



Alpha-idées pour PC 100

Le codage en alphanumérique sur PC 100 est loin d'être économique si l'on suit mot à mot le manuel d'utilisation.

Une programmastuce parue dans L'OI n° 21 vous donnait une méthode assez empirique pour générer des nombres destinés aux registres d'impression, ceci permettant d'obtenir des mises en page agréables et, pourquoi pas, décoratives.

Passons de l'empirique au rationnel. Nous vous proposons tout d'abord un programme qui cherchera pendant vos loisirs (ceci pour dire que l'exécution peut être assez lente) les nombres p et q s'ils existent, tels que la partie fractionnaire de p/q vous donne de l'alphanumérique codé selon vos désirs.

Plus clairement, imaginons que vous ayez besoin d'une ligne de X. La méthode

traditionnelle est de faire 6666666666; OP1; OP2; OP3; OP4; OP5.

Mais vous savez que $2/3 = 0,666...$ Vous pouvez donc obtenir le même résultat en faisant:

2; 3EE 10 +/- = INV EE OP1; OP2; OP3; OP4; OP5.

Ce qui n'est bien sûr pas plus économique si vous ne comptez utiliser que des X; mais si vous décidez de varier les « motifs décoratifs », le sous-programme suivant décodera les nombres p et q que vous aurez mis auparavant dans T et dans X.

000	76	LBL	014	22	INV
001	11	A	015	52	EE
002	32	X:T	016	69	DP
003	55	=	017	01	01
004	32	X:T	018	69	DP
005	05	=	019	02	02
006	22	INV	020	69	DP
007	59	INT	021	03	03
008	65	X	022	69	DP
009	01	1	023	04	04
010	52	EE	024	69	DP
011	01	1	025	05	05
012	00	0	026	92	RTN
013	95	=			

Utilisation: 2; x = t; 3; A donne XXXX...

Congruences modulo n : Kekséksa ? A quoi ça sert ?

Point trop ici de mathématiques pour expliquer les conséquences, mais sachez seulement que tout commence avec une relation d'équivalence qui nous permet d'écrire (dans le cas de modulo 4):

$5 = 1$; $6 = 2$; $7 = 3$; $8 = 0$, $9 = 1$ etc.

Le but recherché :

A travers un exemple simple : nous avons 4 joueurs : 0, 1, 2, 3 et chacun a la main à tour de rôle. On veut donc un compteur qui, chaque fois qu'il est incrémenté « passe la main ». Le cycle est donc :

0, 1, 2, 3, 0, 1, 2... etc. C'est en fait un compteur qui tourne sur lui-même.

Les applications :

Elles sont multiples : simulation d'une aiguille d'horloge (arrivé à 60 on repart à 0); comptage en base n , $n = 10$; remplissage à tour de rôle des registres Op (Op 1, Op2, Op3, Op4), etc. Ce que l'on peut dire, c'est que tout programmeur... amateur en a besoin de manière fréquente.

Le sous-programme suivant, chaque fois qu'il est appelé incrémente la mémoire (01) modulo H. (On peut bien sûr changer le modulo (ligne 4) à sa guise). On obtient ainsi : 0, 1, 2, 3, 0, 1, 2, 3... etc.

1; x = t; 3; A donne PPPP...

Ce sous-programme pourra être utile si l'on sait avec quoi le « nourrir ». Voici un programme qui recherchera, pour x donné, p et q tels que $p/q \cdot 10^{10} = x \pm 1$ près.

Pour les intéressés, l'algorithme de recherche utilise le raisonnement suivant :

soit $y = x \cdot 10^{-10}$. $\exists ? p$ et q ($y = p/q + \epsilon$ et $|\epsilon| < 10^{-10}$). Si oui, alors $qy = p + q \cdot \epsilon$ et $|\epsilon| < q \cdot 10^{-10}$

D'où une recherche pour $q = 1, 2, 3, \dots$ jusqu'à trouver p . Bien sûr p et q existent toujours car $y = x \cdot 10^{-10} = x/10^{10}$

On peut donc prendre $p = x$ et $q = 10^{10}$, on espère seulement que le programme trouvera q assez rapidement car si l'on devait attendre 10^{10} boucles !...

Mode d'emploi

Introduire x (par exemple 3333333333).

