divisez sur votre TI-57

les diviseurs d' un entier

Dans ce troisième article, nous présentons les solutions aux exercices 3, 4, 5 et 6, puis une étude rapide des diviseurs d'un entier.

Voici une solution possible à l'exercice n° 3. Cet exercice est simple si l'on se contente d'un programme fonctionnant hors des cas limites (a = 0 ; a < b ; b = 0). Il se complique légèrement sinon.

Registres	
R0	a;a-b;...;r
R1	q
R7	b

Affichage		
Pas	Codes	Touches
00	32 0	STO 0
01	R/S	R/S
02	32 7	STO 7
03	00	0
04	32 1	STO 1
05	66	x = t
06	51 9	GTO 9
07	86 0	Lbl 0
08	33 0	RCL 0
09	76	X ⩾ t
10	51 1	GTO 1
11	33 1	RCL 1
12	81	Pause
13	33 0	RCL 0
14	81	R/S
15	71	RST
16	86 1	Lbl 1
17	33 7	RCL 7
18	− 34 0	INV SUM 0
19	01	1
20	34 1	SUM 1
21	51 0	GTO 0

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	R/S	a.
06	Introduire b afficher q puis r	R/S	b q. puis r.
07	Pour un autre calcul : . Si clignotement . Sinon aller en 05	CE puis aller en 04	

Le reste de la division est évidemment contenu dans R0 : c'est a lorsque a < b.

Ce programme est lent, et d'autant plus que la différence a − b est grande.

Combien de boucles nécessite une division donnée ?

Une solution à l'exercice n° 4 est présentée page ci-contre. Il s'emploie comme le précédent. Seul l'affichage en ligne 05 est modifié en 0.

Registres	
R0	a
R1	b
R2	q
R7	Multiples de b

Comment fonctionne ce programme ?

Les pas 00 à 09 initialisent les registres. On y trouve en 05-06 le test de nullité de b. Remarquez que q est initialisé à − 1. Sans cette précaution, on obtient en R2 le quotient approché à une unité près par excès.

Les pas 18 à 20 assurent l'affichage du résultat et le retour automatique du pointeur à 00. La boucle de calcul apparait aux pas 10 à 17 : 11 et 12 génèrent les multiples de b, 13 et 14 les comptent, 15 et 16 comparent le dernier multiple de b à a.

Il faut évidemment une fameuse dose de patience pour attendre le résultat d'une division comme 128 436 par 45 par exemple.

Affichage		
Pas	Codes	Touches
00	32 0	STO 0
01	00	0
02	32 7	STO 7
03	81	R/S
04	32 1	STO 1
05	66	x = t
06	51 9	GTO 9
07	01	1
08	84	+/−
09	32 2	STO 2
10	86 0	Lbl 0
11	33 1	RCL 1
12	34 7	SUM 7
13	01	1
14	34 2	SUM 2
15	33 0	RCL 0
16	76	x ≥ t
17	51 0	GTO 0
18	33 2	RCL 2
19	81	R/S
20	71	RST

L'algorithme de calcul des décimales d'un quotient est obtenu en remarquant que pour chacune d'elle, on divise le produit du dernier reste par 10. C'est ce que nous faisons « à la main » quand on « abaisse » un zéro. Cet algorithme peut se décrire ainsi :

1. FAIRE A ← a ; B ← b
2. AFFICHER INT (A ÷ B)
3. R ← A - BxINT (A ÷ B) ; A ← 10xR
4. ALLER EN 2.

Voici un programme basé sur cet algorithme. Il donne les décimales une par une et n'utilise que R7 comme registre de données.

Affichage		
Pas	Codes	Touches
00	65	−
01	43	(
02	14	CE
03	45	÷
04	33 7	RCL 7
05	44	)
06	49	Int
07	36	Pause
08	55	×
09	33 7	RCL 7
10	85	=
11	55	×
12	01	1
13	00	0
14	85	=
15	71	RST

Le fonctionnement de ce programme est évident et n'appelle aucun commentaire.

Voici quelques-uns des résultats demandés :
1 ÷ 81 = 0,012 345 679 012...
Que remarquez-vous ?
1 ÷ 97 = 0,010 309 278 350...
Quelle est la période de ce développement ?
1 ÷ 98 = 0,010 204 081 632...
Même question.

Pour obtenir les décimales 5 par 5, il suffit de multiplier le reste r par 10^5 au lieu de 10. Le programme est modifié comme suit :

00 . . . 11	sans changement	
12	05	5
13	− 18	INV log
14	85	=
15	71	RST

La séquence 5 INV log amène 10^5 à l'affichage.

Après cette modification, les zéros à gauche des « paquets » de décimales ne sont pas affichés. Ainsi, si on demande le développement de 1 ÷ 81 par « paquets » de 5 décimales, le deuxième affichage est 1234, alors qu'il faut lire 01234.

