenêtre avec la commande :

```
ma_fenetre.destroy()
```

On quitte le jeu par destruction de la fenêtre principale *ma\_fenetre*.

- Une fonction **tracer\_piquet()** qui permet de tracer trois piquets comme sur la figure. On pourra utiliser une boucle **for i in range(3) : ...** et l'objet

```
canevas.create_polygon(x0, y0, x1, y1, ..., x0, y0)
```

avec autant de coordonnées  $(x_i, y_i)$  que de sommets sans oublier le retour sur le premier point  $(x_0, y_0)$  afin de fermer la figure.

- Une fonction **tracer\_disque(n)** qui permet de tracer  $n$  disques. On utilisera l'objet

```
create_rectangle(x0, y0, x1, y1)
```

où  $(x_0, y_0)$  et  $(x_1, y_1)$  sont les coordonnées respectives du point supérieur gauche et du point inférieur droite.