

Nom :	DM 05	TS <small>Sup Maths Orale</small>	Janv. 2017
Prénom :			Devoir n° 08

*Le soin et la rédaction seront pris en compte dans la notation. Faites des phrases claires et précises.
L'utilisation de logiciels est autorisée.*

Exercice 1

1 Démontrer que : « n n'est pas multiple de 5. » est équivalent à « $n^4 - 1$ est multiple de 5. »

- Supposons que n n'est pas un multiple de 5 :

On pose la division euclidienne de n par 5, alors $n = 5k + r$ où $0 \leq r < 5$.

Si « n n'est pas multiple de 5. » alors $n = 5k + r$ où $1 \leq r < 5$:

de l'égalité $n = 5k + r$, on déduit $n \equiv r[5]$;

$n \equiv \dots[5]$	0	1	2	3	4
$n^4 \equiv \dots[5]$	0	1	1	1	1
$n^4 - 1 \equiv \dots[5]$	4	0	0	0	0

D'après ce tableau de congruence, si n n'est pas un multiple de 5, alors $n^4 - 1 \equiv 0[5]$, ce qui prouve que « $n^4 - 1$ est multiple de 5. »

- Deuxième étape :

On suppose que $n^4 - 1$ est un multiple de 5.

Prouvons que « n n'est pas multiple de 5. »

on fait un raisonnement par l'absurde : Supposons que n est un multiple de 5 alors $n \equiv 0[5]$ d'où on déduit $n^4 \equiv 0^4[5]$ donc $n^4 - 1 \equiv -1[5]$; ceci contredit l'hypothèse « $n^4 - 1$ est multiple de 5. »; et donc fournit la conclusion.

2 Vérifier que :

- a.** 3 et 3^5 ont le même chiffre des unités.

Pour cela, on montre que 3 et 3^5 sont congrus modulo 10. $3^2 = 9$ et $9 \equiv -1[10]$ donc $3^2 \equiv -1[10]$, puis $3^4 \equiv (-1)^2[10]$ d'où $3^4 \equiv 1[10]$, puis en multipliant par 3 : $3^5 \equiv 3[10]$.

Comme $3^5 \equiv 3[10]$, 3 et 3^5 ont le même chiffre des unités.

- b.** 7^2 et 7^6 ont le même chiffre des unités. $7^2 = 49$ et $49 \equiv -1[10]$ donc $7^2 \equiv -1[10]$, puis $7^6 \equiv (-1)^3[10]$ d'où $7^6 \equiv -1[10]$, et on a montré $7^2 \equiv -1[10]$.

Comme $7^6 \equiv 7^2[10]$, 7^6 et 7^2 ont le même chiffre des unités.

- c.** 2^8 et 2^{12} ont le même chiffre des unités.

$2^8 = 256$ et $2^{12} = 4096$

2^8 et 2^{12} ont le même chiffre des unités qui est 6.

- d.** 4^3 et 4^7 ont le même chiffre des unités.

$4^3 = 64$ et $4^7 = 16384$

4^3 et 4^7 ont le même chiffre des unités qui est 4.

3 Soient a et b deux entiers naturels. Traduire en termes de congruence la propriété : « a et b ont le même chiffre des unités. »

« a et b ont le même chiffre des unités. »ssi $a \equiv b[10]$

- 4** Soit $n \in \mathbb{N}$ et $p \in \mathbb{N}^*$ Démontrer que n^{p+4} et n^p ont le même chiffre des unités.
 Tout d'abord remarquons que d'après le 3 on doit établir que $n^{p+4} \equiv n^p[10]$.

$$\begin{aligned} n^{p+4} \equiv n^p[10] &\iff n^{p+4} - n^p \equiv 0[10] \\ &\iff n^p(n^4 - 1) \equiv 0[10] \end{aligned}$$

On procède alors par disjonction de cas :

- Premier cas n n'est pas un multiple de 5 : d'après la question 1° $n^4 - 1$ est multiple de 5.

