

**JPC**MARS 1987  
NUMERO 42

Le numéro 35 FF.

**A PROPOS DU CLUB**

P. David	Editorial	1
J.J. Dhénin	L'Assemblée Générale	2
	PPC Paris se réunit	3
	Ah ! Vous écrivez !	3
	Courrier du coeur	4
	Courrier des lecteurs	5
L. Ancelet	Choc en retour	5
P. David	Nouveaux produits	6

**HP41**

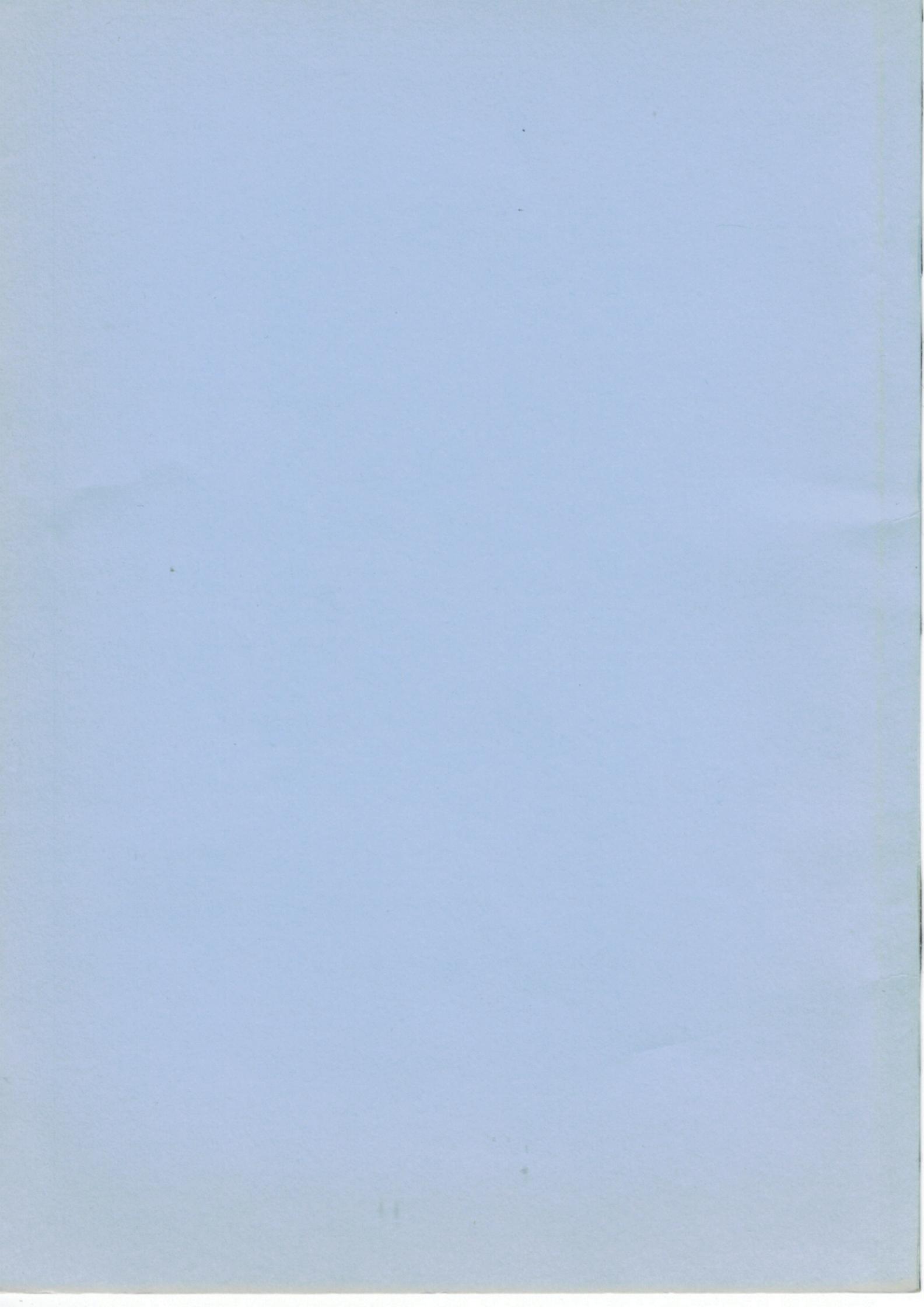
G. Tisserand	Profil de voies urbaines	10
--------------	--------------------------	----

**HP75**

E. Gengoux	Compatibilité des Basic HP71/HP75	12
------------	-----------------------------------	----

**HP71**

X. Bille	Tracer toutes les fonctions	16
S. Vaudenay	Jeu de la vie	19
G. Toublanc	Réfractaire	21
G. Toublanc	FracStory	27
A. Gillet	Débuter en boucles et tests	28
A. Herreman	Agenda	28
E. Gengoux	Dessine-moi un autre mouton	29
	Le coin des Lhex	39



EDITORIAL

Chers amis,

Ce journal est presque fini au moment où j'apprends une triste nouvelle : Richard Nelson, fondateur du mouvement PPC aux Etats-Unis, a décidé de fermer CHHU, son nouveau club. Il ne reste donc plus aucun club valable à vocation internationale dédié aux calculateurs de poche HP.

Cette cessation d'activité est due à des raisons économiques : CHHU ne regroupait pas assez de membres pour survivre. Cela faisait longtemps que plus aucun Journal ne nous parvenait des Etats-Unis.

Les nouvelles de France sont, au contraire, plutôt bonnes. Bien que le nombre définitif de commandes de "modules JPC" ne soit pas encore connu, les résultats provisoires sont très prometteurs.

Ce Journal est le second bénéficiaire de la nouvelle présentation. Nous souhaitons que ces possibilités contribuent à une plus grande lisibilité. N'hésitez pas à nous faire part de vos remarques, suggestions ou critiques sur ce sujet.

Pierre David (37)

## L'ASSEMBLEE GENERALE

Comme il se doit, notre Assemblée Générale Annuelle a eu lieu en Décembre 1986. Annoncée depuis longtemps, elle a réuni tous ceux qui ont pu se libérer, qui n'étaient pas trop éloignés de la capitale et qui ont la fibre associative. Douze pour être précis. A ceux-là s'ajoute un nombre appréciable de mandats qui ont assuré la légitimité de notre réunion.

Notre vénéré Président et néanmoins Trésorier a ouvert la réunion par les paroles de satisfaction et d'encouragement qui conviennent à la bienséance et à l'occasion. Puis il s'est efforcé de nous faire un rapport financier.

Nous avons été sensibles à son optimisme relatif à l'avenir de notre Association. Parmi les faits principaux, soulignons quelques points essentiels :

- Le dernier trimestre est traditionnellement riche en réabonnements. En effet, les préoccupations liées aux congés et à la rentrée universitaire s'estompent, et chacun se frappe le front et la poitrine en s'exclamant : "Bon sang, mais c'est bien sûr ! L'heure est venue de payer ma cotisation à PPC-Paris". Le président nous a annoncé pas moins de vingt réabonnements pour les jours prochains.

- Il y a maintenant deux ans, nous avons entrepris une action commune avec Hewlett-Packard afin de mettre dans chaque emballage de matériel une carte d'information sur le club PPC. Nous ne pouvons que nous féliciter de voir notre attente enfin récompensée. Les premières inscriptions sont arrivées récemment par ce canal.

- Plus de trois cents numéros d'adhérent ont été attribués et nous avons été plus de deux cents membres actifs au même moment au cours de l'année écoulée.

- Outre les abonnements, le Journal se vend bien chez quelques distributeurs, ainsi la Règle à Calcul vend bon an mal an entre vingt et trente exemplaires par mois.

- Cette estimation de la situation a permis à notre Président d'investir dans l'achat d'un lecteur de cassettes et d'un lecteur de disquettes.

- Pour respecter les termes de la loi, le Trésorier Président va rédiger un compte-rendu écrit de l'exercice passé afin que nous puissions le transmettre à la Préfecture.

Pierre David, en sa qualité de Président Adjoint nous a ensuite dressé le rapport moral.

Cette année est caractérisée par une nette amélioration de nos relations avec HP. Nous pouvons voir là se concrétiser un long travail pour que la qualité de notre production au sein du Club n'ait rien à envier à celle des professionnels :

- D'une part la qualité de la présentation de notre Journal, entièrement confectionné sur HP-71 et LaserJet, est la preuve tangible des capacités de cet ordinateur portable. N'oubliez pas que le temps de confection est passé de cent cinquante heures à quarante, grâce à la création d'un ensemble de programmes mis au point essentiellement par Pierre David et Janick Taillandier, mais aussi parce que vos articles nous parviennent maintenant sur support magnétique.

- D'autre part, la qualité du contenu (nous sommes actuellement le Club qui a produit le plus d'applications de qualité sur HP-71 à ce jour) est l'expression d'un esprit associatif appréciable où chacun bénéficie du travail de tous. De ce point de vue, la création du module JPC est un événement qui a un retentissement international dans le petit monde des clubs utilisateurs HP. Dans la même veine, nous allons bientôt proposer la diffusion sur support magnétique des programmes parus. Actuellement plus de sept cents kilo octets de programmes ont été archivés et un catalogue est pratiquement terminé.

Cependant, nous sommes invités à renforcer l'activité du Club au cours de l'année 1987. En effet, les difficultés des Clubs au USA et à Toulouse doivent nous inciter à modérer notre confiance. Notre existence est fondée sur un faisceau de petites actions individuelles. Que chacun d'entre nous essaie de faire un adhérent dans l'année et / ou écrive un article (surtout pour les débutants) et nous améliorerons encore notre apport à tous.

Nous souhaitons également que chacun des membres de province puisse prendre contact avec son distributeur local et négocie un dépôt vente du journal dans son magasin.

Nous avons, bien entendu procédé à l'élection d'un nouveau Bureau. Sont élus :

Président : Philippe Guez  
Président Adjoint : Pierre David  
Trésorier : Janick Taillandier  
Secrétaire : Jean-Jacques Dhénin  
Secrétaire Adjoint : Olivier Arbey

Le nouveau Bureau entre en fonctions à compter du 1<sup>er</sup> Janvier 1987.

Nous avons ensuite débattu de différents projets, tels que la confection d'un numéro 0 de JPC (ce qui ne veut pas dire nul) dont le rôle serait de satisfaire par un exemplaire ad hoc les demandes d'informations qui nous parviennent. Nous avons également évoqué la possibilité de réaliser un ouvrage collectif autour du HP-71 : une sorte d'encyclopédie ou de Vademecum. Toutes les idées ou les propositions seront les bienvenues.

Pour le Secrétariat,  
Jean-Jacques Dhénin (177)

---

## PPC PARIS SE REUNIT UNE FOIS PAR MOIS

Comme vous le savez peut être déjà, PPC Paris se réunit une fois par mois, en plein coeur de Paris. Amenez votre matériel, votre bonne volonté et vos idées ! Plus vous en apporterez, et plus vous en trouverez chez vos collègues de PPC.

Ces réunions se déroulent de manière très libre, aucun ordre du jour, discussion ou autre n'étant imposé. Un membre du bureau est toujours présent. Ainsi, si vous désirez remettre votre article tout frais au Journal, si vous avez des suggestions à faire, si vous voulez vous procurer des anciens numéros de JPC, ce sera en principe toujours possible.

Si donc cela vous intéresse, n'hésitez plus un seul instant, venez nous rejoindre tous les premiers samedis de chaque mois (sauf en période de vacances scolaires) au :

Centre de Jeunesse et de Loisirs Jean Verdier  
11 rue de Lancry  
75010 Paris

et en montant au deuxième étage, vous entendrez des éclats de rire et des discussions passionnées vers la salle 215. Attention, toutefois, de venir entre 16 et 19h.

Pour l'accès en métro, trois possibilités s'offrent à vous :

- Métro Strasbourg Saint Denis :  
Sortie porte St Martin / Bd St Denis, coté pairs  
- Métro République :  
Sortie Bd St Martin, coté pairs  
- Métro Jacques Bonsergent :  
Sortie Bd Magenta, coté impairs.

Ah, j'oubliais ! JPC est (souvent) distribué en avant première lors de ces réunions... A bon entendeur, salut !

Les dates des prochaines réunions sont :

Samedi 7 mars 1987  
Samedi 4 avril 1987  
Samedi 16 mai 1987  
Samedi 6 juin 1987

Pierre David (37)

---

## AH ! VOUS ECRIVEZ

Vous vous sentez en verve, mais vous ne savez pas sous quelle forme "l'équipe de rédaction" souhaite recevoir votre prose. C'est ici que se trouvent les réponses à vos questions.

Dans la mesure du possible, vous devez nous envoyer vos écrits sur support magnétique (carte, cassette ou disquette). Soyez sans crainte, nous vous retournerons vos biens après copie.

## COURRIER DU COEUR

Si vous ne pouvez pas utiliser de support magnétique, ou ne pouvez vous rendre aux réunions, alors et alors seulement faites le sur papier.

Que ce soit sur une feuille de papier, ou sur support magnétique, ne dépassez pas 50 caractères par ligne.

Pour nous épargner du travail, insérez dans votre texte les commandes de formatage suivantes (et non les commandes du formateur HP) :

"^" centre un titre, par exemple :  
^TITRE

"\" (CHR\$(92)) marque le début et la fin d'un paragraphe. Par exemple :

\Début de paragraphe exprimant le contenu de vos idées qui, même si vous en doutez, intéressera certains des membres du Club. Surtout si vous vous sentez débutant. Les articles pour débutants écrits par des débutants sont ceux qui manquent le plus. Fin de paragraphe.\

N'oubliez pas de mettre les accents. Utilisez le jeu de caractères Roman8. Les possesseurs de HP71 utiliseront les redéfinitions de touches ci-dessous, ainsi que le fichier CHARLEX listé dans le coin des Lhex en fin de journal.

Jean-Jacques Dhénin (177)

```
DEF KEY 'fW', CHR$(197);    (é)
DEF KEY 'fE', CHR$(193);    (ê)
DEF KEY 'fR', CHR$(201);    (è)
DEF KEY 'fY', CHR$(203);    (ù)
DEF KEY 'fU', CHR$(195);    (û)
DEF KEY 'fI', CHR$(209);    (ï)
DEF KEY 'fo', CHR$(194);    (ô)
DEF KEY 'f/', CHR$(92);     (\)
DEF KEY 'fA', CHR$(192);    (â)
DEF KEY 'fS', CHR$(200);    (à)
DEF KEY 'fD', CHR$(205);    (è)
DEF KEY 'fJ', CHR$(207);    (ù)
DEF KEY 'fK', CHR$(221);    (ï)
DEF KEY 'f*', CHR$(124);    (|)
DEF KEY 'fC', CHR$(181);    (ç)
```

PPC Paris  
56 rue J.J. Rousseau  
75001 Paris

Vend : (Prix Toutes Taxes)  
Interfaces Vidéo HP82163B neuves (dans leur boîtes, avec documentation). 600 F seulement.

Des lots de 100 cartes magnétiques pour HP-41, HP-67, HP-97 et même HP-65. 100 F seulement.

Des lecteurs de cartes magnétiques HP82400A pour HP-71. 500 F pièce.

---  
France Bilan Informatique  
10 rue Cambon  
75001 Paris

Vend : (Prix Hors Taxes)  
HP71B : 2500 F, lecteurs de disques  
HP9114A : 3500 F, ThinkJet : 2200 F,  
modules 4 Ko Ram : 350 F, module HPIL  
pour HP71 : 560 F, interface HPIL-HPIB :  
1900 F, module Forth / Assembleur pour  
HP71 : 850, module finance : 400 F.

---  
Alain Gillet  
Tél: 43-67-54-44 (après 18h)

Vend :  
Imprimante 82161A, plusieurs rouleaux de papiers : 2000 F, lecteur de code-barre : 500 F, livres : "Programmation synthétique" (Dr. Wickes) : 60 F, "La programmation synthétique c'est facile" (K. Jarret) : 60 F, "Time module solution 1" (HP) : 60 F, "Au fond de la 41" (Dodin) : 60 F.

---  
Jean-Jacques Dhénin  
35, rue Boileau  
92120 Montrouge  
Tel : 43-37-63-50 (bureau)

Vend :  
HP-75 + littérature abondante 3000 F

## COURRIER DES LECTEURS

Thierry Besançon  
9, Rue Champ Chapelle  
77120 Coulommiers

Connaissez-vous la fonction **MEMORY** ? Non ! Alors faites sans tarder **MEMORY** [ENDLINE]. Sur mon HP-71B, cela me renvoie 1. Qu'en dire à part cela ? Et bien, fait étrange, si l'on utilise le programme d'analyse de Lex SYSTCAT (mille excuses pour Jean-Pierre Bondu mais c'était le premier sur le marché et de plus il n'utilise pas l'HP-IL) ainsi que le **PEEK\$** étendu (**PKLEX** ou **DESAL**), on ne trouve aucune trace de **MEMORY** dans les Roms dont les ID sont, sauf erreur de ma part, 00 et 01.

Pourtant, en utilisant **SCANLEX** voilà ce que l'on trouve :

```
TYPE ('MEMORY') --> 0  
ENTRY$ ('MEMORY') --> 0BF92
```

Bizarre, n'est-ce pas ? Pourquoi diable **MEMORY** n'est-il pas groupé avec les autres mot-clefs ? Que fait-il exactement ?

Je n'ai pas pu en savoir plus. Alors je fais appel à tous les fouineurs et surtout aux heureux propriétaires des IDS qui, je pense, pourront y trouver de plus amples renseignements sur cet étrange mot-clef.

Thierry Besançon (292)

### Explication du Dr JPC :

L'explication est très simple. **MEMORY** est compris par le HP-71B comme :

```
MEM OR Y
```

Ce qu'on peut aisément vérifier en tapant dans un programme Basic :

```
10 DISP MEMORY
```

Cette fameuse fonction sera visualisée comme :

```
10 DISP MEM OR Y
```

Comme la mémoire disponible est rarement nulle, la "fonction" **MEMORY** renverra 1. Rien de bien extraordinaire... Le HP-71 ne possède pas les instructions cachées qui ont fait les beaux jours de la HP-41 !

Pierre David (37)

---

## CHOC EN RETOUR INTERFACE MINITEL - HPIL

Ceux d'entre vous qui envisageraient de construire l'interface Minitel/HPIL auront peut-être trouvé le schéma "Module UART" du JPC 34 page 7 un peu confus.

Cette confusion provient de ce que le plan publié correspond à une version "prototype" de cette interface. Voici donc quelques précisions.

