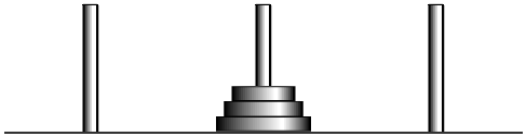
Les tours de Hanoï (1)



« Les tours de Hanoï » est un jeu imaginé par le mathématicien français Édouard Lucas (1842-1891). Ce jeu consiste à déplacer des disques de diamètres différents depuis une tour de « départ » vers une tour d'« arrivée » en passant par une tour « intermédiaire », et en respectant les deux règles suivantes :

- on déplace un seul disque à la fois,
- on place un disque sur un autre disque plus grand ou sur un emplacement vide.

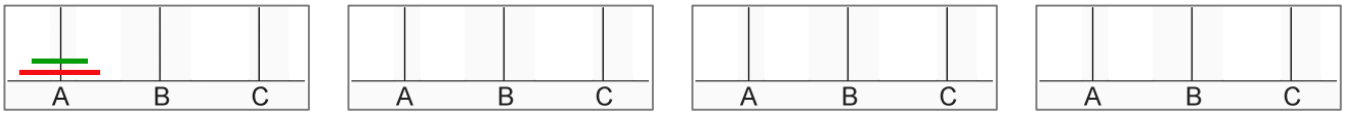
Le jeu est terminé lorsque la totalité des disques ont été déplacés sur la tour d'arrivée.

1) Prise en main du jeu

Une version javascript est en ligne sur le site de la classe : http://www.info-nsi.fr/nsi-algorithmique/hanoi.html

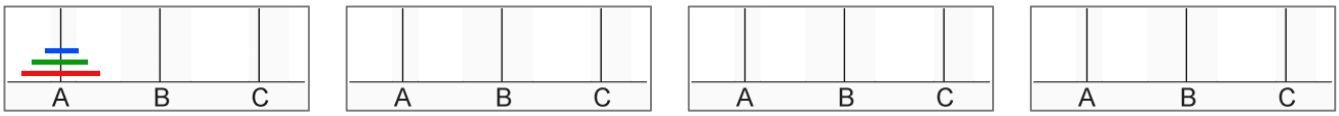
Tester le jeu puis compléter les figures ci-dessous :

a) Reproduire les déplacements des **deux disques** de la tour A vers la tour B (en respectant les règles du jeu).

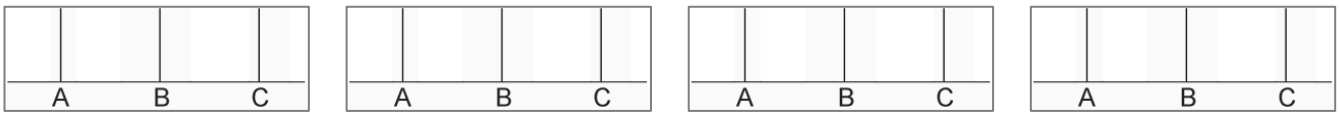


Début | Déplacement 1 | Déplacement 2 | Déplacement 3 et Fin

b) Reproduire les déplacements des **trois disques** de la tour A vers la tour C.



Début | Déplacement 1 | Déplacement 2 | Déplacement 3



Déplacement …….. | Déplacement …….. | Déplacement …….. | Déplacement ……..

2) Simulation des « Tours de Hanoï » avec Python

On veut maintenant simuler « Les tours de Hanoï » avec un programme Python. Chaque tour sera représentée par une liste A, B ou C contenant des nombres classés par ordre croissant.

Par exemple, au départ de l'exercice précédent (trois disques) les listes sont :

A = [3, 2, 1] B = [ ] C = [ ] On remarque que les liste B et C sont vides.

a) Ecrire le code Python qui crée et initialise les listes A, B et C dans le cas de tours de trois disques :

Proposition de code

.....

.....

.....

.....

.....

Correction

.....

.....

.....

.....

.....

b) Les listes A, B et C étant créées, le déplacement d'un disque de A vers B revient à ajouter à la liste B le dernier élément de la liste A (et de supprimer de la liste A l'élément déplacé).

Ecrire une ligne de code Python qui ajoute le dernier élément d'une **liste de depart X non vide** à une **liste d'arrivée Y** et qui supprime cet élément de la liste **X**.

Exemple : `([3, 2, 1], [9])` devient `([3, 2], [9, 1])`

Proposition de code

.....

.....

Correction

.....

c) Ecrire la fonction **deplace2Elements(X,Y,Z)** qui permet de déplacer deux éléments de X vers Y, en passant par Z. On pourra s'aider des résultats des questions 1.a) et 2.b)

Exemple :

`Deplace2Elements([2, 1], [], [3])` retourne `([], [2, 1], [3])`

Proposition de code

```
def deplace2Elements(X,Y,Z):
```

.....

.....

.....

.....

.....

Correction

```
def deplace2Elements(X,Y,Z):
```

.....

.....

.....

.....

.....

d) Ecrire la fonction **deplace3Elements(X,Y,Z)** qui permet de déplacer trois éléments de X vers Y, en passant par Z. On pourra s'aider des résultats des questions 1.b) et 2.c)

Exemple :

`Deplace3Elements([3, 2, 1], [], [4])` retourne `([], [3, 2, 1], [4])`

Proposition de code

```
def deplace3Elements(X,Y,Z):
```

.....

.....

.....

.....

.....

Correction

```
def deplace3Elements(X,Y,Z):
```

.....

.....

.....

.....

.....

e) Ecrire la fonction **deplace4Elements(X,Y,Z)** qui permet de déplacer quatre éléments de X vers Y, en passant par Z. On pourra s'aider des résultats des questions 2.d)

Exemple :

`Deplace4Elements([7, 4, 3, 2, 1], [5], [6])` retourne `([7], [5, 4, 3, 2, 1], [6])`

Proposition de code

```
def deplace4Elements(X,Y,Z):
    .....
    .....
    .....
    .....
```

Correction

```
def deplace4Elements(X,Y,Z):
    .....
    .....
    .....
    .....
```

Remarque :

`Deplace4Elements([4, 3, 2, 1], [], [])` retourne `([], [4, 3, 2, 1], [])`

La fonction **deplace4Elements(X,Y,Z)** permet donc de résoudre la Tour de Hanoï à quatre disques.

f) Comparer les fonctions **deplace3Elements(X,Y,Z)** et **deplace4Elements(X,Y,Z)**.

.....

g) On déduit des questions précédentes que les fonctions deplace3Elements(X,Y,Z), deplace4Elements(X,Y,Z), deplace 5Elements(X,Y,Z) et suivantes peuvent être remplacées par une seule fonction **deplaceNElements(N,X,Y,Z)** où N est le nombre d'éléments à déplacer. Ecrire cette fonction.

Exemple :

`DeplaceNElements(3, [4, 3, 2, 1], [5], [])` retourne `([4], [5, 3, 2, 1], [])`

Proposition de code

```
def deplaceNElements(N,X,Y,Z):
    .....
    .....
    .....
    .....
```

Correction

```
def deplaceNElements(N,X,Y,Z):
    .....
    .....
    .....
    .....
```

h) Quelle condition doit-on imposer à N ?

.....

i) Ecrire et faire fonctionner un programme Python qui résout le problème des tours de Hanoï pour un nombre de variables de tours.

Déposer Hanoi-Votrenom.py, en pièce jointe, dans votre dossier Travaux Dirigés sur OneNote
La page doit nommée Hanoi.
Le programme doit fonctionner, afficher le résultat de chaque déplacement et, en particulier le résultat final.