

Fichier élève

```
In [ ]: """
Amérique du Nord Terminale NSI 2021
Exercice 3

Lien vers le sujet :
https://fr.calameo.com/read/00673140546fcda873cf0
"""
```

```
In [ ]: def total_hors_reduction(tab):
        """Renvoie la somme des valeurs numériques contenues
        dans un tableau tab
        """
        #à compléter

        #tests unitaires
        def test_total_hors_reduction():
            assert total_hors_reduction([]) == 0
            assert total_hors_reduction([2,2]) == 4
            assert total_hors_reduction([1,2]) == 3
            assert total_hors_reduction([-1,2]) == 1
            print("Tests unitaires réussis pour
            total_hors_reduction")

        #à décommenter pour tester
        #test_total_hors_reduction()
```

```
In [ ]: def offre_bienvenue(tab):
        """
        Le site de vente propose la promotion suivante
        comme offre de bienvenue : 20% de réduction
        sur le premier article de la liste,
        30% de réduction sur le deuxième article de la liste
        (s'il y a au moins deux articles)
        et aucune réduction sur le reste des articles
        (s'il y en a).
        Fonction qui en prend paramètre le tableau tab
        des prix des articles du panier d'un client
        et qui renvoie le total à payer lorsqu'on
        leur applique l'offre de bienvenue.
        """
        #à compléter

        #tests unitaires
        def test_offre_bienvenue():
```

```
assert offre_bienvenue([]) == 0
assert offre_bienvenue([100]) == 80.0
assert offre_bienvenue([100, 200]) == 220.0
assert offre_bienvenue([100, 200, 300]) == 520
print("Tests unitaires réussis pour offre_bienvenue")
```

```
#à décommenter pour tester
#test_offre_bienvenue()
```

```
In [ ]: def prix_solde(tab):
        """
        Lors de la période des soldes, le site de vente propose
        les réductions suivantes :
        – si le panier contient 5 articles ou plus, une
        réduction globale de 50%,
        – si le panier contient 4 articles, une réduction
        globale de 40%,
        – si le panier contient 3 articles, une réduction
        globale de 30%,
        – si le panier contient 2 articles, une réduction
        globale de 20%,
        – si le panier contient 1 article, une réduction globale
        de 10%.
        Prend pour argument le tableau tab des prix des articles
        du panier d'un client et renvoyant le total des prix
        de ces articles lorsqu'on leur applique la réduction des
        soldes
        """
```

```
def test_prix_solde():
    assert prix_solde([]) == 0
    assert prix_solde([100]) == 90.0
    assert prix_solde([100, 200]) == 240.0
    assert prix_solde([100, 200, 300]) == 420.0
    assert prix_solde([100, 200, 300, 400]) == 600.0
    assert prix_solde([100, 200, 300, 400, 500]) == 750.0
    print("Tests unitaires réussis pour prix_solde")
```

```
#à décommenter pour tester
#test_prix_solde()
```

```
In [ ]: def minimum(tab):
        """
        Prend en paramètre un tableau de nombres tab
        Renvoie la valeur minimum de ce tableau
        """
        #à compléter

def test_minimum():
    assert minimum([]) == None
```

```
assert minimum([-1]) == -1
assert minimum([-1, 1]) == -1
assert minimum([-1, -2, 1]) == -2
assert minimum([-2, -1, 1]) == -2
assert minimum([1, -1, -2]) == -2
print("Tests unitaires réussis pour minimum")
```

```
#à décommenter pour tester
#test_minimum()
```

```
In [ ]: def offre_bon_client(tab):
        """
        Pour ses bons clients, le site de vente propose
        une offre promotionnelle, à partir de 2
        articles achetés, l'article le moins cher
        des articles commandés est offert
        Prend pour paramètre le tableau des prix
        des articles du panier d'un client
        et renvoie le total à payer lorsqu'on leur applique
        l'offre bon client.
        """
        #à compléter

def test_offre_bon_client():
    assert offre_bon_client([]) == 0
    assert offre_bon_client([100]) == 100
    assert offre_bon_client([50, 200]) == 200
    assert offre_bon_client([100, 200, 300]) == 500
    assert offre_bon_client([200, 100, 300]) == 500
    assert offre_bon_client([200, 300, 100]) == 500
    print("Tests unitaires réussis pour offre_bon_client")

#à décommenter pour tester
#test_offre_bon_client()
```

```
In [ ]: def offre_bloc(tab, debut_bloc):
        """
        Renvoie le total pour un bloc de 3 articles consécutifs
        dans le panier, l'article avec le prix minimal étant
        déduit
        """
        #à compléter

def test_offre_bloc():
    assert offre_bloc([30.5, 15.0, 6.0, 20.0, 5.0, 35.0,
10.5], 0) == 45.5
    assert offre_bloc([30.5, 15.0, 6.0, 20.0, 5.0, 35.0,
10.5], 3) == 55
    print("Tests unitaires réussis pour offre_bloc")

#à décommenter pour tester
```

```
#test_offre_bloc()

def promotion_bloc_trois(tab):
    """
    Prend en paramètre un tableau de prix d'articles dans un
    panier
    Renvoie le prix du panier après déduction du minimum des
    3 premiers
    , du minimum des 3 suivants etc ...
    """
    #à compléter

def test_promotion_bloc_trois():
    assert promotion_bloc_trois([30.5, 15.0, 6.0, 20.0, 5.0,
35.0, 10.5]) == 111
    assert promotion_bloc_trois([30.5, 15.0, 6.0, 20.0, 5.0,
35.0]) == 100.5
    assert promotion_bloc_trois([30.5, 15.0, 6.0, 20.0, 5.0,
35.0, 10.5, 8, 6]) == 119
    print("Tests unitaires réussis pour
promotion_bloc_trois")

#à décommenter pour tester
#test_promotion_bloc_trois()
```

```
In [ ]: def promotion_bloc_trois_glouton(tab):
        """Renvoie le prix optimal pour un panier"""

def test_promotion_bloc_trois_glouton():
    assert promotion_bloc_trois_glouton([30.5, 15.0, 6.0,
20.0, 5.0, 35.0, 10.5]) == 96
    print("Test unitaire réussi pour bloc_trois_glouton")

#à décommenter pour tester
#test_promotion_bloc_trois_glouton()
```