

# Informatique – TP5

Vésale Nicolas - Henrik Thys

## Exercice 1: Premier entier le plus proche.

Donner une fonction **récursive procheEntier** qui prend pour paramètres un entier **n** une liste non vide **L** d'entiers et qui rend le premier (au sens des indices) élément le plus proche de **n**.

```
>>> procheEntier(4, [1,2,3,5,6,7,8])
3
```

## Exercice 2: Palindromes: une fonction récursive.

Un mot est un palindrome s'il se lit de la même façon en partant depuis la fin que depuis le début. Par exemple, « kayak » est un palindrome mais pas « crocodile ». Écrire une fonction **estPalindrome** **récursive** déterminant si un mot est un palindrome ou non.

```
>>> estPalindrome('kayak')
True
>>> estPalindrome('crocodile')
False
```

## Exercice 3: Numérotation romaine.

On rappelle que les chiffres romains ont les valeurs suivantes:

- M= 1000   • D= 500   • C= 100   • L= 50   • X= 10   • V= 5   • I= 1

Que l'on enregistre en utilisant le dictionnaire:

```
dicRom ={ 'M':1000, 'D':500, 'C':100, 'L':50, 'X':10, 'V':5, 'I':1}
```

Pour décoder un **nombre** écrit en chiffres romains, on utilise le principe suivant:

1. Si le nombre n'a qu'un seul chiffre, on renvoie sa valeur.
2. Sinon, si le premier chiffre a une valeur inférieure au deuxième, on soustrait sa valeur aux chiffres restants et sinon, on l'ajoute.

Donner une fonction **récursive romVersDec** qui prend pour paramètre une chaîne de caractères qui représente un nombre écrit en chiffres romains et qui rend sa valeur en décimal.

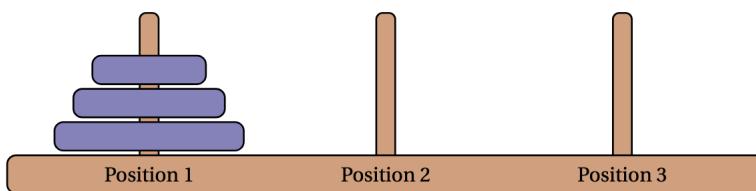
```
>>> romVersDec('MCD')
1400
>>> romVersDec('CXXIV')
124
```

## Exercice 4: Les tours de Hanoï.

Les tours de Hanoï est un jeu de réflexion consistant à déplacer  $n$  disques de diamètres différents d'une position de départ à une position d'arrivée en passant par une position intermédiaire en respectant les règles suivantes :

1. on ne peut déplacer plus d'un disque à la fois ;
2. on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

On suppose que cette dernière règle est également respectée dans la configuration de départ. Le schéma ci-dessous représente le jeu des tours de Hanoï avec  $n = 3$  disques. L'objectif du jeu est de déplacer les disques de la position 1 à la position 3 en respectant les règles décrites ci-dessus.



Pour décrire un mouvement d'un disque au sommet d'une tour à la position  $i$  vers la tour à la position  $j$ , nous utiliserons la notation  $i \rightarrow j$ . Avec ces notations, une solution du jeu pour  $n = 3$  est donnée par la succession de mouvements:

```
1->3, 1->2, 3->2, 1->3, 2->1, 2->3, 1->3
```

Pour résoudre le jeu pour un nombre quelconque  $n \in \mathbb{N}^*$  de disques, on peut construire la solution récursivement de la manière suivante :

1. On déplace les  $n - 1$  disques au-dessus de la tour de la position de départ vers la position intermédiaire en utilisant la récursivité.
2. On déplace le plus grand disque de la position de départ vers la position d'arrivée.
3. On déplace les  $n - 1$  disques de la position intermédiaire vers la position d'arrivée en utilisant la récursivité.

Écrire une fonction récursive `hanoi(n:int, posDep:int, posArr:int) -> list` prenant en argument le nombre de disques  $n$ , leur position au départ `posDep` et leur position d'arrivée `posArr` et qui renvoie la liste des mouvements à effectuer pour résoudre le jeu.

```
>>> hanoi(3, 1, 3)
['1->3', '1->2', '3->2', '1->3', '2->1', '2->3', '1->3']
```

**Rappel:** si `a` est un entier, `str(a)` transforme `a` en chaîne de caractères.