

Evaluation de PALMINTERI Ryan (N°2)

November 1, 2021

Observations

Parfait; 8/8

Note: 20/20

1 Epreuve de l'enseignement de spécialité NSI : Partie Pratique

Exercice 1 (4 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 listes : l'une 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]
```

Ecrire la fonction `mini` qui prend en paramètres le tableau `releve` des relevés et le tableau `date` des dates et qui renvoie la plus petite valeur relevée au cours de la période et l'année correspondante.

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

      def mini(releve, date):
          assert (releve != None) and (annees != None)
          assert len(releve) == len(annees)

          result_index = 0
          max_it = min(len(releve), len(annees))

          for i in range(0, max_it):
              if releve[result_index] > releve[i]:
                  result_index = i
          return (releve[result_index], annees[result_index])
```

```
[9]: # Suite de tests
      assert mini(t_moy, annees) == (12.5, 2016)
```

Commentaires

Code correct. A la ligne 12, on aurait pu commencer l'itération à l'index 1 au lieu de 0 !

Points: 4/4

Exercice 2 (4 points) Un mot palindrome peut se lire de la même façon de gauche à droite ou de droite à gauche : **bob**, **radar**, et **non** sont des mots palindromes.

De même certains nombres sont eux aussi 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 sachant que la fonction `est_nbre_palindrome` s'appuiera sur la fonction `est_palindrome` qui elle-même s'appuiera sur la fonction `inverse_chaine`.

La fonction `inverse_chaine` inverse l'ordre des caractères d'une chaîne de caractères `chaine` et renvoie la chaîne inversée.

La fonction `est_palindrome` teste si une chaîne de caractères `chaine` est un palindrome. Elle renvoie `True` si c'est le cas et `False` sinon. Cette fonction s'appuie sur la fonction précédente.

La fonction `est_nbre_palindrome` teste si un nombre `nbre` est un palindrome. Elle renvoie `True` si c'est le cas et `False` sinon. Cette fonction s'appuie sur la fonction précédente.

```
[10]: def inverse_chaine(chaine):
        result = ""
        for caractere in chaine:
            result = caractere + result
        return result

    def est_palindrome(chaine):
        inverse = inverse_chaine(chaine)
        return chaine == inverse

    def est_nbre_palindrome(nbre):
        chaine = str(nbre)
        return est_palindrome(chaine)
```

```
[11]: # Suite de tests
    assert inverse_chaine('bac') == 'cab'
    assert est_palindrome('NSI') == False
    assert est_palindrome('ISN-NSI') == True
    assert est_nbre_palindrome(214312) == False
    assert est_nbre_palindrome(2133312) == True
```

Commentaire

Parfait.

Points: 4/4