

BACCALAURÉAT

SESSION 2023

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°09

DURÉE 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 relatifs `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)

Soit `tab` un tableau non vide d'entiers triés dans l'ordre croissant et `n` un entier.

La fonction `chercher` ci-dessous doit renvoyer un indice où la valeur `n` apparaît dans `tab` si cette valeur y figure et `None` sinon.

Les paramètres de la fonction sont :

- `tab`, le tableau dans lequel s'effectue la recherche ;
- `n`, l'entier à chercher dans le tableau ;
- `i`, l'indice de début de la partie du tableau où s'effectue la recherche ;
- `j`, l'indice de fin de la partie du tableau où s'effectue la recherche.

L'algorithme demandé est une recherche dichotomique récursive.

Recopier et compléter le code de la fonction `chercher` suivante :

```
def chercher(tab, n, i, j):
    if i < 0 or j > len(tab):
        return None
    elif i > j:
        return None
    m = (i + j) // ...
    if ... < n:
        return chercher(tab, n, ..., ...)
    elif ... > n:
        return chercher(tab, n, ..., ...)
    else:
        return ...
```

L'exécution du code doit donner :

```
>>> chercher([1, 5, 6, 6, 9, 12], 7, 0, 10)
None
>>> chercher([1, 5, 6, 6, 9, 12], 7, 0, 5)
None
>>> chercher([1, 5, 6, 6, 9, 12], 9, 0, 5)
4
>>> chercher([1, 5, 6, 6, 9, 12], 6, 0, 5)
2
```