

# **BACCALAURÉAT**

**SESSION 2023**

---

**Épreuve de l'enseignement de spécialité**

## **NUMÉRIQUE et SCIENCES INFORMATIQUES**

**Partie pratique**

**Classe Terminale de la voie générale**

---

**Sujet n°02**

---

**DURÉE DE L'ÉPREUVE : 1 heure**

**Le sujet comporte 2 pages numérotées de 1 / 2 à 2 / 2  
Dès que le sujet vous est remis, assurez-vous qu'il est complet.**

*Le candidat doit traiter les 2 exercices.*

## EXERCICE 1 (4 points)

Écrire une fonction `indices_maxi` qui prend en paramètre une liste `tab`, non vide, de nombres entiers et renvoie un couple donnant d'une part le plus grand élément de cette liste et d'autre part la liste des indices de la liste `tab` où apparaît ce plus grand élément.

Exemples :

```
>>> indices_maxi([1, 5, 6, 9, 1, 2, 3, 7, 9, 8])
(9, [3, 8])
```

```
>>> indices_maxi([7])
(7, [0])
```

## EXERCICE 2 (4 points)

Cet exercice utilise des piles qui seront représentées en Python par des listes (de type `list`).

On rappelle que l'expression `liste_1 = list(liste)` fait une copie de `liste` indépendante de `liste`, que l'expression `x = liste.pop()` enlève le sommet de la pile `liste` et le place dans la variable `x` et, enfin, que l'expression `liste.append(v)` place la valeur `v` au sommet de la pile `liste`.

Compléter le code Python de la fonction `positif` ci-dessous qui prend une pile `liste` de nombres entiers en paramètre et qui renvoie la pile des entiers positifs dans le même ordre, sans modifier la variable `liste`.

```
def positif(pile):
    pile_1 = ... (pile)
    pile_2 = ...
    while pile_1 != []:
        x = ...
        if ... >= 0:
            pile_2.append(...)
    while pile_2 != ...:
        x = pile_2.pop()
        ...
    return pile_1
```

Exemples :

```
>>> positif([-1, 0, 5, -3, 4, -6, 10, 9, -8])
[0, 5, 4, 10, 9]
```

```
>>> positif([-2])
[]
```