

Exercice 1

On considère une liste `L`.

Écrire un programme qui enlève tous les 2 de la liste.

Par exemple, si la valeur de `L` avant le programme est

`[1, 1, 2, 2, 2, 4, 3, 2, 2, 1]`

alors après l'exécution du programme `L` devra valoir

`[1, 1, 4, 3, 1]`

Exercice 2

Écrire une fonction `enleve_2` qui

- en entrée prend une liste;
- en sortie renvoie une liste, composée des éléments de la liste d'entrée, dans le même ordre mais sans les 2.

Par exemple si `L` vaut `[1, 2, 2, 3, 2]` alors `enleve_2(L)` vaut `[1, 3]` et le fait d'avoir appelé la fonction n'a pas modifié `L`.

Exercice 3

Écrire une fonction `renverse` qui

- en entrée prend une liste;
- en sortie renvoie une liste, composée des éléments de la liste d'entrée mais dans l'ordre inverse.

Par exemple si `L` vaut `[1, 2, 3, 4]` alors `renverse(L)` vaut `[4, 3, 2, 1]` et le fait d'avoir appelé la fonction n'a pas modifié `L`.

Attention : on n'a pas le droit d'utiliser `L.reverse()`.

Exercice 4

Écrire une fonction `renverse_en_place` qui

- en entrée prend une liste;
- ne renvoie rien mais renverse les éléments de la liste.

Par exemple si `L` vaut `[3, 4, 5]`, alors après l'appel de `renverse_en_place(L)`, `L` vaut `[5,4,3]`.

Attention : on n'a pas le droit de créer une deuxième liste dans la fonction.

Exercice 5

Compléter

```
def doublon(L : list) -> bool :  
    """ Renvoie True si une valeur de la liste apparaît au moins deux  
        fois et False si chaque valeur est unique """
```