



« Le défi algorithmique et programmation » de l'IREM, vous est proposé dans le cadre du partenariat entre l'IREM (Institut de Recherche sur l'Enseignement des Mathématiques) de l'Université des Antilles et le Rectorat de l'académie de Guadeloupe.

Catégorie :	Logique	X	Numérique	X	Géométrique	Algorithmique	X
-------------	---------	---	-----------	---	-------------	---------------	---

ENONCÉ

« Défi algorithmique et programmation », Janvier 2020

« le critère de divisibilité par 7 de Chika »



En fin d'année 2019, un jeune élève nigérian a défrayé la chronique. Le jeune Chika Ofili âgé de 12 ans a retrouvé seul une méthode simple pour déterminer si un nombre est divisible par 7.

La méthode est la suivante : On prend le dernier chiffre du nombre dont on cherche à savoir s'il est divisible par 7. On multiplie ce chiffre par 5 et on ajoute ce résultat au nombre de dizaines du nombre de départ. Si le résultat obtenu est divisible par 7, alors le nombre initial est divisible par 7. Sinon, c'est qu'il ne l'est pas.

Illustrons la méthode par un exemple : Cherchons à savoir si le nombre 299⁶ est un multiple de 7 selon la méthode énoncée ci-dessus :

$$299 + 6 \times 5 = 329$$

$$32 + 9 \times 5 = 77$$

$7 + 7 \times 5 = 42$, 42 étant un multiple de 7. On établit ainsi que 2996 est un multiple de 7.

Le défi du mois est de réaliser un programme en Scratch (pour le collège) ou en Python (pour le lycée) qui permettra de dire pour un nombre initialement saisi, s'il est divisible par 7 ou pas en utilisant les contraintes suivantes :

- On utilisera la connaissance des multiples de 7 de 0 à 70.
- On utilisera la méthode que nous nommerons « critère de divisibilité par 7 de Chika »

Pour ce défi, on n'utilisera pas la fonction « modulo » qui donne le reste de la division euclidienne d'un nombre par un autre.

Niveau		du cycle 4 au Lycée				
Champ disciplinaire	Nombres et calcul	X	Géométrie		Fonctions	
	Statistiques et probabilités		Algorithmique et programmation	X		
Compétences mathématiques travaillées	Chercher	X	Modéliser	X	Représenter	X
	Raisonner	X	Calculer	X	Communiquer	
Connaissances et compétences associées	Résoudre des problèmes motivants et substantiels, afin de stabiliser connaissances, méthodes et stratégies.					
Compétences visées pour la mise en oeuvre:	<p>CHERCHER:</p> <ul style="list-style-type: none"> • prélever et organiser les informations nécessaires à la résolution de problèmes à partir de supports variés : textes; • s'engager dans une démarche, observer, questionner, manipuler, expérimenter, émettre des hypothèses, en mobilisant des outils ou des procédures mathématiques déjà rencontrées, en élaborant un raisonnement adapté à une situation nouvelle; • tester, essayer plusieurs pistes de résolution; • expérimenter – en particulier à l'aide d'outils logiciels; • décomposer un problème en sous-problèmes. <p>MODELISER:</p> <ul style="list-style-type: none"> • traduire en langage mathématique une situation réelle (par exemple à l'aide d'équations, de fonctions, de configurations géométriques, d'outils statistiques); • faire une simulation, valider ou invalider un modèle; <p>REPRESENTER:</p> <ul style="list-style-type: none"> • adapter pour traiter un problème ou pour étudier un objet mathématique; • produire et utiliser plusieurs représentations des nombres; • choisir un cadre (numérique, algébrique, géométrique...), changer de registre. <p>RAISONNER:</p> <ul style="list-style-type: none"> • démontrer, trouver des résultats partiels et les mettre en perspective. <p>CALCULER:</p> <ul style="list-style-type: none"> • appliquer des techniques et mettre en œuvre des algorithmes. 					

Pour ceux qui veulent aller plus loin

Encore des critères :

Une promenade au pays des critères de divisibilité. On va y retrouver la star du mois à savoir le critère de divisibilité par 7 mais aussi le classique 11 où encore 41, 67 et les autres ...