On fait une discussion suivant la parité de n

□ n est pair :

dans ce cas n est multiple de 2, donc $n \equiv 0[2]$ puis $n^p \equiv 0[2]$, il existe donc un entier s tel que $n^p = 2s$.

Par ailleurs $(n^4 - 1) \equiv 0[5]$, il existe donc un entier r tel que $n^4 - 1 = 5r$.

$$n^p(n^4 - 1) = 2s \times 5r = 10rs$$

On a donc $n^p(n^4 - 1) \equiv 0[10]$

□ n est impair :

Comme n est impair, on déduit $n \equiv 1[2]$, puis $n^4 \equiv 1[2]$, et donc $n^4 - 1 \equiv 0[2]$, il existe donc un entier s tel que $n^4 - 1 = 2s$.

Par ailleurs $n^4 - 1$ est multiple de 5, $n^4 - 1$ est divisible par 2 et par 5 donc de 10.

On a donc $(n^4 - 1) \equiv 0[10]$ puis On a donc $n^p(n^4 - 1) \equiv 0[10]$

- Deuxième cas n est un multiple de 5 : On fait une discussion suivant la parité de n

□ n est pair :

dans ce cas n est multiple de 2 et de 5, donc de 10, car 2 et 5 sont premiers entre eux.

il en est de même pour n^p donc pour $n^p(n^4 - 1)$.

On a donc $n^p(n^4 - 1) \equiv 0[10]$

□ n est impair :

Comme n est impair, on déduit $n \equiv 1[2]$, puis $n^4 \equiv 1[2]$, et donc $n^4 - 1 \equiv 0[2]$, il existe donc un entier s tel que $n^4 - 1 = 2s$.

Par ailleurs $n^4 - 1$ est multiple de 5, $n^4 - 1$ est divisible par 2 et par 5 donc de 10.

On a donc $(n^4 - 1) \equiv 0[10]$ puis On a donc $n^p(n^4 - 1) \equiv 0[10]$

Exercice 2

On considère les suites et définies pour tout entier n naturel par :

$$u_0 = 0 \quad ; \quad v_0 = 1 \quad ; \quad \begin{cases} u_{n+1} = 6u_n - v_n \\ v_{n+1} = u_n + 4v_n \end{cases}$$

On définit la suite de matrices (X_n) par la relation $X_n = \begin{pmatrix} u_n \\ v_n \end{pmatrix}$

- 1** a. Déterminer la matrice A telle que pour tout entier naturel n , on ait :

$$X_{n+1} = A \cdot X_n$$

$$\begin{cases} u_{n+1} = 6u_n - v_n \\ v_{n+1} = u_n + 4v_n \end{cases} \iff \begin{pmatrix} u_{n+1} \\ v_{n+1} \end{pmatrix} = \begin{pmatrix} 6 & -1 \\ 1 & 4 \end{pmatrix} \begin{pmatrix} u_n \\ v_n \end{pmatrix}$$

- b. Démontrer par récurrence la relation suivante pour tout entier naturel n :

$$X_n = A^n \cdot X_0.$$

- Initialisation : comme $A^0 = I_2$; et $A^0 X_0 = I_2 X_0 = X_0$; ainsi la propriété est vraie au rang 0.

- Transmission de l'hérédité :

soit $k \geq 0$, on suppose que $X_k = A^k X_0$; on doit prouver que la propriété est vraie au rang $k + 1$; c'est-à-dire : $X_{k+1} = A^{k+1} X_0$

$$\begin{aligned}
X_{k+1} &= AX_k \\
&= A \cdot A^k X_0 \text{ d'après HR} \\
&= A^{k+1} X_0
\end{aligned}$$

- Conclusion : On a donc vérifié que la propriété est vraie au rang 0 et qu'il y a transmission de l'hérédité ; le théorème de récurrence s'appliquant, on a pour tout $n \geq 0$, $X_n = A^n X_0$

2 a. Déterminer la matrice J telle que $A = 5I + J$ où I est la matrice identité d'ordre 2.