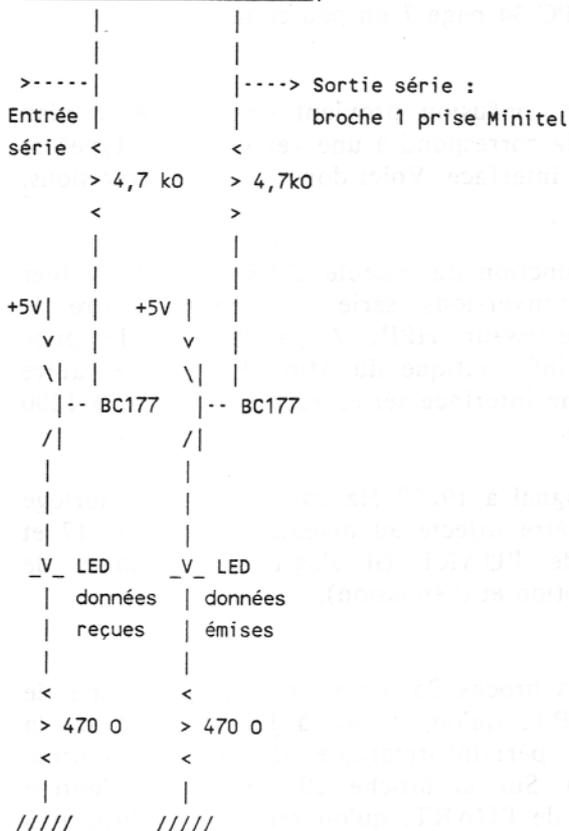
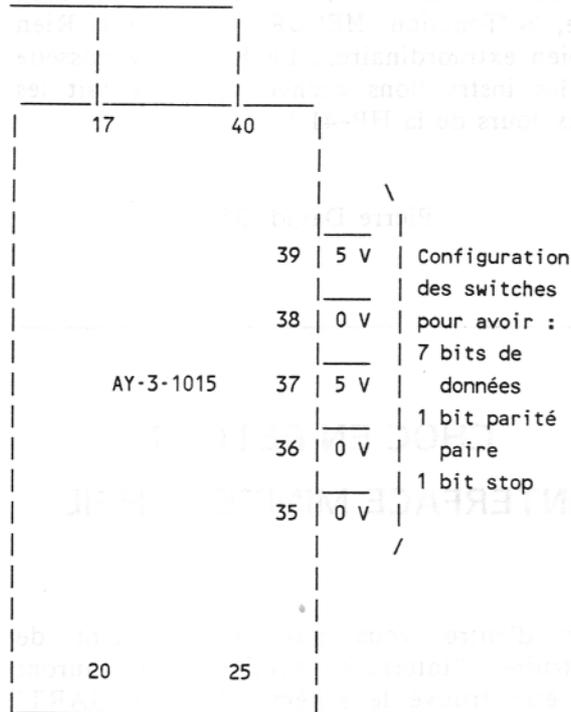
La fonction du module UART est d'effectuer les conversions série / parallèle entre le convertisseur HPIL / parallèle et la prise péri-informatique du Minitel, qui n'est autre qu'une interface série, au niveau TTL, à 1200 bauds.

Le signal à 19200 Hz issu du module horloge doit être injecté au niveau des broches 17 et 40 de l'UART (il s'agit des horloges de réception et d'émission).

Sur la broche 25, on trouve la sortie série de l'UART, qu'on reliera à la broche 1 de la prise péri-informatique du Minitel (entrée série). Sur la broche 20, on trouve l'entrée série de l'UART, qu'on reliera à la broche 3 de la prise péri-informatique du Minitel (sortie série).

La broche 4 de la prise péri-informatique sert à indiquer au Minitel la présence d'un périphérique quand elle est au niveau 0 volt. On pourra donc la relier directement à la masse, c'est-à-dire à la broche 2 de la prise péri-informatique.

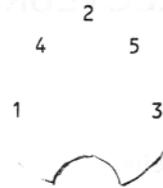
Horloge 1200 bauds



masse (broches 2 et 4 prise Minitel)

/////

Prise Minitel :



- Broche 1 : Périphérique --> Minitel.
- Broche 2 : Masse.
- Broche 3 : Minitel --> Périphérique.
- Broche 4 : Périphérique en transmission.
- Broche 5 : Minitel sous tension.

Les broches 4 et 5 sont actives à 0 V. Sur les futurs Minitel, la broche 5 sera une source de +5 V pour les périphériques à faible consommation.

Sur le schéma publié dans le JPC 34, les LEDs vertes et jaunes s'éteignaient au passage des données, alors qu'il eût été plus logique qu'elles s'allumassent à leur passage. Il suffit pour cela de remplacer les transistors NPN par des PNP, et bien entendu de les monter dans le bon sens !

Enfin, les 5 interrupteurs reliés aux broches 35 à 39 de l'UART permettent de configurer la liaison série dans le mode 7 bits de données, 1 bit de parité paire, 1 bit de stop.

Lionel Ancelet

## NOUVEAUX PRODUITS

La firme américaine HHP (Hand Held Products) a annoncé ces jours-ci des nouveaux modules mémoire pour le HP-71. Ces modules, enfichables à la place du lecteur de cartes, ont une capacité beaucoup plus importante que les précédents modules. La gamme complète est donc maintenant :

- 160 Ko Ram (\$750)
- 128 Ko Ram avec batterie de sauvegarde (\$625)
- 64 Ko Ram avec batterie de sauvegarde (\$295)
- 32 Ko Ram avec batterie de sauvegarde (\$165)

32 Ko Eprom (\$115)

32 Ko Eprom + 32 Ko Ram (\$275)

32 Ko Eprom + 64 Ko Ram (\$375)

Notons que le module 96 Ko Ram n'est plus au catalogue, et que les prix (du moins aux Etats-Unis) sont devenus presque raisonnables.

Annoncé également un nouvel interface HP-IL / RS232. Compatible avec l'interface HP (HP-82164A), il dispose de buffers plus grands (8 Ko) et est alimenté par des batteries. Le prix : \$449.

Ne pas confondre avec l'interface présentée dans JPC 40, fabriquée par "Firmware Specialists" (Réf : FSI164A).

La date de disponibilité reste à déterminer.

Pierre David (37)



Hand Held Products is recognized worldwide for its lines of intelligent portable systems, memory modules and related products used in conjunction with Hewlett-Packard hand held computers.

**HAND HELD™**  
**PRODUCTS**

### Description:

Hand Held Products HP-IL/RS-232 Battery-Powered Interface gives you portable power for up to 30 hours of continuous interaction (15 hours minimum) between Hewlett-Packard Interface Loop and RS-232 devices. The Interface enables a number of HP computers (HP-41, HP-71, HP-110 and others) to communicate with other computers, terminals, printers, or modems without AC power in even the most remote locations.

Compact and lightweight, this serial communications interface is physically identical to the HP 82164A. Functionally compatible as well, it will respond to the entire 82164A remote mode instruction set, with local and remote operating states for complete flexibility in configuring the interface.

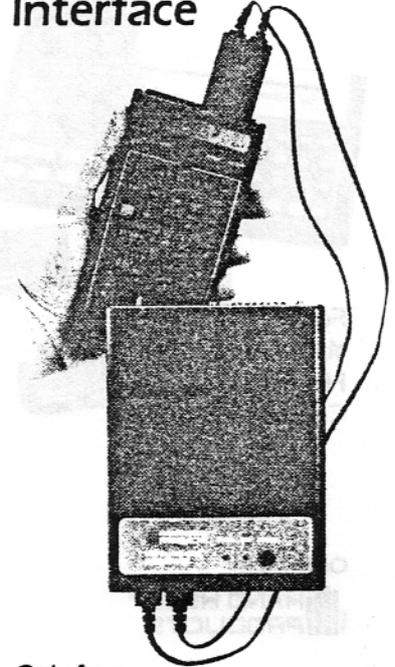
In addition to standard RS-232 and HP-IL features, the interface has extended transmit and receive buffers with 8K byte capacities. Special function keys permit direct user interaction with the interface and the setting of certain states through manual means. Control logic in the HP-IL/RS-232 Interface stores operating information, implements user-selectable operating modes, and controls the flow and interpretation of data within the interface.

### Features:

- Compact, Lightweight and Completely Portable
- Rechargeable 500 mA Ni-Cad Battery
- Programmable Battery Conservation Shutdown Timer
- Battery-Low Warning Indicator
- Hardware Protected from Over-Discharge
- Uses Standard HP Wall Transformer to Recharge
- Extended Transmit and Receive Buffers (8K Byte Capacity)
- EIA Standard Bipolar RS-232 Signals, Jumper-Selectable Option for TTL Level

# NEW!

## HP-IL/RS-232 Battery Powered Interface



Only from

**HAND HELD™**  
**PRODUCTS**

### Specifications:

#### Power Requirements:

- Power Source . . . One Ni-Cad Battery, Rechargeable
- Operating Current . . . 500 mA, maximum (350 mA typical)

#### Environmental Limits

- Operating Temperature Range . . . 0° to 65° C (32° to 131° F)
- Storage Temperature Range . . . -40° to 75° C (-40° to 167° F)

#### Physical Dimensions

- Length x Width x Height . . . 6.3 x 4.7 x 1.1 Inches
- Weight . . . 15 Ounces

#### RS-232 Features:

- Asynchronous Data Communications
- Full-Duplex and Half-Duplex Operation
- Standard Hardware Handshakes
- Programmable Control of Non-Standard Handshakes
- Programmable Software Handshakes (ENQ, ACK, XON/XOFF)
- Selectable Baud Rates (50 to 19,200)
- Selectable Word Lengths (5, 6, 7, or 8 bits)
- Parity Checks (none, even, odd, always 1, always 0)
- Selectable Stop Bits (1 or 2)
- Echo Capability
- Jumper Block Configurable as DTE or DCE
- User-Definable Protocols
- Compatible with Direct-Connect Modems & Acoustic Couplers

#### HP-IL Features:

- Talker and Listener Capabilities
- Detailed Status Information
- Programmable Asynchronous Service Requests (including programmable service request options)
- Listener Not Ready for Data
- Simple Auto-Addressing

# NEW!

## High Capacity Memory Modules



For Your HP71 Hand Held Computer

Only from  
**HAND HELD™ PRODUCTS**

## More Memory. . . . Up to 160K Bytes!

Unleash the power of your HP71 with Hand Held Products' new 71 M/M Series high capacity extended Memory Modules. Choose from seven available configurations offering a variety of features to meet your specific needs. Enjoy the flexibility of EPROM—ideal for developing your own programs—Or multiply your HP71's data storage in RAM to more than ten times its present capacity. With battery back-up and card-reader port installation, the 71 M/M Series offers features you can't get in memory modules from any other supplier.

### Your Choice of . . .

- RAM: 160K Bytes  
128K Bytes with Battery Back-up  
64K Bytes with Battery Back-up  
32K Bytes with Battery Back-up
- EPROM: 32K Bytes
- EPROM/RAM: 32K/32K or 32K/64K Bytes

### Features

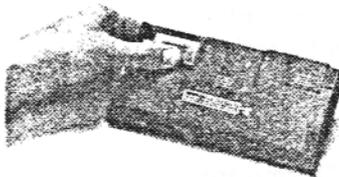
- Highest Capacity RAM in a Single Module
- Battery Back-up in Most RAM Modules Retains Data in Case of HP71 Power Loss
- Self-Contained Modules Reside in Card Reader Port Leaving Valuable Front Ports Free for Application Modules
- Easy Plug-in Installation, No Tools Required
- RAM May Be Used in Main Memory or Addressed as Independent RAM
- Battery Life Up to Five Years Under Normal Use



## Installing Your Memory Module

71 M/M Series extended Memory Modules are installed in the card-reader port of your HP71. Installation is simple, requires no tools and can be accomplished in seconds. Here's all you do:

- 1) Turn off the HP71.
- 2) Turn the HP71 over and remove the cover to the card reader port.
- 3) Remove the plastic insert.
- 4) Install the Memory Module by aligning the gold pins in the HP71 with the sockets in the Modules. Press firmly on the back of the Module until it fits snugly in place.
- 5) Turn the HP71 right side up. Check to be certain the name plate on the Memory Module is flush with the front of the HP71.
- 6) Replace the cover to the card reader port.



**HAND HELD™ PRODUCTS**  
Post Office Box 2388  
Charlotte, NC 28211  
Phone (704) 541-1380  
Telex 810-621-0380

## Ordering Information

Model Number	Description	Price
71 M/M 160R	160K RAM	\$750
71 M/M 128R	128K RAM	625
71 M/M 64R	64K RAM	295
71 M/M 32R	32K RAM	165
71 M/M 32E	32K EPROM	115
71 M/M 32/64	32K EPROM 64K RAM	375
71 M/M 32/32	32K EPROM 32K RAM	275
HHP 71 M/M Battery Replacement Assembly		10

\*Volume Discounts Available  
\*Visa and MasterCard Accepted

## Memory Modules from the Leader

Hand Held Products is recognized worldwide for its lines of intelligent portable systems, memory modules and related products used in conjunction with Hewlett-Packard hand held computers.

## Using Your Extended RAM

Hand Held Products' 71 M/M Series extended RAM modules can be utilized in two ways:

### {1} Main Memory

When the module is in place, you have instantly added 160K, 128K, 64K or 32K bytes of memory to your HP71's main RAM.

### {2} Independent Memory

Your increased memory may also be partitioned or placed in independent RAM, residing in port 5 of the HP71. Memory is partitioned in 32K byte increments and addressed in these locations:

- 32K . . . Port 5.0
- 64K . . . Ports 5.0 and 5.01
- 128K . . . Ports 5.0, 5.01, 5.02, and 5.03
- 160K . . . Ports 5.0, 5.01, 5.02, 5.03, and 5.04

## RAM Battery Back-Up

The 32K, 64K and 128K RAM modules include a self-contained lithium battery. This feature allows you to safely remove the Memory Module from your HP71 and still retain its memory without fear of data loss.

The 71 M/M RAM's battery operates only when the module is removed from the HP71; there is no battery drain while it is installed in the HP71. When installed, all RAM modules are powered by the HP71's own power source. Should the power source fail, however, the battery back-up safeguards your data until power is resupplied.

## Battery Life

Each 71 M/M Series RAM module includes a snap-on cover which must be placed on the module immediately upon its removal from the HP71. The cover must remain on the module at all times when not installed in the HP71 to preserve battery life.

The module will retain its memory for up to five years with normal use (installed in your HP71 about 80% of the time). Expected life when removed from the HP71 is as follows: 32K, six years; 64K, three years; 128K, 18 months. Hand Held Products guarantees seven months minimum battery life for all products.

To replace the battery in your Memory Module, all that's required is a 71 M/M battery replacement assembly. This assembly, available from Hand Held



Products, comes complete with everything you need and installs in minutes.

## Using Your EPROM Carrier

If you wish to add EPROM capability to your HP71, three different Memory Modules are available:

- 32K EPROM Carrier
- 32K EPROM Carrier with 32K RAM
- 32K EPROM Carrier with 64K RAM

EPROM functions are identical in all three versions.

You may install up to 32K of EPROM in your 71M/M EPROM Carrier. The Carrier will accept the following: Intel compatible, CMOS, 27256, 200 nanosecond or faster.

### Addressing the Memory. . .

Memory in each EPROM Carrier version is partitioned in 32K increments and addressed in these locations:

- 32K EPROM . . . Port 5.0
- 32K EPROM/32K RAM . . . Ports 5.0 and 5.01
- 32K EPROM/64K RAM . . . Ports 5.0, 5.01, and 5.02

### Programming the Memory. . .

Hand Held Products offers SDS (System Development Software) for programming your 71 M/M EPROM. The EPBURN SDS may be purchased for a nominal charge if you wish to burn your own programs into EPROM.

Hand Held Products can also perform quick EPROM burns for you at our facilities. Call or write if you'd like additional information on this low cost service.

# PROFIL DE VOIES URBAINES

Voici les notations usuelles des termes employés et le numéro du registre les concernant.

(1) -> "niveau" pour le numéro de la classe du revêtement de la voie à mesurer ;

(2) revêtement : béton de ciment, bitume, etc.  
 (3) revêtement : granules de goudron, asphalté, etc.

(4) pavage en granit : moindres petites (0.10) ou grosses (0.14) ;

(5) chaussée pavée en échantillon ;

Ce choix place V en R01, V étant l'altitude d'un point appelé "niveau", pose à 1.00 mètre des bordures et formant le cariveau.

## Profil de voies urbaines

10

(6) -> "niveau" pour le niveau des hauteurs des fils d'eau, stocks en R03 ;

Les trois éléments de calcul ainsi entités permettent par l'axe d'axe D en R02, c'est le "démarrage" du point haut de la voie, appelé F en R04. Ce point de la façade permet de réaliser de part et d'autre deux plans profils normaux, en les traitant séparément comme tels.

Les deux plans de hauteur A à l'axe de la chaussée calculée, et W étant la cote du point équidistant à A et B.

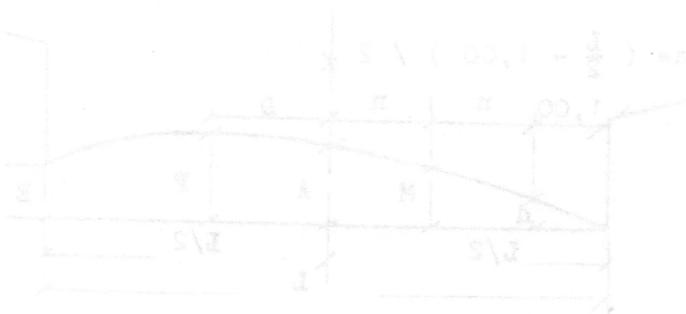
Si les cariveaux sont de même hauteur la cote de A sera également celle de B.

Ces calculs sont relatifs au calcul des classes, Conditions et Charges Générales imposées aux entrepreneurs des travaux publics de la Ville de Paris (Annexe III du C.C.G.).

Gilbert Tisserand (24)

## HP41

G. Tisserand



## PROFIL DE VOIES URBAINES

Voici les significations succinctes des termes employés et le numéro du registre les concernant.

[A] -> "CHOIX (1-3)?" entrer le numéro de la classe du revêtement désiré, pour voir le menu : [R/S] ;

[1] revêtements lisses : béton de ciment, bétons bitumineux, granulats de synthèse, asphalte coulé ou pavés d'asphalte ;

[2] pavage en granit : mosaïques petites (0.10) ou grosses (0.14) ;

[3] chaussée pavée en échantillon.

Ce choix place  $d$  en R01,  $d$  étant l'altitude d'un point appelé défense, posé à 1.00 mètre des bordures et formant le caniveau.

