Les factorielles : croître et multiplier

La curiosité est-elle un vilain défaut ? Si l'on n'a pas peur des grands nombres et si l'on utilise une TI 58 ou 59, voici un programme de calcul de (grosses !) factorielles... à l'unité près.

■ Factorielle ! Dans le " dictionnaire imaginaire des mathématiques farfelues ", on trouve la définition suivante : *produit de la multiplication par lui-même d'un nombre qui rapetisse jusqu'à ce qu'il n'en reste plus rien.*

C'est clair ? Non ! Pour un nombre entier positif n, n! (prononcez " ène-factorielle ") est le résultat que l'on obtient en effectuant le produit de ce nombre par tous les entiers positifs qui le précèdent.

D'où $n! = n \times (n-1) \times (n-2) \times \ldots 3 \times 2 \times 1$. Dans la pratique, bien entendu, on s'épargne la dernière multiplication (par 1) qui n'apporte rien. C'est ainsi que $6! = 6 \times 5 \times 4 \times 3 \times 2 = 720$.

La première propriété du nombre obtenu en calculant n! est (La Palisse n'aurait pas fait mieux) d'être divisible par tous les entiers compris entre 1 et n, ce qui fournit aux amateurs l'occasion de construire un algorithme qui détermine si un nombre est ou non le résultat d'un calcul de factorielle.

Une autre particularité des factorielles est la facilité avec laquelle on peut les simplifier quand elles apparaissent dans des fractions, ce qui permet souvent des calculs " à la main " (comprenez : sans votre micropoche préféré).

L'exemple ci-dessous est volontairement choisi pour se simplifier de lui-même, mais il arrive parfois qu'avec la meilleure volonté du monde, le meilleur papier et le meilleur crayon, le calcul devienne un travail de Romain. C'est là que le micropoche entre en jeu et vous verrez qu'il peut dépasser — et de loin — les dix chiffres traditionnellement autorisés par l'affichage.

A quoi ça sert ?

Les factorielles se rencontrent fréquemment dans les problèmes de dénombrement et de probabilités.

Dénombrement : mathématiquement, n! est le nombre de bijections sur un ensemble de n éléments. Cela ne vous dit rien ? Alors prenons un exemple : soit un ensemble de pions numérotés de 1 à 3. On veut savoir combien il existe de manières différentes pour les ranger dans des cases repérées A, B, C.

On trouve les ordres suivants :

A	B	C
1	2	3
1	3	2
2	1	3
2	3	1
3	1	2
3	2	1

Ce qui fait 6 possibilités, et comme par hasard, $6 = 3 \times 2 \times 1 = 3!$ Plus généralement, pour n pions numérotés de 1 à n, on a toujours n! possibilités. Un tiercé se courant avec 10 chevaux peut donc donner à l'arrivée 10! = 3628800 ordres différents.

Probabilités : intuitivement, pour un jeu de hasard, si l'on a 100 cas possibles et que dans ces 100 cas un seul permet de gagner, on dit que l'on a une chance sur 100. Les pro-

Exemple : calcul de

$$\frac{30! \times 8! \times 3!}{28! \times 9!}$$

Si l'on calcule la valeur du numérateur, puis celle du dénominateur, on obtient :

$$\frac{6,4169972 \quad 10^{37}}{1,1063788 \quad 10^{35}} = 580$$

Mais si l'on remarque que, par exemple :

