

TP-Le code ISBN

Nom : _____

Le code ISBN d'un livre

L'ISBN est un numéro international normalisé permettant l'identification d'un livre dans une édition donnée.

ISBN est l'acronyme d'International Standard Book Number

L'ISBN a été conçu pour simplifier le traitement informatisé des livres : les libraires peuvent passer des commandes standardisées, les distributeurs ont le même code pour traiter les commandes et les retours, les différentes opérations de gestion dans les bibliothèques et centres de documentation sont également facilitées. Par ailleurs, le caractère international de cette numérotation constitue, à l'étranger également, une référence unique pour tous les professionnels du livre.

L'ISBN identifie donc de façon unique un livre quel que soit son support de publication, imprimé ou numérique.

Deux livres peuvent avoir le même titre, mais auront des codes ISBN différents

L'ISBN est un numéro à 13 chiffres depuis le 1er janvier 2007 (les ouvrages antérieurs avaient un ISBN à 10 chiffres).



Les parties de l'ISBN: 978 2 1234 5680 3

- **978**, signifie que l'article est un livre ('Bookland') (979 sera aussi employé en fonction des besoins...)
Dans un code barre, les 3 premiers chiffres indiquent le pays producteur, comme 978 et 979 ne sont pas utilisés, on les a attribués aux livres...
- La deuxième partie de l'ISBN est le numéro identifiant le groupe national, linguistique ou géographique. Le 2 identifie la francophonie dans son ensemble.
- **1234**: C'est un numéro qui identifie un éditeur en particulier.
- **5680**: C'est le numéro d'identification d'une édition donnée d'une publication produite par un éditeur en particulier.
- **3**: La cinquième partie de l'ISBN est **la clé de contrôle**. Il s'agit de la partie finale de l'ISBN qui est reliée aux précédents caractères de la chaîne par un algorithme mathématique et qui permet d'en vérifier la validité.
La clé peut varier de 0 à 9.

Le problème :

Étant donné un code ISBN, est-il valide?

Dans un premier temps, examinons comment est calculée la clé de contrôle:

Les 12 premiers chiffres de l'ISBN (n)	9	7	8	2	1	2	3	4	5	6	8	0
Pondération (p)	1	3	1	3	1	3	1	3	1	3	1	3
Multiplication $n \times p$	9	21	8	6	1	6	3	12	5	18	8	0

 **À FAIRE 1:**

On additionne les résultats de la dernière ligne :

On calcule le reste de la division par 10 de la somme obtenue :

Si ce reste est 0, alors la clé de contrôle est 0, sinon la clé de contrôle est alors le résultat de $10 - \text{reste}$, soit ici :

Donc en comparant le résultat et la clé de contrôle du code ISBN, on peut affirmer que le code ISBN est valide ou pas.

C'est un cas typique où l'informatique est d'une aide précieuse...

- Toute personne pourra vérifier la validité d'un code ISBN, sans connaître les calculs à faire.
- On entre le code ISBN dans une zone de saisie, et un programme nous informe de la validité du code...

Dans un second temps, écrivons un **algorithme** qui permet de savoir si un code ISBN est valide ou pas.

Il s'agit de décrire **la méthode** pour résoudre ce problème:

 **À FAIRE 2:**

Quelles sont les données en entrées?

La variable somme reçoit

La variable reste reçoit

Si la variable reste est égale à 0 alors la variable cle reçoit

Sinon la variable cle reçoit

*Si la variable cle est égale
tourne*

alors on retourne

Sinon on re-

Dans un troisième temps, il faudra traduire cet algorithme dans un langage de programmation (Python dans notre cas).

