

```
"""
Author: Pascal Padilla
Source: correction de l'exercice 1 du sujet 11 des épreuves pratiques NSI 2022
"""
```

```
from doctest import testmod
```

```
def recherche(tab: list, n: int) -> int:
    """Recherche dichotomique dans un tableau trié tab
    de l'élément n.

    Args:
        tab (list): tableau trié de nombres entiers non vide
        n (int): nombre entier à chercher

    Returns:
        int: si n est dans tab, renvoie l'indice
            sinon renvoie -1

    Exemples et tests:
    >>> recherche([2, 3, 4, 5, 6], 5)
    3
    >>> recherche([2, 3, 4, 6, 7], 5)
    -1
    """
    i = 0
    j = len(tab) - 1

    # Invariants à chaque tour de boucle :
    # * les éléments avant l'indice i sont inférieurs à n
    # * les éléments après l'indice j sont supérieurs à n
    # * si n est dans tab, alors il est dans tab[i..j]
    # Initialisation de i et de j
    # * i est 0 et j est dernier indice
    # Condition d'arrêt de la boucle:
    # * soit on trouve n
    # * soit tab[i..j] devient l'intervalle vide
    #     ; i dépasse j
    #     ; i > j

    while not i > j:
        # principe de la recherche dichotomique
        # comparer n avec la valeur médiane de tab[i..j]

        # indice médian
        i_median = (i + j) // 2

        # comparaison

        # si on trouve la valeur cherchée
        # on arrête le programme/la boucle
        # et on renvoie l'indice
        if tab[i_median] == n:
            return i_median

        # non trouvé donc
        # * si n est supérieur alors on cherche dans la
        #     zone qui succède à l'élément médian
        if n > tab[i_median]:
            i = i_median + 1

        # * sinon n est inférieur, alors on cherche dans la
        #     zone qui précède l'élément médian
        else:
            j = i_median - 1

    # Variant de boucle : l'intervalle [i..j] diminue à chaque fois

    # si on sort de la boucle,
    # c'est que l'élément n'a pas été trouvé
    return -1
```

```
# Tests avec des assertions
assert recherche([2, 3, 4, 5, 6], 5) == 3
assert recherche([2, 3, 4, 6, 7], 5) == -1

# Tests avec la bibliothèque doctest
testmod()
```