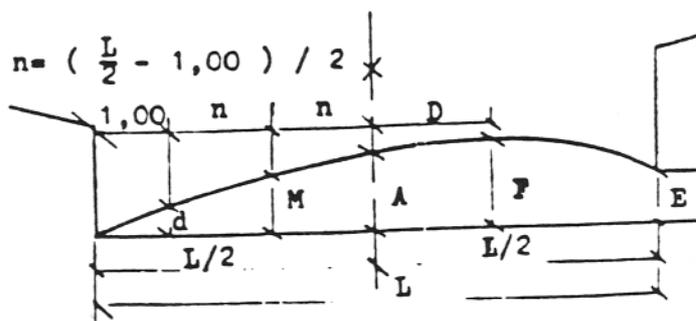
[R/S] -> "LARGEUR?" entrer la largeur de la chaussée qui sera stockée en R02.

[R/S] -> "EPAULEMENT?"  $E$  = différence des hauteurs des fils d'eau, stocké en R03.

Les trois éléments de calcul ainsi entrés permettent par [R/S] d'afficher  $D$  en R05, c'est le "désaxement" du point haut de la voie, appelé  $F$  en R04. Ce déport de la flèche permet de rétablir de part et d'autre deux demi profils normaux, en les traitant séparément comme tels.

[R/S] affiche la hauteur  $A$  à l'axe de la chaussée calculée, et  $M$  étant la cote du point équidistant à  $A$  et  $d$ .

Si les caniveaux sont de même hauteur la cote de  $A$  sera également celle de  $F$ .



Ces calculs sont admis au cahier des Clauses, Conditions et Charges Générales imposées aux entrepreneurs des travaux publics de la Ville de Paris (titre III du C.C.C.G.).

Gilbert Tisserand (54)

01\*LBL "PROFIL"

02\*LBL A

FIX 3

04\*LBL 00

CF 22 "CHOIX<1-3>?" TONE 9 PROMPT

FS?C 22 GTO IND X AON "1-ASPHALTE" PSE

"2-MOSAIQUE" PSE "3-PIERRE" PSE AOFF GTO 00

20\*LBL 01

5 E-2 GTO 04

23\*LBL 02

6 E-2 GTO 04

26\*LBL 03

72 E-3

28\*LBL 04

STO 01 TONE 9 "LARGEUR?" PROMPT STO 02

2 \* 5 - X<>Y 4 / \* STO 03 TONE 9

"EPAULEMENT?" PROMPT X>Y? GTO 05 STO 04

STO 03 "D=" RCL 01 / STO 05 ARCL 05 " F="

2 \* RCL 02 X<>Y - ENTER^ STO 07 X^2 X<>Y

E - / RCL 01 4 / \* ST+ 04 ARCL 04 PROMPT

RCL 02 2 / RCL 05 - X^2 RCL 02 RCL 05

2 \* - E - / RCL 05 X^2 RCL 05 2 \*

RCL 02 + E - / - RCL 01 \* RCL 03 +

"A=" ARCL X RCL 01 - 625 E-3 \* RCL 01 +

" M=" ARCL X PROMPT GTO A

116\*LBL 05

"E. MAX=" ARCL 03 PROMPT END

Une différence majeure : le HP-75 n'admet pas le dimensionnement variable (on peut en fait plus facilement en faire un tableau de données). Rom-Math, encore faut-il faire des calculs au préalable pour le maximum d'éléments de la table. Ceci, évidemment, ce qui revient à réorganiser la logique de la manipulation. Le HP-75 fait donc des variables dynamiquement.

Les instructions d'insertion multiple de la ROM ne sont pas adaptées car le HP-75 n'a pas de table de données.

FONCTIONS DE CARACTÈRES  
CARACTÈRES

HP75

E. Gengoux

L'utilisation directe de sous-chaînes est moins souple avec le HP-75, ainsi qu'avec les autres ROM-Math. (même avec la ROM-Math d'origine). AS(3) (2,5) ou HEXS(A) (2,5) ne font donc passer par une variable intermédiaire sans dimension.

LEFT\$, RIGHT\$ et MID\$ ne sont pas adaptés car le HP-75 n'a pas de chaîne locale. De même pour LEFT\$, RIGHT\$ et STRINGS (mais le HP-75 est tout de même plus souple au niveau de l'utilisation des sous-chaînes, qu'il compare largement cette absence).

PUT, qui était pour argument tout au moins la caractéristique de la ROM-Math, ne peut donc plus être utilisé pour l'écriture de caractères spéciaux.

Le format keys est un format texte directement éditable avec le HP-75.

AGOS (comme avec) du HP-75 ne peut être utilisé directement en HP-75 à l'aide de HTAB de la ROM-Math.

Les instructions d'attente de réponse au clavier (KEYS ou WAITKEYS(+)) sont le HP-75. KEYS pour le HP-75 et les autres ROM-Math (présent) ont des possibilités légèrement différentes.

Enfin, l'ordre des arguments de POS est inversé sur les deux machines.

HP-71 / HP-75

L'ajout au recensement de chaque ligne du programme de HP-75 vers le HP-71, ainsi que l'inverse d'ailleurs, et j'ai pu découvrir que, dans certains cas, ce n'était pas gagné d'avance. Comme il est toujours intéressant de savoir récupérer de bons programmes de l'une sur l'autre machine et que, à moins de connaître rien n'avait été écrit sur le sujet, hormis un article dans PC-ET (N°12/1977), limité à la comparaison des caractéristiques physiques des deux machines, je vous propose un premier inventaire des différences. Je ne prend pas en compte le sujet, mais en mentionnant une aide à ceux qui souhaitent faire de telles conversions. De sorte que les deux machines, qui sont souvent pratiquées comme presque totalement compatibles, ne le sont en fait que si l'on n'utilise ni fichiers, ni fonction art.

Compatibilité des Basic HP71/HP75 12

Bref, du basic idéal et sans surprise tel qu'on n'en voit que rarement dans le pratique.

L'inventaire a été organisé par grand sujet, et ne se limite pas à la machine "de base", les cas posés par les modèles ou les les plus courants (ROM-Math, Vitesse et édit de la ROM-Math) ont été inclus. Bon voyage, donc.

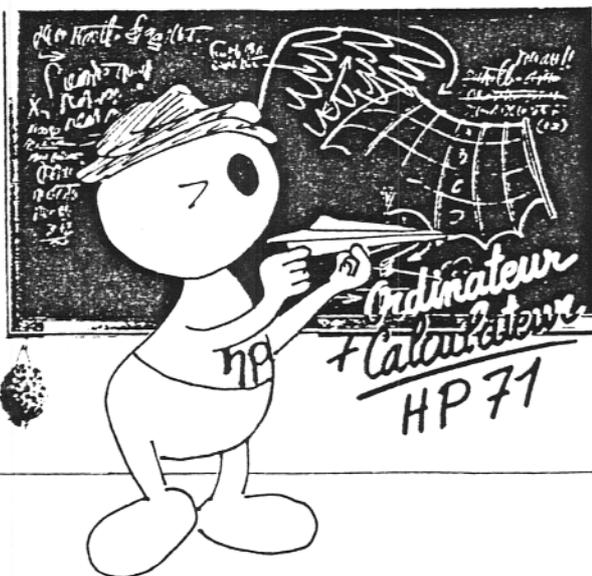


## COMPATIBILITE DES BASIC

### HP-71 / HP-75

J'ai eu récemment à adapter divers programmes du HP-75 vers le HP-71, ainsi que l'inverse d'ailleurs, et j'ai pu découvrir que, dans certains cas, ce n'était pas gagné d'avance... Comme il est toujours intéressant de pouvoir récupérer de bons programmes de l'une sur l'autre machine et que, à notre connaissance, rien n'avait été écrit sur le sujet hormis un article dans *PPC-CJ* (V3N1P27), limité à la comparaison des caractéristiques physiques des deux machines, je vous propose un premier inventaire des difficultés. Il ne prétend pas épuiser le sujet, mais au moins constituer une aide à ceux qui auraient à faire de telles conversions. On verra que les deux machines, qui sont souvent présentées comme presque totalement compatibles, ne le sont en fait que si l'on n'utilise ni fichiers, ni fonctions particulières, ni accès à la boucle... Bref, du Basic idéal et sans surprise tel qu'on n'en voit que rarement dans la pratique...

L'inventaire a été organisé par grand sujet, et ne se limite pas à la machine "de base" : les cas posés par les modules ou Lex les plus courants (Rom I/O, Visicalc et Math du 75 notamment) ont été inclus. Bon voyage, donc.



## CONSTANTES, VARIABLES ET TABLEAUX

Une différence majeure : le HP-75 n'admet pas le dimensionnement variable (on peut tout au plus redimensionner un tableau avec la Rom Maths, encore faut-il l'avoir dimensionné au préalable pour le maximum d'éléments qu'il aura à accueillir, ce qui restreint singulièrement la souplesse de la manipulation). Le HP-71, lui, gère ses variables dynamiquement...

Les instructions d'affectation multiple du 75 ne sont pas admises par le 71 : ainsi,  $A, B, C, D = 3.1416$  est interdit !

## FONCTIONS DE CHAINES DE CARACTERES

L'utilisation directe de sous-chaînes est moins souple avec le HP-75 : ainsi, il est interdit (même avec la Rom I/O) d'écrire  $AS(3)[2,5]$  ou  $HEX$(A)[2,5]$  : il faut donc passer par une variable intermédiaire sans dimension...

$LEFT$, RIGHT$$  et  $MID$$  ne sont pas admis par le 71 (sauf avec module Forth). De même pour  $LTRIM$$  et  $RTRIM$$  (mais le 71 est tellement plus souple au niveau de l'utilisation des sous-chaînes, qu'il compense largement cette absence).

$PUT$ , qui admet pour argument tout numéro de caractère avec le 75, n'accepte que des codes de touches avec le 71 (on ne peut donc pas s'en servir pour forcer un caractère spécial).

Le fichier **keys** est un fichier texte directement éditable avec le 75.

$ASC$(chaîne hexa)$  du 75 ne peut se traduire directement en 71 qu'à l'aide de  $HTA$$  du Lex DESAL.

Les instructions d'attente de réponse au clavier ( $WKEY$$  ou  $WAITKEY$(#)$  pour le 75,  $KEYWAIT$$  pour le 71 si le Lex approprié est présent) ont des possibilités légèrement différentes.

Enfin, l'ordre des arguments de  $POS$  est inversé sur les deux machines.

## DRAPEAUX ET CHANGEMENTS DE BASE

Il n'y a pas d'indicateurs accessibles à l'utilisateur sur le 75 ; ou plutôt, ils doivent être simulés en armant, désarmant ou testant des bits dans une chaîne ascii (fonctions **FLAG\$** et **FLAG?** de la Rom I/O) ou dans un nombre entier (fonction **BIT** de ladite Rom). Pas d'indicateurs système sur le 75, sauf avec **PEEK\$** et **POKE**, mais c'est hyper dangereux !

La fonction de conversion généralisée **TOBASE** de la Rom I/O n'a pas d'équivalent direct 71. Le type octal n'existe pas non plus sur le 71.

Plus vicieux sans doute, car affectant le résultat des conversions décimal / hexadécimal : le 75 travaille en complément à deux (et est limité aux entiers signés compris dans l'intervalle [-32767..+32767]), alors que le 71 travaille en non signé, dans l'intervalle [0..+1048575] ! Problème découvert en essayant de convertir **BAR75C**, de Michael Markov... De ce fait, les fonctions **FLAG?** du 75, et **FLAG** du 71, ne donnent pas le même résultat !

Pour le 75, une chaîne ascii et une chaîne hexadécimale sont deux types de données différents (pas pour le 71).

La Rom I/O (75) est très riche en fonctions de traitement de bits (décalages, complémentation, inversion, calcul booléen...).

## STRUCTURES DE CONTROLE

C'est là que le 71 révèle qu'il est un petit être surdoué... Puisqu'il accepte labels, sous programmes, structuration, etc. que son aîné ignore ou n'accepte qu'avec des restrictions, via la Rom I/O : un seul ordre **SUB** au début du programme, qui est ainsi un fichier indépendant, invoqué par **CALL** externe.

Par contre, le 75 avec Rom I/O accepte tous les branchements calculés (**GOSUBX**, **GOTOX...**), et surtout l'ordre **EXIT** (sortie de boucle propre, mais il a été remédié depuis à ce manque, cf. **JPCLEX**).

L'instruction **DO ERROR** n'existe pas pour le 71 (sauf si utilisation de **JPCLEX**, ou simulation par **BEEP @ DISP MSG\$(#) @ END**).

Le 75 admet la syntaxe :

**ON condition liste d'inst. liées par des @**  
alors que le 71 n'admet que **ON ... GOTO n.**

## LES FICHIERS

C'est le gros morceau ! En effet, le 75 admet quatre types de fichiers (données, texte, **Lif1** et **VisiCalc**), qui ne sont pas directement compatibles avec ceux du 71 ou de la 41 sans conversion préalable. Ainsi, pour passer un fichier **Text** du 71 au 75, il faut :

- Numérotter les lignes de 1 en 1, sur quatre positions plus un espace, sans oublier de mettre des zéros pour cadrer à quatre chiffres les numéros de lignes, puis enfin :
- Convertir du type **Lif1** au type **Text**.

Ceci puisque le 75 ne peut pas exploiter directement un fichier ascii (**Lif1**). Non plus qu'un fichier **Data** ou **Sdata** d'ailleurs, puisqu'il attend ses données sous la forme de lignes **DATA** d'un fichier **Basic**, ce qui permet de les éditer directement en mémoire, mais interdit l'accès direct sur disquette. Différence complète de philosophie, on le voit, qui oblige à faire les conversions de fichiers en plusieurs étapes, soit avec passage par un intermédiaire **Texte** sur disquette, soit en transmettant les données par la boucle, avec un programme de conversion / reformatage aux deux bouts...

Le type **VisiCalc**, capable d'emmagasiner valeurs, labels et formules, n'avait pas d'équivalent sur le 71, jusqu'à l'apparition du logiciel *Workbook-71* (de Richard Harvey, chez **EduCALC**). Il permet, comme **VisiCalc**, de traiter directement ou d'éditer données et formules, ou de laisser ce soin à un programme utilisateur en **Basic** qui saura accéder à toute case de la feuille. Ce type de fichier (**Data 71**) Il comporte également un éditeur de texte plein écran (pour les heureux possesseurs du vidéo 80 colonnes). Les fichiers utilisés (**Data 71**) se convertissent bien dans le sens 71 -> 75 ; dans l'autre sens, il faut extraire les données et labels de la feuille **VisiCalc** au moyen des extensions **Basic** de **VisiCalc75**, les placer dans un fichier **Text** ou **Basic** et convertir cet intermédiaire.

Le type **HPAF**, qui est un cas particulier de fichier **DATA-71**, se convertit en fichier 75 sans trop de problèmes en procédant par étapes : les en-têtes (les fameux tags, où se trouvent les descriptions des zones à convertir et leur type), puis les données elles-mêmes.

Tout cela a l'air assez flou, mais je peux vous assurer qu'on peut très bien, par exemple, passer des données VisiCalc (75, donc) au module Curve Fitting (du 71) et vice versa...

Importante limitation du 75 : il ne sait pas lire directement de fichier sur support externe, donc, pas d'accès direct comme le 71 sait le faire. Par contre, la Rom I/O lui permet d'être beaucoup plus souple en mémoire : insertions, échanges, gestion directe des pointeurs et longueurs de fichier, de ligne... et édition directe de tout fichier ! Le 71 ne peut approcher cette souplesse que muni du Lex DATALEX : voir *HPAF Utilities* de Corvallis.

Notez au passage que le 71 n'accepte pas les fichiers Basic (constitués uniquement d'instructions Data) : il ne peut pas accéder à des DATA externes. Par contre, il lit directement des données de HP-41 (fichiers Sdata), chose que son grand frère ne sait pas faire, sauf avec les deux Lex de J. Y. Hervé, MADLEX et DATA41. Un Français ! Pour une fois...

#### INSTRUCTIONS D'ENTREE/SORTIE

Les READ# et PRINT# s'utilisent de façon similaire. Il n'y a pas d'équivalent direct de INSERT#, DELETE#, SEEK#, LINE# et de la plupart des fonctions avancées de gestion de fichiers de la Rom I/O du 75, qu'il faut donc simuler par des boucles de recopie (à moins que vous n'utilisiez DATALEX, déjà cité, ou l'éditeur de texte...). Notez aussi que le 71 ne tolère pas les dépassements de fin de fichier ou de longueur d'enregistrement, alors que le 75, lui, s'en moque : il travaille en longueur variable et en gestion dynamique !

#### ACCES A LA BOUCLE HP-IL

Quatre grandes différences : assignation des appareils sur la boucle, mode de contrôle, syntaxe des instructions SEND et ENTER (pour le 75, seulement si vous avez la Rom I/O : le Lex HPILCMD5 ne donne pas cette possibilité) et les deux mots particuliers du 75, SENDIO et ENTIO\$.

Contrôle de la boucle : le 75 ne peut être que contrôleur (cette partie de la Rom I/O n'a jamais été achevée : au détour de la notice, vous découvrirez peut-être les mots magiques qui auraient l'air de pouvoir permettre... Mais,

las, ils sont analysés et acceptés, mais ça s'arrête là ! Tout ce qu'on peut faire, c'est placer le 75 en mode récepteur et lui faire analyser les frames au passage, avec RIO et WIO (qui est dans la Rom I/O, voir le programme SCOPE...).

Pour la désignation des appareils sur la boucle, se souvenir que le 75 ne connaît que celles de la forme ":Mx" où M est une lettre désignant la classe de l'appareil (Display, Mass, Printer, Interface, Controller...) et x un numéro d'ordre, et que l'assignation est manuelle (ASSIGN IO), sauf si la ROM I/O est présente et AUTOIO actif. Pas de noms génériques tels que :PRINTER, pas de désignation par numéro dans la boucle ou par l'ID d'accessoire (:%64, par exemple).

Quelques différences mineures de syntaxe entre les SEND et ENTER des deux machines (attention à la désignation des unités, aux blancs, et à l'utilisation de SDA)

SENDIO et ENTIO\$ du 75 sont, il faut le dire, très puissants, tolérants et concis : pas besoin d'envoyer explicitement de MTA, MLA, ils ne refusent pas certaines frames (tels SDA). Voyons comment les simuler sur 71 :

SENDIO :

SEND *liste*

Là, ça reste encore assez simple...

ENTIO\$ :

*n*=DEVADDR(...)

SFLAG -23

SEND UNL UNT TALK *n* DDT *n*

ENTER :LOOP ; *liste*

Voilà. Encore une fois, c'est un premier jet : si un des courageux qui m'ont suivi sans faillir jusqu'ici découvre d'autres pièges, qu'il m'en fasse part, un addendum sera publié. Et le mois prochain, si vous êtes sages, on vous dira tout sur le PMS et sur l'art de développer des logiciels en Eproms. A bientôt.

