

Coding Python Party Challenge

édition 2020

Algorithme de Luhn



Exercice 9 ★ ★ ★

Algorithme de Luhn

Énoncé

Sur un site de ventes en ligne, il peut être utile de vérifier la validité des numéros de cartes bancaires.

On peut pour cela utiliser l'algorithme de Luhn.

Pour un numéro de carte donné, son fonctionnement est le suivant :

1. On multiplie par deux un chiffre sur deux du premier vers le dernier en commençant par le premier.
Si une fois multiplié le chiffre est supérieur à neuf, retirer neuf.
2. Additionner tous les chiffres obtenus entre-eux.
3. On divise le résultat par 10. Si le reste de la division est égal à 0, alors le numéro de carte est valide.

Exemple :

Prenons le numéro de carte suivant : 6703 1111 2222 3334.

N° de carte	6	7	0	3	1	1	1	1	2	2	2	2	3	3	3	4
Multiplier par 2 (1 chiffre sur 2 du premier vers le dernier, en commençant par le premier)	6 x 2 = 12	7	0 x 2 = 0	3	1 x 2 = 2	1 x 2 = 2	1 x 2 = 2	1 x 2 = 2	2 x 2 = 4	2	2 x 2 = 4	2	3 x 2 = 6	3	3 x 2 = 6	4
Si chiffre > 9, retirer 9	12 - 9 = 3	7	0	3	2	1	2	1	4	2	4	2	6	3	6	4
Somme de tous les chiffres	3 + 7 + 0 + 3 + 2 + 1 + 2 + 1 + 4 + 2 + 4 + 2 + 6 + 3 + 6 + 4 = 50 50															
Est divisible par 10 ? (signifie que le reste de la division est égal à 0)	50 mod 10 = 0 VRAI															
Conclusion	Le numéro de carte est valide.															

Créer un programme permettant de tester la validité d'un numéro de carte bancaire donné en entrée. Il retournera True si le numéro est valide, False sinon. Assurez-vous également que le numéro comporte bien 16 chiffres.

Données fournies en entrée

Les données fournies en entrée dans la variable `lines` sont :

```
lines[0]
```

```
# Numéro de carte bancaire quelconque (type : chaîne de caractères)
```

Exemples d'exécution

Valeur de la variable <code>lines</code>	Résultat attendu
<code>lines = ["6703111122223334"]</code>	True
<code>lines = ["6803111122223333"]</code>	True
<code>lines = ["6776111854223134"]</code>	False
<code>lines = ["6103111122423339"]</code>	False