Rappelons l'algorithme:

```
Données : code ISBN
( nombres de 13 chiffres )
Résultat : Validité du code
code_isbn ← le code ISBN
cle_controle ← le 13ème chiffre
cle_calculée ← le calcul de la clé
On teste la validité de la clé
retourner validité
```

Des questions se posent :

```
Comment représenter l'ISBN?
Comment atteindre le 13ème chiffres?
Comment calculer la clé?
Comment calculer une somme?
Comment calculer le reste?
...
```

Représentation du code ISBN:

Comme c'est l'utilisateur qui donne le code ISBN, l'entrée sera une chaîne de caractères.

```
code=input("Entrez le code à 13 chiffres")
print(type(code))
```

Produira :

```
<class 'str'>
```

Une chaîne de caractère étant itérable, on peut accéder à chacun des caractères via leurs indices.

Par exemple : Pour accéder au dernier caractère, puisque nous connaissons la longueur de la chaîne, on utilise l'instruction `code[12]` (le 1^{er} étant d'indice 0, le 13^{ème} est d'indice 12)

On peut aussi et cela est pratique lorsque l'on ne connaît pas à priori la longueur de la chaîne, utiliser l'instruction `code[-1]`, qui donne accès au dernier caractère.

```
# ces deux instructions donnent le même résultat
code=input("Entrez le code à 13 chiffres")
print(code[12])
print(code[-1])
```

Nous avons donc la variable `code` qui contient une chaîne de 13 caractères, nous savons accéder à chacun d'entre eux et nous pouvons transformer chacun de ces caractères en nombre avec la fonction `int()`.

À FAIRE 3:

Pour s'en convaincre tester ce code avec le code 9782123456803 et expliquer les affichage obtenus.

```
code=input("Entrez le code à 13 chiffres")
print(3*code[-1])
print(3*int(code[-1]))
```

Les calculs :

Notre code est une chaîne de 13 caractères :code = "9782123456803"

En détails :

indice	0	1	2	3	4	5	6	7	8	9	10	11	12
chiffre	9	7	8	2	1	2	3	4	5	6	8	0	3

- Il faut faire la somme (s1) des caractères (transformés en nombre) d'indices (0, 2, 4, 6, 8, 10).
- Il faut faire la somme (s2) pondérée par 3 des caractères (transformés en nombre) d'indices (1, 3, 5, 7, 9, 11).
- Il faudra ajouter s1 et s2.

Vérifier que le code ci-dessous effectue bien le calcul de s1 :

```
code = "9782123456803"
s1=0 # on initialise s1 à 0
for i in range(0,11,2):
    s1 = s1+int(code[i])
```

Remarque: L'instruction `for i in range(0,11,2)` : signifie que la boucle démarre à l'indice 0 pour aller jusqu'à l'indice 11 non compris (soit l'indice 10) et cela avec un pas de 2.

À FAIRE 4:

Écrire une boucle qui calcule s2.

Rappel : Pour calculer le reste de la division de a par b , on utilise l'instruction: `reste = a%b`.

À FAIRE 5:

La clé de contrôle se calcule comme suit:

- On détermine le reste de la division de $s = s1 + s2$ par 10.
- Si ce reste est nul alors la clé est égale à 0.
- Sinon la clé est égale à $10 - \text{reste}$.

Écrire le code qui calcule cette clé de contrôle

Pour finir... :

Il ne reste plus qu'à comparer cette clé calculée avec le 13^{ème} chiffre du code ISBN, s'il y a égalité alors le code est valide, sinon il n'est pas valide.

Votre travail : Écrire une fonction `validite(code)` qui prend en paramètre un code ISBN entré par l'utilisateur, et qui renvoie **True** si ce code est valide et **False** sinon. On prendra soin de spécifier la fonction.

Un plus...

Si l'utilisateur se trompe en entrant par exemple 12 chiffres, Il faudrait que le programme signifie l'erreur.

Pour cela on utilise l'instruction (à mettre au début de la fonction):

assert (**len**(code) == 13) , 'Le code ISBN doit être de 13 chiffres'

Si l'utilisateur ne rentre pas un code ISBN de 13 caractères, il en sera informé par l'affichage:

```
AssertionError: Le code ISBN doit être de 13 chiffres
```