Eric Gengoux (108)

# CERCLES, PARABOLES, TRACER TOUTES LES FONCTIONS

## FORTH

- X. Bille
- S. Vaudenay

- Tracer les fonctions 16
- Jeu de la vie 19

## ASSEMBLEUR

- G. Toublanc

- Réfractaire 21

## BASIC

- G. Toublanc
- A. Gillet
- A. Herreman
- E. Gengoux

- FracStory 27
- Débuter en boucles et tests 28
- Agenda 28
- Dessine-moi un autre mouton 29

- E. Gengoux

- Exemple de résultat de "BARGRAF2" 31

- S. Vaudenay
- G. Toublanc
- A. Gillet
- A. Herreman
- E. Gengoux

- Programme "VIEB" 32
- Programme "TFRAC" 33
- Programme "COMPTEUR" 33
- Programme "AGENDA" 34
- Programme "BARGRAF2" 35

## LE COIN DES LHGX

39

# CERCLES, PARABOLES, TRACER TOUTES LES FONCTIONS

Dessiner des courbes  $y=f(x)$  était le pari. Il fallait donc pouvoir gérer et afficher un plan de résolution supérieur à 132x8. Grâce à ce programme, tous les types de courbes (de la droite à l'hyperbole en passant par les courbes en polaires) peuvent être tracés dans un plan de 132x128.

## RENTRE LE PROGRAMME

Le programme s'étale sur trois langages (Forth, Basic, Assembleur). Ecrire d'abord le code de la fonction MIX. L'assembler. Augmenter la capacité du dictionnaire de 2000 quartets ou plus, entrer le contenu de la partie Forth, la compiler.

## MODE D'EMPLOI

Vous disposez de cinq fonctions évoluées (plus d'autres qui sont détaillées plus avant).

**CLEAR** efface l'écran résolution.

**AXIS** dessine des axes, qu'importe où est située l'origine du repère. Les graduations de 5 en 5 sont repérées par un point blanc le long de ces axes.

**DISPLAY** affichage du plan de résolution. L'écran est alors une fenêtre coulissante. Une pression sur [^] (resp. [v]) fait coulisser l'écran de 4 lignes vers le "haut" (resp. "bas"). Les extrémités sont repérées par "up" (resp. "down"). Une pression sur g[^] (resp. g[v]) place la fenêtre tout en haut (resp. en bas). Une pression sur [SPC] sort du programme.

**MODE** sert à définir l'origine du repère du plan, ainsi que les échelles des axes. Affichage de :

X-Origin (I) 66

qui signifie que l'origine des X se situe, en valeur absolue, en 66 ( $1 \leq X \leq 132$ ). Le I pour rappeler qu'il s'agit de la I-stack. Une pression sur [v] visualise l'origine des Y de la même manière.

Exemple : modifier l'origine des Y ( $1 \leq Y \leq 128$  en valeur absolue) :

- appuyer sur [RUN]. Rien ne se passe.  
- entrer 23 [ENDLINE]. Affichage de :  
Y origin (I) 23

Visualiser les échelles :

Une pression sur [v]. Affichage de :  
X-ratio (F) 1  
F pour dire qu'il s'agit de la F-stack.

Changer la valeur :

- appuyer sur [RUN]. Rien ne se passe, vous l'aviez pressenti, hein ?  
- taper .25 [ENDLINE]. Affichage de :  
X-ratio (F) .25

Pour revoir l'origine des Y, un appui sur [^]. Pour réafficher ces résultats, utiliser les flèches [^], [v], g[^], g[v]. Pour sortir du programme, taper [SPC].

Maintenant, la manière de tracer les courbes. Exemple :  $y=3x+5$ . Créer un mot Forth ayant pour paramètres (X -- X Y), ceux-ci étant sur la F-stack. Ainsi :

: DROITE FENTER 3. F\* 5. F+ ;

La question que vous vous posez : pourquoi faire si compliqué ? Cette méthode permet de tracer des courbes paramétrées : prenons le cercle avec  $x=40\cos t$ ,  $y=40\sin t$ ,  $t$  en degrés. Dans la mesure où le repère est défini orthonormé :

: CERCLE 3. F\* FENTER

COS 40. F\* X<>Y SIN 40. F\* ;

Si la notation polonaise inverse ne vous convient pas, alors réalisez un programme Basic :

: FONC " X=FORTHF @ RUN 10" BASICX " Y" BASICF ;

Bon, et alors ? Faites tracer votre fonction par **DRAW**. Exemples : **DRAW CERCLE**, **DRAW DROITE**, **DRAW FONC**, etc. Quand la main vous revient, vous avez accès aux autres mots.

Effets de bord. Les valeurs calculées peuvent sortir de la matrice 132x128. En ce cas, vous en serez averti par des X ou des Y selon que l'excédent se trouve sur l'un des deux axes associés.

## LE PROGRAMME

Il y 3 modules de fonctions :

- l'affichage avec MIX, DISPLAY
- les normalisations CLEAR, AXIS, MODE
- les écritures avec COORDI, DRAW

COORDI est un mot qui place un point dans le plan, sachant ses coordonnées X, Y et l'origine du repère.

MIX, partie assembleur, sert à extraire le poids faible de l'octet d'une certaine adresse, le poids fort d'un autre ailleurs, 132 fois séquentiellement. Cette suite d'octets construite à cheval sur deux ensembles sera affichée. D'où l'origine du coulisant de l'écran de 4 lignes en 4 lignes.

### CONCLUSION

Ce programme a un point faible : sa lenteur. Dix secondes ou plus pour tracer une courbe (autant pour les axes), c'est long. Mais ce vocabulaire est une base, bien sûr, on doit pouvoir améliorer, optimiser.

Xavier Bille (203)

#### Partie Assembleur :

```

FORTH
here EQU #2FB93 là où est stockée la valeur
* de HERE
WORD 'MIX' (ad1 ad2 -- ad3)
ADDOX
R0=A sauvegarde de D0 en R0
D0=(5) here calcul de l'adresse de PAD :
* HERE + 5A
A=DAT0 A
LC(5) #5A
A=A+C A
D0=A
R1=A R1 := ^ PAD
C=DAT1 A
C=C+1 A ad2 := ad2+1
DAT1=C A récupérer LE bon quartet !
LC(2) #84 prémice de boucle 1 à 132
D1=D1+ 5 D1 sur ad1
deb R2=C
GOSUB adres récupérer le quartet indexé
* par ad2
ASL B rotation sur 1 bit
D1=D1- 5 ad1
GOSUB adres récupérer le quartet indexé
* par ad1
DAT0=A B on range l'octet construit
* à l'ad3

```

```

D0=D0+ 2 on se place 1 octet de +
* sur ad3
D1=D1+ 5 ad2
C=R2
C=C-1 B
?C#0 B 132 boucles ?
GOYES deb
C=R1 on range ad3 dans la pile
DAT1=C A
C=R0
D0=C
RTNCC

```

\* extraction d'un quartet à 1 adresse

```

adres C=DAT1 A adresse
CD1EX
A=DAT1 1 A := (C)
D1=D1+ 2 permet de pointer sur
* l'octet suivant
CD1EX
DAT1=C A
RTN

```

#### Partie Forth :

HEX

```

: GRAPH ( repérage dans le plan graphique)
CREATE 108 * NALLOT
DOES> SWAP 108 * + ;

```

```

: GDISP 84 " GDISP FORTH$" BASICX ;

```

```

10 GRAPH TABLE

```

```

: DISPLAY ( -- )
0

```

```

BEGIN ( *1*)

```

```

DUP 2/ 2DUP 2* -

```

```

IF DUP 1+ TABLE SWAP TABLE

```

```

MIX

```

```

ELSE TABLE THEN

```

```

GDISP

```

```

BEGIN ( *2*)

```

```

KEY CASE

```

```

13 OF DUP 1E = ( [v] )

```

```

IF ." Down ." CR ELSE 1+ THEN

```

```

1,

```

```

ENDOF

```

```

12 OF DUP 0= ( [^] )

```

```

IF ." Up ." CR ELSE 1- THEN

```

```

1,

```

```

ENDOF

```

```

20 OF DROP 0, ." end" ENDOF

```

```

( [SPC] )

```

```

14 OF DROP 0 1, ENDOF

```

```

( g[^] )

```

```

15 OF DROP 1E 1, ENDOF

```

```

      ( g[v] )
    -1 SWAP
  ENDCASE
  0= UNTIL ( *2* )
  0= UNTIL ; ( *1* )

CREATE ORIGIN 4 NALLOT
42 ORIGIN C! 40 ORIGIN 2+ C!

: COORDI ( Y X -- )
  ORIGIN C@ +
  DUP 84 < OVER -1 > AND
  IF 2* SWAP
    ORIGIN 2+ C@ - NEGATE
    DUP 80 < OVER -1 > AND
    IF DUP 8 / TABLE ROT + SWAP
      8 MOD ( calcul de 2^ )
      CASE
        0 OF 1 ENDOF
        1 OF 2 ENDOF
        2 OF 4 ENDOF
        3 OF 8 ENDOF
        4 OF 10 ENDOF
        5 OF 20 ENDOF
        6 OF 40 ENDOF
        7 OF 80 ENDOF
      ENDCASE
    OVER C@ OR SWAP C!
  ELSE
    2DROP ." Y" THEN
  ELSE
    2DROP ." X"
  THEN ;

: CLEAR ( -- )
  0 TABLE 840 0 FILL ; IMMEDIATE

: AXIS ( -- )
  ORIGIN C@ NEGATE DUP 83 +
  BEGIN ( *axe des X* )
    DUP 5 MOD
    IF 0 OVER COORDI THEN
      1- 2DUP >
    UNTIL 2DROP
    ORIGIN 2+ C@ DUP 7F -
    BEGIN ( *axe des Y* )
      DUP 5 MOD
      IF DUP 0 COORDI THEN
        1+ 2DUP <
      UNTIL 2DROP ; IMMEDIATE

FVARIABLE XRATIO 1. XRATIO STO
FVARIABLE YRATIO 1. YRATIO STO

: MODE ( -- )
  0,
  BEGIN ( *1* )
  OVER CASE

```

```

  0 OF IF XRATIO STO THEN
    ." X-ratio (F) " XRATIO RCL F.
  ENDOF
  1 OF IF YRATIO STO THEN
    ." Y-ratio (F) " YRATIO RCL F.
  ENDOF
  2 OF IF SWAP DUP 84 < OVER -1 > AND
    IF ORIGIN C! ELSE DROP THEN THEN
    ." Origin X (I) " ORIGIN C@ .
  ENDOF
  3 OF IF SWAP DUP 80 < OVER -1 > AND
    IF ORIGIN 2+ C! ELSE DROP THEN THEN
    ." Origin Y (I) " ORIGIN 2+ C@ .
  ENDOF
  ABORT" Error"
  ENDCASE
  BEGIN ( *2* )
  KEY CASE
  20 OF DROP 0, ENDOF
  F OF CR DEPTH >R PAD 2+ EXPECT96
  SPAN @ DUP 0>
  IF PAD C! PAD NUMBER DEPTH R> - 1 >
    IF DROP OVER 2 <
      IF ITOF ELSE SWAP THEN
        THEN -1
      ELSE R> 2DROP 0
    THEN
    1,0
  ENDOF
  12 OF 1- DUP 0< - 0 1, ENDOF
  13 OF 1+ DUP 3 > + 0 1, ENDOF
  1 SWAP
  ENDCASE
  0=
  UNTIL CR ( *2* )
  0=
  UNTIL ; ( *1* )

VARIABLE FUNCTION

: DRAW ( -- , draw |___| )
  ' FUNCTION !
  ORIGIN C@ NEGATE DUP 83 +
  BEGIN
    DUP ITOF XRATIO RCL F*
    FUNCTION @ EXECUTE
    YRATIO RCL F* FTOI RDN
    XRATIO RCL F/ FTOI
    COORDI
    1- 2DUP >
  UNTIL 2DROP ;

DECIMAL

```

## JEU DE LA VIE

Bonjour tout l'monde. Devant la baisse du taux d'insultes que le Journal reçoit pour moi, relative à la baisse du taux d'articles, je suis contraint, au grand regret des anthropologues spécialistes en singes volants, d'en pondre un nouveau.

L'ânerie de ce mois-ci consiste à faire un jeu de la vie sur HP-71. Comme chacun sait, le jeu de la vie se joue sur un quadrillage dans lequel on met des "cellules". Chaque cellule a 8 cases voisines qui peuvent être occupées ou non par d'autres cellules. Bien évidemment, on appellera voisine une cellule qui occupe une case voisine. Le jeu de la vie consiste à étudier l'évolution d'une population de cellules.

Une génération est la forme d'une population à un instant donné. On peut établir la génération suivante en utilisant les règles ci-dessous :

- une case qui a trois ou quatre voisines est occupée, à la génération suivante par une cellule ;
- une case qui n'a ni trois ni quatre voisines est vide à la génération suivante.

Le programme que je vous propose aujourd'hui calculera les générations successives d'une population dont vous aurez donné une génération et les visualisera sous vos yeux émmmerveillés. Pour l'utiliser, c'est très simple. Entrez les deux fichiers (Basic et Text), entrez le fichier KEYWAIT, célèbre aujourd'hui dans l'histoire de l'émancipation assembleur du 71, prenez soin d'avoir mis le module Forth et de ne pas avoir laissé trainer vos élucubrations forthées dans le fichier Forthram, et lancez le programme Basic.

Après une attente de trente secondes, le temps de mettre tout en ordre, le programme passe en mode édition, c'est à dire qu'il vous demande d'introduire la première génération de votre population.

Pour ce faire, vous sélectionnez les cases à modifier au moyen des touches fléchées, puis vous frappez [ENDLINE]. Alors vous choisissez de remplir [->] ou de vider [-<] la case. Une fois la génération introduite, frappez [ENDLINE] deux fois, pour sortir de l'éditeur.

Le programme vous demande alors si vous voulez commencer le calcul (répondre au moyen des deux flèches).

Vous comprendrez facilement la suite, mais sachez que le message expansion signifie que votre population s'étend en dehors du quadrillage que tolère la machine.

Pour prendre un exemple concret, si la génération 0 d'une population est :

\*\*  
\*\*

toutes les générations le seront également. C'est une population "constante".

Si la génération 0 à la forme d'un carré de trois cellules de coté, la population sera périodique (vous ne pourrez le vérifier qu'à la main, car les étapes intermédiaires débordent du quadrillage du 71. Toutefois, la partie Forth du programme a été prévue pour n'importe quel quadrillage, à la condition de ne pas essayer de le visualiser.

Amusez-vous bien avec ce programme, et n'oubliez pas de faire part de vos études sur ces populations au Journal.

Serge Vaudenay (124)

```
( CTABLE {x y ---} "x y CTABLE nom" : crée un )
( tableau "nom" de dimension x*y bits )
( nom : {x y --- d adr} l'exécution de "nom" )
( fournie l'adresse et le numéro b du bit de )
( l'octet correspondant de la case xy )
```

```
: CTABLE
  CREATE 2DUP C, C, * 8 / 1+ ALLOT
  DOES> ROT OVER C@ * ROT + 4 /MOD ROT + 2+ 2+
;
```

```
( B@ : {b adr ---n} retourne l'état n du bit b )
( de l'octet adr )
```

```
: B@ C@ 2* SWAP 1+ 0
  DO 2/
  LOOP 1 AND
;
```

```
( B! : {n b adr ---} place le bit b de l'octet )
( adr à l'état n )
```

```

: B! 1 ROT 1+ 0
DO 2*
LOOP 2/ 2DUP NOT SWAP C@ AND ROT 2SWAP *
ROT OR SWAP C!
;

```

( dimensions du tableau utilisé par le programme )

```

DECIMAL 18 CONSTANT x 8 CONSTANT y HEX
x y CTABLE TABLE
x y CTABLE TABLE2

```

( PLP: {n x y---} place la case xy de TABLE2 )  
( à l'état n )

```

: PLP TABLE2 B! ;

```

( RGP: {x y---n} retourne l'état n de la case )  
( xy de TABLE )

```

: RGP TABLE B@ ;

```

( ?PION: {n---b} dit si la case à n voisine )  
( sera occupée ou non )

```

: ?PION
CASE
3 OF 1 ENDOF
4 OF 1 ENDOF 0 SWAP
ENDCASE
;

```

```

: CMTF
2OVER ROT + >R + R> RGP >R ROT R> + ROT ROT ;

```

( TABLE2TO1: {---} copie la table 2 )

```

: TABLE2TO1
0 0 TABLE2 0 0 TABLE DUP 2- DUP 2- C@ SWAP C@
* 8 / 1 + ROT DROP CMOVE DROP ;

```

( T+1: {---} calcule la génération suivante de )  
( TABLE )

```

: T+1 x 1- 1
DO y 1- 1
DO 0 J I
-1 -1 CMTF
0 -1 CMTF
1 -1 CMTF
-1 0 CMTF
1 0 CMTF
-1 1 CMTF
0 1 CMTF
1 1 CMTF
ROT ?PION ROT ROT PLP
LOOP
LOOP TABLE2TO1
;

```

( NEANT: {---} vide les tables )

```

: NEANT
x 0 DO y 0 DO 0 J I PLP LOOP LOOP TABLE2TO1 ;

```

( VIEW: {---} visualise TABLE à condition )  
( qu'elle ait la dimension "standard" )

```

: VIEW
0 0 TABLE 2E328 12 CMOVE DROP ;

```

( ?FIN: {---b} si TABLE à la dimension )  
( "standard", dit s'il y a expansion )

```

: ?FIN
1 1 TABLE C@ 0= 0= 10 1 TABLE C@ 0= 0= ROT
2SWAP 2DROP OR x 1- 1
DO I 1 RGP OR I 6 RGP OR
LOOP
;

```



## REFRACTAIRE

Convertir un nombre décimal en fraction a été résolu deux fois dans JPC :

- en Basic par Pierre David (JPC 20) : fichier **FRAC** ou sous-programme **FR**.

- en Forth par Alain Herreman (JPC 34) : mot Forth **FRC**.

Alors voici le troisième larron : le mot Basic **FRAC\$**.

Quelques remarques concernant **FRAC** et **FRC** :

- **FRAC** (Pierre) : au lieu de **INTEGER A, B** faire **DIM A, B** (et donc **DIM A, B, X**) si l'on ne veut pas déclencher des rafales d'overflow ou de /zero ou obtenir 9.00001 -> 99999/99999 ! pour la précision de  $10^{-6}$ . Depuis sa publication, l'auteur avait d'ailleurs remédié à cela. D'autre part pour une précision de  $10^{-6}$  on obtient :  
4.000001 -> 4000001/1000000  
3.999999 -> 4/1  
4.0000003 -> 13333333/3333333  
on aurait préféré la même réponse logique 4/1.

