Module pytest

Le but de cette activité est d'utiliser le module pytest de Python pour vérifier le bon fonctionnement de fonctions selon leurs spécifications.
L'idée est de créer un fichier contenant les fonctions à tester, puis un autre fichier de tests qui importera le module pystest ainsi que le fichier précédent et qui contiendra les tests à effectuer.

⚠ Bien respecter les noms des fichiers et des fonctions demandées dans les questions suivantes, en particulier dans le fichier des fonctions. Ils seront utilisés tels quels pour toute autre personne (élève ou enseignant) souhaitant vérifier leur bon fonctionnement.

1. Créer un fichier `fonctions.py`. On y définira les fonctions suivantes (sans utiliser les fonctions prédéfinies en Python, en particulier sans utiliser la fonction `max`) :
   (a) La fonction `maximum(liste)` qui prend comme argument une liste de nombres (de type `float` ou `int`) `liste` et retourne le maximum de cette liste, s'il existe, et lève l'exception `ValueError` (avec l'instruction `raise ValueError`) si la liste est vide.
   (b) La fonction `maximum_inf(liste,v)` qui prend en argument une liste et une valeur `v` et qui retourne le maximum de la liste qui est strictement inférieur à `v` s'il en existe un, et lève l'exception `ValueError` sinon.
   (c) La fonction `deuxieme_max(liste)` qui prend en argument une liste et retourne la deuxième plus grande valeur de la liste (différente de la plus grande), s'il en existe une, et lève l'exception `ValueError` sinon. Cette fonction utilisera les deux fonctions précédentes.
2. Créer un fichier `tests.py`, dont le contenu initial est le suivant :

```
import pytest
from fonctions import deuxieme_max as dm

class TestMax:
        def test01(self):
                assert dm([2,7,5,9,5,3,1]) == 7

        def test02(self):
                with pytest.raises(ValueError):
                        dm([])
```

   (a) Ajouter des tests, soit avec des `assert`, soit pour les cas où l'exception `ValueError` doit être levée, éventuellement d'autres cas où d'autres exceptions pourraient être levées, comme `TypeError`.
   On pourra également effectuer des tests sur les fonctions intermédiaires `maximum(liste)` et `maximum_inf(liste,v)`.
   Imaginer un maximum de cas très particuliers qui pourraient se produire, en jouant sur la taille de la liste et sur les valeurs dans la liste.
   (b) Lancer alors les tests en exécutant, dans un terminal, en étant dans le répertoire contenant les deux fichiers, la commande : `pytest tests.py` ou `pytest-3 tests.py`.
3. Corriger les erreurs trouvées lors de l'exécution de ces codes, jusqu'à réussir les tests.

4. Pour les plus rapides et efficaces :
   (a) Ajouter une fonction `deuxieme_max_bis(liste)` qui fait la même chose que la fonction `deuxieme_max(liste)` mais sans utiliser de fonction intermédiaire.
   (b) Ajouter les tests pour cette fonction dans le fichier de tests et lancer à nouveau les tests.
   (c) Corriger et répéter le processus jusqu'à réussir les tests.