

Thème : Les nombres premiers

Corrigé de l'activité 2. Algorithmes sur les nombres premiers (3 exercices)

Exercice 1 : Tester la primalité (exercice identique à l'exercice 3 de l'activité 1 « Autour des nombres premiers »)

- 1) 107 n'est pas divisible par 2, 3, 5, ..., 103 : c'est donc un **nombre premier**.
- 227 n'est pas divisible par 2, 3, 5, ..., 223 : c'est donc un **nombre premier**.
 - 375 est divisible par 5, ce n'est donc pas un nombre premier.
 - 377 est divisible par 13, ce n'est donc pas un nombre premier.
 - 379 n'est pas divisible par 2, 3, 5, ..., 373 : c'est donc un **nombre premier**.
 - 571 n'est pas divisible par 2, 3, 5, ..., 569 : c'est donc un **nombre premier**.
- 2) Pour 107, il n'est pas utile de tester tous les nombres premiers de 2 jusqu'à 103. En effet, il est certes nécessaire de tester la division par 2, 3, ...

Mais par exemple, le test de la division par 73 est inutile : $\text{PartEnt}(107/73) = 1$.

Les élèves auront tendance à énoncer le fait qu'il faut "s'arrêter" à $107/2$.

Il faudra relancer le débat en demandant si l'on ne peut pas faire mieux...

Et ce jusqu'à obtenir que si n est composé alors son plus petit diviseur noté p est inférieur ou égal à \sqrt{n} .

Preuve : On suppose que n a au moins un autre diviseur que 1 et lui-même. On appelle p le plus petit de ces diviseurs.

Donc $1 < p < n$ où p est un diviseur de n . Alors :

- Il existe un second diviseur q tel que $pq = n$
- Ce second diviseur est tel que $p \leq q$. Donc $1 < p \leq q < n$

$$p < p^2 \leq pq$$

$$p^2 \leq n$$

$$p \leq \sqrt{n}$$

$\sqrt{107} \approx 10,3$. Donc il faut et il suffit de tester la division par tous les nombres premiers inférieurs ou égaux à 7.

- 3) On fait fonctionner l'algorithme sur le papier :

Pour $N = 19$,

1^{er} tour de boucle Tant que
2^e tour de boucle Tant que
3^e tour de boucle Tant que
4^e tour de boucle Tant que
Sortie de boucle car la condition I ne divise pas N est vraie
et la condition $I \leq E(\sqrt{N})$ est fausse

N	I
19	
	2
	3
	4
	5

L'affichage est PREMIER.

Pour $N = 21$,

1^{er} tour de boucle Tant que
 2^e tour de boucle Tant que
 Sortie de boucle car la condition I ne divise pas N est fausse
 et la condition $I \leq E(\sqrt{N})$ est vraie

N	I
21	
	2
	3

L'affichage est NON PREMIER.

Pour $N = 53$,

1^{er} tour de boucle Tant que
 2^e tour de boucle Tant que
 3^e tour de boucle Tant que
 4^e tour de boucle Tant que
 5^e tour de boucle Tant que
 6^e tour de boucle Tant que
 7^e tour de boucle Tant que
 Sortie de boucle car la condition I ne divise pas N est vraie
 et la condition $I \leq E(\sqrt{N})$ est fausse

N	I
53	
	2
	3
	4
	5
	6
	7
	8

L'affichage est PREMIER.

4) a) Programme TESTA

```

Programme TESTA sur TI
Prompt N
If N ≤ 3
Then
If N=2 ou N=3
Then
Disp "N PREMIER"
Else
Disp "N NON PREMIER"
End
Else
2→I
While (partEnt(N/I)≠N/I) et
(I≤ partEnt(√(N)))
I+1→I
End
If partEnt(N/I)=N/I
Then
Disp "N NON PREMIER"
Else
Disp "N PREMIER"
End
End
  
```

```

Programme TESTA sur Casio
"N="?→N
If N≤ 3
Then
If N=2 Or N=3
Then
"N PREMIER"
Else
"N NON PREMIER"
IfEnd
Else
2→I
While (Intg(N÷I)≠N÷I) And
(I≤ Intg(√(N)))
I+1→I
WhileEnd
If Intg(N÷I)=N÷I
Then
"N NON PREMIER"
Else
"N PREMIER"
IfEnd
IfEnd
ClrText
  
```

b) En utilisant ce programme, on affiche :

- 19 et 53 sont premiers et 21 n'est pas premier.
- 2011 et 2017 sont premiers et 2013, 2015 et 2019 ne pas premiers.

c)	TI83	TI83 CE Premium	Casio Graph 35+ USB
<u>TESTA</u> : Durée du test de la primalité de 7 237 031	75 s	58 s	31 s

- 5) a) Modification de l'algorithme pour tester la division par 2 et pour ensuite ne pas tester la division par les autres nombres pairs :

Déclaration des variables : N et I sont des entiers

Début de l'algorithme

Saisir N
Si $N \leq 3$
Alors
 Si $N = 2$ ou $N = 3$
 Alors
 Afficher " N PREMIER"
 Sinon
 Afficher " N NON PREMIER"
 Fin Si
Sinon
 Si 2 divise N
 Alors
 Afficher " N NON PREMIER"
 Sinon
 I prend pour valeur 3
 Tant que (I ne divise pas N) et ($I \leq \sqrt{N}$)
 I prend la valeur $I + 2$
 Fin Tant que
 Si I divise N
 Alors
 Afficher " N NON PREMIER"
 Sinon
 Afficher " N PREMIER"
 Fin Si
Fin Si
Fin de l'algorithme

