

"""

Author: Pascal Padilla

Source: correction de l'exercice 2 du sujet 16 des épreuves pratiques NSI 2022

Remarques:

- \* quand vous créez vos propres programmes, utiliser des noms de variables en minuscules (sauf pour les CONSTANTES)

"""

```
def positif(T):
    # l'idée de l'algo (pour laisser T inchangée) est:
    # 1. copier T dans une autre pile (T2)
    # 2. dépiler T2 dans T3 en ignorant les nombres négatifs
    # La pile T3 contient les bons éléments, mais dans l'ordre inverse
    # 3. dépiler T3 dans T2 (qui est vide au début) pour rétablir l'ordre
    # des éléments

    # BOUCLE n°1
    # invariant:
    # * T3 contient tous les éléments dépilés de T2, sauf les négatifs
    # * T3 et T2 contiennent tous les éléments positifs de T

    # initialisation:
    # * T2 : copie de T
    # * T3 : pile vide
    T2 = list(T)
    T3 = []

    # condition d'arrêt:
    # * T2 n'as plus aucun élément
    while T2 != []:
        # dépiler un élément de T2
        x = T2.pop()

        # et l'empiler dans T3 seulement s'il est positif
        if x >= 0:
            T3.append(x)

    # BOUCLE n°2
    # invariant: T3 et T2 contiennent tous les éléments positifs de T

    # initialisation: T2 est vide (c'est déjà le cas, mais on le formalise)
    T2 = []

    # condition d'arrêt: T3 est vide
    while T3 != []:
        x = T3.pop()
        T2.append(x)

    # on vérifie que 'T' est inchangée
    print('T = ', T)

    # on renvoie T2
    return T2

# Test de l'énoncé ajouté pour tester la fonction
assert positif([-1, 0, 5, -3, 4, -6, 10, 9, -8]) == [0, 5, 4, 10, 9]

# Autre façon de tester avec un affichage
print() # pour sauter une ligne
print("Autre test possible avec un affichage:")
print( positif([-1, 0, 5, -3, 4, -6, 10, 9, -8]) )
```