

BACCALAURÉAT

SESSION 2024

Épreuve de l'enseignement de spécialité

NUMÉRIQUE et SCIENCES INFORMATIQUES

Partie pratique

Classe Terminale de la voie générale

Sujet n°35

DURÉE DE L'ÉPREUVE : 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 (10 points)

On a relevé les valeurs moyennes annuelles des températures à Paris pour la période allant de 2013 à 2019. Les résultats ont été récupérés sous la forme de deux tableaux (de type `list`) : l'un pour les températures, l'autre pour les années :

```
t_moy = [14.9, 13.3, 13.1, 12.5, 13.0, 13.6, 13.7]
annees = [2013, 2014, 2015, 2016, 2017, 2018, 2019]
```

Écrire la fonction `annee_temperature_minimale` qui prend en paramètres ces deux tableaux et qui renvoie la plus petite valeur relevée au cours de la période et l'année correspondante.

On suppose que la température minimale est atteinte une seule fois.

Exemple :

```
>>> annee_temperature_minimale(t_moy, annees)
(12.5, 2016)
```

EXERCICE 2 (10 points)

Un mot palindrome peut se lire de la même façon de gauche à droite ou de droite à gauche : *kayak*, *radar*, et *non* sont des mots palindromes.

De même certains nombres ont des écritures décimales qui sont des palindromes : 33, 121, 345543.

L'objectif de cet exercice est d'obtenir un programme Python permettant de tester si un nombre est un nombre palindrome.

Pour remplir cette tâche, on vous demande de compléter le code des trois fonctions ci-dessous qui s'appuient les unes sur les autres :

- `inverse_chaine` : qui renvoie une chaîne de caractères inversée ;
- `est_palindrome` : qui teste si une chaîne de caractères est un palindrome ;
- `est_nombre_palindrome` : qui teste si un nombre est un palindrome.

Compléter le code des trois fonctions ci-dessous.

```
def inverse_chaine(chaine):  
    '''Retourne la chaine inversée'''  
    resultat = ...  
    for caractere in chaine:  
        resultat = ...  
    return resultat  
  
def est_palindrome(chaine):  
    '''Renvoie un booléen indiquant si la chaine ch  
    est un palindrome'''  
    inverse = inverse_chaine(chaine)  
    return ...  
  
def est_nombre_palindrome(nombre):  
    '''Renvoie un booléen indiquant si le nombre nombre  
    est un palindrome'''  
    chaine = ...  
    return est_palindrome(chaine)
```

Exemples :

```
>>> inverse_chaine('bac')  
'cab'  
>>> est_palindrome('NSI')  
False  
>>> est_palindrome('ISN-NSI')  
True  
>>> est_nombre_palindrome(214312)  
False  
>>> est_nombre_palindrome(213312)  
True
```