- **FRC** (Alain) : si l'algorithme utilisé est très simple et a l'avantage d'être compréhensible par tous, il pêche par sa lenteur à partir de 3 décimales (3.14159 **FRC** : pas encore de réponse au bout d'une demi-heure !).

Côté expression des résultats, il me semble préférable d'avoir 4 au lieu de 4/1.

**FRAC\$** utilise le même type de programmation que **FRAC** (Pierre) mais sans les inconvénients cités.

Deux options sont possibles : choix d'une précision absolue ou du nombre d'itérations de la routine.

Rapidité et confort se paient : coût 300 octets environ.

## MODE D'EMPLOI

### FRAC\$ :

conversion d'un nombre décimal en une chaîne alphanumérique de l'approximation fractionnaire ou non de ce nombre.

syntaxe : **FRAC\$ (N)**

précision implicite :

si  $|N| \geq 1$  :  $10^{-10}$

si  $|N| < 1$  :  $10^{-10+\text{exposant de } N}$

syntaxe : **FRAC\$ (N, n)**

$1 \leq n \leq 99$  précision optionnelle  $10^{-n}$

si  $n = 0$  Titan impose la précision implicite

syntaxe : **FRAC\$ (N; -n)**

$1 \leq n \leq 99$  ordre de l'approximation

ou nombre d'itérations de la routine plus une

## EXEMPLES

**FRAC\$ (5)**

-> 5

**FRAC\$ (PI, 12)**

-> 1146408/364913

(valeur : PI, delta = 0)

**FRAC\$ (PI, 2)**

-> 22/7 (une vieille connaissance)

(valeur : 3.1428..., delta =  $1.26 \cdot 10^{-3}$ )

**FRAC\$ (PI)**

-> 312689/99532

(valeur :  $\text{PI} + (3 \cdot 10^{-11})$ , delta =  $3 \cdot 10^{-11}$ )

**FRAC\$ (-1.25E-15)**

->  $-1/8 \cdot 10^{14}$ , delta = 0

**FRAC\$ (4.001, 3)**

-> 4, delta =  $-1 \cdot 10^{-3}$

**FRAC\$ (4.001, 4)**

-> 4001/1000, delta = 0

**FRAC\$ (3.999, 3)**

-> 4, delta =  $1 \cdot 10^{-3}$

**FRAC\$ (3.1031031031)**

-> 3100/999, delta = 0

**A\$=FRAC\$ (SQR(2), 2) @**

**B\$=FRAC\$ (SQR(3), 2) @**

**DISP "( "&A\$&" ) / ( "&B\$&" )"**

-> ( 17/12 ) / ( 19/11 )

(valeur : 0.82017..., delta =  $3.67 \cdot 10^{-3}$ )

FRAC\$(5/7+8/9-13/11)

-> 292/693, delta = 0

FRAC\$(PI, -2)

-> 22/7

FRAC\$(PI, -3)

-> 333/106

FRAC\$(PI, -10)

-> 1146408/364913

valeur PI

Je dois remercier ici Pierre David qui m'a remis sur le droit chemin : j'avais créé un mot FRAC(N) qui ne savait qu'afficher des résultats, ceux-ci n'étant pas récupérables. Janick Taillandier, à qui je dois la routine A2STR, a été l'artisan de la fiabilité de FRAC\$ car il a su trouver les contre-exemples qui mirent cette fonction en difficulté dans ses premières versions.

### ANNEXES

1- FRC (Alain) est lent car si  $N = A/B$  le nombre d'itérations est égal à  $\min(A,B)$  et Forth perd sa rapidité car FRC calcule en virgule flottante donc avec les routines du Basic. A tout cela s'ajoute le type de précision qui est pénalisant.

FRC demande que :

$$|A - B/N| \leq 10^{-n}$$

$$\text{d'où } |A/B - N| \leq |10^{-n} / B|$$

donc c'est une précision relative.

Par exemple, si  $B > 10^6$  et  $n=6$ , la précision absolue correspondante est d'au moins  $10^{-12}$  ! Ceci nous vaut des gags du genre :

$$3/82646281 \rightarrow 6/165292562$$

2- l'algorithme des réduites de fractions continues, utilisé pour FRAC et FRAC\$ n'est pas simple, aussi voici quelques explications.

Soit un nombre décimal :  $N = \text{INT}(N) + \text{FP}(N)$

posons  $H_0 = N$

$$\text{FP}(N) = 1 / H_1$$

$$H_0 = a_0 + 1/H_1$$

d'où  $H_1 = 1 / (H_0 - \text{INT}(H_0))$

avec  $a_0 = \text{INT}(H_0)$

$$H_1 = a_1 + 1 / H_2$$

d'où  $H_2 = 1 / (H_1 - \text{INT}(H_1))$

avec  $a_1 = \text{INT}(H_1)$

etc... 
$$\left| \begin{array}{l} H_{i+1} = 1 / (H_i - \text{INT}(H_i)) \\ a_{i+1} = \text{INT}(H_i) \end{array} \right|$$

et 
$$N = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + \dots}}}$$

posons  $N-1 = 1 \quad D-1 = 0$  etc...

réduites successives :

1ère 
$$\frac{N_0}{D_0} = \frac{a_0}{1}$$

2ème 
$$\frac{N_1}{D_1} = a_0 + \frac{1}{a_1} = \frac{a_1 a_0 + 1}{a_1} = \frac{a_1 N_0 + N_1}{a_1 D_0 + D_1}$$

3ème 
$$\frac{N_2}{D_2} = a_0 + \frac{1}{a_1 + \frac{1}{a_2}} = \frac{a_2(a_1 a_0 + 1) + a_0}{a_2 a_1 + 1}$$

$$\frac{a_2 N_1 + N_0}{a_2 D_1 + D_0}$$

etc...

$N_i$  ème réduite 
$$\left| \begin{array}{l} \frac{N_{i+1}}{D_{i+1}} = \frac{a_{i+1} N_i + N_{i-1}}{a_{i+1} D_i + D_{i-1}} \end{array} \right|$$

On démontre que  $N_{i+1} / D_{i+1}$  converge vers N.

### EXEMPLE

$$N = 3,14 = 3 + 0,14$$

1ère approximation :

$$N_0 / D_0 = 3 / 1 \quad \text{delta} = 0,14$$

$$H_1 = 1 / 0,14 = 7,1428\dots$$

$$a_1 = 7$$

2<sup>ème</sup> approximation :

$$N_1 / D_1 = ( 7*3 + 1 ) / ( 7*1 + 0 )$$

$$= 22 / 7 \quad \text{delta} = -2,85..E-3$$

$$H_2 = 1 / 0,1428... = 6,99..986$$

$$a_2 = 6$$

3<sup>ème</sup> approximation :

$$N_2 / D_2 = ( 6*22 + 3 ) / ( 6*7 + 1 )$$

$$= 135 / 43 \quad \text{delta} = 4,65..E-4$$

$$H_3 = 1 / 0,99..986 = 1,00..01$$

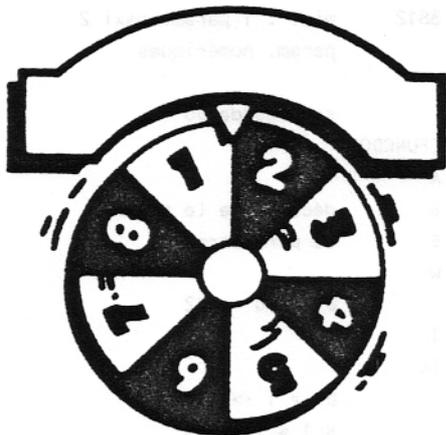
$$a_3 = 1$$

4<sup>ème</sup> approximation :

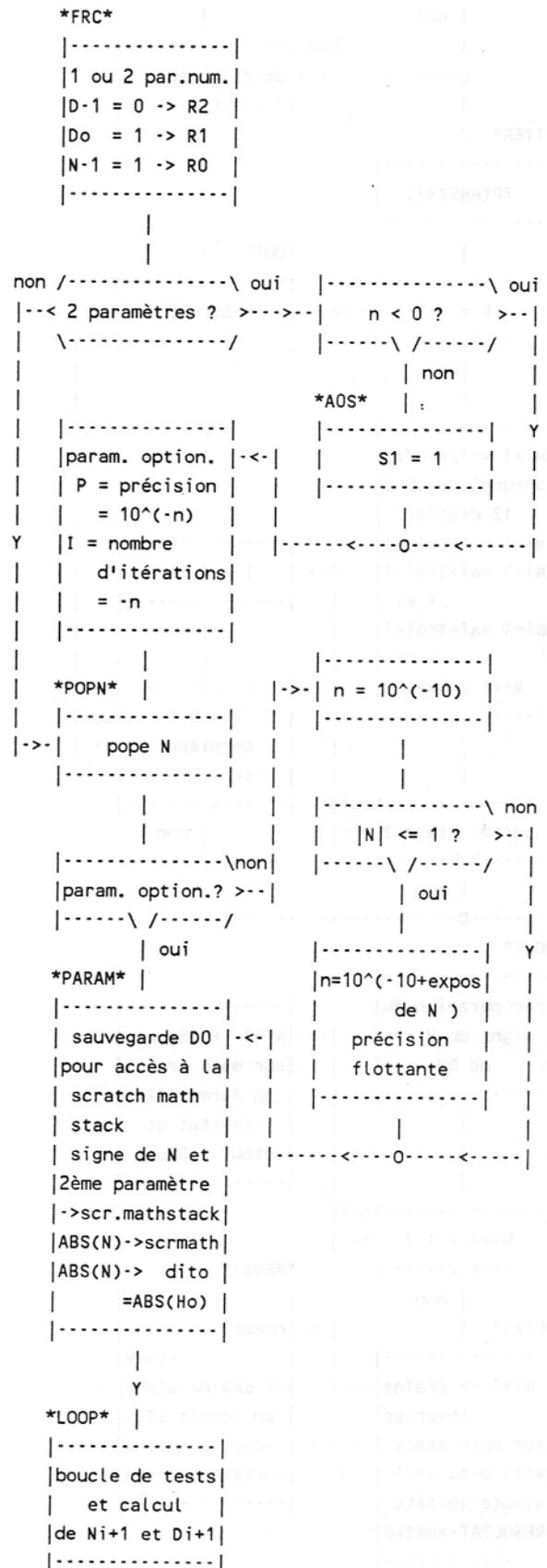
$$N_3 / D_3 = ( 1*135 + 22 ) / ( 1*43 + 7 )$$

$$= 157 / 50 \quad \text{delta} = 0$$

La convergence est obtenue :  $N = N_3 / D_3$



## ORGANIGRAMME DE FRACLEX



```

*LOOP* |-----|
|-----\ non | delta = | |
|> param. itéra.? >---|Ni+1/Di+1 - N||
|-----\ /-----/ |-----|
| | oui | |
| | non /-----|
| 0-----<-----< delta<=précis.?)
| | \-----\ /-----|
*ITER* | | oui
|-----| |
| FP(ABS(Hi)) | Y
|-----| | |
| | *OUT* |
|-----\ oui |-----|
| FP = 0 ? >--->---| RESULTAT |<--|
|-----\ /-----/ |-----|
| | non | |
| | | |
|-----| |
|Hi+1 = 1/FP(Hi)| ^
|arrondi de 15 à|
| 12 digits |
|ai-1 = IP(Hi-1)| |-----|
|Ni+1 =ai+1*Ni+1 |>---| I = I + 1 |
| + Ni | |-----|
|Di+1 =ai+1*Di+1 | |
| + Di | |
| Ni+1 / Di+1 | |-----|
|-----| |
| | I = 0 ? \oui|
| | dernière >---|
| | itération ? /
|-----\oui| |-----\ /-----|
| param. itéra ? >---| | non
|-----\ /-----/ |
| | non | Y
|-----<-----<-----|
*OUT*
|-----|
|récupération du| |-----|
| signe de N et |>---|Ni+1 -> chaîne |
| de D0 | |sur math stack |
|-----| | en format STD |
| | | résultat et |
| | | retour à Basic |
|-----| |-----|
|-----\oui| |
| Di+1 = 1 ? >---|
|-----\ /-----/ *REVST*
| | non |-----|
*DIF1* |>---|nombre -> math |
|-----| | stack |
| Di+1 -> chaîne|>---| -> chaîne alpha |
| inversée| | en format STD |
|sur math stack |>---<---|>chaîne invers|
|Ni+1 dito Di+1 | |en-tête enlevée|
|ajoute en-tête | |-----|
|RESULTAT->BASIC|
|-----|

```

```

LEX 'FRACLEX'
ID #E1
MSG 0
POLL 0
AD2-12 EQU #0C35F
ADHEAD EQU #181B7
CLRFRC EQU #0C6F4
DSPFMT EQU #2F6DC
DV2-12 EQU #0C4A8
EXPR EQU #0F23C
FUNCD0 EQU #2F8BB
MP2-12 EQU #0C432
POP1R EQU #0E8FD
RCLW1 EQU #0E981
RCLW2 EQU #0E9BE
RCSCR EQU #0E954
REV$ EQU #1B38E
SPLITA EQU #0C6BF
STR$SB EQU #18149
STSCR EQU #0E92C
TST12A EQU #0D476
URES12 EQU #0C994
XINV15 EQU #0C33E Normalement 1/X15
XXHEAD EQU #1A44E
XYEX EQU #0C697
ENTRY FRC
CHAR #F
KEY 'FRAC$'
TOKEN 88
ENDTXT
NIBHEX 8812 mini : 1 param. maxi 2
* param. numériques
FRC ADOEX sauvegarde D0
D0=(5) (FUNCD0)
DAT0=A A
C=C-1 S décrémente le nombre
C=C-1 S de paramètres
C=0 W
R2=C D-1 = 0 -> R2
P= 1
C=P 14
R1=C Do = 1 -> R1
R0=C N-1 = 1 -> R0
ST=0 0 repère paramètre précision
GOC POPN 1 seul paramètre alors POPN
GOSBVL POP1R pope et teste n (précision
* ou nombre d'itérations)
D1=D1+ 16 actualise D1
?A=0 S paramètre précision ?
GOYES AOS oui alors on conserve le
* repère précision (S0=0)
ST=1 0 repère itération

```

AOS	A=0	S		ITER	GOSUB	AR3	Ni -> scratch math stack
	?A=0	X	1 seul digit ?		GOSBVL	RCLW2	X = ABS(Hi)
	GOYES	aslc2	oui alors shift 2 fois		GOSUB	clrfrc	A(W) = IP(ABS(Hi))
	ASLC		non alors shift 3 fois				
aslc2	ASLC			out	GOC	OUT	FP(Hi) = 0 alors résultat
	ASLC		ici A(B) = n		GOSUB	A-1S	A(W) = FP(ABS(Hi))
	C=A	B	C(W) = 10^n		GOSBVL	SPLITA	A(W) -> X
	C=-C	X	C(W) = 10^(-n)		GOSBVL	XINV15	X = 1/FP(ABS(Hi))
					*		= ABS(Hi+1)
POP	GOSBVL	POP1R	pope et teste N		GOSUB	ROUND	ABS(Hi+1) arrondi à 12
	?C#0	B	C(B) = n ?		*		digits -> A(W)
	GOYES	PARAM	oui alors précision donnée		GOSBVL	RCSCR	fait la place pour Hi+1
	*		ou nombre d'itérations		GOSUB	STOfrc	ABS(Hi+1) -> scr. math st.
	*		demandé		*		puis A(W) = IP(ABS(Hi+1))
	LCHEX	499			*		= ai+1
	C=C-A	X	C(X) = 499 - exposant de N		C=R3		C(W) = ABS(Ni)
	LCHEX	990	C(W) = 10^(-10)		GOSUB	mp2-12	A(W) = ai+1 * ABS(Ni)
	GONC	PARAM	exposant de N > 0 alors		C=R0		C(W) = ABS(Ni-1)
	*		C(W) = précision		GOSUB	AD	A(W) = ABS(Ni+1)
	C=C+A	X	exposant < 0 alors préc.		*		= ai+1*ABS(Ni)+ABS(Ni-1)
	*		flottante 10^(-10 + expos.	AR3EX			R3 = ABS(Ni+1)
	*		de N )	*			A(W) = ABS(Ni)
				R0=A			RO = Ni remplace Ni-1
PARAM	C=A	S	récupère le signe de N	GOSBVL	RCLW1		X = ABS(Hi+1)
	DAT1=C	W	sauve le signe de N et le	GOSUB	clrfrc		A(W) = ai+1
	*		paramètre précision ou	C=R1			C(W) = Di
	*		itérations sur la math	GOSUB	mp2-12		A(W) = ai+1 * Di
	*		stack	C=R2			C(W) = Di-1
	A=0	S	A(W) = ABS(N)	GOSUB	AD		A(W) = Di+1 = ai+1*Di+Di-1
	GOSUB	STO	sauvegarde dans la scratch	AR1EX			R1= Di+1 A(W) = Di
	*		math stack	R2=A			R2 = Di remplace Di-1
	GOSUB	stscr	sauvegarde de ABS(Ho) =	GOSUB	AR3		X = ABS(Ni+1) -> scr.mstack
	*		ABS(N)	A=B	M		A(W) = ABS(Ni+1)
	GOSUB	clrfrc	A(W) = IP(ABS(N))	C=R1			C(W) = Di+1
	R3=A		IP(ABS(N)) = ABS(No) -> R3	GOSBVL	DV2-12		X = ABS(Ni+1) / Di+1
LOOP	?ST=0	0	option précision ?	GOSUB	ROUND		A(W) = ABS(Ni+1)/Di+1
	GOYES	PREC	oui saut au test précision	*			arrondi à 12 digits
	A=DAT1	B	A(B) = - paramètre	*			scrmstk -> ABS(Ni+1) -> R3
	*		itérations	GOTO	LOOP		
	A=A+1	B	incrémentaion				
	GOC	out	dernière itération	OUT	A=R3		A(W) = ABS(Ni+1)
	*		alors résultat	C=DAT1	W		C(S) = signe de N
	DAT1=A	B	sauve le nombre	A=C	S		A(W) = Ni+1
	*		d'itérations restantes	R3=A			R3 = Ni+1
	GONC	ITER	saute le test précision	D0=(5) (FUNCD0)			récupération de D0
PREC	GOSUB	A-1S	A(W) = -IP(ABS(N))	C=DAT0	A		
	*		ou -ABS(Ni/Di)	D0=C			
	*		X = ABS(N)	C=R1			C(W) = Di+1
	*		A(W) = ABS(N)	P=	14		
	*		X = ABS(N) - ABS(Ni/Di)	C=C-1	P		
	*		= delta	C=C-1	W		si Di+1 = 1 alors carry
	*		A(W) = ABS(delta)	P=	0		nécessaire pour STR\$00
	C=DAT1	W	C(W) = précision et signe	ST=0	1		les blancs sont supprimés
	*		de N	GONC	DIF1		Di+1 # 1 alors résultat
	C=0	S	C(W) = précision	*			avec Di+1
	P=	3	pour TEST12A	GOSUB	A2STR		Ni+1 ou N en chaine
	GOSBVL	TST12A	ABS(delta) <= précision ?	GOTO	expr		et fin
	GOC	out	oui alors résultat				

```

DIF1  A=R1          A(W) = Di+1
      GOSUB REVST  Di+1 sur la math stack ->
      *           chaîne alpha inversée
      *           -> en-tête enlevée
      C=R1         C(A) = D1 (fin de chaîne)
      RSTK=C       sauve D1 sur pile retours
      LCASC '/'    C(B) = /
      D1=D1- 2    prépare la pile à recevoir
      *           /
      DAT1=C B    / sur la pile
      A=R3         A(W) = Ni+1
      D1=D1- 16   prépare la pile à recevoir
      *           Ni+1
      GOSUB REVST Ni+1 -> pile -> chaîne
      *           alpha -> chaîne inversée
      *           -> entête enlevée
      C=RSTK       on récupère D1 (fin de
      *           chaîne
      R1=C         R1(A) = D1 (fin de chaîne)
      *           nécessaire pour ADHEAD
      GOSBVL ADHEAD ajoute l'entête
      GOSBVL REV$ renverse la chaîne
      expr GOVLNG EXPR résultat et retour au BASIC

```

```

REVST GOSUB A2STR converti A(W)
      GOSBVL REV$ chaîne inversée
      P= 0         nécessaire pour XXHEAD
      GOVLNG XXHEAD en-tête enlevée

```

```

STOfrc GOSUB STO  A(W) -> scr. math st.
clrfrfc GOSBVL CLRFRFC X = IP(X)
A=B A=B M X -> A(W)
      RTN

```

```

mp2-12 GOSBVL MP2-12 X = A(W) * C(W)
      GOTO A=B X -> A(W) et retour

```

```

AR3 A=R3 A(W) = ABS(Ni)
STO GOSBVL SPLITA A(W) -> X
stscr GOVLNG STSCR X -> scr. math st.