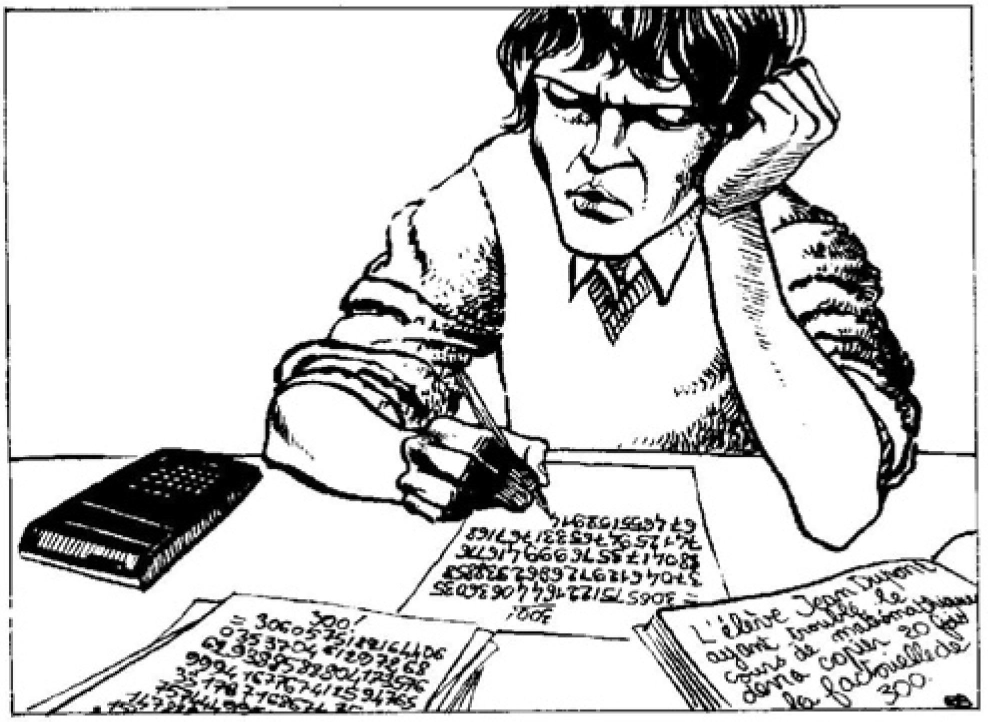
$$\frac{30!}{28!} = \frac{30 \times 29 \times 28 \times 27 \times \ldots \times 3 \times 2}{28 \times 27 \times \ldots \times 3 \times 2} = 30 \times 29$$

On peut alors simplifier notre fraction ainsi :

$$\frac{30! \times 8! \times 3!}{28! \times 9!} = \frac{30 \times 29 \times 3 \times 2}{9} = \frac{90 \times 29 \times 2}{9} = 10 \times 29 \times 2 = 580$$

Calcul effectué sans autre accessoire qu'un papier et un crayon.

Les factorielles : croître et multiplier



babilistes emploient la formule suivante : " la probabilité de gain est égal à 1/100 ". Pour un jeu où l'on gagne tout le temps (lucratif peut-être, mais quel ennui !) la probabilité est égale à 1. Pour un jeu où l'on a aucune chance de gain (encore plus ennuyeux que le précédent), la probabilité est égale à 0.

La probabilité de gain est donc égale à :

$$\frac{\text{nombre de cas favorables}}{\text{nombre de cas possibles}}$$

Prenons un exemple, construisons un jeu avec les trois pions numérotés : on mise sur une possibilité, par exemple 2, 3, 1, puis on range au hasard les pions dans les 3 cases, et l'on gagne si les pions se retrouvent dans l'ordre sur lequel on avait misé. On a donc 3! cas possibles, et un seul cas favorable. La probabilité de gain est donc égal à

$\frac{1}{3!}$, soit $\frac{1}{6}$

Sorties du contexte des applications probabilistes, les factorielles ont intéressé et amusé de tout temps les mathématiciens, en herbe ou... chevronnés. Il est vrai que cette série de multiplications envahissantes (personne n'est encore parvenu à la mettre sous forme d'expression ne dépendant que de n) continue, au même titre que les nombres premiers, à fasciner les chercheurs en théorie des nombres.

Le problème majeur est, sans compter les risques d'erreur, d'effectuer des multiplications à n'en plus finir si l'on veut savoir à tout prix, à l'unité près, les valeurs de factorielles élevées.

Chercheurs acharnés, économisez vos nuits blanches, la TI 58 ou 59 travaillera pour vous.

Un programme court mais des résultats gigantesques

Très bon exercice pour le programmeur débutant : une boucle simple tout simplement et l'on a un programme de calcul de n!.

Mais, dès que l'on dépasse 13!, survient la triste réalité du résultat aux chiffres significatifs tronqués et honteusement remplacés par une puissance de 10. Où sont les nombres immenses promis par nos ordinateurs ?

Enfin, on s'y fait. Puis on pousse plus loin, de plus en plus grand et... crac ! Comment ? Mon ordinateur miniaturisé serait incapable d'avaler plus que 69!, même en tronquant ? Il ne faut pas se résigner. Avec un autre programme, on pourra obtenir les factorielles de tous les nombres compris entre 1 et 490, avec un résultat exact à l'unité près.

La première lecture de la liste des instructions montre une abondance de HIR : le programme sort un peu des sentiers battus du Texas.

Vous avez donc, dorénavant, la faculté de connaître exactement le nombre d'arrivées possibles pour un tiercé jusqu'à... 490 partants.

Dans un premier temps, il faut introduire, en partition 10 Op 17 (4 Op 17 pour TI 58) les 160 lignes qui composent le programme. Dans le cas d'un fonctionnement sans imprimante, on remplacera les lignes 144 et 151 par R/S et les lignes 152 à 159 par la séquence : CLR R/S.

