

Ah !

si vous aviez su...

La multiplication des mémoires

■ Les registres de mémoire d'un ordinateur de poche sont capables de stocker des valeurs aussi grandes que 9.10^9 avec dix chiffres significatifs, cela en base 10. Mais bien souvent, l'utilisateur mobilise un registre entier pour stocker un nombre plus petit que 100, soit deux chiffres sur dix.

A chaque fois que l'on travaillera sur un intervalle borné $I = [0, x[$ où x est inférieur à 100, l'astuce suivante permettra d'éviter tout ce gâchis. Si l'on connaît leur borne supérieure, on peut coder plusieurs données numériques dans une même mémoire. La méthode consiste à exprimer chaque valeur sous la forme de la $n^{\text{ième}}$ décimale d'une grandeur convertie en base (x).

La routine suivante pour TI-58 et 59 utilise le registre R(01) pour la base (x) et R(00) comme adresse du registre d'arrivée. Elle permet de

Des mémoires bien pleines

Utilitaire pour TI-58 et 59

Auteur Michel Arditti

Copyright l'Ordinateur de poche et l'auteur

```
000 76 LBL
001 43 RCL
002 42 STD
003 02 02
004 69 DP
005 32 32
006 73 RC*
007 00 00
008 55 ÷
009 53 (
010 43 RCL
011 01 01
012 45 Y*
013 43 RCL
014 02 02
015 54 )
016 42 STD
017 03 03
018 95 =
019 52 EE
020 22 INV
021 52 EE
022 59 INT
```

```
023 75 -
024 53 (
025 24 CE
026 55 ÷
027 43 RCL
028 01 01
029 54 )
030 59 INT
031 65 *
032 43 RCL
033 01 01
034 95 =
035 92 RTN
036 76 LBL
037 42 STD
038 71 SBR
039 43 RCL
040 94 +/-
041 85 +
042 32 X!T
043 95 =
044 65 *
045 43 RCL
046 03 03
047 95 =
048 74 SM*
049 00 00
050 91 R/S
```

Exemple d'utilisation

Pour coder la suite 26, 19, 11, 13, 5, 16, 11, 14 dans le registre (R05), entrer au clavier :

5 STO 00

0 STO 05

27 STO 01

puis $26 \times \approx t 1$ SBR STO

$19 \times \approx t 2$ SBR STO

$11 \times \approx t 3$ SBR STO

(...)

$14 \times \approx t 8$ SBR STO

Pour revoir la donnée $n^{\circ} x$:
x SBR RCL

coder une trentaine de valeurs binaires, ou une vingtaine de chiffres entre (0) et (x) dans un même registre, mais cette quantité dépend des valeurs stockées. Elle est maximale lorsque les valeurs sont entrées par ordre décroissant, puisque (N_1, \dots, N_n) est codé comme suit : $N_n \times (\text{base})^{n-1} + \dots + N_2 \times (\text{base})^1 + N_1 \times (\text{base})^0$.

En règle générale, il faut éviter que la valeur contenue dans le registre d'arrivée soit supérieure à 9×10^9 (9×10^{12} pour TI-58 et 59). Le programme étant court permet de libérer de nombreux registres précieux, comme le montre l'exemple ci-dessus.

□ Michel Arditti