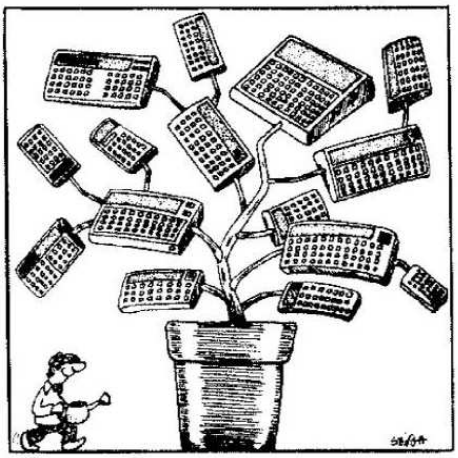
Un pot commun pour toutes les machines



Nombres composés, théorème de Fermat et TI-57

Reste de la division de 2^{n-1} par n
Programme pour TI-57
Auteur Gilles Nicolas

00	32	0	STO 0
01	45		÷
02	02		2
03	85		=
04	49		2nd Int
05	32	1	STO 1
06	02		2
07	32	2	STO 2
08	01		1
09	32	3	STO 3
10	86	0	2nd Lbl 0
11	19		2nd C.t.
12	33	2	RCL 2
13	23		x^2
14	61	2	SBR 2
15	32	2	STO 2
16	31	1	RCL 1
17	66		2nd x=t
18	51	1	GTO 1
19	45		÷
20	02		2
21	85		=
22	32	7	STO 7
23	49		2nd Int
24	32	1	STO 1
25	66		2nd x=t
26	51	0	GTO 0
27	33	2	RCL 2
28	55		×
29	33	3	RCL 3
30	85		=
31	61	2	SBR 2
32	32	3	STO 3
33	51	0	GTO 0
34	86	1	2nd Lbl 1
35	33	3	RCL 3
36	81		R/S
37	71		RST
38	86	2	2nd Lbl 2
39	32	4	STO 4
40	45		÷
41	33	0	RCL 0
42	85		=
43	49		2nd Int
44	55		×
45	33	0	RCL 0
46	85		=
47	−34	4	INV SUM 4
48	33	4	RCL 4
49	−61		INV SBR

■ Les nombres premiers continuent à faire l'objet des mêmes recherches : trouver des nombres premiers toujours plus grands en des temps toujours plus brefs. Si les gros ordinateurs sont capables de prouesses, les poquettes, sans pouvoir rivaliser, se débrouillent assez bien (sur HP-41, voir *l'Op* n° 1 page 54).

La TI-57, grâce au théorème de Fermat, peut rendre de précieux services aux lycéens. Elle leur permet de vérifier ce théorème et de dire, avec certitude de certains nombres, qu'ils ne sont pas premiers. Ce théorème de Fermat peut être énoncé ainsi : « Si p est un nombre premier et a un nombre quelconque non divisible par p (ce qui est forcément vrai si a est inférieur à p), alors ($a^{p-1} - 1$) est un multiple de p ». Ce qui peut aussi s'écrire "$a^{p-1} \equiv 1$ (mod. p)" et se dit "a^{p-1} est congru à 1 modulo p".

Utilisation du programme
Introduire n au clavier, presser sur RST puis sur R/S. Au bout de quelques instants, la TI-57 affiche le reste de la division de 2^{n-1} par n. Exemple 6981 RST R/S ; résultat = 4663. Par conséquent, 2^{6980} est congru à 4663 modulo 6981.

Ainsi, tout nombre p qui ne vérifie pas l'assertion $a^{p-1} \equiv 1$ (mod. p) est un nombre composé (c'est-à-dire que p n'est pas premier). La TI-57 nous dira, par exemple, que le nombre 9991 vérifie : $2^{9990} \equiv 3362$ (mod. 9991). Le nombre 9991 n'est donc pas premier. Et vous pouvez vérifier que 9991 est en fait le produit de 97 par 103. Mais, la TI-57 étant limitée par son affichage (8 chiffres seulement), le nombre p à tester ne peut avoir plus de 4 chiffres.

La méthode utilisée par le programme consiste à décomposer le nombre (n-1) en une somme de puissances de 2, dans l'ordre croissant des exposants. Comme $2^{2i+1} = (2^{2i})^2$, si l'on pose : $u_1 = 2^2$, $u_2 = u_1^2$, ..., $u_p = u_{p-1}^2$ et si l'on désigne par $\bar{u}_1$, $\bar{u}_2$, ..., $\bar{u}_p$ les restes des u_1, u_2, ..., u_p dans la division par n, on fera calculer $\bar{u}_i$ à chaque boucle (pas 12, 13 et sous-programme), le reste cherché étant égal à $\varepsilon_1 \bar{u}_1 \times \varepsilon_2 \bar{u}_2 \ldots \times \varepsilon_p \bar{u}_p$, et chaque ε_i valant 1 ou 0.

Ces ε_i sont les coefficients de 2^i dans la décomposition de (n-1) en une somme de puissances de 2 (coefficients obtenus grâce au test du pas 25 : on part de (n-1)/2, à chaque boucle on divise par 2 et l'on prend la partie entière jusqu'à ce que l'on arrive à zéro ; on est alors renseigné par le test du pas 17). Ce programme, qui utilise au maximum les possibilités de la TI-57, donne la réponse rapidement.

☐ Gilles Nicolas