TP n°2 : les fonctions en Python.

1. Documentation des fonctions en Python et un peu de traitement de caractères.

Pour faciliter la lecture et le test des fonctions, on peut utiliser les docstrings qui permettent de documenter les programmes et de préciser les attendus des fonctions.

Quelques commandes sur les chaînes de caractères en python :

```
>>> a="zert"
>>> type(a)
<class 'str'>
>>> a[0]
'z'
>>> a[3]
't'
>>> len(a) # renvoie la longueur de la chaîne de caractères
4
>>> a[len(a)-1]
't'
>>> a[-1]
't'
>>> "a"+a
'azert'
>>> a+"y"
'zerty'
>>> for caract in a:
    print(caract)

z
e
r
t
>>> for i in range(len(a)):
    print(a[i])

z
e
r
t
```

Activité 1.

On considère la fonction ci-dessous et sa docstring, compléter la pour qu'elle réponde à celle-ci.

```python
def inverse(message):
    ''' inverse une chaîne de caractères
    message : type str
    return : retourne le message inversé (str )

    >>> inverse("azerty")
    "ytreza"
    >>> inverse("non")
    "non"
    '''
```

Activité 2

Modifier la fonction précédente pour qu'elle traite aussi bien les chaînes de caractères que les entiers. La valeur retournée sera du même type que la valeur passée en paramètre.

Activité 3

Ecrire une fonction permettant de déterminer si un nombre entier ou une chaîne de caractères est un palindrome.

```python
def palindrome(message):
    ''' indique si le message est un palindrome (se lit de droite à gauche ou de gauche
à droite )
    message : type str ou int
    return : True si message est un palindromme, False sinon.

    >>> palindrome("azerty")
    False
    >>> palindrome("non")
    True
    >>> palindrome(12)
    False
    >>> palindrome(1221)
    True
    '''
```

2. Conjecture de Syracuse, ou de Collatz, ou d'Ulam et découverte des tuples

> "Les mathématiques ne sont pas encore prêtes pour de tels problèmes". Paul Erdos

Soit N un entier naturel, on applique les instructions suivantes :

- si N est pair , on le divise par 2,
- si N est impair, on le multiplie par 3 et on ajoute 1.

- On applique à nouveau au nombre obtenu cet algorithme.

1. a) Appliquer ces instructions au nombre $N = 14$. Que remarquez vous ?
   b) Appliquer ces instructions à un entier de votre choix. Que remarquez-vous ?
2. Peut-on définir cette série d'instructions comme un algorithme ? Que faut-il modifier à cette série d'instructions pour constituer un algorithme ?
3. Compléter la fonction python suivante qui retourne ce que l'on appelle le vol du nombre N, c'est à dire la suite des nombres obtenus en appliquant cet algorithme jusqu'à obtenir 1.

```
def vol(n):
  ''' retourne le vol du nombre n en appliquant l'algorithme de syracuse
  : return : un tuple

  >>> vol(1)
  (1,4,2,1)
  >>> vol(5)
  (5,16,8,4,2,1)
  '''
```

4. On désire à présent ecrire une fonction donnant la hauteur maximale d'un vol.

```
def hauteux_max_vol(n):
   '''
   retourne la hauteur max du vol de n
   n: int
   return : int
   '''
  >>> hauteur_vol(8)
   8
  >>> hauteur_vol(5)
   16
  '''
```

5. Ecrire à présent une fonction déterminant la longueur du vol d'un nombre entier n.

```
def longueur_vol(n):
  '''
  retourne la longueur vol et la hauteur maximale
  return : int
   n : int
   >>> longueur_vol(8)
   4
  >>> longueur_vol(5)
  6
  '''
```

6. a) Ajouter une fonction vol_longueur_max(n) qui détermine pour quelle valeur de k , comprise entre 1 et n, on a le vol le plus long.
   b) En déduire quel est l'entier n compris entre 1 et 100 inclus qui a le vol le plus long. Et pour n entre 1 et 1 000 ?
7. a) Ajouter une fonction vol_hauteur_max(n) qui détermine pour quelle valeur de k, comprise entre 1 et n, on a la hauteur de vol la plus grande.
   b) En déduire quel est l'entier n compris entre 1 et 100 inclus qui a le vol le plus haut. Et pour n entre 1 et 1 000 ?

> "Une course au record est engagée de la plus grande valeur vérifiant la conjecture. Le record de 2009 tenu par T. Oliveira e Silva est de $5 \times 2^{60} \approx 5.8 \times 10^{18}$(Oliveira, 2010, [46])."
>
> Luc-Olivier Pochon, Alain Favre : https://hal.archives-ouvertes.fr/hal-01593181/document

TP inspiré de : https://fr.wikipedia.org/wiki/Conjecture_de_Syracuse