Sur TI 59, on pourra calculer n! pour n inférieur ou égal à 490. Sur TI 58, je dois vous avouer que je n'ai pas eu le temps de rechercher la limite supérieure, mais si vous êtes curieux, je suis certain que vous le ferez vous-même.

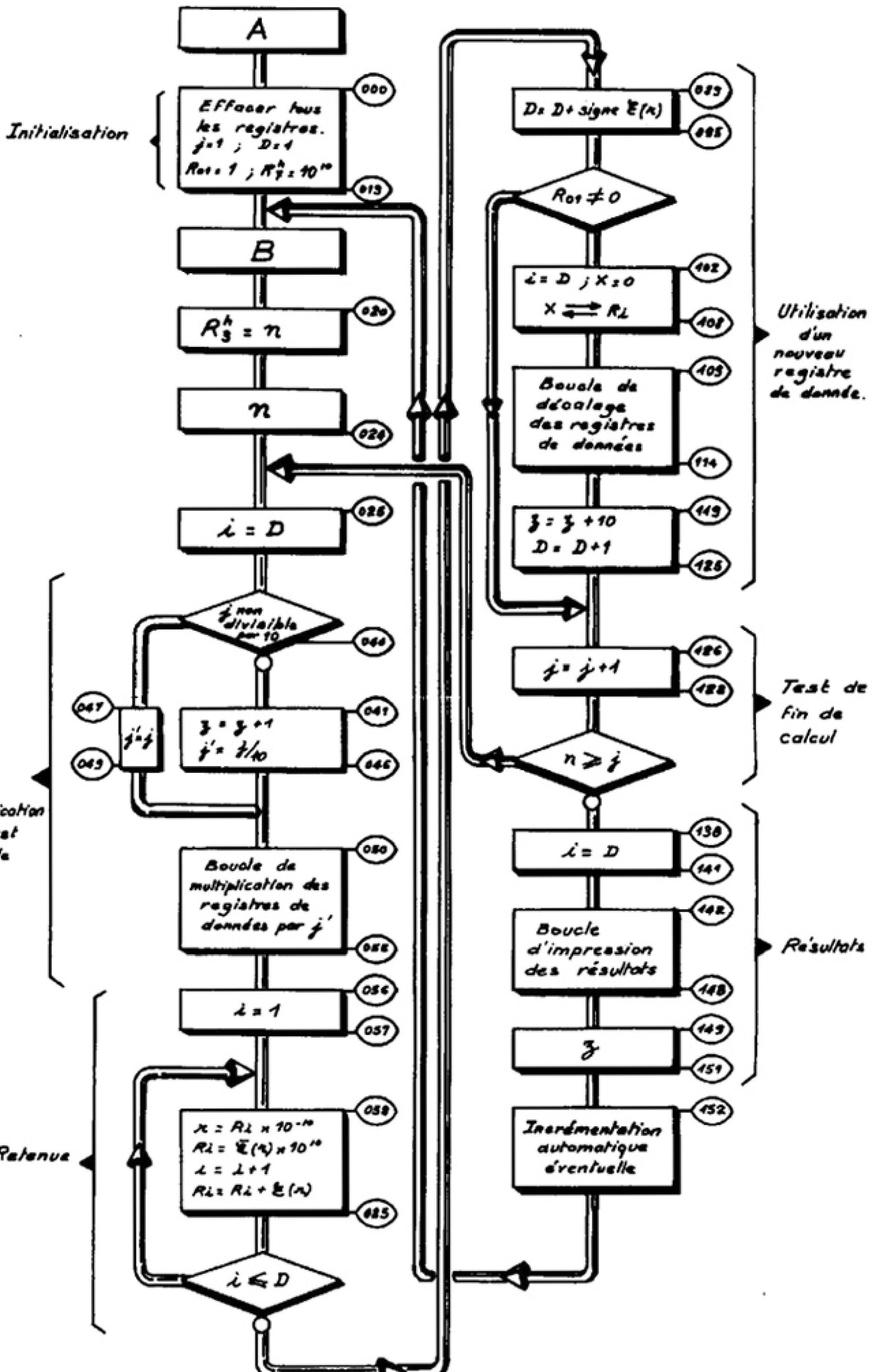
Une fois le programme engrangé dans le micropoche, il suffit d'afficher le nombre dont on désire connaître la factorielle et de presser sur la touche A. Le temps d'exécution varie évidemment selon la grandeur de n. A titre indicatif le calcul de 3! se fait en 13 secondes, celui de 20! demande une minute et demie, celui de 35! 3 minutes et 25 secondes, et pour obtenir 490! vous devrez attendre huit heures. Cela peut paraître beaucoup, mais le même calcul à la main réclamerait sans doute des mois et quelques kilos de papier.