Presser sur A. En fin d'exécution RCL 05 donne p ; RCL 02 donne q . Ce programme peut être utilisé en tant que sous-programme : on pourra ainsi lui donner une liste de nombres à « digérer » pendant la nuit et récupérer les résultats au matin (il faut amortir le matériel).

A priori, rien ne s'oppose à ce que x soit quelconque mais si vous introduisez le premier mot venu codé en alpha, vous risquez de devoir attendre longtemps (pour mémoire : 10^{10} boucles, c'est long !).

Cela ne coûte rien d'essayer surtout si vous faites de beaux rêves pendant que votre préférée travaille pour vous.

Revenons au domaine de la décoration. On a besoin

liste du programme :

000	76	LBL	018	69	DP	036	00	00
001	11	A	019	22	22	037	52	EE
002	65	X	020	43	RCL	038	22	INV
003	01	1	021	02	02	039	52	EE
004	52	EE	022	65	X	040	22	INV
005	01	1	023	43	RCL	041	58	FIX
006	00	0	024	04	04	042	42	STD
007	94	+/-	025	95	=	043	05	05
008	42	STD	026	32	X:T	044	75	-
009	04	04	027	43	RCL	045	43	RCL
010	95	=	028	02	02	046	03	03
011	22	INV	029	65	X	047	95	=
012	52	EE	030	43	RCL	048	50	I×I
013	42	STD	031	01	01	049	77	GE
014	01	01	032	95	=	050	00	00
015	01	1	033	42	STD	051	18	18
016	42	STD	034	03	03	052	92	RTN
017	02	02	035	58	FIX			

dans ce cas de lignes comportant le même caractère ou la même suite de caractères répétée plusieurs fois. C'est là que le programme excelle car un nombre comportant une suite répétitive assez courte (2 ou 3 chiffres) donne en général p et q assez rapidement (c'est mathématique mais ça se prouve).

En approfondissant dans cette direction, et en s'aidant des mathématiques, on s'aperçoit que pour obtenir un nombre à deux chiffres répétés plusieurs fois (c'est en fait une suite infinie), il suffit de diviser ce nombre par 99 (pourvu qu'il soit inférieur à 99).

Par exemple $42/99 = 0,424242...$

D'où le programme suivant :

000	76	LBL
001	11	A
002	69	DP
003	21	21
004	04	4
005	75	-
006	43	RCL
007	01	01
008	95	=
009	69	DP
010	10	10
011	49	PRD
012	01	01
013	92	RTN

Pour obtenir 1, 2, 3, 4, 1, 2, 3, 4, etc., il suffit de placer la séquence Op21 (lignes 2 et 3) juste avant INV/SBR.

Antoine Jennet

000	76	LBL	008	94	+/-	016	69	DP
001	11	A	009	95	=	017	03	03
002	55	÷	010	22	INV	018	69	DP
003	09	9	011	52	EE	019	04	04
004	09	9	012	69	DP	020	69	DP
005	52	EE	013	01	01	021	05	05
006	01	1	014	69	DP	022	92	RTN
007	00	0	015	02	02	"	"	"

Mode d'emploi :

Introduire un code alpha (2 chiffres). Presser sur A. Le résultat est une ligne de 20 caractères du signe alphanumérique considéré.

Un peu plus loin dans la même voie :

42/999 = 0,042042...

42/9999 = 0,00420042...

En continuant ainsi, on augmente le nombre de zéros entre chaque groupe de chiffres. L'application est immédiate : tracer un tableau sans être obligé de bloquer des registres avec de l'alphanumérique. (N'ou-

blions pas qu'une mémoire de donnée équivaut à 8 pas de programme).

Pour étendre les possibilités qu'on devine dans ce programme, il suffit de le multiplier en l'adaptant selon l'effet désiré. On peut ainsi tracer des cases pour un tableau, en choisissant le nombre de cases que l'on souhaite. On obtient alors un programme à 2 entrées : l'une pour le traçage des lignes, l'autre pour les cases.

N'oubliez pas que les blancs alphanumériques correspondent à des zéros

(un blanc correspond à 2 zéros). Pour remplir le tableau, il suffit donc d'additionner les codes alpha correspondant aux cases à ceux correspondant à ce que vous voulez mettre dedans.

J'espère que ces quelques idées pourront vous aider dans vos alpha-recherches. Elles vous permettront déjà de réaliser des mises en pages originales sinon « professionnelles », et surtout pas trop coûteuses.

Antoine Jennet