```

```

ROUND GOSBVL uRES12 X 15 digits -> C(W) arrondi
      *           à 12 digits
      A=C W
      GOSBVL RCSCR scr.mstack -> ABS(Ni) -> Y
      C=D M Y -> C(W)
      R3=C ABS(Ni) -> R3
      RTN

```

```

A-1S A=A-1 S positif -> négatif
      GOSBVL RCLW2 X = 2e niveau scr.mstack
      A=B M X -> A(W)
      AD GOSBVL AD2-12 X = A(W) + C(W)
      A=0 S valeur absolue
      GOTO A=B X -> A(W) et retour

```

```

A2STR DAT1=A W nombre sur la pile
      AD1EX sauve D1

```

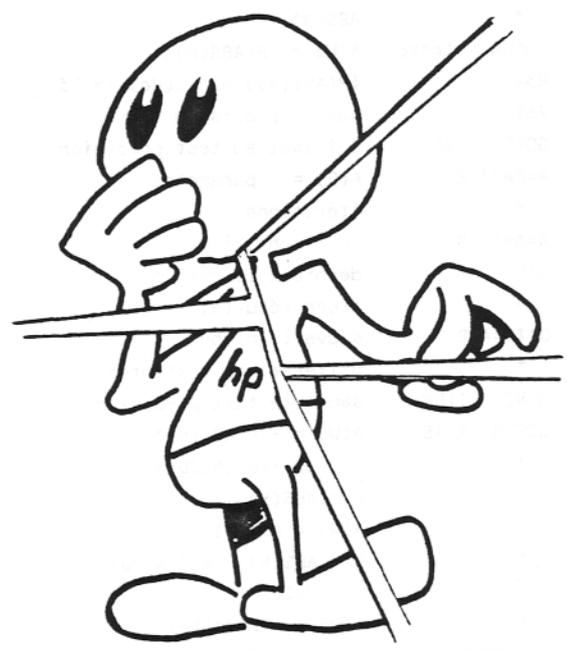
```

D1=(5) (DSPFMT)
C=DAT1 S C(S) = format courant
R4=C sauve dans R4(S)
C=0 A impose le ..
DAT1=C 1 .. format STD
AD1EX restaure D1
GOSBVL =STR$SB fait la conversion
AD1EX sauve D1
D1=(5) DSPFMT
C=R4
DAT1=C S restaure le format courant
AD1EX restaure D1
RTN
END

```

Et bien fractionnez maintenant...

Guy Toublanc (276)



## FRACSTORY

La conversion d'un nombre décimal en fraction a été abordé (côté Lex) dans l'article "Réfractaire" de ce JPC. Ce titre faisait allusion à différentes difficultés rencontrées.

Je voudrais attirer l'attention sur les solutions et leurs limites d'utilisation. En effet, sauf cas favorables, nous n'obtiendrons que des approximations car les résultats sont tributaires des arrondis successifs et du nombre de chiffres acceptés par les registres du microprocesseur.

Si Titan calcule avec des mantisses de 15 chiffres, les entrées et les sorties se font après arrondi et troncature à 12 chiffres.

Deux exemples typiques :

1/999997993279 1/RES -> 999997993274

1/999997993270 1/RES -> 999997993274

donc le douzième chiffre n'est pas fiable :

1/82646281E4 1/RES -> 826462810002

Aussi ne comptez pas qu'un algorithme miracle vous donne :

FRAC\$(1/999997993277) ->  
1/999997993277

FRAC\$(1/82646281E4) ->  
1/826462810000)

FRAC\$(999999967/82646303) ->  
999999967/82646303

Pour cette dernière donnée FRAC\$ renvoie :  
1677389/13863 pour delta = 0, car avec 12  
chiffres ces 2 fractions sont équivalentes.

Pour expérimenter et y voir plus clair je vous  
livre un tout petit fichier Basic TFRAC.

### MODE D'EMPLOI DE TFRAC

Nécessite FRACLEX.

CALL PFRAC(N)

Renvoie les différentes fractions équivalentes  
à N pour une précision de  $10^{-1}$  à  $10^{-n}$  jusqu'à  
ce que delta = 0 une pression sur une touche  
autre que [ATTN] arrête la routine et renvoie la  
dernière approximation avec émission d'un  
BEEP.

CALL AFRAC(N)

Renvoie les fractions équivalentes à N de la  
1<sup>ère</sup> itération jusqu'à ce que delta=0. Pression  
sur une touche : dito PFRAC.

### EXEMPLES

CALL PFRAC(PI)

->

p 1 22/7  
- 3 3.14

p 2 22/7  
- 3 3.14

p 3 333/106  
- 5 3.1415

p 12 1146408/364913 3.14159265359 0.00

donc affichage successivement de :

la précision imposée :

$10^{-1}$  -> p 1 à  $10^{-12}$  -> p 12

et la fraction correspondante ;

sur la 2<sup>ème</sup> ligne :

- delta : n,...E-j -> -j

et la valeur de la fraction arrondie à j-1

Après interruption affichage de tout ceci mais  
avec delta exprimé en FIX 2.

CALL AFRAC(PI)

->

a 1 3  
- 1 3.08

a 1 22/7  
- 3 3.14

a 2 333/106  
- 5 3.1415  
:  
:  
a 10 1146408/364913 ...dito PFRAC...

On voit que **FRAC\$** vient à bout de **PI** en 9 itérations alors que pour **FRC** (mot Forth de Alain Herreman) il faudrait plus de 364913 itérations !

J'espère avoir apporté quelques éclaircissements et donné un outil assez fiable.

Guy Toublanc (276)

je me suis rappelé de l'article de Pierre David. (JPC 13 et 14) sur les ordres Basic du 71. Donc, **P\$[I,I]** prend la sous-chaine **P\$** composée des caractères indiqués par les variables **I**. Comme elles ont la même valeur on se déplace bien de lettre en lettre à chaque itération de la boucle.

Enfin, en finale, on donne le résultat en imbriquant les deux variables (**L\$** et **N**) dans une phrase.

En espérant que cet article permettra aux débutants de progresser.

Alain Gillet (199)

---

## POUR DEBUTER EN BOUCLE ET TEST

Soit à résoudre ce problème : on veut connaître le nombre de lettres (espaces compris) dans une phrase ou un mot.

- La variable de comparaison est **L\$**. Elle contient la lettre que l'on veut compter.

- La boucle est construite avec les ordres **FOR ... TO** et **NEXT**, dont le compteur est la variable **D**, la valeur de celle-ci étant définie par l'ordre **LEN(P\$)**. **LEN** est l'ordre permettant de compter le nombre de caractères, **P\$** est la variable contenant la phrase. La longueur de **P\$** est au maximum de 32 caractères. Pour prendre en compte une phrase dont le nombre de caractères est supérieur à 32, on ajoute en début de programme l'ordre **DIM**. On ajoute en début de programme l'ordre suivant : **DIM P\$[X]**. **X** représentant le nombre de caractères que l'on veut allouer à la variable **P\$**.

- La partie du programme permettant la comparaison est construite avec **IF ... THEN**. Le compteur est la variable **N** suivant **THEN** sous la forme **N=N+1**. **IF**, lui, est suivi de la variable **P\$[I,I]**. En effet, pour prendre caractère par caractère, il faut prendre (à défaut des ordres **MID\$, LEFT\$, RIGHT\$**) **P\$[I,I]**, ce qui m'a posé un problème, mais

---

## AGENDA

Une désignation générale du HP-71 est : ordinateur de poche. Mais vous avouerez tout de même qu'une fois que le HP-71 se trouve dans votre poche, il n'y a plus guère de place pour autre chose. Alors, plutôt que d'agrandir vos poches ou de vous en rajouter dans le dos, je vous propose un petit programme qui fera de la place dans votre poche. Non, votre HP-71 ne sera pas transformé en portefeuille (bien que de temps à autre j'aimerais qu'il me distribuât quelques billets). Je ne crois pas non plus qu'un policier vous demandant vos papiers (cela se fait beaucoup ces jours-ci) apprécierait si vous lanciez **RUN IDENTITE** la machine affichant vos nom, adresse en certifiant que ces informations sont conformes à votre carte d'identité. Je vous propose simplement un programme qui remplacera votre agenda sur une durée de quinze jours. Ce programme est très pratique, presque utile !

En plus d'être utile ce programme comporte autre chose d'assez exceptionnel : il utilise un Lex ! (le Lex **DATELEX** de François Le Grand dans le JPC 31 page 35). Vous avouerez que nombreux sont les Lex dans JPC et peu nombreux sont les programmes en Basic qui les utilisent.

En lançant ce programme, le 71 vous demande un jour : si le jour que vous désirez est un des sept suivants tapez simplement le jour ("mercredi" par exemple), si ce jour n'appartient pas aux sept jours qui suivent, tapez en plus "en 8" ("mercredi en 8" par exemple). Si le jour qui vous intéresse est le jour même, tapez simplement [ENDLINE]. Ceci étant fait, un point d'interrogation s'affiche (si vous voulez mieux comme affichage je vous laisse faire), vous avez alors trois possibilités :

- vous désirez rentrer un rendez-vous, alors tapez simplement votre texte puis [ENDLINE].
- vous voulez consulter le programme du jour choisi, tapez directement [ENDLINE], alors s'affichent, un par un, vos rendez-vous (vous tapez sur [ENDLINE] pour passer au suivant) si vous ne désirez pas tous les faire défiler tapez sur la touche [F1].
- vous voulez annuler un rendez-vous, alors tapez **annulation** : vos rendez-vous s'affichent avec des numéros de ligne. A la fin la machine vous demande Annulation ligne n, si vous ne désirez plus rien annuler, tapez [0]. Comme précédemment vous pouvez éviter de tout faire défiler en tapant sur [F1].

Si vous ne voulez plus jouer avec votre agenda tapez fin à la question jour:. Par jour, le texte est limité à 255 caractères. Vous pouvez très facilement changer cette limite, mais si vous dépassez la limite fixée vous aurez le droit à Journée surchargée !.

Pour utiliser ce programme AGENDA, il vous faut créer un fichier AGENDAF :

```
CREATE DATA AGENDAF, 16
puis stocker dans le premier enregistrement la
date du jour où vous rentrez ce programme :
ASSIGN #1 TO AGENDAF
```

```
puis :
PRINT #1,0;jj.mmaaaa
et enfin remplir de blancs les autres
enregistrements en exécutant :
FOR I=1 TO 15 @
  PRINT #1,I;" " @
NEXT I
et pour terminer : ASSIGN #1 TO *
```

Je pense que ce programme est assez simple pour se passer d'explication. N'hésitez pas à le modifier à votre convenance.

Alain Herreman (200)

## DESSINE-MOI UN AUTRE MOUTON...

Vous vous souvenez de GRAPHLEX et du programme en Basic qui l'accompagnait ? Bon, d'accord, c'était en Juin (JPC 35 page 38), autant dire déjà assez loin... N'empêche, ce programme est vraiment très pratique (merci à son auteur, Pierre David !), sauf peut-être pour la gestion des données. En effet, ce sous programme ne gère pas les données, et il est rapidement apparu nécessaire de pouvoir :

- sauvegarder ces données dans un fichier séparé,
- éditer ce fichier de façon interactive (c'est à dire ajouter ou retirer des données),
- enfin, échanger ces données avec d'autres logiciels, statistiques par exemple, qui sont "beaucoup moins doués du côté des graphiques"...

Le standard en la matière étant "HPAF", c'est bien entendu lui qui a été choisi. Rappelons qu'il est utilisé par les modules Finance, Ajustement de courbes, Circuits et Query71, qu'il a déjà été décrit dans ces colonnes (par Michel WEIL, voir JPC 30 page 12), et que son usage est fortement recommandé par HP. Il existe, comme agrément supplémentaire, d'excellents utilitaires de manipulation ou reconditionnement de ces fichiers (cf. Bibliothèque de Corvallis : *Data Management Pac* et *HPAF Utilities*), qui peuvent tenir au large dans une Eprom 32 Ko...

Rappelons aussi qu'un fichier HPAF est un fichier Data, donc utilisable avec Basic, auquel on a ajouté un en-tête spécial (qui permet certains contrôles et reconnaissances automatiques, et donne la disposition des zones alpha et numériques dans la ligne, la longueur du fichier, etc.), plus à la fin du fichier un "bloc descripteur" contenant des animaux un peu particuliers, les "Tags" (Tagex Octivorus V.). Ce bloc est d'ailleurs facultatif, nous l'utiliserons pour logger titre et légendes...

Le programme proposé contient donc un "éditeur", qui permet de créer, rapatrier ou modifier un fichier HPAF au format voulu, et utilise des "menus" à dessein très proches de ceux de l'éditeur contenu dans le module d'ajustement de courbes. Il contient donc les blocs KBD (saisie au clavier), LOAD

(chargement et modifications éventuelles) et SAVE (transformation de la matrice de données en fichier HPAF), plus une routine pour les légendes et un bloc de listage de contrôle (PRINT). Il peut traiter des fichiers contenant ou non un "descriptor block", mais, travaillant sur une matrice de données, il ignore les zones de type chaîne. En revanche, il n'a pas besoin de DATALEX pour insérer ou supprimer des enregistrements ou les redimensionner.

La structure du fichier est la suivante (une ligne de données par année) :

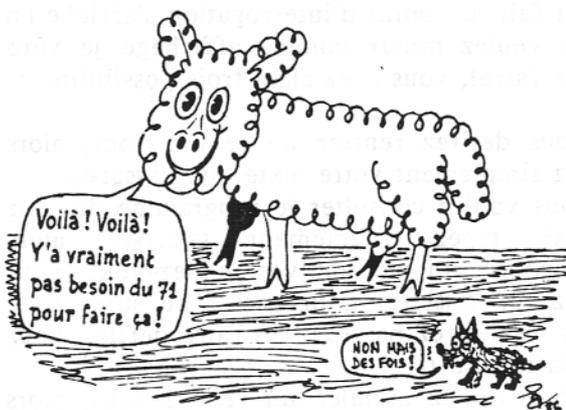
No.enreg.	Contenu
0	HPAFNNN...(B fois, nombre de 'barres')
1	<A, nombre d'années>
2	<A+3, position du 1er 'Tag'>
3 à A+2	données (une ligne par année, ... une zone par 'barre')
	Début du 'Descriptor Block', contenant:
A+3	Tag "TITLE"
A+4 à	Tags "LEGENDS" (un par ligne)
A+B+3	

A la suite de l'éditeur, on retrouve le sous-programme GRAPH de Pierre David (avec les mêmes noms de variables et numéros de lignes), toujours limité à 6 types de barres (voir son article si vous désirez en mettre davantage, ce que l'éditeur permet sans modification), mais avec un nombre quelconque d'années.

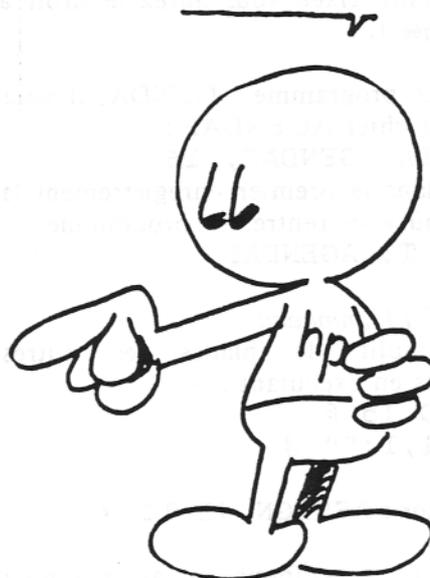
L'exemple joint montre que l'on peut faire un usage très professionnel et sérieux de ce logiciel (analyse de l'évolution comparée du trafic annuel SNCF pour chacune des deux classes). Pour ma part, je rêve de pouvoir un jour disposer (comme on approche du bi-centenaire de 1789, l'heure est aux cahiers de doléances...) de barres empilées, de barres allant de part et d'autre de zéro, voire d'un "super-Lex graphique" permettant d'utiliser la 2225B pour tracer des courbes (pour le moment, je fais de la recopie d'écran graphique avec la Pac Screen). Mais ça nécessitera une reprise en profondeur de la routine GRAPH et du programme, à tout le moins...

