

# CONCOURS BLANC : INFORMATIQUE

## AUTOUR DE DONNÉES MÉTÉOROLOGIQUES

### CORRIGÉ

## 1 Variations autour du minimum

### Question 1

Corrigé

- $\text{len}(t) : 5$
- Valeurs prises par  $i : 0, 1, 2, 3, 4$

Itération $i$	$t[i]$	$p$	$t \leq p ?$	$p$
0	6	6	OUI	6
1	2	6	OUI	2
2	15	2	NON	2
3	2	2	OUI	2
4	15	2	NON	2

- Valeur retournée : 2

### Question 2

Corrigé

Invariant de boucle proposé : à chaque itération de boucle  $i$ ,  $p$  contient le minimum de la liste  $t[0 : i]$ ,  $i$  inclus.

1. L'algorithme faisant appel à une boucle for dans laquelle la taille de  $t$  ne varie jamais et dans laquelle  $i$  est incrémentée à chaque itération, la boucle se terminera après  $\text{len}(t)$  itérations.
2. Avant d'entrer dans la boucle,  $p = t[0]$ .
3. À la première itération, en entrant dans la boucle,  $p = t[0]$ .  $p$  est bien la valeur minimale de la portion de tableau de 1 élément.
4. À la  $i$ ème itération, on considère que  $p$  est le minimum de la liste  $t[0 : i]$ . À l'itération  $i+1$ , si  $p$  est inférieur à  $t[i+1]$  alors la  $p$  prend la valeur de  $t[i+1]$ .  $p$  est donc la valeur minimale de la liste  $t[0 : i+1]$ .
5. À l'itération  $\text{len}(t)$ ,  $p$  est donc la valeur minimale de  $t$ .

### Question 3

Corrigé

Au cours de l'algorithme, on parcourt une seule fois chacune des valeurs de la liste. La complexité de l'opération de recherche du minimum est donc linéaire  $\mathcal{O}(n)$ .

### Question 4

Corrigé

La ligne 7 peut être remplacée par `if t[i] >= p :`

### Question 5

Il ne faut plus stocker la valeur minimale mais l'indice de la valeur minimale dans la liste.

```
def position_mini(t):
    """Calcule l'indice du minimum d'un tableau d'entiers
    ou de flottants."""
    if len(t) == 0:
        return None
    p = 0
    for i in range(len(t)):
        # Ou for i in range(1, len(t)):
        if t[i] <= t[p]:
            p = i
    return p
```

Corrigé



### Question 6

Dans la question précédente, l'indice renvoyé est l'indice de la dernière valeur rencontrée.

Si on souhaite retourner le premier indice, il faut utiliser une inégalité stricte : `if t[i] < p :`

Corrigé

### Question 7

Pour rechercher un minimum dans une liste de liste, il faut rechercher le minimum dans chacune des sous-listes. Pour cela, il faut donc avoir recours à deux boucles imbriquées.

Corrigé

### Question 8

```
def mini2D(t):
    if len(t) == 0:
        return None
    p = t[0][0]
    for i in range(len(t)):
        m = mini(t[i])
        if m < p:
            p = m
    return p
```

Corrigé



### Question 9

Corrigé

Pour une liste de taille  $n \times n$  il faut rechercher la valeur minimale en utilisant la fonction `mini` dans  $n$  listes. La complexité est donc quadratique et donc  $C(n) = \mathcal{O}(n^2)$ .

### Question 10

Corrigé



```
def chaine_mini(t):
    if len(t) == 0:
        return None
    p = 0
    for i in range(len(t)):
        if t[i][1] < t[p][1]:
            p = i
    return t[p][0]
```

1  
2  
3  
4  
5  
6  
7  
8

### Question 11

Corrigé



```
def majores_par(t,):
    cpt = 0
    for i in range(len(t)):
        if t[i] < seuil:
            cpt = cpt + 1
    return cpt
```

1  
2  
3  
4  
5  
6

## 2 Manipulation de données

### Question 12

Corrigé

Un caractère est codé par un octet en ASCII. En comptant 3 caractères pour le numéro du jour, 4 caractères pour la température minimale et 4 caractères pour la température maximale, les 2 points virgules et le retour chariot, il faut donc 14 octets pour coder une ligne et 5110 octets pour stocker les mesures sur une année.

On peut ajouter les 17 octets d'en-tête.

### Question 13

Corrigé

La ligne 5 permet d'ignorer les lignes (de commentaires) contenant le caractère `#`. La ligne 6 permet de diviser une ligne en 3 chaînes de caractères. La ligne est séparée par les points virgules.

Corrigé

jours est une liste d'entiers. Tmin et Tmax sont des listes de flottants.

#### Question 14

Corrigé

La méthode append permet d'ajouter un élément à la liste sans changer sa référence (son adresse).

Lors de la concaténation, un nouvel objet (avec une nouvelle adresse est crée). On peut supposer que lorsque la liste grandie, une concaténation et la création d'un nouvel objet prendra davantage de temps que l'utilisation de la méthode append.

#### Question 15

Corrigé



```
def moyenne(a,b):
    if (len(a)!=len(b)):
        return None
    moy = []
    for i in range(len(a)):
        moy.append((a[i]+b[i])/2)
    return moy
```

1  
2  
3  
4  
5  
6  
7

#### Question 16

Corrigé

```
>>> Tmoy = moyenne(Tmin,Tmax)
```

1

#### Question 17

Corrigé

```
>>> nb_jours_ge=majores_par(Tmoy,0)
```

1

#### Question 18

Corrigé

```
import numpy as np
import matplotlib.pyplot as plt
plt.plot(t,Tmin)
plt.plot(t,Tmax)
plt.plot(t,Tmoy)
plt.show()
```

1  
2  
3  
4  
5  
6