

HIRISEZ VOS PROGRAMMES !

■ Si vous êtes utilisateur de TI-58/59, vous connaissez sans doute l'existence de la fonction HIR. Il s'agit d'une instruction multivalente, qui n'est pas révélée par les notices ou modes d'emploi, car elle est manifestement réservée aux calculs internes. En tout cas, c'est une fonction occulte, en principe interdite à l'utilisateur, mais que l'ingéniosité de quelques pionniers fouineurs a mise pratiquement à nu et à la disposition de tous.

Nous vous en rappelons l'essentiel et vous proposons (page suivante) un petit programme à titre d'exemple.

La fonction HIR, accessible uniquement en mode LRN, a pour objet de substituer aux instructions relatives aux mémoires (STO, RCL, SUM, Prd) d'autres instructions dont l'intérêt principal est de ne consommer que des pas de programme, mais pas de contrepartie de réservation de mémoire. Cette propriété permet de soulager la partition mémoire, voire de travailler en partition zéro (0 Op 17).

Six fonctions, huit mémoires

Ainsi, la fonction HIR donne accès à 6 fonctions et à 8 « mémoires » que nous appellerons registres HIR (en fait, ce sont des registres de gestion de parenthèses détournés de leur fonction). Pour introduire une instruction en mode HIR, il faut faire suivre HIR d'un nombre de deux chiffres occupant un seul pas de programmé. Comme on le voit avec le tableau de correspondance ci-contre, le premier chiffre désigne la nature de la fonction, le deuxième (X) le numéro du registre. Par exemple HIR 14 est équivalent à RCL 4. Les registres adressables vont de 1 à 8, ou plutôt de 8 à 1, car il est impératif de les assigner dans l'ordre décroissant.

Comment écrire un programme en mettant à profit l'instruction HIR ? Mes mésaventures[®] personnelles me



poussent à vous conseiller vivement de l'écrire d'abord en langage clair et, une fois qu'il tourne parfaitement, de le transcrire en HIR. En effet, la mise au point d'un programme écrit directement en HIR se révèle pratiquement impossible, à tout le moins déraisonnable. N'oubliez pas en effet que si vous voulez contrôler la valeur d'une variable dans un programme écrit en HIR, la touche RCL est inopérante, que ce soit en mode calcul ou en mode LRN.

Tableau de correspondance

Instruction HIR	Instruction AOS
HIR 0X	STO X
HIR 1X	RCL X
HIR 3X	SUM X
HIR 4X	Prd X
HIR 5X	INV SUM X
HIR 6X	INV Prd X

Dans la colonne de gauche, X représente le numéro du registre interne. Dans celle de droite, il représente le numéro d'une mémoire de données.

Pour transcrire votre programme, utilisez les instructions en HIR selon les clés du tableau de correspondance, puis introduisez le programme en appliquant les règles suivantes :

- l'instruction HIR a pour code 82. Comme HIR n'existe pas au clavier,

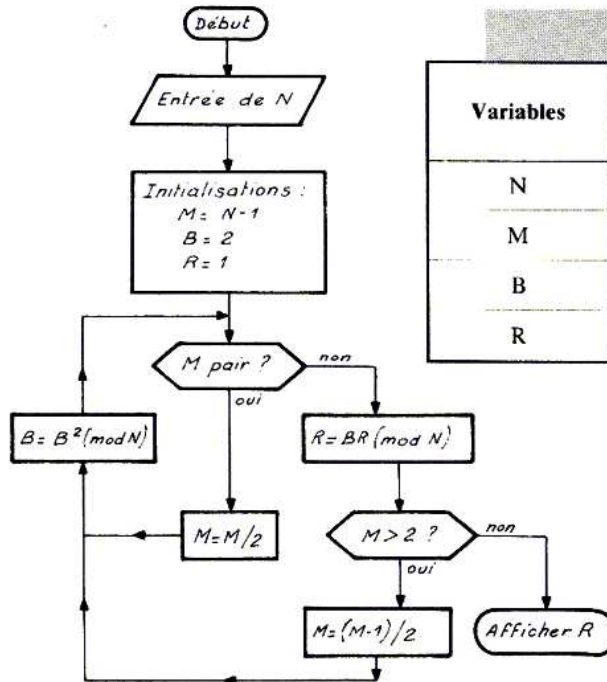
il faut entrer le code 82 en un seul pas de programme, ce qui ne peut pas se faire en introduisant successivement 8 et 2. Tapez donc STO 82 et annulez STO. Il reste 82, donc HIR. En pratique, il convient de presser successivement sur les touches STO 8 2 BST BST 2nd DEL SST .