A bientôt pour la suite, donc...

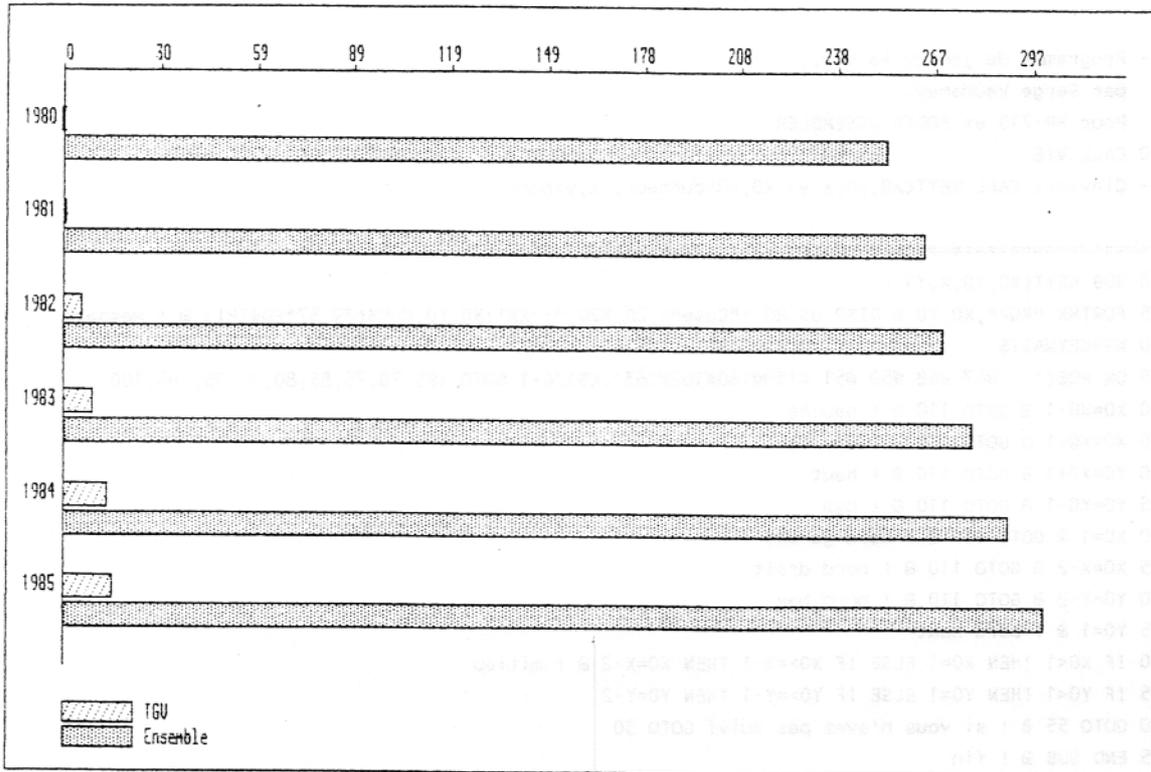
Eric Gengoux (108)



EST-CE QU'ON PEUT DESSINER UN NIBBLE ?

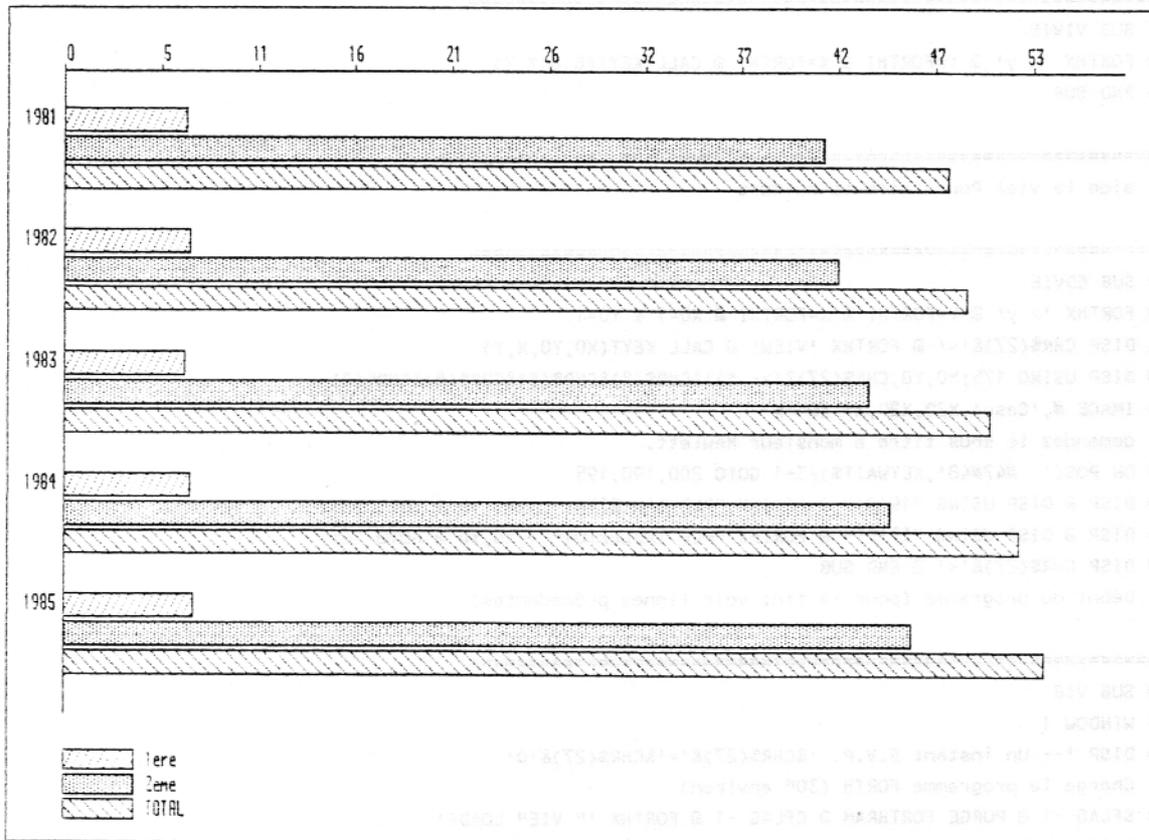


### Gdes Lignes



Millions de voyageurs transportés

### TRAFIC PAR CLASSE



Résultats en milliards de UK

Programme "VIEB" (Jeu de la Vie, nécessite FORTH + Lex KEYWAIT)

- Programme de jeu de la vie.  
par Serge Vaudenay.  
Pour HP-71B et FORTH ASSEMBLER
- ```
40 CALL VIE
```
- Clavier: CALL KEYT(x0,y0,x,y) x0,y0=curseur; x,y=bord.

```
=====
50 SUB KEYT(X0,Y0,X,Y)
55 FORTHX "RGP",X0,Y0 @ DISP USING "'Case"X,2D,X2D,":'XK';X0,Y0,CHR$(79-37*FORTH) @ ! message
60 K$=KEYWAIT$
65 ON POS(' #47 #48 #50 #51 #159#160#162#163',K$)/4+1 GOTO 125,70,75,85,80,90,95,105,100
70 X0=X0-1 @ GOTO 110 @ ! gauche
75 X0=X0+1 @ GOTO 110 @ ! droite
80 Y0=Y0+1 @ GOTO 110 @ ! haut
85 Y0=Y0-1 @ GOTO 110 @ ! bas
90 X0=1 @ GOTO 110 @ ! bord gauche
95 X0=X-2 @ GOTO 110 @ ! bord droit
100 Y0=Y-2 @ GOTO 110 @ ! bord bas
105 Y0=1 @ ! bord haut
110 IF X0<1 THEN X0=1 ELSE IF X0>=X-1 THEN X0=X-2 @ ! milieu
115 IF Y0<1 THEN Y0=1 ELSE IF Y0>=Y-1 THEN Y0=Y-2
120 GOTO 55 @ ! si vous n'avez pas suivi GOTO 30
125 END SUB @ ! fin
=====
```

- vive la vie! Si vous voulez examiner votre culture, CALL VIVIE

```
=====
135 SUB VIVIE
140 FORTHX 'x y' @ Y=FORTH) @ X=FORTH) @ CALL KEYT(0,0,X,Y)
145 END SUB
=====
```

- aide la vie! Pour faire la culture

```
=====
155 SUB EDVIE
160 FORTHX 'x y' @ Y=FORTH) @ X=FORTH) @ X0=1 @ Y0=1
165 DISP CHR$(27)&'<' @ FORTHX 'VIEW' @ CALL KEYT(X0,Y0,X,Y)
170 DISP USING 175;X0,Y0,CHR$(27)&'>- *')&CHR$(8)&CHR$(8)&CHR$(8)&CHR$(8)
175 IMAGE #,'Case',X2D,X2D,X' (O 'K
- demandez le sous titre a monsieur Hewlett.
185 ON POS(' #47#48',KEYWAIT$)/3+1 GOTO 200,190,195
190 DISP @ DISP USING "19'0'" @ FORTHX "PLP TABLE2TO1",0,X0,Y0 @ GOTO 165
195 DISP @ DISP USING "19'*'" @ FORTHX "PLP TABLE2TO1",1,X0,Y0 @ GOTO 165
200 DISP CHR$(27)&'<' @ END SUB
=====
```

- Début du programme (pour la fin: voir lignes précédantes)

```
=====
210 SUB VIE
215 WINDOW 1
220 DISP '--- Un instant S.V.P. '&CHR$(27)&'>'&CHR$(27)&'Q'
- Charge le programme FORTH (30" environ)
230 SFLAG -1 @ PURGE FORTH RAM @ CFLAG -1 @ FORTHX "' VIE" LOADF'
235 FORTHX 'NEANT'
=====
```

```

240 WINDOW 1 @ DISP CHR$(27)&'<&CHR$(27)&'R-- Edition du tableau.'
245 DISP @ WINDOW 1,19
  - Edition
255 CALL EDVIE @ FORTHX '?FIN' @ IF FORTH THEN 300
260 DISP CHR$(27)&'<-- Tableau:' @ FORTHX 'VIEW' @ WAIT 1
  - Vérification
270 DISP CHR$(27)&'>&CHR$(27)&'R? O.K. ( O - N )&CHR$(8)&CHR$(8)&CHR$(8)&CHR$(8)&CHR$(8)
275 ON POS(' #47#48',KEYWAIT$)/3+1 GOTO 320,285,305
  - Calcul de la génération suivante
285 DISP '-- Calcul... ' &CHR$(27)&'>&CHR$(27)&'Q' @ FORTHX 'T+1 VIEW'
  - Teste les débordements
295 FORTHX '?FIN' @ IF NOT FORTH THEN 260
300 DISP '-- Expansion...' &CHR$(27)&'<' @ WAIT 2
305 DISP CHR$(27)&'>&CHR$(27)&'R? Encore ( O - N )&CHR$(8)&CHR$(8)&CHR$(8)&CHR$(8)&CHR$(8)
310 ON POS(' #47#48',KEYWAIT$)/3+1 GOTO 320,240,320
  - fini les gars!
320 WINDOW 1 @ DISP CHR$(27)&'<Fin.' @ PURGE FORTH RAM @ END SUB

```

\*\*\*\*\*

Programme "TFRAC" (Exemples d'utilisation de FRAC\$, Necessite FRACLEX)

```

=====
10 SUB AFRAC(N) @ CALL FRA(N,0) @ END
=====
20 SUB PFRAC(N) @ CALL FRA(N,1) @ END
=====
30 SUB FRA(N,O) @ J=0 @ IF O THEN I=1 ELSE I=-1
40 J=J+I @ F$=FRAC$(N,J) @ V=VAL(F$) @ D=V-N
50 IF NOT D OR KEYDOWN THEN 80
60 GOSUB 90 @ DISP @ F=EXPONENT(D) @ DISP F;" "; @ FIX ABS(F)-1 @ DISP V @ STD
70 WAIT 1 @ GOTO 40
80 BEEP @ GOSUB 90 @ DISP " ";V; @ FIX 2 @ DISP " ";D @ STD @ END
90 IF O THEN DISP "p";J; ELSE DISP "a";-J;
100 DISP " ";F$; @ WAIT 1 @ RETURN

```

\*\*\*\*\*

Programme "COMPTEUR" (Calcule le nombre d'occurrences de lettre dans une phrase)

```

- COMPTEUR DE LETTRE
GILLET ALAIN 4/2/86
P$ : Variable contenant la phrase
L$ : Variable contenant la lettre à compter
D : Variable contenant la longueur de la phrase
N : Variable contenant le nombre de lettre
I : Incrément pour la boucle
80 PRINT "COMPTEUR D'UNE LETTRE"
90 WAIT .8

```

```

- On demande la phrase
110 INPUT "Donner une phrase: ";P$
- On demande la lettre à compter
130 INPUT "Lettre a compter? ";L$
- On prend la longueur de la phrase
150 D=LEN(P$)
- On initialise la variable du compteur
170 N=0
- Début de la boucle
190 FOR I=1 TO D
- Si la lettre est celle compter alors on la comptabilise dans N
210 IF P$[I,1]=L$ THEN N=N+1
- Si non ,et même si,alors on prend la lettre suivant
230 NEXT I
- Quand I=D alors on affiche le nombre de lettre
250 PRINT "Nombre de ";L$;" contenu dans la phrase: ";N
260 END

```

\*\*\*\*\*

Programme "AGENDA" (Agenda en Basic, necessite DATELEX)

```

10 DESTROY ALL @ DELAY 0,.3 @ ASSIGN #1 TO AGENDAF @ DIM A$[256] @ DISP "Prgm agenda sur 15 j."
20 READ #1,0;D1
30 D0=VAL( DATE$[7] )+VAL( DATE$[4,5] )/100+(VAL( DATE$[1,2] )+1900)/1000000
40 D=DDAYS(D1,D0)
50 IF D<=15 THEN 70
60 FOR I=1 TO 15 @ PRINT #1,I;" " @ NEXT I @ GOTO 90
70 FOR I=D+1 TO 15 @ READ #1,I;A$ @ PRINT #1,I-D;A$ @ NEXT I
80 FOR I=16-D TO 15 @ PRINT #1,I;" " @ NEXT I
90 PRINT #1,0;D0
100 LC ON @ DIM B$[96]
110 INPUT "Jour: ";D$ @ X=POS(D$," en 8") @ IF D$="" THEN D$=DOW$(D0) @ N=1 @ GOTO 160
120 N=(POS("dimanchelundi***mardi***mercredijeudi***vendredisamedi",D$[1,X-1+(X=0)*8]) - 1)/8
130 IF D$="fin" THEN ASSIGN #1 TO * @ LC OFF @ OFF @ END
140 IF N<0 THEN 110
150 N=N-DOW(D0)+((DOW(D0)+1>N)+(X#0))*7+1
160 LINPUT B$ @ READ #1,N;A$ @ IF B$="" OR B$="annulation" THEN 190
170 IF LEN(A$)+LEN(B$)>254 THEN BEEP @ DISP "Journee surchargee!" @ WAIT 2 @ GOTO 110
180 A$=A$&B$&"@" @ PRINT #1,N;A$ @ GOTO 110
190 RESTORE @ READ #1,0;D @ D=DATE+(D,N-1)
200 FOR I=1 TO IP(100*FP(D)) @ READ E$ @ NEXT I
210 DISP D$[1,X+(X=0)*8];IP(D);E$&"." @ WAIT 1 @ X=0 @ I=0
220 I=I+1 @ Y=POS(A$,"@",X+1) @ IF NOT Y THEN 270
230 DISP I;" ";A$[1+X,Y-1]
240 IF NOT KEYDOWN THEN 240
250 IF KEYDOWN("#18") THEN 270
260 X=Y @ DISP @ GOTO 220
270 DISP @ IF B$="" THEN 110 ELSE INPUT "Annulation ligne n";J @ X=0 @ Y=0
280 FOR I=1 TO J @ X=Y @ Y=POS(A$,"@",X+1) @ NEXT I @ A$=A$[1,X]&A$[Y+1] @ PRINT #1,N;A$
290 GOTO 110

```

```

=====
300 DATA janvier,fevrier,mars,avril,mai,juin,juillet,aout,septembre,octobre,novembre,decembre

```

\*\*\*\*\*

Programme "BARGRAF2" (Gestion des donnees HPAF + graphique)

- BARGRAF2 gestion de donnees et edition d'histogrammes sur ThinkJet.

Utilise GRAPHLEX, contient le sous-programme GRAPH (JPC35 p47).

