

EXERCICE 2 (4 points)

Principaux thèmes abordés : algorithmique (recherche dichotomique) et langages et programmation (récursivité)

On veillera à mettre sur la copie toutes les réponses.

Partie A : La recherche dichotomique

1. La recherche d'un élément dans un tableau avec une méthode dichotomique ne peut se faire que si le tableau est trié.
 - a) Vrai
 - b) Faux
2. Le coût d'un algorithme de recherche dichotomique est :
 - a) Constant : Complexité $O(1)$
 - b) Linéaire : Complexité $O(n)$
 - c) Logarithmique : Complexité $O(\log(n))$
3. Justifier pourquoi l'entier `fin - deb` est un variant de boucle qui montre la terminaison du programme de recherche dichotomique de l'**annexe 1 de l'exercice 2**.

Partie B : La recherche dichotomique itérative

Le programme de recherche dichotomique de l'**annexe 1 de l'exercice 2** est utilisé pour effectuer des recherches dans une liste.

Dans l'ensemble de cette partie, on considère la liste : `lnoms = ["alice", "bob", "etienne", "hector", "lea", "nathan", "paul"]`.

1. Expliquer pourquoi en ligne 2, on a «`fin = len(liste) - 1`» plutôt que «`fin = 6`».
2. En Python, l'opérateur `//` donne le quotient de la division euclidienne de deux nombres entiers.
Proposer un algorithme pour obtenir ce quotient.
3. Donner la trace complète de l'exécution `rechercheDicho("lea", lnoms)` en complétant le tableau ci-dessous sur votre copie :

Variables			Condition	Valeur renvoyée
deb	Fin	M	<code>deb <= fin</code>	

4. Sur votre copie, modifier le code du corps de la fonction `rechercheDicho()` pour qu'elle renvoie aussi la position (indice) de l'élément cherché ou `-1` si l'élément n'est pas trouvé.

On pourra indiquer sur la copie le numéro des lignes modifiées, à supprimer ou à insérer s'il y a lieu.

Partie C : La recherche dichotomique récursive

1. Donner la définition d'une fonction récursive en programmation.
2. Écrire en langage naturel ou en python, l'algorithme de recherche dichotomique d'un élément dans une liste, triée de façon croissante, en utilisant une méthode récursive. Il renverra `True` si l'objet a été trouvé, `False` sinon.

Exercice 2 - Annexe 1

On considère la fonction de recherche dichotomique suivante :

```
def rechercheDicho (elem, liste):
    """
    Cette fonction indique si un élément se trouve dans un
    tableau.
    Elle utilise la méthode de recherche dichotomique.
    Elle prend en arguments :
    - elem : élément à rechercher de type string
    - liste : liste d'éléments de type string triée
    par ordre croissant
    Elle renvoie un booléen correspondant à la présence ou
    non de l'élément
    """
1     deb = 0
2     fin = len(liste)-1
3     m = (deb+fin)//2
4     while deb <= fin :
5         if liste[m] == elem :
6             return True
7         elif liste[m] > elem :
8             fin = m-1
9         else :
10            deb = m+1
11            m = (deb+fin)//2
12    return False
```