- b) Programmation sur calculatrice :

Programme TESTB sur TI

Prompt N
If $N \leq 3$
Then
 If $N=2$ ou $N=3$
 Then
 Disp " N PREMIER"
 Else
 Disp " N NON PREMIER"
 End
Else
 If partEnt($N/2$)= $N/2$
 Then
 Disp " N NON PREMIER"
 Else
 $3 \rightarrow I$
 While (partEnt(N/I) $\neq N/I$) et ($I \leq \sqrt{N}$)
 $I+2 \rightarrow I$
 End
 If partEnt(N/I)= N/I
 Then
 Disp " N NON PREMIER"
 Else
 Disp " N PREMIER"
 End
End

Programme TESTB sur Casio

"N="? $\rightarrow N$
If $N \leq 3$
Then
 If $N=2$ Or $N=3$
 Then
 "N PREMIER" \blacktriangleleft
 Else
 "N NON PREMIER" \blacktriangleleft
 IfEnd
Else
 If Intg($N \div 2$)= $N \div 2$
 Then
 "N NON PREMIER" \blacktriangleleft
 Else
 $3 \rightarrow I$
 While (Intg($N \div I$) $\neq N \div I$) And ($I \leq \sqrt{N}$)
 $I+2 \rightarrow I$
 WhileEnd
 If Intg($N \div I$)= $N \div I$
 Then
 "N NON PREMIER" \blacktriangleleft
 Else
 "N PREMIER" \blacktriangleleft
 IfEnd
IfEnd
ClrText

- c) L'amélioration du temps d'exécution du programme est très significative. La durée d'exécution du programme est divisée par 2. C'est une très nette amélioration pour de grandes valeurs de N .

	TI83	TI83 CE Premium	Casio Graph 35+ USB
<u>TESTB</u> : Durée du test de la primalité de 7 237 031	38 s	29 s	16 s

Exercice 2 : Liste des nombres premiers inférieurs à 1000

1. L'exécution du programme LISTPREM dure environ :

	TI83	TI83 CE Premium	Casio Graph 35+ USB
<u>LISTPREM</u> : Durée de la construction de la liste LP de tous les nombres premiers jusqu'à 1009	120 s	140 s	54s

2. Dans la liste Liste6, on peut lire 2, 3, 5 ... 997, 1 009. On a donc bien obtenu les nombres premiers inférieurs ou égaux à 1009.

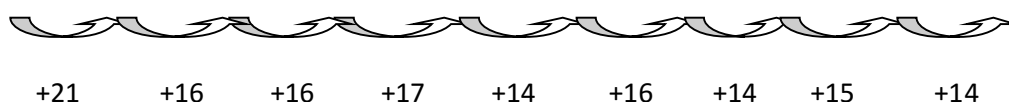
3. En utilisant la liste Liste6 des nombres premiers inférieurs ou égaux à 1009, on obtient :

P	100	200	300	400	500	600	700	800	900	1 000
$\pi(P)$	25	46	62	78	95	109	125	139	154	168

On ne peut donc pas parler d'équi-répartition des nombres premiers.

En effet, si c'était le cas, chaque centaine comporterait le même nombre de nombres premiers. Or :

P	100	200	300	400	500	600	700	800	900	1 000
$\pi(P)$	25	46	62	78	95	109	125	139	154	168



Il semble que la densité des nombres premiers diminue quand N augmente.

Exercice 3 : Recherche des diviseurs premiers inférieurs ou égaux à 1000

1) Voici les programmes pour calculatrices TI et Casio de cet algorithme :

Programme DIVISPRES sur TI	Programme DIVISPRES sur Casio
Prompt N 1 → I 0 → P While P < 997 et P ≤ N L6(I) → P If partEnt(N/P)=N/P Then Disp "DIVISIBLE PAR",P End I+1 → I End	"N=" ? → N 1 → I 0 → P While P < 997 And P ≤ N List 6[I] → P If Intg(N÷P)=N÷P Then "DIVISIBLE PAR":P ◀ IfEnd I+1 → I WhileEnd ClrText

2) L'exécution du programme DIVISPRES, permet d'obtenir :

N	12	13	985	997	1 000	1 009
Affichage : DIVISIBLE PAR	2 ; 3	13	5 ; 197	997	2 ; 5	aucun

3) Le programme DIVISPRES donnera tous les diviseurs premiers des entiers N dont la décomposition en produit de facteurs premiers ne comporte aucun nombre premier strictement supérieur à 997. Par exemple, testons le nombre $N = 2 \times 3 \times 5 \times 7 \times 11 \times 13 \times 17$.

L'affichage est alors : DIVISIBLE PAR 2 ; 3 ; 5 ; 7 ; 11 ; 13 ; 17.

4) Le programme donnera certains diviseurs premiers mais pas tous pour tous les nombres qui admettent au moins un diviseur premier strictement supérieur à 997. Par exemple, testons le nombre $N = 2 \times 3 \times 5 \times 1009$. L'affichage est alors : DIVISIBLE PAR 2 ; 3 ; 5. Le programme ne donne donc pas 1 009 qui est un nombre premier.

5) Le programme ne donnera aucun diviseur premier pour tous les nombres dont la décomposition en produit de facteurs premiers ne comporte que des termes strictement supérieurs à 1 009. Par exemple, testons le nombre $N = 1\,013 \times 1\,019$. Le programme n'affiche alors aucun diviseur premier alors que pourtant 1 013 et 1 019 sont deux nombres premiers.