

DS 3 - Première NSI

Consignes :

- Votre code doit être bien structuré et les fonctions correctement définies.
- L'indentation entre vos lignes de programmation sera prise en compte dans le barème.
- Vous n'avez pas le droit d'utiliser dans vos programmes les fonctions `min` et `max` de python, ni la notion `**` pour fusionner les listes de dictionnaires.

Question 1

Écrire une fonction `trouver_maximum` qui retourne le maximum d'une liste donnée.

Exemple d'entrée :

```
trouver_maximum([3, 7, 2, 9, 4]) # retourne 9
```

À recopier et compléter :

```
def trouver_maximum(liste):  
  
    # compléter  
  
    return max_val
```

Question 2

Utiliser la fonction `trouver_maximum` pour implémenter une fonction `tri_inverse` qui trie une liste par ordre décroissant.

Exemple d'entrée :

```
tri_inverse([3, 7, 2, 9, 4]) # retourne [9, 7, 4, 3, 2]
```

À recopier et compléter :

```
def tri_inverse(liste):  
  
    # compléter  
  
    return resultat
```

Question 3

Écrire une fonction `tri_insertion_inverse` qui trie une liste par ordre **décroissant** en utilisant la méthode du tri par insertion.

Exemple d'entrée :

`tri_insertion_inverse([3, 7, 2, 9, 4])` # retourne `[9, 7, 4, 3, 2]`

À recopier et compléter :

```
def tri_insertion_inverse(liste):  
  
    # compléter  
  
    return liste
```

Question 4

À partir du dictionnaire suivant :

```
donnees = [ {"produit": "Pomme", "prix": 1.2},  
             {"produit": "Banane", "prix": 0.8},  
             {"produit": "Orange", "prix": 1.5},  
             {"produit": "Mangue", "prix": 2.0} ]
```

Créer une fonction `tri_par_prix` qui trie la liste `donnees` par prix croissant.

À recopier et compléter :

```
def tri_par_prix(liste):  
  
    # compléter  
  
    return liste
```

Question 5

À partir du nouveau dictionnaire suivant :

```
codes_barres = [ {"produit": "Pomme", "code_barre": "1234567890123"},  
                 {"produit": "Banane", "code_barre": "9876543210987"},  
                 {"produit": "Orange", "code_barre": "2345678901234"},  
                 {"produit": "Mangue", "code_barre": "5678901234567"} ]
```

Créer une fonction `def fusionner_listes` qui réalise une jointure entre la liste `donnees` de la question 4 et la liste `codes_barres`, en utilisant la clé commune "produit" pour fusionner les informations des deux listes dans une nouvelle liste ?

À recopier et compléter :

```
def fusionner_listes(liste1, liste2):  
  
    # à compléter  
  
    return liste
```