• nous avons vu que HIR doit être obligatoirement suivi d'un code de deux chiffres (le premier étant éventuellement un zéro) également en un seul pas. Si ce code correspond à une touche, il suffit de l'actionner. Ainsi, pour introduire HIR 13, tapez HIR selon la technique exposée au paragraphe précédent, puis le faire suivre de C (13 est le code de C). Mais pour HIR 64, c'est plus délicat : vous pouvez, après HIR, introduire la série de touches correspondant au code 64 (2nd Prd 2nd Ind), soit taper STO 64 et annuler l'instruction STO, comme pour l'introduction de HIR.

N'oubliez surtout pas d'assigner les registres HIR par ordre décroissant, à partir de 8.

Pour illustrer le mécanisme de la pile de registres accessibles grâce à HIR, nous vous proposons un programme de test de Fermat écrit en AOS normal, puis transcrit en HIR.

Rappelons que le test de Fermat a pour objet de déterminer si un nombre N est composé ou non. Il suffit pour cela de calculer le reste R de la division de B^N par N, B étant un nombre quelconque, non divisible par N (on prend généralement $B = 2$). Si R est différent de 1, N est composé ; si R est égal à 1, N est présumé premier (avec une faible marge d'incerti-



Variables	Registres	
	En AOS	En HIR
N	00	8
M	01	7
B	02	6
R	03	5

résultat par N) sont effectuées par le même sous-programme E.

Le programme en HIR occupe deux pas de moins que le programme normal (79 pas contre 81) en raison de la compacité des instructions HIR 57 et HIR 67 par rapport à leurs homologues normaux

INV SUM 7 et INV Prd 7. En outre, il tourne un peu plus vite : pour $N = 1\ 037$, le résultat ($R = 815$) est affiché en 30'9" par le programme normal et en 28'5" par le programme HIR.

Notez surtout que le programme HIR n'utilise que quatre variables et peut être introduit en partition zéro, laissant libres 401 pas sur TI-58. Mais il vous est parfaitement loisible de panacher les instructions en HIR et en AOS normal dans le corps d'un programme, à plus forte raison dans deux programmes distincts.

Pierre Ladislas GEDO

Test de Fermat

Programme en AOS normal	Transcription en HIR
(00-04) Lbl A CP STO 00	(00-04) Lbl A CP HIR 08
(05-09) - 1 = STO 01	(05-09) - 1 = HIR 07
(10-12) 2 STO 02	(10-12) 2 HIR 06
(13-15) 1 STO 03	(13-15) 1 HIR 05
(16-17) GTO D	(16-17) GTO D
(18-23) Lbl B 1 INV SUM 01	(18-22) Lbl B 1 HIR 57
(24-29) Lbl C 2 INV Prd 01	(23-27) Lbl C 2 HIR 67
(30-35) RCL 02 SBR E STO 02	(28-33) HIR 16 SBR E HIR 06
(36-46) Lbl D RCL 01 ÷ 2 = INV Int x = t C	(34-44) Lbl D HIR 17 ÷ 2 = INV Int x = t C
(47-52) RCL 03 SBR E STO 03	(45-50) HIR 15 SBR E HIR 05
(53-59) RCL 01 - 2 = x ≥ t B	(51-57) HIR 17 - 2 = x ≥ t B
(60-62) RCL 04 R/S	(58-60) HIR 15 R/S
(63-80) Lbl E × RCL 02 - (CE ÷ RCL 00) Int × RCL 00 = INV SBR	(61-78) Lbl E × HIR 16 - (CE ÷ HIR 18) Int × HIR 18 = INV SBR

L'utilisation de HIR et des registres internes de la calculatrice permet ici d'économiser 4 mémoires de données.