Mode d'emploi			
N°	Données ou instructions	Touches	Affichage
01 . . 04			
05	Introduire b	x t	?
06	Introduire a Afficher INT (a ÷ b) puis les décimales	R/S	décimales
07	Pour un autre calcul	R/S CLR	0.
	Aller à 04		

Ne pas omettre le CLR à la ligne 07 du mode d'emploi. Sans cette précaution, le comportement du calculateur devient « bizarre ». Le programme, en effet, ne s'arrête pas automatiquement. L'arrêt manuel peut se faire en n'importe quel pas. Il arrive donc que les registres internes du calculateur ne soient pas vides. Le passage à un autre calcul provoque alors ceux en attente sur les nouvelles données, d'où parfois des résultats fantaisistes. L'instruction CLR a pour but de « nettoyer » les registres internes.

La solution de l'exercice n° 6 est simple si on se contente de chercher la période du développement de l'INVERSE d'un entier. Si on cherche un programme fonctionnant dans tous les cas, le problème devient plus compliqué.

Voici une solution possible. Ce programme fonctionne dans « presque » tous les cas. *Quand ne fonctionne-t-il pas ?* (Essayez, par exemple, 469 ÷ 300).

Affichage		
Pas	Codes	Touches
00	32 0	STO 0
01	00	0
02	32 2	STO 2
03	81	R/S
04	33 1	STO 1
05	61 2	SBR 2
06	33 3	STO 3
07	86 0	Lbl 0
08	00	0
09	66	x = t
10	51 1	GTO 1
11	01	1
12	34 2	SUM 2
13	61 2	SBR 2
14	34 3	RCL 3
15	− 66	INV x = t
16	51 0	GTO 0
17	34 2	RCL 2
18	86 1	Lbl 1
19	81	R/S
20	71	RST
21	86 2	Lbl 2
22	34 0	RCL 0
23	65	−
24	43	(
25	34 0	RCL 0
26	45	÷
27	34 1	RCL 1
28	44	)
29	49	Int
30	55	×
31	34 1	RCL 1
32	85	=
33	55	×
34	01	1
35	00	0
36	85	=
37	33 0	STO 0
38	33 7	STO 7
39	− 61	INV SBR

Registres	
R0	a
R1	b
R2	I
R3	$10xr_1$, avec r_1 le reste de la première division
R7	10xr

Les pas 21 à 39 constituent le sous-programme qui multiplie par 10 le reste d'une division et enregistre le résultat en R0 pour une nouvelle division et en R7 pour le test d'arrêt.

Les pas 00 à 04 permettent l'introduction des données et initialisent I (R2) à 0.

05 appelle le sous-programme de calcul du reste de la division de a par b.

De 07 à 10, on teste la nullité de ce reste. S'il est nul, c'est que le quotient ne présente pas de période (cas d'un quotient exact par exemple). D'où le transfert à 19 et 20 qui assurent l'arrêt sur 0 et l'initialisation automatique du pointeur pour un nouveau calcul.

Si ce reste n'est pas nul, 11 et 12 incrémentent I de 1, 13 calcule le nouveau reste multiplié par 10 et 14-15 le comparent au premier reste obtenu. S'ils sont égaux, on a trouvé la période. 17 rappelle I pour l'affichage et l'arrêt. Sinon, on recommence en 07 une nouvelle boucle.

On peut bien sûr essayer tous les entiers...

L'exercice n° 2 permettait de reconnaître si un entier b est un diviseur d'un entier a. On souhaite à présent, faire afficher par le calculateur la liste des diviseurs de a.

Ex. n° 7 : Ecrivez un programme qui affiche la liste des diviseurs d'un entier a introduit en entrée.

Une méthode consiste à diviser a par tous les entiers qui lui sont inférieurs. On fait afficher les nombres pour lesquels le quotient et sa partie entière sont égaux. On arrête l'itération lorsque la liste des entiers inférieurs à a est épuisée.

Ce programme est très lent, puisque chaque liste demande un nombre de divisions égal à a.

Il est possible de le rendre plus efficace. En effet, si b est un diviseur de a, le quotient de a par b est aussi un diviseur de a. On sait, d'autre part, que b et q ne peuvent être simultanément supérieurs à $\sqrt{a}$. On peut donc limiter la recherche des diviseurs de a aux entiers inférieurs à $\sqrt{a}$. Pour étudier 17 par exemple, il suffit de le diviser par 2, 3 et 4 puisque $\sqrt{17} \simeq 4,1$.

Ex. n° 8 : Ecrivez un programme qui utilise ces remarques pour afficher les diviseurs d'un entier.

Voilà donc pour les diviseurs d'un entier. Nous espérons que ces quelques lignes vous auront donné d'autres idées de programmes. Le prochain article vous permettra d'étudier différents algorithmes de calcul du PGCD de deux entiers.

Bons programmes.

Christophe Haro

Mode d'emploi			
N°	Instructions ou données	Touches	Affichage
05	Introduire le dividende a	R/S	0.
06	Introduire le diviseur b		b
	Afficher I	R/S	I.
07	Pour un autre calcul, aller en 05		