

"""

Author: Pascal Padilla

Source: correction de l'exercice 1 du sujet 16 des épreuves pratiques NSI 2022

"""

```
from doctest import testmod
```

```
def maxi(tab: list) -> tuple[int, int]:  
    """Renvoie le plus grand élément du tableau 'tab'  
    et son indice.
```

```
    Args:
```

```
        tab (list): tableau à parcourir
```

```
    Returns:
```

```
        tuple[int, int] : couple de nombres entiers :  
            * valeur maximale du tableau  
            * indice de cette valeur
```

```
    Exemples et tests:
```

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

```
# invariant de boucle
```

```
# * v_max est la valeur maximale de la portion tab[0..i]
```

```
# * i_max est l'indice de v_max
```

```
# initialisation (avec la première valeur du tableau):
```

```
# * i ↗ 0
```

```
# * v_max ↗ tab[0]
```

```
# * i_max ↗ 0
```

```
i_max = 0
```

```
v_max = tab[0]
```

```
# condition d'arrêt (tout le tableau est parcouru):
```

```
# * i == len(tab)
```

```
for i in range(1, len(tab)):
```

```
    # mise à jour de v_max et i_max si la
```

```
    # valeur courante du tableau est plus grande
```

```
    # que le maximum trouvé jusqu'à présent
```

```
    if tab[i] > v_max:
```

```
        i_max = i
```

```
        v_max = tab[i]
```

```
# sortie de boucle : tout le tableau est parcouru
```

```
# et grâce aux invariants, nous avons la valeur
```

```
# maximale et son indice.
```

```
return (v_max, i_max)
```

```
# Tests de l'énoncé: méthode classique
```

```
assert maxi([1,5,6,9,1,2,3,7,9,8]) == (9,3)
```

```
# Tests de l'énoncé: méthode avec doctest
```

```
testmod()
```