

Divisibilité par 7 (Méthode de Horner)

On se propose de concevoir une interface graphique permettant de saisir un nombre **X** formé de 5 à 20 chiffres, puis de vérifier s'il est divisible par 7 en utilisant le principe suivant :

Etape1 : Former, à partir du nombre **X**, un nouveau nombre **Y** en faisant correspondre à chaque chiffre de **X** le reste de sa division euclidienne par 7.

Exemple

X =	5	5	2	7	5	7	9	8	2
	↓	↓	↓	↓	↓	↓	↓	↓	↓
Y =	5	5	2	0	5	0	2	1	2

Etape2 : Former, à partir du nombre **Y**, un nouveau nombre **Z** en faisant correspondre à chaque tranche de deux chiffres de **Y** (en commençant à partir de la droite), le reste de sa division euclidienne par 7.

Exemple

Y =	5	5	2	0	5	0	2	1	2
	↓	↓	↓	↓	↓	↓	↓	↓	↓
Z =	5	3	5	2	5				

- En effet, pour la tranche **12** de **Y**, lui correspond dans **Z** la valeur **5** (reste de la division euclidienne de **12** par 7).

- De même pour les autres tranches.

- La dernière tranche se compose d'un seul chiffre car le nombre de chiffres de **Y** est impair.

Etape3 : Appliquer au nombre **Z**, la méthode de **Horner** définie par l'algorithme suivant de la fonction **Horner(Z)**.

Fonction Horner(Z : Chaîne de caractères) : Entier

DEBUT

M ← 0

Tant que Z <> "" Faire

CH ← Z[0]

*M ← (M * 2 + Valeur(CH)) Mod 7*

Z ← Sous_chaine(Z, 1, Long(Z))

Fin Tant que

Retourner M

FIN

T.D.O.L

Objet	Type/Nature
M	Entier
CH	Chaîne de caractères

Le nombre **X** est divisible par 7 lorsque la fonction **Horner**, appliquée au nombre **Z**, retourne la valeur **zéro**.

Créer un programme en Python et l'enregistrer sous le nom **DivHorner** dans lequel on demande :

- a. d'implémenter l'algorithme de la fonction **Horner**.
- b. de développer une fonction **Etape1(X)** qui simule l'Etape1 pour retourner le nombre **Y**.
- c. de développer une fonction **Etape2(Y)** qui simule l'Etape2 pour retourner le nombre **Z**.
- d. développer une fonction qui permet de saisir un nombre **x** de 5 à 20 chiffres
- e. Vérifier si ce nombre est divisible par 7 ou non