$$A = 5I + J \iff J = A - 5I = \begin{pmatrix} 6 & -1 \\ 1 & 4 \end{pmatrix} - \begin{pmatrix} 5 & 0 \\ 0 & 5 \end{pmatrix} = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix}$$

b. Calculer J^2 puis A^2 .

$$J^2 = \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix} = \begin{pmatrix} 1-1 & -1+1 \\ 1-1 & -1+1 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$$

$$\begin{aligned}
A^2 &= (5I + J) \cdot (5I + J) \\
&= 5I \cdot 5I + 5I \cdot J + J \cdot 5I + J^2 \\
&= 25I + 5J + 5J \\
&= 25I + 10J
\end{aligned}$$

$$J^2 = \begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix} \text{ et } A^2 = \begin{pmatrix} 35 & -10 \\ 10 & 15 \end{pmatrix}$$

3 Démontrer par récurrence que, pour tout entier naturel n supérieur ou égal à 1, on a : $A^n = n5^{n-1}J + 5^n I$.

- Initialisation : comme $A^1 = A$; et $1 \times 5^{1-1}J + 5^1 I = J + 5I = A$; ainsi la propriété est vraie au rang 1.
- Transmission de l'hérédité : soit $k \geq 1$, on suppose que $A^k = k5^{k-1}J + 5^k I$; on doit prouver que la propriété est vraie au rang $k + 1$; c'est-à-dire : $A^k = k5^{k-1}J + 5^k I$

$$\begin{aligned}
A^{k+1} &= A^k \times A \\
&= (k5^{k-1}J + 5^k I) \cdot (J + 5I) \text{ d'après HR} \\
&= k5^{k-1}J^2 + k5^{k-1}J \cdot 5I + 5^k I \cdot J + 5^k I \cdot 5I \\
&= k5^{k-1} \times 5J + 5^k J + 5^k \times 5I \\
&= k5^k \times J + 5^k J + 5^{k+1} I \\
&= (k+1)5^k \times J + 5^{k+1} I
\end{aligned}$$

- Conclusion : On a donc vérifié que la propriété est vraie au rang 1 et qu'il y a transmission de l'hérédité ; le théorème de récurrence s'appliquant, on a pour tout $n \geq 1$, $A^n = n5^{n-1}J + 5^n I$

4 Exprimer u_n et v_n en fonction de n .

On a montré $X_n = A^n X_0$ et $A^n = n5^{n-1}J + 5^n I$.

$$\begin{aligned}
A^n &= n5^{n-1}J + 5^n I \\
&= n5^{n-1} \begin{pmatrix} 1 & -1 \\ 1 & -1 \end{pmatrix} + 5^n \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \\
&= \begin{pmatrix} n5^{n-1} & -n5^{n-1} \\ n5^{n-1} & -n5^{n-1} \end{pmatrix} + \begin{pmatrix} 5^n & 0 \\ 0 & 5^n \end{pmatrix} \\
&= \begin{pmatrix} 5^n + n5^{n-1} & -n5^{n-1} \\ n5^{n-1} & 5^n - n5^{n-1} \end{pmatrix}
\end{aligned}$$

On a ainsi :

$$\begin{aligned} X_n = \begin{pmatrix} u_n \\ v_n \end{pmatrix} &= A^n X_0 \\ &= \begin{pmatrix} 5^n + n5^{n-1} & -n5^{n-1} \\ n5^{n-1} & 5^n - n5^{n-1} \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ &= \begin{pmatrix} -n5^{n-1} \\ 5^n - n5^{n-1} \end{pmatrix} \end{aligned}$$

$$u_n = -n5^{n-1} \text{ et } v_n = 5^n - n5^{n-1}.$$

Exercice 3

Algorithme de Luhn et carte de crédit



Voici un algorithme :