- [Critères de divisibilité par 7 suite](#), (wikipédia)

Proposition de démonstration de la méthode de Chika par un collègue du site [mathix.org](#)

Tout nombre entier peut se décomposer de la forme $a \times 10 + b$ avec $b < 10$

$$a \times 10 + b = 0[7]$$

En multipliant par 5:

$$\Leftrightarrow a \times 50 + 5 \times b = 0 [7]$$

comme on sait que $a \times 49 = 0[7]$ car 49 est un multiple de 7,

on a

$$\Leftrightarrow a + 5 \times b = 0[7]$$

$a + 5 \times b$ correspond au 2e nombre dont il faut tester la divisibilité

On recommence le processus autant de fois que nécessaire jusqu'à clairement reconnaître un multiple connu de 7

Structure de l'algorithme / proposition de solution

On demande à l'utilisateur de saisir, le **nombre** que l'on souhaite tester afin de savoir s'il est divisible par 7 ou pas.

`multiples_de_sept=[0,7,14,21,28,35,42,49,56,63,70]`

`n←nombre`

Tant que `n>70` faire

`n1 (le nombre de dizaines de n)`

`n2 (le nombre des unités de n)`

`n=n1+5xn2`

Si `n` est dans la liste `multiples_de_sept` :

 Afficher le nombre est divisible par 7

 sinon

 Afficher le nombre n'est pas divisible par 7

Proposition de solution au collège avec SCRATCH

Etape 1:



Création de la liste qui va recevoir les multiples de 7,

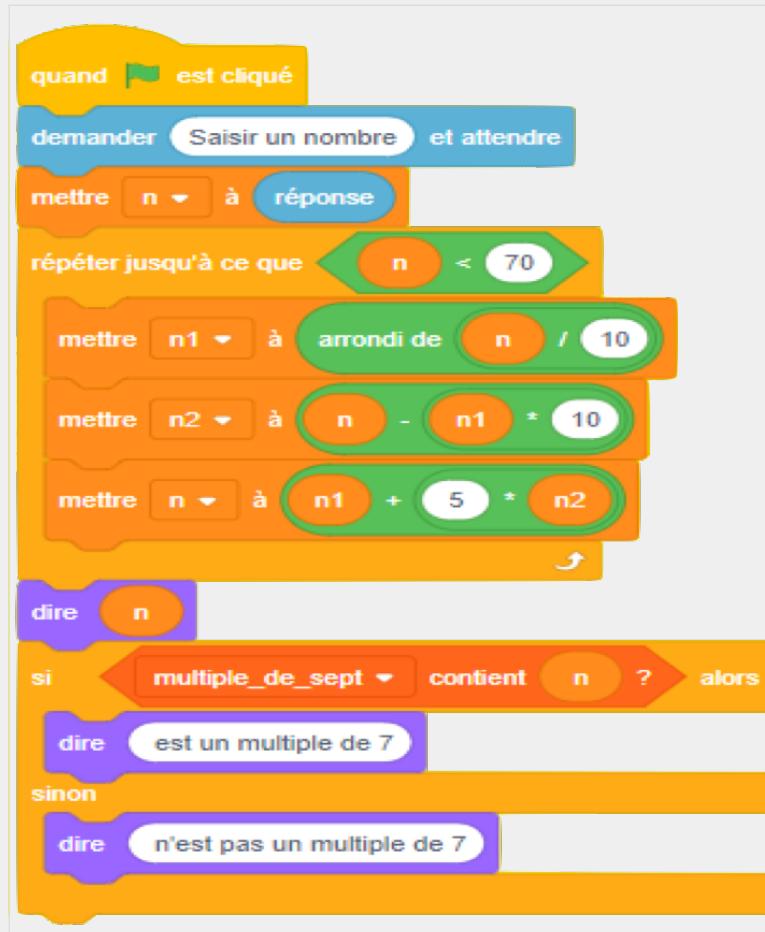


on remplit la liste avec les multiples de 7 de 0 à 70

liste « `multiple_de_sept` » complétée, affichage obtenu,

Proposition de solution au collège avec SCRATCH

Etape 2:



saisie du nombre entier à saisir

Tant que $n > 0$ faire :

$n1$ est la partie entière de la division de n par 10,

$n2$ vaut $n - n1 \times 10$

n devient $n1 + 5 \times n2$

Si n fait partie de la liste multiple_de_sept,

Sinon

« LYCEE »

Proposition de solution du défi avec Python

Version texte (à copier - coller)	Description
multiple_de_sept=[0,7,14,21,28,35,42,49,56,63,70] nombre=int(input("saisir le nombre")) n=nombre while n>70: n1=int(n//10) n2=n-n1*10 print(n1, " ", n2) n=n1+5*n2 print(n) if n in multiple_de_sept: print(nombre," est un multiple de 7") else: print(nombre," est un multiple de 7")	<p><i>Création de la liste des multiples de 7</i> <i>Saisie du nombre entier à tester</i></p> <p><i>Tant que $n > 70$ faire :</i> <i> n1 est la partie entière de la division de n par 10, et n2 vaut $n - n1 \times 10$</i> <i> Afficher n1 et n2</i></p> <p><i> n devient n1 + 5xn2</i> <i> Afficher n</i></p> <p><i>Si n fait partie de la liste multiple_de_sept,</i> <i> Afficher le nombre de départ suivi de</i> <i> « est multiple de 7 »</i></p> <p><i>Sinon</i> <i> Afficher le nombre de départ suivi de</i> <i> « n'est pas multiple de 7 »</i></p>

