beaucoup de nombres premiers sur votre TI-57

Améliorer la recherche d'un nombre premier, ou tout simplement détecter si un nombre l'est ; afficher les nombres premiers supérieurs à un entier donné ; générer des nombres premiers avec 8 pas de programme ; voilà le menu proposé ce mois-ci à votre TI-57 ou à une autre calculatrice programmable.

L'exercice n° 14 (voir L'OI n° 36) propose un programme qui « reconnaît » un nombre premier. Pour cela, il affiche un « message » si le nombre étudié est premier, et sinon l'un de ses diviseurs. Voici une solution possible. Le « message » choisi est le nombre étudié, précédé du signe « - » s'il est premier, mais toute autre convention, comme faire afficher 1, 0, ou le nombre lui-même, conviendrait parfaitement dans ce cas. L'algorithme du programme est le suivant :

```
01  lire N
02  Q ← N/2
03  Si (Q = Int (Q)) Alors
    aller à 10 ; * N est pair
04  DIV ← 1
05  DIV ← DIV + 2
06  Q ← N/DIV
07  si (Q = Int (Q)) alors
    aller à 10 ; * N non premier
08  si (DIV < Int (√N))alors
    aller à 05
09  écrire -N ; aller à 11
10  écrire Q
11  fin
```

Affichage		
Pas	**Codes**	**Touches**
00	32 0	STO 0
cas des entiers pairs		
01	45	÷
02	02	2
03	85	=
04	32 7	STO 7
05	49	Int
06	66	x=t
07	512	GTO 2
initialisations		
08	33 0	RCL 0
09	24	√x
10	49	Int
11	32 3	STO 3
12	01	1
13	32 1	STO 1
boucle de calculs		
14	86 1	Lbl 1
15	02	2
16	34 1	SUM 1
17	33 0	RCL 0
18	45	÷
19	33 1	RCL 1
20	85	=
21	32 7	STO 7
22	49	Int
23	66	x=t
24	51 2	GTO 2
25	33 3	RCL 3
26	32 7	STO 7
27	33 1	RCL 1
28	-76	INV x ≥ t
29	51 1	GTO 1
résultat		
30	33 0	RCL 0
31	84	+/−
32	86 2	Lbl 2
33	81	R/S
34	71	RST

Mode d'emploi			
N°	**Instructions ou données**	**Touches**	**Affichage**
01	Passer en mode « programme »	LRN	
02	Introduire le programme		
03	Passer en mode « calcul »	LRN	
04	Initialiser le pointeur	RST	0.
05	Introduire N Afficher le résultat	R/S	N −N ou Q.
06	Pour un autre calcul, aller en 05		

Registres	
RO	N
R1	DIV
R3	Int ($\sqrt{N}$)
R7	Tests

On trouve alors que 11 est un diviseur de 417, 19 est un diviseur de 1 007, 3 est un diviseur de 513 et 7 un diviseur de 581. Les autres entiers proposés sont premiers.

Voici l'algorithme et le programme commenté d'une solution à l'exercice n° 15. Nous vous proposons un programme qui affiche la suite des entiers premiers supérieurs à un entier donné INF, fourni initialement. Ainsi, on pourra faire afficher, par exemple, la suite des entiers premiers supérieurs à 1 000 en introduisant 1 000 pour INF. On évite ainsi d'engendrer le début de cette suite si on le possède déjà.

```
01  lire INF
02  INF ← Int(INF/2)x2 + 1
03  N ← INF
04  N ← N+2
05  DIV ← 1
06  DIV ← DIV + 2
07  Q ← N/DIV
08  si (Q=Int(Q)) alors aller à 04
09  si (DIV < Int(√N)) alors aller à
    06
10  écrire N ; aller à 04
11  fin
```

Remarquez dès à présent que ce programme « boucle », son arrêt étant manuel.

Affichage		
Pas	Codes	Touches
00	45	÷
01	02	2
02	85	=
03	49	Int
04	55	x
05	02	2
06	75	+
07	01	1
08	85	=
09	32 0	STO 0
10	86 1	Lbl 1
11	02	2
12	34 0	SUM 0
13	33 0	RCL 0
14	24	$\sqrt{x}$
15	49	Int
16	32 3	STO 3
17	01	1
18	32 1	STO 1
Calculs sur N		
19	86 2	Lbl 2
20	02	2
21	34 1	SUM 1
22	33 0	RCL 0
23	45	÷
24	33 1	RCL 1
25	85	=
26	32 7	STO 7
27	49	Int
28	66	x=t
29	51 1	GTO 1
30	33 3	RCL 3
31	32 7	STO 7
32	33 1	RCL 1
33	-76	INV x ≥ t
34	51 2	GTO 2
35	33 0	RCL 0
36	36	Pause
37	51 1	GTO 1

$6i \pm 1$, avec i entier naturel supérieur ou égal à 1. On élimine ainsi les multiples de 2 et 3 et le programme est plus rapide, du moins on peut l'espérer.

