

BACCALAUREAT

SESSION 2022

Épreuve de l'enseignement de spécialité

NUMERIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°21

DUREE DE L'ÉPREUVE : 1 heure

**Le sujet comporte 3 pages numérotées de 1 / 3 à 3 / 3
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

Le candidat doit traiter les 2 exercices.

EXERCICE 1 (4 points)

Programmer la fonction `multiplication`, prenant en paramètres deux nombres entiers `n1` et `n2`, et qui renvoie le produit de ces deux nombres.

Les seules opérations autorisées sont l'addition et la soustraction.

Exemples :

```
>>> multiplication(3,5)
```

```
15
```

```
>>> multiplication(-4,-8)
```

```
32
```

```
>>> multiplication(-2,6)
```

```
-12
```

```
>>> multiplication(-2,0)
```

```
0
```

EXERCICE 2 (4 points)

Recopier et compléter sous Python la fonction suivante en respectant la spécification. On ne recopiera pas les commentaires.

```
def dichotomie(tab, x):  
    """  
        tab : tableau d'entiers trié dans l'ordre croissant  
        x : nombre entier  
        La fonction renvoie True si tab contient x et False sinon  
    """  
  
    debut = 0  
    fin = len(tab) - 1  
    while debut <= fin:  
        m = ...  
        if x == tab[m]:  
            return ...  
        if x > tab[m]:  
            debut = m + 1  
        else:  
            fin = ...  
    return ...
```

Exemples :

```
>>> dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33],28)  
True  
>>> dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33],27)  
False
```

```
def multiplication(n1, n2):  
    if n1 == 0 or n2 == 0:  
        return 0  
  
    if n1 == 1:  
        return n2  
  
    if n1 > 1:  
        return n2 + multiplication(n1 - 1, n2)  
    else:  
        return multiplication(n1 + 1, n2) - n2  
  
print(multiplication(3, -5))
```

```
def dichotomie(tab, x):
    """
        tab : tableau d'entiers trié dans l'ordre croissant
        x : nombre entier
        La fonction renvoie True si tab contient x et False sinon
    """

    debut = 0
    fin = len(tab) - 1
    while debut <= fin:
        m = (fin + debut) // 2
        if x == tab[m]:
            return True
        if x > tab[m]:
            debut = m + 1
        else:
            fin = m - 1
    return False

assert dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33], 28) == True
assert dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33], 27) == False
assert dichotomie([], 27) == False
assert dichotomie([27], 27) == True
```