

Exercice 1 : Applications directes du cours

- On pose : $u_0 = 3$ et $\forall n \in \mathbb{N}, u_{n+1} = u_n^2 - 2$.
Ecrire une fonction récursive d'argument n qui renvoie u_n .
- Ecrire une fonction récursive d'argument n qui renvoie $\sum_{k=1}^n k$.
- On rappelle que la méthode d'exponentiation rapide est basée sur la remarque suivante :

$$x^n = \begin{cases} x^{n/2} \cdot x^{n/2} & \text{si } n \text{ est pair,} \\ x \cdot x^{(n-1)/2} \cdot x^{(n-1)/2} & \text{si } n \text{ est impair,} \end{cases}$$

c'est-à-dire :

$$x^n = \begin{cases} x^{n/2} \cdot x^{n/2} & \text{si } n \text{ est pair,} \\ x \cdot x^{n/2} \cdot x^{n/2} & \text{si } n \text{ est impair.} \end{cases}$$

- En suivant cette méthode, écrire une fonction récursive `expRap` d'arguments x et n qui renvoie x^n .
On veillera bien à minimiser le nombre d'appels récursifs.
- Etudier, à la main et en effectuant un balisage, les appels successifs lors de l'exécution de `expRap(2,9)` et `expRap(2,11)`.

Exercice 2 : Dichotomie et récursivité

On rappelle que si L est une liste, alors :

$$L[i : j] = [L[i], L[i+1], \dots, L[j-1]].$$

et :

$$L[:j] = [L[0], L[1], \dots, L[j-1]] \text{ et } L[i:] = [L[i], L[i+1], \dots, L[\text{len}(L)-1]].$$

- Ecrire une fonction dichotomique et récursive d'arguments l une liste d'entiers triée par ordre croissant et x un entier, qui renvoie `True` si x est dans l et `False` sinon.
- Ecrire une fonction dichotomique et récursive d'arguments l une liste de flottants qui renvoie le maximum de l .

Exercice 3 : Tours de Hanoï

Le jeu des tours de Hanoï est un jeu de réflexion qui consiste à déplacer des disques, un à un, sur 3 tiges.



Le but est de déplacer toute la tour d'une tige à une autre. La règle est d'un grand disque ne peut jamais se trouver au dessus d'un plus petit.

Voici une solution pour le cas de 3 disques :

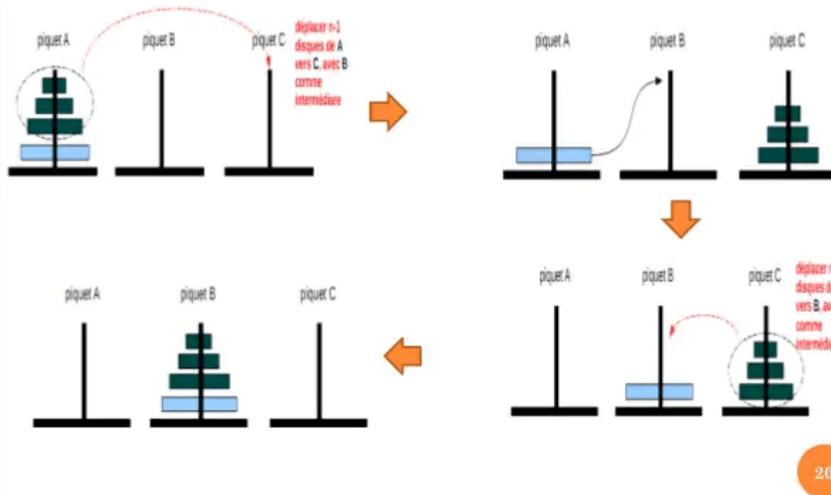
Mouvement	Position	Mouvement	Position																		
	<table style="margin: auto;"> <tr><td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;"> </td><td style="text-align: center;"> </td></tr> <tr><td colspan="3" style="text-align: center;"> </td></tr> </table>	A	B	C								<table style="margin: auto;"> <tr><td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;"> </td><td style="text-align: center;"> </td></tr> <tr><td colspan="3" style="text-align: center;"> </td></tr> </table>	A	B	C						
A	B	C																			
A	B	C																			
Position initiale		4 : A vers C																			
	<table style="margin: auto;"> <tr><td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;"> </td><td style="text-align: center;"> </td></tr> <tr><td colspan="3" style="text-align: center;"> </td></tr> </table>	A	B	C								<table style="margin: auto;"> <tr><td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;"> </td><td style="text-align: center;"> </td></tr> <tr><td colspan="3" style="text-align: center;"> </td></tr> </table>	A	B	C						
A	B	C																			
A	B	C																			
1 : A vers C		5 : B vers A																			
	<table style="margin: auto;"> <tr><td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;"> </td><td style="text-align: center;"> </td></tr> <tr><td colspan="3" style="text-align: center;"> </td></tr> </table>	A	B	C								<table style="margin: auto;"> <tr><td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;"> </td><td style="text-align: center;"> </td></tr> <tr><td colspan="3" style="text-align: center;"> </td></tr> </table>	A	B	C						
A	B	C																			
A	B	C																			
2 : A vers B		6 : B vers C																			
	<table style="margin: auto;"> <tr><td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;"> </td><td style="text-align: center;"> </td></tr> <tr><td colspan="3" style="text-align: center;"> </td></tr> </table>	A	B	C								<table style="margin: auto;"> <tr><td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;"> </td><td style="text-align: center;"> </td></tr> <tr><td colspan="3" style="text-align: center;"> </td></tr> </table>	A	B	C						
A	B	C																			
A	B	C																			
3 : C vers B		7 : A vers C																			
	<table style="margin: auto;"> <tr><td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;"> </td><td style="text-align: center;"> </td></tr> <tr><td colspan="3" style="text-align: center;"> </td></tr> </table>	A	B	C								<table style="margin: auto;"> <tr><td style="text-align: center;">A</td><td style="text-align: center;">B</td><td style="text-align: center;">C</td></tr> <tr><td style="text-align: center;"> </td><td style="text-align: center;"> </td><td style="text-align: center;"> </td></tr> <tr><td colspan="3" style="text-align: center;"> </td></tr> </table>	A	B	C						
A	B	C																			
A	B	C																			