Voici ce programme. Il ne diffère du précédent que par la façon dont il engendre la suite des nombres N.

```
01   I ← 0
02   F ← 1
03   I ← I+1
04   calcul de N
05   affichage de N s'il est pre-
     mier
06   calcul de N
07   affichage de N s'il est pre-
     mier
08   aller à 03
100  calcul de N
101  F ← -F
102  N ← 6x1 + F
103  retour
200  affichage de N s'il est pre-
     mier
201  DIV ← 1
202  DIV ← DIV + 2
203  Q ← N/DIV
206  si (Q=Int(Q)) alors aller à
     209 ; *N non premier
207  si (DIV < Int(√N)) alors
     aller à 202
208  écrire N
209  retour
```

Mode d'emploi			
N°	Instructions ou données	Touches	Affichage
01 ·· ·· 04	comme le programme précédent		
05	Introduire INF		INF
06	Afficher les entiers premiers supérieurs à INF	R/S	N.
07	Pour un autre calcul Aller en 04	R/S CLR	N. O.

Registres	
RO	N étudié
R1	DIV
R3	Int ($\sqrt{N}$)
R7	Tests

Remarquez que ce programme ne donne ni 2, ni 3 pour INF = 1. Ceci provient des initialisations choisies.

L'exercice n° 16 proposait d'améliorer la recherche des entiers premiers en ne testant que les nombres impairs de la forme

Affichage		
Pas	Codes	Touches
00	01	1
01	32 5	STO 5
Programme principal		
02	86 1	Lbl 1
03	01	1
04	34 4	SUM 4
05	61 4	SBR 4
06	61 5	SBR 5
07	61 4	SBR 4
08	61 5	SBR 5
09	51 1	GTO 1

Affichage		
Pas	Codes	Touches
Calcul de N		
10	86 4	Lbl 4
11	33 5	RCL 5
12	84	+/-
13	32 5	STO 5
14	33 4	RCL 4
15	55	x
16	06	6
17	75	+
18	33 5	RCL 5
19	85	=
20	32 0	STO 0
21	24	$\sqrt{x}$
22	49	Int
23	32 3	STO 3
24	-61	INV SBR
Affichage de N s'il est premier		
25	86 5	Lbl 5
26	01	1
27	32 1	STO 1
28	86 2	Lbl 2
29	02	2
30	34 1	SUM 1
31	33 0	RCL 0
32	45	÷
33	33 1	RCL 1
34	85	=
35	32 7	STO 7
36	49	Int
37	66	x=t
38	51 3	GTO 3
39	33 3	RCL 3
40	32 7	STO 7
41	33 1	RCL 1
42	-76	INV x ≥ t
43	51 2	GTO 2
44	33 0	RCL 0
45	36	Pause
46	86 3	Lbl 3
47	-61	INV SBR

Mode d'emploi			
N°	Instructions ou données	Touches	Affichage
01 .. 04	comme le programme précédent		
05	Initialiser les registres de données	INV C.t	0.
06	Afficher la suite des entiers premiers	R/S	ent. premiers

Registres	
R0 R1 R3	Comme précédent programme
R4	I
R5	F = ± 1
R7	Tests

Ce programme est donc décomposé en trois parties :

. Un sous-programme qui engendre la suite des entiers de la forme 6i ± 1, implanté aux pas 10 à 24 ; de 11 à 13, en enregistre en R5 1 ou -1 alternativement ; 14 à 20 calculent N et l'enregistrent en R0 ; 21 à 24 calculent la partie entière de $\sqrt{N}$ et l'enregistrent en R3.

. Un sous-programme (25 à 49) teste N et l'affiche s'il est premier. C'est le corps du programme précédent (voir les pas 09 à 37).

. Le programme principal enfin, des pas 00 à 09, assurent l'incrémentation de I et l'appel des 2 sous-programmes. Comme pour les précédents, l'arrêt du programme est manuel.

En remarquant que la suite des N=6 × I ± 1 est engendrée par la récurrence :

$$D_0 \leftarrow 2 \; ; \; N_0 \leftarrow 1$$
$$D_i \leftarrow 6 - D_{i-1}$$
$$N_i \leftarrow N_{i-1} + D_i$$

on peut écrire, comme le proposait l'exercice n° 17, un programme *a priori* plus performant, dont voici le principe :

```
01  D ← 2
02  N ← 1
03  D ← 6-D
04  N ← N+D
05  affichage de N s'il est premier
06  aller à 03
07  fin
```

Les registres utilisés seront :

Registres	
R0 R1 R3	Comme précédents programmes
R6	D
R7	Tests

Le mode d'emploi est le même que celui du précédent programme.