Copie d'écran

```
main.py  ⚏  ⚑ saved
1  multiple_de_sept=[0,7,14,21,28,35,42,49,56,63,70]
2  nombre=int(input("saisir le nombre :"))
3  n=nombre
4  while n > 70:
5      n1=int(n//10)
6      n2=n-n1*10
7      print(n1, " ", n2)
8      n=n1+5*n2
9      print(n)
10     if n in multiple_de_sept:
11         print(nombre," est un multiple de 7")
12     else:
13         print(nombre," n'est pas un multiple de 7")|
```

exemple

Vous pouvez tester le script ci-dessus sur [Python 3 - repl.it](https://repl.it/) (site permettant de faire du python 3 en ligne sans installation) en faisant un Copier - Coller dans la fenêtre centrale puis lancer le programme en cliquant en haut sur RUN.

Il faut vérifier après le collage la mise en forme (l'indentation, les guillemets qui souvent sont à retaper, ...) car sinon vous aurez des messages d'erreurs.

Prenons l'exemple d'une somme à remettre de : 252

Le résultat obtenu par le programme est : $1 \times 50 + 1 \times 10 + 1 \times 5 + 2 \times 2 = 69 \text{ €}$

affichage obtenu

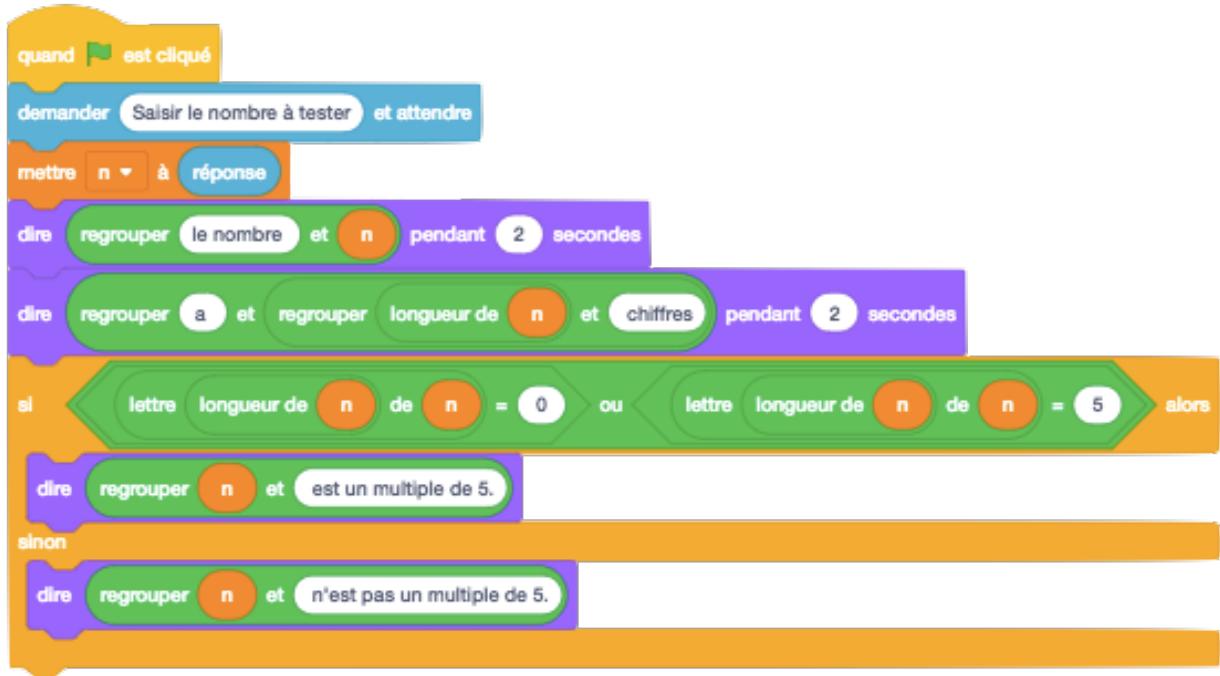
```
saisir le nombre :252
25    2
35
252  est un multiple de 7
```

