divisez sur votre TI 57

du PGCD...

aux nombres premiers entre eux

Nous allons voir maintenant l'application des exercices précédents pour aboutir à un programme rapide de calcul du PGCD. Une utilisation directe : les nombres premiers entre eux.

L'exercice n° 9 proposait de calculer le PGCD de deux entiers en utilisant la définition de ce nombre. Le programme proposé ici utilise la méthode adoptée pour l'exercice n° 7 pour trouver les diviseurs d'un entier. Mais il ne fonctionne pas si le premier nombre introduit est nul. Vous n'aurez aucune peine à vous convaincre de la rapidité du calcul si l'on introduit d'abord le plus petit des deux nombres.

Registres	
RO	d = diviseur étudié
R1	a
R2	b
R3	a ÷ d ou b ÷ d
R7	Int (R3)

Affichage		
Pas	Codes	Touches
Enregist. des données		
00	32 0	STO 0
01	32 1	STO 1
02	81	R/S
03	32 2	STO 2
Le diviseur. de a est-il div. de b.		
04	66 0	Lbl 0
05	33 2	RCL 2
06	61 2	SBR 2
07	– 66	INV x = t
08	51 1	GTO 1
si oui, résultat		
09	33 0	RCL 0
10	81	R/S
11	71	RST
calcul d'un diviseur de a		
12	66 1	Lbl 1
13	56	Dsz
14	33 1	RCL 1
15	61 2	SBR 2
16	66	x = t
17	51 0	GTO 0
18	51 1	GTO 1
sous-programme de calcul du quotient par d		
19	66 2	Lbl 2
20	45	÷
21	33 0	RCL 0
22	85	=
23	32 3	STO 3
24	49	Int
25	22	x ⇄ t
26	33 3	RCL 3
27	– 61	INV SBR

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 a ≠ 0	R/S	a.
06	Introduire b Afficher PGCD (a ; b)	R/S	PGCD (a ; b).
07	Pour un autre calcul, aller en 05		

Le calcul du PGCD par soustractions successives peut se faire à l'aide d'un programme très court si on n'exige pas son fonctionnement pour a = b, a = o ou b = o. Seul R7(t) est utilisé.

Mode d'emploi			
N°	Instructions ou données	Touches	Affichage
05	Introduire a≠0	x ⇄ t	...
06	Introduire b≠a, b≠0 Afficher PGCD (a;b)	R/S	b. PGCD.
07	Pour un autre calcul, aller en 05		

Affichage		
Pas	Codes	Touches
00	22	x ⇄ t
01	– 76	INV x≥t
02	22	x⇄t
03	65	-
04	34 7	RCL 7
05	85	=
06	– 66	INV x=t
07	71	RST
08	81	R/S
09	71	RST

Malgré sa concision, ce programme reste très lent, et ce d'autant plus que la différence entre a et b est grande. L'algorithme d'Euclide permet d'écrire des programmes très rapides. En voici un ci-dessous, à utiliser

Affichage		
Pas	Codes	Touches
00	32 0	STO 0
01	81	R/S
02	32 1	STO 1
03	86 1	Lbl 1
04	33 0	RCL 0
05	65	–
06	43	(
07	14	CE
08	45	÷
09	33 1	RCL 1
10	44	)
11	49	Int
12	55	x
13	33 1	RCL 1
14	32 0	STO 0
15	85	=
16	32 1	STO 1
17	– 66	INV x=t
18	51 1	GTO 1
19	33 0	RCL 0
20	81	R/S
21	71	RST

comme celui de l'exercice n° 9, à ceci près que le second nombre introduit, b, doit être différent de 0.

On peut encore améliorer la rapidité de ce programme en éliminant les enregistrements de b dans a (pas 14) et du reste de la division dans b (pas 16). Voici la description de ce programme que nous vous laissons écrire. Il occupe environ 40 pas dans la mémoire programme d'une TI 57.

```
01 ' PGCD version 4
02 '
10   Lire A ; lire B
19 '
20   A ← A – B x INT (A÷B)
22   Si A = 0 alors aller en
     30 ; 'Fini
24   B ← B – A x INT (B÷A)
26   Si B ≠ 0 alors aller en 20
29 '
30   Afficher A+B
32 Terminer
```

Pour étudier la rapidité des différentes versions, vous pouvez les appliquer à 1 000 couples de nombres choisis par programme au hasard.

Les entiers étrangers

Les nombres premiers et les généralisations sur la notion de nombre premier ont inspiré une foule de recherches très fécondes. Les résultats sont souvent passionnants.

Nous commençons aujourd'hui une série sur les nombres premiers ou sur l'extension de la notion de nombre premier. Le but reste, bien entendu, la programmation d'une calculatrice.

Commençons par les entiers étrangers, avec un rappel de définition : « *Nous dirons que deux entiers sont étrangers, ou encore premiers entre eux, si leur PGCD est 1.* » 16 et 15 sont étrangers, puisque le PGCD (16 ; 15) = 1. 12 et 16 ne sont pas étrangers, puisque le PGCD (12 ; 16) = 4.

Deux termes consécutifs de la suite de Fibonacci sont premiers entre eux.

Soit n un naturel quelconque, 10 par exemple. Considérons tous les entiers inférieurs à n. On compte ceux qui sont étrangers à n et on note E(n) leur nombre.

Par exemple, pour n = 10, on a E(n) = 4 puisque 1, 3, 7 et 9 lui sont étrangers.

E(n) est donc, par définition, le nombre d'entiers inférieurs et étrangers à n.

Ex. n° 12 : Ecrivez un programme donnant E(n) pour n introduit en entrée.

Complétez une table donnant n et E(n) pour tous les entiers n jusqu'à 20, ou plus si vous en avez le courage.

Que remarquez-vous ?

Reprenez une table donnant n et E(n) pour n = 2, 2^2, 2^3, 2^4... Même travail avec les puissances successives de 3, de 5, de 7.

Alors que remarquez-vous ?

On note S(n) la somme des naturels inférieurs et étrangers à n.

Par exemple S(10) = 1+3+7+9 soit S(10) = 20.

Ex. n° 13 : Ecrivez un programme donnant S(n) pour n introduit en entrée.

Complétez une table donnant n, E(n), S(n) et $\frac{E(n)}{S(n)}$

Que remarquez-vous ?

Etant donné un entier n, on note d1, d2,..., dk ses diviseurs. On pose S'(n) = E(d1) + E(d2)+... + E(dk).

Complétez une table donnant n, E(n) et S'(n) pour tous les n inférieurs à 20 par exemple. Vous pouvez pour cela utiliser la table de l'exercice précédent et le programme de l'exercice n° 8 ou :

Ex. n° 13 bis : Ecrire un programme qui calcule et affiche S'(n) pour tout n inférieur à 20 (par exemple).

(Si la mémoire programme de votre calculatrice est suffisante.)

E(n) est appelé l'indicateur d'Euler. Nous le retrouverons lorsque nous étudierons plus particulièrement les nombres premiers. D'ici là, bonnes recherches et bons programmes !

Christophe Haro