Une fois l'exécution terminée, on a une série de nombres que l'on affiche successivement en appuyant sur R/S. Le dernier nombre obtenu représente une puissance de 10, par exemple 5. Cela indique que l'on doit ajouter 5 zéros au résultat final pour obtenir n!.

La série de nombres affichés représente n! par tranches de 10 chiffres (tout nombre comportant moins de 10 chiffres devant être complété à gauche par des zéros).

Dans la version " avec imprimante ", ces résultats sont automatiquement imprimés.

Organigramme général:



Les factorielles : croître et multiplier

Exemple : Calcul de 17!
17.
3556.
8742809600.
1.

Le premier nombre imprimé est 17 (cela permet aux étourdis de savoir de quelle factorielle il s'agit). En mettant le deuxième et le troisième " bout à bout ", on obtient : 35568742809600. le dernier nombre imprimé est 1, ce qui signifie que l'on doit encore ajouter un zéro à notre résultat intermédiaire. Donc, 17! = 355687428096000

Pour un nouveau calcul, celui de m!, plusieurs cas se présentent :

- si m<n, reprendre le mode d'emploi précédent,
- si m=n (pour éventuellement revoir les résultats), faire GTO 138 ; R/S,
- si m>n, introduire m et presser sur B.

Ceux et celles qui disposent de l'imprimante peuvent obtenir des factorielles successives sans réintroduire à chaque fois un nouveau nombre n. Il suffit de remplacer la ligne 153 par 1 ou 2 ou 3... ou 9, selon la progression désirée, puis de reprendre la suite d'opérations décrite dans le mode d'emploi, pour obtenir les valeurs successives de n!, (n+**a**)!, (n+**2a**)!, (n+**3a**)! etc... où **a** est le chiffre qui aura été placé à la ligne 153. (Ne sortez pas de chez vous si l'exécution est commencée, il en va de la vie de votre stock de papier thermique.)

Algorithmes : globalement, on a deux algorithmes imbriqués. Le premier consiste à incrémenter un registre " multiplicateur ", que l'on notera j. On a alors n! = 1×2×3×... n, où j varie de 1 à n.

Le deuxième permet la gestion des registres en " multiple précision ". Un pointeur que l'on notera i permet d'indexer le registre sur lequel on travaille. Ce pointeur peut prendre les valeurs comprises entre 1 et D, où D indique le " degré de remplissage " des registres, c'est-à-dire le numéro le plus élevé des registres de calcul qui deviennent,

Calcul de factorielles
Auteur : Bernard Kokanosky


```
000  76  LBL
001  11  A
002  47  CMS
003  32  X:T
004  69  OP
005  00  00
006  01  1
007  82  HIR
008  06  06
009  82  HIR
010  05  05
011  42  STO
012  01  01
013  52  EE
014  01  1
015  00  0
016  82  HIR
017  07  07
018  25  CLR
019  32  X:T
020  76  LBL
021  12  B
022  82  HIR
023  03  03
024  99  PRT
025  82  HIR
026  15  15
027  42  STO
028  00  00
029  82  HIR
030  16  16
031  55  ÷
032  01  1
033  00  0
034  95  =
035  22  INV
036  59  INT
037  22  INV
038  67  EQ
039  00  00
040  47  47
041  01  1
042  82  HIR
043  38  38
044  93  .
045  01  1
046  65  ×
047  82  HIR
048  16  16
049  95  =
050  64  PD*
051  00  00
052  97  DSZ
053  00  00
054  00  00
055  50  50
056  69  OP
057  20  20
058  73  RC*
059  00  00
060  55  ÷
061  82  HIR
062  17  17
063  72  ST*
064  00  00
065  95  =
066  82  HIR
067  04  04
068  22  INV
069  59  INT
070  64  PD*
071  00  00
072  69  OP
073  20  20
074  82  HIR
075  14  14
076  59  INT
077  74  SM*
078  00  00
079  24  CE
080  82  HIR
081  15  15
082  75  -
083  43  RCL
084  00  00
085  95  =
086  77  GE
087  00  00
088  58  58
089  82  HIR
090  14  14
091  59  INT
092  69  OP
093  10  10
094  82  HIR
095  35  35
096  43  RCL
097  01  01
098  22  INV
099  67  EQ
100  01  01
101  26  26
102  82  HIR
103  15  15
104  42  STO
105  00  00
106  25  CLR
107  63  EX*
108  00  00
109  69  OP
110  30  30
111  63  EX*
112  00  00
113  69  OP
114  20  20
115  97  DSZ
116  00  00
117  01  01
118  09  09
119  01  1
120  00  0
121  82  HIR
122  38  38
123  01  1
124  82  HIR
125  55  55
126  01  1
127  82  HIR
128  36  36
129  82  HIR
130  13  13
131  75  -
132  82  HIR
133  16  16
134  95  =
135  77  GE
136  00  00
137  25  25
138  82  HIR
139  15  15
140  42  STO
141  00  00
142  73  RC*
143  00  00
144  99  PRT
145  97  DSZ
146  00  00
147  01  01
148  42  42
149  82  HIR
150  18  18
151  99  PRT
152  98  ADV
153  05  5
154  85  +
155  82  HIR
156  13  13
157  95  =
158  61  GTO
159  12  B
160  00  0
161  00  0
```