Eléments de différenciation et exploitations possibles	
Rappels des contenus des programmes sur les attendus – Cycle 4	Attendus de fin de cycle - écrire, mettre au point et exécuter un programme simple. Ecrire, mettre au point, exécuter un programme. Connaissances : notions d'algorithme et de programme ; notion de variable informatique ; déclenchement d'une action par un événement ; séquences d'instructions, boucles, instructions conditionnelles. Compétences associées : écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné.
Passage de scratch à Python en lycée	La démarche algorithmique est, depuis les origines, une composante essentielle de l'activité mathématique. Au collège, en mathématiques et en technologie, les élèves ont appris à écrire, mettre au point et exécuter un programme simple. Il s'agira d'aider les élèves à réussir le passage de Scratch à Python .
Piste de prolongement	Modifier le programme pour travailler sur la notion de base. Voir comment on peut utiliser les principes de l'algorithmique créé pour écrire un nombre entier en binaire ou dans une base quelconque.
Piste de différenciation	Inscrire les apprentissages de l'algorithmique dans la progressivité du cycle 3 au cycle 4 puis du cycle 4 au lycée. Proposer différents niveaux pour ce défi. On pourra commencer avec un nombre limité de billets et de pièces puis traiter tous les cas de figures avec notamment des sommes avec des valeurs décimales. Ce défi peut se transformer en jeu consistant à trouver une somme donnée (avec le minimum de billets et pièces) à partir de billets et de pièces disponibles.
Situation 1: « Critères de divisibilité par 2 et 5 »	
Autres pistes	Défi intermédiaire1 : Réaliser un programme qui permet de savoir si un nombre est un multiple de 5.
	Défi intermédiaire2 : Réaliser un programme qui permet de savoir si un nombre est un multiple de 2.
Rq: Pour le collège avec SCRATCH, de nombreuses stratégies sont possibles, nous vous proposons ici d'utiliser de découvrir et d'utiliser les fonctions « longueur de ... » et « lettre ... de ... »	
Défi intermédiaire1	
Coup de pouce: - Utiliser les fonctions	
 (permet de déterminer la longueur d'un nombre, d'un mot ou d'une chaîne de caractère)	
 (donne le n ^{ième} chiffre ou lettre d'un nombre, d'un mot ou d'une chaîne de caractère en partant de la gauche)	
Défi intermédiaire2	
Même coup de pouce que pour le 1 en pensant cette fois-ci à créer la liste {0;2;4;6;8}	

Situation 1: « Critères de divisibilité par 5 »

Proposition de solution avec SCRATCH

Défi intermédiaire1



Remarque :

L'utilisation de la variable « n » n'était pas nécessaire, nous aurions pu utiliser tout du long « réponse ». Toutefois, habituer les élèves à systématiser le recours aux variables est formateur.

Le bloc « regrouper » permet ici d'afficher ensemble du texte et une variable.

regrouper et

Différenciation possible :

Proposer aux élèves les blocs qu'ils doivent utiliser.



Situation 1: « Critères de divisibilité par 5 »

1ère proposition de solution avec Python

Version texte (à copier - coller)	Description
<pre>nombre=int(input("saisir le nombre entier à tester")) m=len(str(nombre)) print(nombre," est un nombre a ",m," chiffres") A=nombre-(nombre//10)*10 print("le chiffre des unités de « ,nombre," est « ,A) if A==5 or A==0: print(nombre," est divisible par 5") else: print(nombre," n'est pas divisible par 5")</pre>	<p><i>Saisie du nombre à tester</i> <i>Longueur de « nombre » vu comme une chaîne de caractère (str)</i> <i>Afficher ...</i> <i>A est le chiffre des unités du nombre</i> <i>Afficher « le chiffre des unités du nombre à tester est A »</i> <i>Test si A=5 ou A=0</i> <i>Afficher « 'nombre' est divisible par 5 »</i> <i>Sinon</i> <i>Afficher « 'nombre' n'est pas divisible par 5 »</i></p>

Remarque :

- Pour cette proposition de solution, le nombre saisie est un entier « `int(input("saisir le..."))` ».
- On découvre `len(...)` qui permet de déterminer la longueur d'une chaîne de caractère, c'est pourquoi on transforme temporairement « `nombre` » qui est un entier en chaîne de caractère à l'aide de `str(nombre)`. Afin de pouvoir annoncer le nombre de chiffre du nombre à tester (*cette action n'est pas nécessaire pour le critère mais nous fait découvrir quelques nouvelles instruction de Python*)
- `A=nombre-(nombre//10)*10` permet de déterminer le chiffre des unités de nombre.
- `a//b` permet d'obtenir le quotient entier de la division de a par b.

