

- 1_SPE NSI_4 DS03 Recherche Dichotomique Codage texte	Le 24/01/2022 Durée: 1h55
Calculatrice non autorisée Aucun document autorisé Réponses sur feuilles doubles numérotées	

Exercice 1: (10 points) <https://www.mathexien.com>

1.1. Décrire ce qu'est un système d'encodage de caractères et donner le nom de deux systèmes d'encodage.

1.2. Ecrire une instruction qui affiche sur une ligne: le caractère "a" , son code en décimal, son code en hexadécimal et son code en binaire.

```
>>> a 97 0x61 0b110001
```

1.3. Implémenter une fonction qui convertit une lettre minuscule en majuscule.

```
>>> assert majuscule( "a" ) == "A"
```

1.4. Implémenter une fonction qui renvoie le minimum et le maximum d'une liste de nombres donnée en paramètre.

```
>>> assert minMax( [ 7 , 4 , 2 , 10 , 5 ] ) == [ 2 , 10 ]
```

1.5. Implémenter une fonction qui recherche si une valeur est dans une liste de nombres donnée en paramètre. La liste est non triée.

```
>>> assert recherche( [ 7 , 4 , 2 , 10 , 5 ] , 5 ) == True  
>>> assert recherche( [ 7 , 4 , 2 , 10 , 5 ] , 1 ) == False
```

1.6. Implémenter une fonction qui recherche linéairement si une valeur est dans une liste de nombres données en paramètre. La liste est triée en ordre croissant. Votre implémentation doit en tirer bénéfice.

```
>>> assert recherche( [ 2 , 4 , 5 , 7 , 10 ] , 5 ) == True  
>>> assert recherche( [ 2 , 4 , 5 , 7 , 10 ] , 1 ) == False
```

Licence CC BY-NC-SA 4.0

Annexe

```
1 # 1 spé NSI - DS03 - Exercice 2
2
3 def dichotomie(tab: list, x: int) -> bool:
4     '''
5     Paramètres:
6     -----
7     tab (list): tableau trié dans l'ordre croissant
8     x (int): nombre entier
9     Retour:
10    -----
11    Renvoie True si tab contient x et False sinon
12    '''
13    # YYY
14    if x < tab[0] :
15        return False
16
17    # YYY
18    if (x < tab[0]) or x > tab[-1] :
19        return False
20
21    debut = 0 # YYY
22    fin = len(tab) - 1 # YYY
23    # YYY
24    while debut <= fin:
25        m = (debut + fin) // 2 # YYY
26        # YYY
27        if x == tab[m] :
28            return True
29        # YYY
30        if x > tab[m] :
31            debut = m + 1
32        # YYY
33        else :
34            fin = m
35    return False
36
37
38 assert dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33], 28) == True
39 assert dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33], 27) == False
40 assert dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33], 1) == False
41 assert dichotomie([], 28) == False
```