Affichage		
Pas	Codes	Touches
00	02	2
01	32 6	STO 6
02	01	1
03	32 0	STO 0
04	86 1	Lbl 1
05	06	6
06	65	-
07	33 6	RCL 6
08	85	=
09	32 6	STO 6
10	34 0	SUM 0
11	33 0	RCL 0
12	24	$\sqrt{x}$
13	49	Int
14	32 3	STO 3
Affichage de N s'il est premier		
15 .. 34	reprendre les pas 26 à 45 du programme précédent (en modifiant le pas 38)	
35	51 1	GTO 1

Attention : remplacer le pas 38 51 3 GTO 3 du programme précédent par :
27 51 1 GTO 1

Des formules pour les nombres premiers

Passons à l'étude de quelques formules donnant des nombres premiers. On ne connaît pas de formule qui donne tous les entiers premiers et seulement eux. Certaines, cependant, permettent de calculer *des* nombres premiers inférieurs à un entier qui dépend de la formule utilisée. En voici 2.

1. On pose $P(n) = n^2 - n + 41$, n étant un entier naturel. On voit immédiatement que $P(41) = 41^2$ n'est pas premier. *Vérifiez que P(n) est premier pour n = 1, 2, 3,... 40.* Pour cela :

> **Ex. n° 18 : écrivez un programme qui calcule P(n) pour tous les entiers n de 1 à 40.**

Complétez, à l'aide de ce programme, une table donnant n et P(n).

a) Afin d'éviter le calcul de n^2, vous pouvez utiliser la factorisation de Hörner et écrire la formule :

$$P(n) = n(n-1) + 41$$

Cette remarque conduit à une première version du programme.

b) Un petit calcul devrait vous permettre d'améliorer ce programme, il suffit de démontrer que :

$$P(n+1) - P(n) = 2n.$$

> **Ex. n° 19 : Modifiez votre programme en utilisant cette remarque.**

c) Voici enfin une remarque qui améliore encore les performances du programme. Puisque la différence de deux valeurs consécutives de P est 2n, les différences consécutives sont données par les multiples de 2.

Le début de la table donnant n, P(n), et P(n+1)–P(n) pour quelques valeurs de n devrait vous suggérer quelques améliorations.

> **Ex. n° 20 : Modifiez votre programme en utilisant cette nouvelle remarque.**

Si vous vous contentez de surveiller l'affichage, en vous dispensant du test d'arrêt, 8 pas de programme et le seul registre de données t (R7) sont suffisants !

2. On pose $Q(n) = N^2 - 79n + 1601$. Cette formule, très connue également, donne des entiers premiers pour n = 1, 2, 3, ..., 79.

> **Ex. n° 21 : Ecrivez un programme qui calcule et affiche Q(n) pour n = 1, ..., 79.**

Comme pour la formule précédente, vous pouvez, si cela vous intéresse, décomposer votre étude en trois parties :

a) Calcul de Q(n) par $Q(n) = n(n-79) + 1601$.

b) Refaire ce programme en constatant que :

$Q(n+1) = Q(n) + 2n - 78$

c) Optimiser en constatant que la différence de deux valeurs successives de Q est un multiple de 2, à partir de – 78.

Combien d'entiers premiers distincts obtenez-vous à l'aide de Q ?

3. Les nombres de Fermat sont les entiers donnés par la formule :

$$F(n) = 2^{(2^n)} + 1$$

dans laquelle n est un entier naturel. Par exemple, on a :

$F(2) = 2^{(2^2)} + 1 = 2^4 + 1 = 17.$

En 1665, Fermat a émis l'hypothèse que F(n) est premier pour tout entier n, mais en 1732, Euler a démontré que $F(5) = 2^{32} + 1$ est divisible par 641. Legendre, à son tour, a établi en 1780 que 274177 est un diviseur de F(6). Depuis, on a pu démontrer que F(n) n'est pas premier pour tout entier n entre 7 et 16, et pour bien d'autres valeurs, telles que 18, 28, 36, ...

> **Ex. n° 22 : Ecrivez un programme donnant le début de la suite des nombres de Fermat.**

Optimisez ce programme au point de vue du temps de calcul.

4. D'autres formules.

a) La suite semble donner des entiers premiers :

$S(n) = 1 + 2 \times 3 \times 5 \times 7 \times 11 \times ... \times n$ (attention n premier).

> **Ex. n° 23 : Ecrivez un programme qui calcule S(n) et l'affiche s'il est premier.**

b) Etudiez les nombres entiers qui s'écrivent : T(n) = 11...1 avec n chiffres égaux à 1.

Remarquez que T(n) n'est pas premier lorsque n > 2 est pair. *Pourquoi ? Et lorsque n est multiple de 3 ?*

Nous n'osons pas vous proposer d'écrire un programme testant les nombres T(n) pour n impair supérieur à 3. En effet, on est vite limité par les petites possibilités de TI.57. Dans ces limites, seul T(2) est premier. *Et ensuite ?*

Si le problème vous intéresse, et si vous disposez d'un calculateur plus puissant que la TI.57, vous pouvez écrire un programme permettant de travailler sur de grands nombres, mais cela dépasse le cadre de cet article.

Voilà pour des formules donnant des nombres premiers. Il en existe bien d'autres et nous vous renvoyons à l'abondante littérature qui fleurit sur le sujet, si cela vous intéresse.

Le mois prochain, nous étudierons des exercices sur la décomposition d'un entier en produit de facteurs premiers.

Christophe Haro