TD : Programmation Fonctionnelle en Python TNSI

Toutes les fonctions demandées devront être écrites avec **le paradigme de programmation fonctionnelle**, c'est-à-dire sans utiliser de boucles explicites et en évitant les effets de bord (fonctions pures).

Exercice 1 : Calculer la somme en style fonctionnel

Écrire une fonction `somme(liste)` qui prend en paramètre une liste de nombres et renvoie la somme de tous les éléments de la liste.

Exercice 2 : Filtrer les nombres pairs en style fonctionnel

Écrire une fonction `filtrer_pairs(liste)` qui prend en paramètre une liste de nombres et renvoie une nouvelle liste contenant uniquement les nombres pairs.
On commencera par écrire une fonction `est_pair(x)` qui renvoie `True` si `x` est pair, `False` sinon.

Exercice 3 : Appliquer une fonction à chaque élément en style fonctionnel

1. Écrire une fonction `carre(x)` qui renvoie le carré de `x`.
2. Écrire une fonction `appliquer_fonction(liste, fonction)` qui applique une fonction donnée à chaque élément d'une liste et renvoie la nouvelle liste. La fonction `appliquer_fonction` devra être générique et pouvoir appliquer n'importe quelle fonction à chaque élément de la liste.

Pour la tester, utiliser :

```
# Test
assert appliquer_fonction([1, 2, 3], carre) == [1, 4, 9]
```

Exercice 4 : Vérifier si tous les éléments sont positifs en style fonctionnel

Écrire une fonction `tous_positifs(liste)` qui prend en paramètre une liste de nombres et renvoie `True` si tous les éléments de la liste sont positifs, sinon `False`.
On commencera par écrire une fonction `est_positif(x)` qui renvoie `True` si `x` est positif, `False` sinon.

Exercice 5 : Longueur maximale d'une chaîne en style fonctionnel

Écrire une fonction `longueur_maximale(liste)` qui prend en paramètre une liste de chaînes de caractères et renvoie la longueur de la chaîne la plus longue.

On commencera par écrire une fonction `longueur(chaine)` qui renvoie la longueur de la chaîne `chaine` (sans utiliser `len()`).