Situation 1: « Critères de divisibilité par 5 »

2ème proposition de solution avec Python

Version texte (à copier - coller)	Description
<pre>nombre=input("saisir le nombre entier à tester : ") if nombre[-1] in {'0','5'}: print(nombre," est divisible par 5") else: print(nombre," n'est pas divisible par 5")</pre>	<p><i>Saisie du « nombre » à tester</i> <i>On test si le dernier caractère de « nombre » est 0 ou 5</i></p>

Remarque :

- Pour cette proposition de solution, le nombre saisie n'est pas considéré comme un entier mais comme une chaîne de caractère « `input("saisir le...")` ».
- « `nombre[-1]` » donne le dernier caractère de « `nombre` », pour nous ici le chiffre des unités.
- On test si le dernier caractère est 0 ou 5 avec `if nombre[-1] in {'0','5'}`, bien penser aux apostrophes '...' car ici 0 et 5 sont considérés comme des caractères et non des nombres.

Situation 1: « Critères de divisibilité par 2 »

1ère proposition de solution avec Python

Version texte (à copier - coller)

```
multiple_de_2=[0,2,4,6,8]
nombre=input("saisir le nombre entier à tester: "))
m=len(str(nombre))
print(nombre," est un nombre à ",m," chiffres")
A=nombre-int(nombre//10)*10
print("le chiffre des unités est ",A)
if A in multiple_de_2:
    print(nombre," est divisible par 2")
else:
    print(nombre," n'est pas divisible par 2")
```

Description

*Création de la liste des multiples de 2
Saisie du nombre à tester
Longueur de « nombre » ...
Afficher ...
A est le chiffre des unités du nombre
Afficher le chiffre des unités
Test si A fait partie de la liste multiple_de_2
Afficher « 'nombre' est divisible par 2 »
Sinon
Afficher « 'nombre' n'est pas divisible par 2 »*

Remarque:

Cette solution première solution est construite comme la première du critère de divisibilité par 5 on y a juste ajouté en amont la création de la liste des multiples inférieur à 10.

Situation 1: « Critères de divisibilité par 2 »

2ème proposition de solution avec Python

Version texte (à copier - coller)

```
nombre=input("saisir le nombre entier à tester : ")
if nombre[-1] in {'0','2','4','6','8'}:
    print(nombre," est divisible par 2")
else:
    print(nombre," n'est pas divisible par 2")
```

Description

*Saisie du « nombre » à tester
On test si le dernier caractère de « nombre » est 0 ou 5*

Remarque:

Solution analogue à la deuxième solution proposée pour la divisibilité par 5.

Situation 1: « Critères de divisibilité par 2 »

Proposition de solution avec SCRATCH

Défi intermédiaire1

Etape 1:



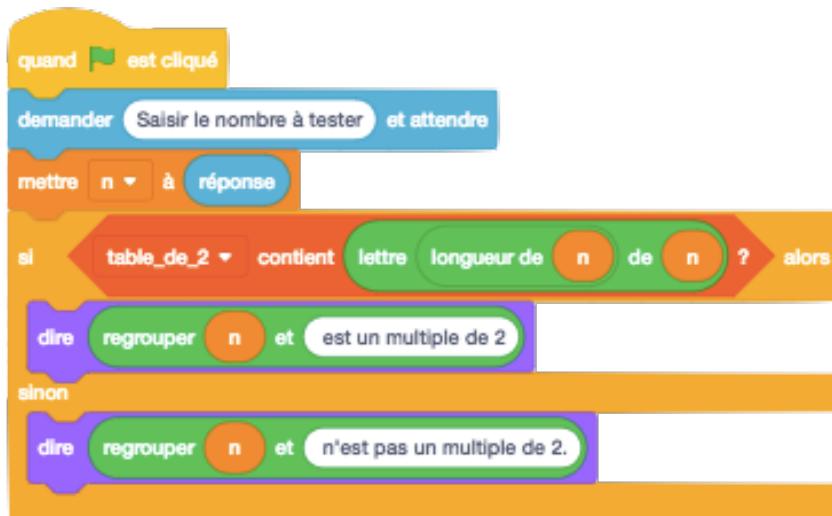
On commence par créer la liste « table_de_2 » dans l'onglet « variables », puis on remplit cette liste.

On entre les 6 valeurs de la liste manuellement.

Affichage obtenu:

table_de_2
1 0
2 2
3 4
4 6
5 8
6 10
+ longueur 6 =

Etape 2: « Script du test de divisibilité par 2 »



Entrer le nombre à tester

On affecte la valeur à « n »

On test si le dernier chiffre du nombre « n » est dans la liste « table_de_2 »