après l'exécution, les registres de résultats.

Le programme utilise largement les possibilités de l'instruction HIR. Les registres internes accessibles grâce à cette instruction sont notés R_1^h, R_2^h, etc... Dans le cas présent, on utilise les registres R_3^h à R_8^h.

Rappel : Pour un registre interne donné, R_3^h par exemple, HIR 03 a le même rôle que STO 03 si l'on considère R_3^h comme R 03. De même, HIR 13 correspond à SUM 03 (voir *l'Op* n° 1, pages 60 à 62). Ces trois possibilités sont les seules utilisées dans le programme.

Les registres sont répartis en deux groupes. Pour les registres de données, on a :

R 00 : i (pointeur)
R 01 : Registres contenant n! par
R 02 tranches de 10 chiffres (jus-
R 99 qu'à R 39 sur TI 58).

Le dernier registre utilisé contient z, nombre de zéros à ajouter au résultat final.

Quant aux registres internes, ce sont :

R_3^h n (constant)
R_4^h Stockage de retenues en cours de calcul
R_5^h D " degré de remplissage des registres de données "
R_6^h J (Algorithme de calcul)
R_7^h 10^{10} (constant)
R_8^h Z (Nombre de zéros à ajouter au résultat).

On remarque que le programme, bien que relativement court (160 lignes), est cependant très touffu. Cela mérite quelques explications.

Ainsi, sur l'organigramme, on a utilisé des notations qui ne sont peut-être pas familières à tout un chacun : ε (x) est la partie entière de x, obtenue avec l'instruction Int. De même, ε (x) est la partie fractionnaire de x, obtenue avec la séquence INV Int. X est le registre d'affichage. X⇌Ri est l'échange du contenu de l'affichage avec celui du registre Ri, obtenu avec l'instruction Exc.

Pour terminer, et à titre de vérification du bon fonctionnement du programme, voyez ci-contre ce que donne le calcul de 300 !

Mettez tout cela bout à bout (n'oubliez pas les 70 zéros à la fin) et diviser le résultat successivement par 300, 299,... etc, jusqu'à 2.

300.	5980776521.
30605.	2708224376.
7512216440.	3944912012.
6360353704.	8678675368.
6129726862.	3057122936.
9388588804.	8194364995.
1735769994.	6460498166.
1677674125.	4502277165.
9476533176.	18517654.
7168674655.	6469340112.
1529142247.	2260347297.
7573349939.	2406633325.
1478887017.	8583506870.
2636886426.	1501697941.
3907759003.	6885035375.
1542268429.	2137554910.
2790697455.	2891264071.
9841225476.	5715483028.
9302719546.	2284937952.
400801221.	6365801452.
5776252176.	3523315693.
8542559653.	6482233436.
5690350678.	7992545940.
8725264321.	9527682060.
8962642993.	8062232812.
6520457644.	3873838808.
8830388909.	1704960000.
7539434896.	70.
2543605322.	

Si, si, vous verrez, ça tombe toujours juste.

☐ Bernard Kokanosky

Bogue corrigée est à moitié pardonnée

■ A la page 69 de *l'Op* n° 5, dans la liste de « la petite montre pour Casio 702 P », la ligne 130 était erronée. Les lettres majuscules M et H se ressemblent beaucoup sur l'imprimante du 702, et nous les avons confondues à trois reprises dans cette fatidique ligne 130 (13 × 10) dont voici la bonne version : 130 M = O : H = H + 1 : IF H>23 THEN 150

Voilà qui devrait nous permettre de ne pas perdre trop de temps.

D'autre part, dans le programme de calcul de factorielles (page 58 de *l'Op* 5), le registre interne n° 8 (HIR 8) n'est jamais remis à zéro dans le cas de calculs successifs. Le remède est simple : il suffit d'insérer HIR 58 après l'instruction HIR 18 des pas 149 et 150.

l'Op