

DS 2

Informatique pour tous, première année

Julien REICHERT

Les programmes sont à écrire en Python et l'alignement vertical doit être rigoureux. En particulier, il s'agit de matérialiser la position du début de la ligne en cas d'ajouts (moyennement recommandés).

Exercice 1 : On cherche à écrire une fonction dont la valeur de retour est le « miroir » de son entrée, qu'on suppose avoir les propriétés d'une séquence (itérable, indexable, supportant la longueur).

1) Expliquer pourquoi le code ci-dessous provoque une erreur si l'entrée est une chaîne de caractères :

```
def miroir(seq):
    n = len(seq)
    seq2 = seq
    for i in range(n): # i de 0 à n-1
        seq2[n-1-i] = seq[i] # indice i de seq -> indice i en partant de la fin de seq2
    return seq2
```

2) Expliquer pourquoi il est incorrect si l'entrée est une liste en détaillant son comportement sur la liste [6,6,4].

3) Corriger le code en respectant au choix l'une des deux spécifications suivantes (en la précisant) : la valeur de retour est le miroir de l'entrée OU il n'y a pas de valeur de retour et la chaîne en entrée est transformée en son miroir.

4) [Bonus] Écrire un code de deux lignes (la première étant `def miroir(seq):`) pour résoudre l'exercice (là aussi, les deux spécifications sont possibles).

Les deux exercices suivants se basent sur la connaissance du module `numpy`.

Exercice 2 : Écrire une fonction simulant `arange` en se servant de `linspace` et vice-versa. On pourra supposer que les trois arguments principaux des deux fonctions sont obligatoires et les autres sont interdits.

On rappelle les spécifications de ces deux fonctions issues de `numpy` : `arange(deb,fin,pas)` retourne un tableau contenant `deb`, `deb+pas`, ..., `deb+k*pas` tel que `deb+k*pas < fin ≤ deb+(k+1)*pas` ; `linspace(deb,fin,n)` retourne un tableau de taille `n` (50 par défaut) dont le premier élément est `deb`, le dernier est `fin` et les différences entre deux éléments consécutifs sont toutes identiques (on ne parlera pas des autres arguments optionnels).

Une partie importante de la notation sera consacrée au raisonnement mathématique (si le programme n'est pas bon), ce qui fait que donner des exemples non triviaux de résultats de ces deux fonctions et de correspondances de l'une à l'autre serait pris en compte.

Exercice 3 : Écrire une fonction retournant une matrice `M` à `n` lignes et `m` colonnes telle que `M[i][j]` vaille `|j-i|`. Il est possible (et recommandable) de se servir d'une certaine fonction présentée en cours...