Les variables sont l'entier naturel i , le réel S et la liste L .

```
S ← 0
L ← le numéro de votre carte bancaire
Pour i allant de 1 à 16
  Lire L[i]
Fin Pour
Pour i allant de 1 à 8
  L[2*i - 1] ← 2 * L[2*i - 1]
  Si L[2*i - 1] > 9 alors L[2*i - 1] ← L[2*i - 1] - 9
FinSi
Fin Pour
Pour i allant de 1 à 16
  S ← S + L[i]
Fin Pour
Si reste(S, 10) = 0 alors
  afficher "Le numéro de carte est valide."
Sinon
  afficher "Le numéro de carte n'est pas valide."
FinSi
```

- 1 Quel est le rôle de cet algorithme ?
L'algorithme de Luhn (modulo 10) est une formule de validation de nombres utilisée notamment pour les cartes bancaires et les numéros administratifs (SIRET).
- 2 Décrire très précisément le fonctionnement de l'algorithme.
L'utilisateur saisit dans l'ordre les numéros de sa carte bancaire.

On double tous les chiffres de rang pair.

Si le double d'un chiffre est supérieur ou égal à 10, le remplacer par ce double moins 9. On calcule la somme des chiffres. Le chiffre de contrôle est égal à $(10 - \text{sum} \% 10) \% 10$.

Exemple : Le nombre 853X, avec X, nombre que l'on souhaite calculer.

Prendre le 3, le doubler pour obtenir 6.

Prendre le 5, non doublé.

Enfin, doubler le 8 ($8 * 2 = 16$ et $16 - 9 = 7$) pour obtenir 7.

La somme est $6 + 5 + 7 = 18$. Comme $18 \equiv 8$ modulo 10, calculer $10 - 8 = 2$, la clé de contrôle est donc 2. Ainsi 8532 est valide selon Luhn.

3 Écrire un programme utilisant cet algorithme sur votre logiciel d'algorithmique préféré ou votre calculatrice programmable.

Voici cet algorithme traduit en python :

- Un premier algorithme où l'on saisit les chiffres du code 1 à 1 :

```
1 # Algorithme de Luhn
2 L=[]
3 for i in range(1,17):
4     n=input("Ecris successivement les chiffres de ta carte , valide apres chaque s
5     n=int(n)
6     L.append(n)
7
8 print(L)
9 for k in range(0,16):
10     if k%2==0:
11         L[k]=2*L[k]
12         if L[k]> 9:
13             L[k]=L[k]-9
14     else:
15         L[k]=L[k]
16 S=0
17 for j in range(0,16):
18     S=S+L[j]
19 if S%10==0:
20     print("Le code de votre carte est valide.")
21 else:
22     print("Le code de votre carte n'est pas valide.")
```

- Un deuxième sous forme de fonction :

```
1 # Une fonction pour l'algo de Luhn
2 def algo_Luhn(x):
3     L=[]
4     for lettre in x:
5         L.append(int(lettre))
6
7
8
9     print(L)
10    for k in range(0,16):
11        if k%2==0:
12            L[k]=2*L[k]
13        if L[k]> 9:
14            L[k]=L[k]-9
15        else:
16            L[k]=L[k]
17    S=0
18    for j in range(0,16):
```

```

19     S=S+L[j]
20     if S%10==0:
21         return "Le code de votre carte est valide."
22     else:
23         return "Le code de votre carte n'est pas valide."

```

4 Emprunter la carte bleue d'un de vos parents et tester son numéro avec votre programme.

5 Tester les deux numéros suivants :

5209499211024837

5135136065996745

- En utilisant le deuxième algorithme :
La saisie de
algo_Luhn('5209499211024837') donne le retour :
[5, 2, 0, 9, 4, 9, 9, 2, 1, 1, 0, 2, 4, 8, 3, 7]

"Le code de votre carte n'est pas valide."

- En utilisant le deuxième algorithme :
La saisie de
algo_Luhn('5135136065996745') donne le retour :
[5, 1, 3, 5, 1, 3, 6, 0, 6, 5, 9, 9, 6, 7, 4, 5]
'Le code de votre carte est valide.'