```
# Exercice 1

def somme(a, b):
    """
    Calcule la somme de a et b
    """
    return a + b

# Exercice 2

def moyenne(t):
    """
    Calcule la moyenne des valeurs de la liste T passée en paramètre
    """
    somme = 0
    for indide in range(len(t)):
        somme = somme + t[indide]
    return somme / len(t)

# Exercice 3

def indice_maxi_tab(T):
    """
    Renvoie l'indice de la première occurence de la valeur
    maximale du tableau T. T est supposé non vide.
    """
    # TEST A ECRIRE ICI
    assert type(T) == list, "Le paramètre T doit être une liste de nombre"
    assert len(T) != 0, "La liste T ne peut pas être vide"

# Exercice 4

def quotient(a, b):
    '''
    Renvoie la valeur du quotient de a par b, b étant non nul.
    '''
    # TEST A ECRIRE ICI
    assert type(a) == int and type(b) == int, "a et b doivent être des nombres"
    assert b != 0, "b ne peut pas être nul"

    #Exercice 5

def multiplication(a, b):
    '''
    Renvoie le produit de a par b,
    où a et b sont deux nombres quelconques.
    '''
    return a * b

# jeu de tests à écrire ici

# On pourrait essayer de passer des valeurs autres que des nombres...
assert multiplication(1,1) == 1
assert multiplication(0,1) == 0
assert multiplication(1,0) == 0
assert multiplication(0,0) == 0
assert multiplication(5,5) == 25

# Exercice 6

def somme(t):
    '''
    Renvoie la somme des éléments du tableau t,
    t étant un tableau de nombres
```

```
    '''
    s = 0
    for i in range(len(t)):
        s = s + t[i]
    return s

assert somme([]) == 0
assert somme([0]) == 0
assert somme([1]) == 1
assert somme([1,2,3,4,5]) == 15

# Exercice 7

# à vous de jouer !
# L'erreur viens du fait que quand i est égal à len(t)-1, la comparaison cherche
la valeur i+1 dans la liste, donc dépasse la dernière valeur
# Pour corriger ça, on peut modifier le range
# Il faut aussi modifier le sense de l'inégalité

def est_croissant(t):
    """
    Renvoie True si le tableau t est trié dans l'ordre croissant,
    et False sinon.
    """
    for i in range(len(t)-1):
        if t[i+1] < t[i]:
            return False
    return True

def tests_est_croissante():
    """
    Fonction lançant des tests pour la fonction est_croissante
    """
    # jeu de tests à écrire ici
    assert est_croissant([0,1])
    assert est_croissant([10,11,12])
    assert not est_croissant([3,2])
tests_est_croissante()
```