TD_parcours_sequentiel_de_tableau

April 8, 2021

1 Algorithme de parcours d'un tableau

1.1 Objectif du TP

Ecrire en langage python les fonctions suivantes

[2]:
```
def recherche(liste, valeur):
    """
    Description de la fonction : Détermine si la valeur est présente dans la␣
 ↪liste
    paramètre liste (list)
    paramètre valeur (type quelconque)
    Return (bool)
    """
    # A compléter

def recherche_indice(liste, valeur):
    """
    Description de la fonction : Renvoie l'indice de la valeur recherchée (None␣
 ↪si valeur non présente)
    paramètre liste (list) : liste non vide
    paramètre valeur (type quelconque)
    Return (int ou Nonetype)
    """
    # A compléter

def recherche_minimum(liste):
    """
    Description de la fonction : Renvoie la plus petite valeur de la liste
    paramètre liste (list) : liste non vide d'entiers(int), de nombres␣
 ↪réels(float) ou de chaîne de caractères(str)
    Return (int)
    """
    # A compléter

def recherche_maximum(liste):
    """
    Description de la fonction : Renvoie la plus grande valeur de la liste
```

```
    paramètre liste (list) : liste non vide d'entiers(int), de nombres␣
↪réels(float) ou de chaîne de caractères(str)
    Return (int)
    """
    # A compléter

def moyenne(liste):
    """
    Description de la fonction : Calcule la moyenne des valeurs contenues dans␣
↪la liste
    paramètre liste (list) : liste non vide d'entiers(int) ou de nombres␣
↪réels(float)
    Return (int)
    """
    # A compléter
```

1.2 Travail à faire

Pour atteindre l'objectif, il faut :

1. **Ecrire** l'algorithme **en français** (façon "recette de cuisine"). Cette étape est indispensable : ne pas passer à la question suivante sans l'avoir fait!!
2. **Traduire** votre algorithme en langage python sur ordinateur. Normalement, si votre algorithme a bien été écrit à la question précédente, cette étape est simple : il suffit simplement de respecter la syntaxe python.
3. **Tester** le bon fonctionnement de votre fonction sur quelques listes (voir ci-dessous).

1.2.1 Indices :

D'après le cours d'introduction aux algorithmes :

- Faut-il parcourir les éléments de la liste OU parcourir les indices des éléments de la liste ?
- Certaines **astuces** peuvent être nécessaires (compteur, accumulateur)

1.3 Tests des fonctions

Pour tester vos fonctions, vous pouvez générer des listes d'entiers tirés au hasard. (Le code est donné ci-dessous)

[3]:
```
# Création d'une liste de 100 entiers tirés au hasard entre 0 et 500
from random import randrange
liste = []
for i in range(100):
    liste.append(randrange(500))

print(liste)
```

```
[241, 214, 95, 228, 81, 462, 314, 285, 202, 293, 3, 397, 62, 345, 75, 488, 162,
344, 206, 163, 257, 67, 255, 75, 384, 75, 89, 262, 275, 449, 431, 90, 27, 65,
```

```
301, 291, 135, 45, 448, 200, 409, 186, 469, 340, 25, 100, 404, 205, 368, 494,
21, 454, 186, 339, 262, 341, 358, 45, 198, 265, 300, 195, 89, 338, 37, 379, 391,
250, 357, 269, 355, 145, 198, 181, 249, 275, 457, 74, 420, 283, 239, 376, 460,
395, 199, 207, 17, 362, 176, 406, 44, 293, 73, 117, 457, 329, 266, 407, 43, 374]
```

1.3.1 Exemples de tests

[5]:
```
recherche(liste, 35)
```

[5]:
```
False
```

[6]:
```
recherche_indice(liste, 45)
```

[6]:
```
37
```

[7]:
```
recherche_minimum(liste)
```

[7]:
```
3
```

[8]:
```
recherche_maximum(liste)
```

[8]:
```
494
```

[9]:
```
moyenne(liste)
```

[9]:
```
248.32
```