On souhaite déplacer une tour de n disques de la tige i vers la tige j .
On notera k la tige qui n'est ni i , ni j . Le principe de la solution récursive est le suivant :

- on déplace récursivement $n - 1$ disques de la tige i à la tige k ,
- on déplace le disque restant de la tige i à la tige j ,
- on déplace récursivement $n - 1$ disques de la tige k à la tige j .

EXEMPLE DES TOURS DE HANOI

CONCEPTION DE LA SOLUTION RÉCURSIVE



Ecrire une fonction récursive d'arguments n, i, j qui affiche les déplacements à effectuer.

Exercice 4 : Somme des chiffres

Dans cet exercice, on pourra utiliser les commandes suivantes :

- `str` : conversion en chaîne de caractères,
- `int` : conversion en entier.

Par exemple :

```
>>> str(154)
'154'
>>> int('154')
154
```

1. Ecrire une fonction `somme` d'argument un entier n qui renvoie la somme des chiffres de n .

2. Ecrire une fonction récursive `SommeSomme` d'argument un entier n qui renvoie la somme de la somme, ... des chiffres de n , où on a sommé jusqu'à obtenir une somme à un chiffre.

Par exemple : `SomeSome(154)=1`.

Exercice 5 : Le compte est bon

Le but du jeu est le suivant : on tire 6 nombres parmi : 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 25, 50, 75, 100. En utilisant les quatre opérations élémentaires, on cherche à obtenir un nombre cible. On précise que chaque nombre ne peut être obtenu qu'une seule fois.

Pour résoudre le problème, on utilisera le procédé suivant :

- on considère 2 nombres de la liste,
- on fait une opération entre ces deux nombres,
- si l'objectif est atteint, la fonction s'arrête,
- sinon, le résultat de l'opération remplace, dans la liste, les 2 nombres considérés au départ,
- on trie le nouveau tableau par ordre des croissant (on pourra utiliser les méthodes `sort` qui trie par ordre croissant et `reverse` qui inverse une séquence),
- on recommence le procédé avec le nouveau tableau (appel récursif),
- si l'objectif n'est pas atteint, on considère une autre opération,
- si l'objectif n'est toujours pas atteint, on considère deux autres éléments de la liste.

Ecrire une fonction récursive `LeCompteEstBon` d'arguments `t` liste de 6 nombres parmi 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 25, 50, 75, 100 triée par ordre décroissant et `obj` nombre représentant l'objectif à atteindre et qui renvoie, si elle existe, la solution du problème.

Par exemple :

```
>>> LeCompteEstBon([100,25,10,6,5,2],53)
63 - 10 = 53
126 / 2 = 63
131 - 5 = 126
125 + 6 = 131
100 + 25 = 125
```