BACCALAUREAT

SESSION 2022

Épreuve de l'enseignement de spécialité

NUMERIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°35

DUREE DE L'EPREUVE : 1 heure

Le sujet comporte 3 pages numérotées de 1 / 3 à 3 / 3
Dès que le sujet vous est remis, assurez-vous qu'il est complet.

Le candidat doit traiter les 2 exercices.

EXERCICE 1 (4 points)

Ecrire une fonction qui prend en paramètre un tableau d'entiers non vide et qui renvoie la moyenne de ces entiers. La fonction est spécifiée ci-après et doit passer les assertions fournies.

```
def moyenne (tab):
    '''
        moyenne(list) -> float
        Entrée : un tableau non vide d'entiers
        Sortie : nombre de type float
        Correspondant à la moyenne des valeurs présentes dans le
        tableau
    '''

assert moyenne([1]) == 1
assert moyenne([1,2,3,4,5,6,7]) == 4
assert moyenne([1,2]) == 1.5
```

EXERCICE 2 (4 points)

Le but de l'exercice est de compléter une fonction qui détermine si une valeur est présente dans un tableau de valeurs triées dans l'ordre croissant.

L'algorithme traite le cas du tableau vide.

L'algorithme est écrit pour que la recherche dichotomique ne se fasse que dans le cas où la valeur est comprise entre les valeurs extrêmes du tableau.

On distingue les trois cas qui renvoient `False` en renvoyant `False,1` , `False,2` et `False,3`.

Compléter l'algorithme de dichotomie donné ci-après.

```
def dichotomie(tab, x):
    """
        tab : tableau trié dans l'ordre croissant
        x : nombre entier
        La fonction renvoie True si tab contient x et False sinon
    """
    # cas du tableau vide
    if ...:
        return False,1

    # cas où x n'est pas compris entre les valeurs extrêmes
    if (x < tab[0]) or ...:
        return False,2
```

```
    debut = 0
    fin = len(tab) - 1
    while debut <= fin:
        m = ...
        if x == tab[m]:
            return ...
        if x > tab[m]:
            debut = m + 1
        else:
            fin = ...
    return ...
```

Exemples :

```
>>> dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33],28)
True
>>> dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33],27)
(False, 3)
>>> dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33],1)
(False, 2)
>>> dichotomie([],28)
(False, 1)
```

```python
def moyenne(tab):
    '''
        moyenne(list) -> float
        Entrée : un tableau non vide d'entiers
        Sortie : nombre de type float
        Correspondant à la moyenne des valeurs présentes dans le
        tableau
    '''
    somme = 0
    n = len(tab)
    for i in range(n):
        somme += tab[i]
    return somme/n

assert moyenne([1]) == 1
assert moyenne([1,2,3,4,5,6,7]) == 4
assert moyenne([1,2]) == 1.5
```

```python
def dichotomie(tab, x):
    """
        tab : tableau trié dans l¿ordre croissant
        x : nombre entier
        La fonction renvoie True si tab contient x et False sinon
    """
    # cas du tableau vide
    if tab == [] :
        return False, 1

    # cas où x n'est pas compris entre les valeurs extrêmes
    if (x < tab[0]) or (x > tab[len(tab)-1]):
        return False, 2

    debut = 0
    fin = len(tab) - 1
    while debut <= fin:
        m = (debut + fin) // 2
        if x == tab[m]:
            return True
        if x > tab[m]:
            debut = m + 1
        else:
            fin = m - 1
    return False, 3


assert dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33],28) == True
assert dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33],27) == (False, 3)
assert dichotomie([15, 16, 18, 19, 23, 24, 28, 29, 31, 33],1) == (False, 2)
assert dichotomie([],28) == (False, 1)
```