(c)Eric GENGOUX 05\01\1987

40 DIM T\$(30)

50 DISP 'Data: Kbd Load'

60 K\$=UPRC\$(KEYWAIT\$) @ IF NOT POS('KL',K\$) THEN BEEP @ GOTO 50

70 ON POS('KL',K\$) GOSUB 'KBD','LOAD'

80 DISP 'Working...' @ BEEP 400,.5

100 DISP 'Save Print Graph Exit'

110 K\$=UPRC\$(KEYWAIT\$) @ IF NOT POS('SPGE',K\$) THEN BEEP @ GOTO 100

120 ON POS('SPGE',K\$) GOSUB 'SAVE','PRINT','GRAF','END'

130 GOTO 100

=====  
140 'END': DISP 'End of Job' @ BEEP 400,.5 @ BEEP 300,.5 @ END

=====  
142 'GRAF': CALL GRAPH(T(,),A,B,X,X\$(1),T\$,0) @ RETURN

=====  
160 'KBD': GOSUB 'LEGENDS'

170 DISP 'Entrée valeurs' @ BEEP 400,.5 @ WAIT .5

180 FOR I=X TO X+A-1 STEP 1

190 I2=I-X+1

200 FOR J=1 TO B

210 L\$="T("&STR\$(I)&","&STR\$(J)&")="

220 DISP L\$; @ INPUT T(I2,J)

230 NEXT J

240 NEXT I

250 RETURN

=====  
270 'LOAD': INPUT 'LOAD: File name? ';F\$

280 IF LEN(K\$)>8 THEN BEEP @ GOTO 270

290 ASSIGN #1 TO F\$

300 READ #1,0;H\$

310 IF H\$[1,4]# 'HPAF' THEN ASSIGN #1 TO \* @ DISP 'File not HPAF' @ POP @ END

320 B=LEN(H\$)-4

330 READ #1,1;A @ DIM T(A,B),X\$(B)

340 READ #1,2;A9 @ IF A9=0 THEN GOSUB 'LEGENDS' @ GOTO 410

350 READ #1,A+3;Z\$,Z,T\$ ! Titre

360 READ #1,A+4;Z\$,Z,X\$(1) ! Legendes

370 FOR I=2 TO B

380 READ #1,A+I+3;X\$(I) ! Legendes (suite)

390 NEXT I

400 DIM T2(B)

410 FOR I=1 TO A ! Valeurs

420 READ #1,I+2;T2( )

430 FOR J=1 TO B

440 T(I,J)=T2(J)

450 NEXT J

460 NEXT I

```

480 DISP 'Add new values Y/N'
490 K$=UPRC$(KEYWAIT$) @ IF NOT POS('YN',K$) THEN BEEP @ GOTO 490
500 IF K$='N' THEN GOTO 600
510 INPUT 'How many years?';D
520 DIM T(A+D,B),T2(B)
530 FOR I=1 TO D
540 FOR J=1 TO B
550 DISP 'T('&STR$(X+A+I-1)&','&STR$(J)&')=';
560 INPUT T(A+I,J)
570 NEXT J
580 NEXT I
590 A=A+D @ GOSUB 'SAVE'
600 RETURN

```

```

=====
620 'SAVE': ON ERROR GOTO 660
630 INPUT 'SAVE: File name?';F$
640 IF LEN(F$)>8 THEN BEEP @ GOTO 630
650 CREATE DATA F$,A+10,8*B+15 @ GOTO 690
660 OFF ERROR @ BEEP @ DISP 'File exists. Kill Y/N'
670 K$=UPRC$(KEYWAIT$) @ IF NOT POS('YN',K$) THEN BEEP @ GOTO 670
680 IF K$='Y' THEN PURGE F$ @ GOTO 650
690 OFF ERROR @ ASSIGN #1 TO F$ @ DIM T2(B)
700 H$='HPAF' @ FOR I=1 TO B @ H$=H$&'N' @ NEXT I
710 PRINT #1,0;H$
720 PRINT #1,1;A @ PRINT #1,2;A+3
730 FOR I=1 TO A
740 FOR J=1 TO B
750 T2(J)=T(I,J)
760 NEXT J
770 PRINT #1,I+2;T2( )
780 NEXT I
790 PRINT #1,A+3;'TITLES',1,T$
800 PRINT #1,A+4;'FIELDS',B,X$(1)
810 FOR I=2 TO B @ PRINT #1,A+I+3;X$(I) @ NEXT I
820 ASSIGN #1 TO *
830 RETURN

```

```

=====
850 'PRINT': ! Modifée 05/1/87
860 DIM Z0$(120),Z1$(120)
861 PRINT CHR$(27)&'&d@'
865 FOR I=1 TO 120 @ Z0$(I,I)=' ' @ NEXT I
870 Z1$=Z0$
875 FOR I=1 TO B @ Z1$[8*(I-1)+1,8*I]=X$(I) @ NEXT I
880 PRINT 'Année ' & Z1$
885 FOR I=1 TO A
890 Z1$=Z0$ @ Z1$[1,4]=STR$(X+I-1)
895 FOR J=1 TO B @ Z1$[8*J,8*(J+1)]=STR$(T(I,J)) @ NEXT J
900 PRINT Z1$
910 NEXT I
920 RETURN

```

```

=====
940 'LEGENDS': INPUT 'Title: ';T$

```

```

950 INPUT 'Nb. of years ' ; A
960 INPUT '1st year: ' , '1980' ; X
970 INPUT 'Nb. of bars' ; B @ IF B > 6 THEN BEEP @ GOTO 970
980 DIM T(A,B), X$(B) [8]
990 FOR I=1 TO B
1000 INPUT 'Legend ' , STR$(I) ; X$(I)
1010 NEXT I
1020 RETURN

```

=====

```

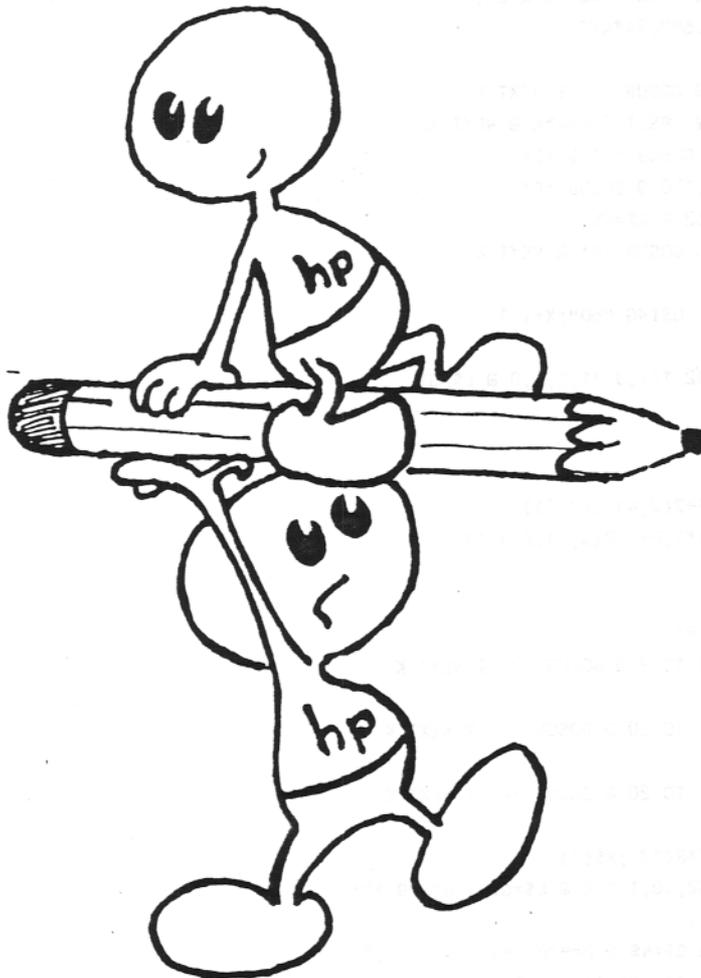
1040 SUB GRAPH(T(,), A, B, X, X$(,), T$, M)
1050 N=6 @ INTEGER Z(N,4) @ FOR J=1 TO N @ FOR K=1 TO 4 @ READ Z(J,K) @ NEXT K @ NEXT J
1060 DIM G$[80], K$[80], L$[80], N$[80]
1070 FOR I=1 TO 20 @ N$=N$&CHR$(0)&CHR$(0)&CHR$(0)&CHR$(0) @ NEXT I
1080 PWIDTH INF @ ENDLINE "" @ PRINT CHR$(27)&"R640S"&CHR$(27)&"&kOS"
1090 IF M THEN 1110
1100 M=-INF @ FOR I=1 TO A @ FOR J=1 TO B @ M=MAX(T(I,J), M) @ NEXT J @ NEXT I
1110 Y=M/.9 @ F=608/Y
1120 PRINT TAB(40-LEN(T$)/2); CHR$(14)&T$&CHR$(15)&CHR$(13)&CHR$(10)&CHR$(10)
1130 PRINT CHR$(27)&"&k2S"
1140 G$=N$ @ GLINE 1,639,1,1,0 @ GOSUB 'P'
1150 G$=N$ @ FOR K=1 TO 20 @ GOSUB 'P' @ NEXT K
1160 FOR K=0 TO 10 @ PRINT TAB(12*K+3);
1170 PRINT USING "#,6D"; 54*K/F
1180 NEXT K
1190 FOR K=1 TO 12 @ GOSUB 'P' @ NEXT K
1200 FOR K=0 TO 10 @ GPSET 32+54*K @ NEXT K
1210 FOR K=1 TO 3 @ GOSUB 'P' @ NEXT K
1220 GLINE 32,590,1,1,0 @ GOSUB 'P'
1230 G$=N$ @ GPSET 32 @ K$=G$
1240 FOR K=1 TO 20 @ GOSUB 'P' @ NEXT K
1250 FOR I=1 TO A
1260 IF X THEN PRINT USING "6D"; X+I-1
1270 FOR J=1 TO B
1280 G$=K$ @ GLINE 32, T(I,J)*F, 1, 1, 0 @ L$=G$
1290 GOSUB 'P'
1300 P=IP(RND*Z(J,3))
1310 FOR K=1 TO 12
1320 G$=N$ @ P=MOD(P+Z(J,4), Z(J,3))
1330 GLINE 32, T(I,J)*F, P+1, Z(J,2), Z(J,1)
1340 GOSUB 'P'
1350 NEXT K
1360 G$=L$ @ GOSUB 'P'
1370 G$=K$ @ FOR K=1 TO 2 @ GOSUB 'P' @ NEXT K
1380 NEXT J
1390 G$=K$ @ FOR K=1 TO 20 @ GOSUB 'P' @ NEXT K
1400 NEXT I
1410 G$=N$ @ FOR K=1 TO 20 @ GOSUB 'P' @ NEXT K
1420 FOR J=1 TO B
1430 PRINT @ PRINT TAB(18); X$(J)
1440 G$=N$ @ GLINE 32,40,1,1,0 @ L$=G$ @ GOSUB 'P'
1450 P=IP(RND*Z(J,3))
1460 FOR K=1 TO 10 @ G$=N$ @ P=MOD(P+Z(J,4), Z(J,3))
1470 GLINE 32,40, P+1, Z(J,2), Z(J,1) @ GOSUB 'P'
1480 NEXT K
1490 G$=L$ @ GOSUB 'P'
1500 G$=N$ @ FOR K=1 TO 2 @ GOSUB 'P' @ NEXT K

```

```
1510 NEXT J
1520 G$=N$ @ FOR K=1 TO 10 @ GOSUB 'P' @ NEXT K
1530 G$=N$ @ GLINE 1,639,1,1,0 @ GOSUB 'P'
1540 ENDLIN @ PRINT @ PRINT @ END
```

```
=====
1550 'P': GPSET 1 @ GPSET 640 @ PRINT CHR$(27)"*b80w"&G$ @ RETURN
- Table de parametres de motifs
```

```
=====
1570 DATA 8,1,9,4
1580 DATA 3,1,4,2
1590 DATA 6,2,7,8
1600 DATA 6,2,7,6
1610 DATA 1,1,2,1
1620 DATA 9,1,5,4
```



## LE COIN DES LHEX

Comme de coutume, cette rubrique contient la liste des codes hexadécimaux des fichiers Lex parus ce mois-ci.

Rappelons ce qu'est un fichier Lex : c'est un programme pour le HP71, en assembleur, qui apporte de nouvelles fonctions. Celles-ci sont utilisables directement, ou dans des programmes Basic.

Pour bénéficier de ces nouvelles fonctions, vous n'avez pas besoin de programmer vous-même en assembleur, ni de posséder un module Forth/Assembleur.

Il suffit de recopier le petit programme basic "MAKELEX" ci-dessous, de le lancer et de recopier les codes du fichier Lex désiré. Quand vous avez fini, les nouvelles fonctions sont accessibles, après avoir éteint et rallumé votre HP71.

Si l'erreur "Erreur de somme" apparaît, vérifiez la ligne que vous avez introduite.

Vous trouverez donc, outre le Lex FRACLEX de la rubrique assembleur, le Lex CHARLEX nécessaire à la rédaction de votre article (voir "Ah ! Vous écrivez !"), et les Lex utilisés par les programmes de la rubrique "Basic".

P. David (37)

### CHARLEX

KEYWAIT KEYWAIT\$ 82001

FRACLEX FRAC\$ 225088

DATELEX DATE+ 225051 DDAYS 225052 DMY 225053  
DOW\$ 225054 DOW 225055 MDY 225056

GRAPHLEX GLINE 225078 GPSET 225079

```
10 CALL MLEX @ SUB MLEX @ SFLAG -1 @ PURGE AH @ INPUT "Nb. d'octets: ";N @ LC OFF
20 CREATE DATA AH,1,N-4 @ A=HTD(ADDR$("AH")) @ B=A @ GOSUB 130
30 Q=1 @ X=0 @ INPUT "000: ",P$;A$ @ C$=A$ @ S=0 @ GOSUB 90
40 Q=2 @ X=1 @ GOSUB 80 @ A$=A$&C$ @ A=A+37 @ N=N*2+37 @ Q=3 @ SFLAG 5 @ FOR X=2 TO N DIV 16-1
50 GOSUB 80 @ C$=C$[5*SFLAG(5)+1] @ POKE DTH$(A),C$ @ A=A+16-5*SFLAG(5,0) @ NEXT X @ Q=4
60 DISP DTH$(X)[3]; @ INPUT ": ",P$[1,MOD(N,16)];C$ @ GOSUB 90
70 POKE DTH$(A),C$ @ POKE DTH$(B),A$ @ CFLAG -1 @ END
80 DISP DTH$(X)[3]; @ INPUT ": ",P$;C$
90 DISP DTH$(X)[3]; @ INPUT " sm ","---";D$
100 M=S @ FOR Z=1 TO LEN(C$) @ M=NUM(C$[Z])+M+1 @ NEXT Z
110 IF D$=DTH$(MOD(M,4096))[3] THEN GOSUB 130 @ S=M @ RETURN
120 DISP "Erreur de somme" @ BEEP @ P$=C$ @ POP @ ON Q GOTO 30,40,50,60
130 P$="-----" @ RETURN
```

CHARLEX ID#E1 624 octets

0123456789ABCDEF sm

000: 34841425C4548502 35E  
 001: 802E009022512078 6B1  
 002: 5E4001E000000000 9F5  
 003: FE0000000800001F D4F  
 004: F31BF961400032BF 0E2  
 005: 38F14A11DB10AD23 47C  
 006: 07D532BF8FD7911 82F  
 007: 11AD754D7A101743 BB2  
 008: 11014D1CB15D0000 F1D  
 009: 71450375FF864834 29A  
 00A: 5655581008355654 5F1  
 00B: 5810002455565870 93E  
 00C: 0026555658700836 C90  
 00D: 5556581008364545 FE6  
 00E: 4A30000A49724000 339  
 00F: 0808094A2C180814 6A2  
 010: A464242008355455 9FC  
 011: 581000054C714000 D42  
 012: 0C3142404C700832 09E  
 013: 41414A70002078A0 3F6  
 014: 2F30000000000000 721  
 015: 0000000000000000 A31  
 016: 0000000000000000 D41  
 017: 0000000000000000 051  
 018: 0000000000000000 361  
 019: 0000000000000000 671  
 01A: 0000000000000000 981  
 01B: 0000000000000000 C91  
 01C: 0000000000000000 FA1  
 01D: 0000000000000000 2B1  
 01E: 0000000000000000 5C1  
 01F: 0000000000000000 8D1  
 020: 0000000000000000 BE1  
 021: 000000000000080C F0C  
 022: 1A28080008080A2C 276  
 023: 180008040E340800 5BF  
 024: 08001E3018000000 8F9  
 025: 0000000000000000 C09  
 026: 0000000000000000 F19  
 027: 0000000000000000 229  
 028: 0201000000010200 53F  
 029: 0000000201020000 854  
 02A: 0001000100000002 B68  
 02B: 0102010000000000 E7C  
 02C: 0000000000000000 18C  
 02D: 045E755142400101 4D8  
 02E: 0101010000000000 7EB  
 02F: 0000000000000000 AFB  
 030: 0000070507000000 E1E  
 031: 00000000083444C4 15C  
 032: 44400D7901112D70 4BC  
 033: 050D750509700000 806  
 034: 0D70000000384540 B49  
 035: 4020014E322E3140 E9D

036: 084E794142400000 1ED  
 037: 00000000002E4559 52B  
 038: 3200000000000000 840  
 039: 0000000000000026 B58  
 03A: 5556587008365556 EB7  
 03B: 5810083645464830 208  
 03C: 0832414248700024 549  
 03D: 5655587008345655 8A6  
 03E: 5810083446454830 BF5  
 03F: 0C3042414C700024 F4A  
 040: 5556587008355654 2A7  
 041: 5810083546444830 5F6  
 042: 0C3142404C700025 94C  
 043: 5455587008355455 CA6  
 044: 5810083544454830 FF4  
 045: 0C3140414C700875 356  
 046: 14141870000A4972 6A7  
 047: 40000E3159454E30 A07  
 048: 0C7A0F7949400024 D7F  
 049: 5554587000084A71 0DB  
 04A: 40000C523A262D10 43C  
 04B: 0424587458400875 793  
 04C: 1415187000094A70 AE3  
 04D: 4000083544454830 E27  
 04E: 0C3140414C300C74 18F  
 04F: 5655545000054C71 4E6  
 050: 40000 5DF

KEYWAIT ID#52 55 octets

0123456789ABCDEF sm

000: B454957514944502 366  
 001: 802E000122512078 6B1  
 002: 3700025101000000 9D4  
 003: F710000000000000 D02  
 004: 0E1000FFB4549575 08D  
 005: 14944542101FF001 3ED  
 006: 361081371098F2C6 75D  
 007: 0045111913511813 A99  
 008: 48D8ACA18F127006 E35  
 009: BDF F04

✓ FRACLEX ID#E1 325 octets

0123456789ABCDEF sm

000: 64251434C4548502 35C  
 001: 802E000122512078 6A7  
 002: F82001E858500000 A07  
 003: F710000000000000 D35  
 004: 0A1000F964251434 093  
 005: 42851FF88121321B 40F  
 006: BB8F2140A4EA4EAF 7E3  
 007: 210A2180CE109108 B4D  
 008: 8404B28FDF8E017F F04  
 009: 94850850AC093850 278  
 00A: 810810810AE6BB8A 60C

00B: FDF8E096E5132994 9BE  
 00C: B3232099550A32AC D40  
 00D: 61557AC076517951 0B5  
 00E: 743110386011148B 411  
 00F: 644E21495917A511 782  
 010: 577AC2238F674D04 B17  
 011: 1179118FEB9E077F EC2  
 012: 04F671318FFB6C08 265  
 013: FE33C076018F459E 60C  
 014: 07FC011B78D01187 994  
 015: 3111231008F189E0 CF5  
 016: 74B011979B011A74 06D  
 017: F012110273B0AD41 3E0  
 018: 198F8A4C073B0695 778  
 019: F1131577ACA1031B B03  
 01A: BB8F21461341192E E8C  
 01B: A0EA7E208415A072 21D  
 01C: B066301117330119 569  
 01D: 0631F21C114D1131 8CF  
 01E: CF7A10071098F7B1 C6A  
 01F: 818FE83B18DC32F0 01E  
 020: 70708FE83B1208DE 3C0  
 021: 44A17A108F4F6C0A 765  
 022: D4018F234C063FF1 AFE  
 023: 138FFB6C08DC29E0 ECO  
 024: 8F499C0AFA8F459E 294  
 025: 0ADB10B01A4C8FEB 65C  
 026: 9E0AD48FF53C0ACO A26  
 027: 6EAF15171331FCD6 DD1  
 028: F2157410CD215D01 14F  
 029: 338F941811331FCD 4DF  
 02A: 6F211C155413301 808

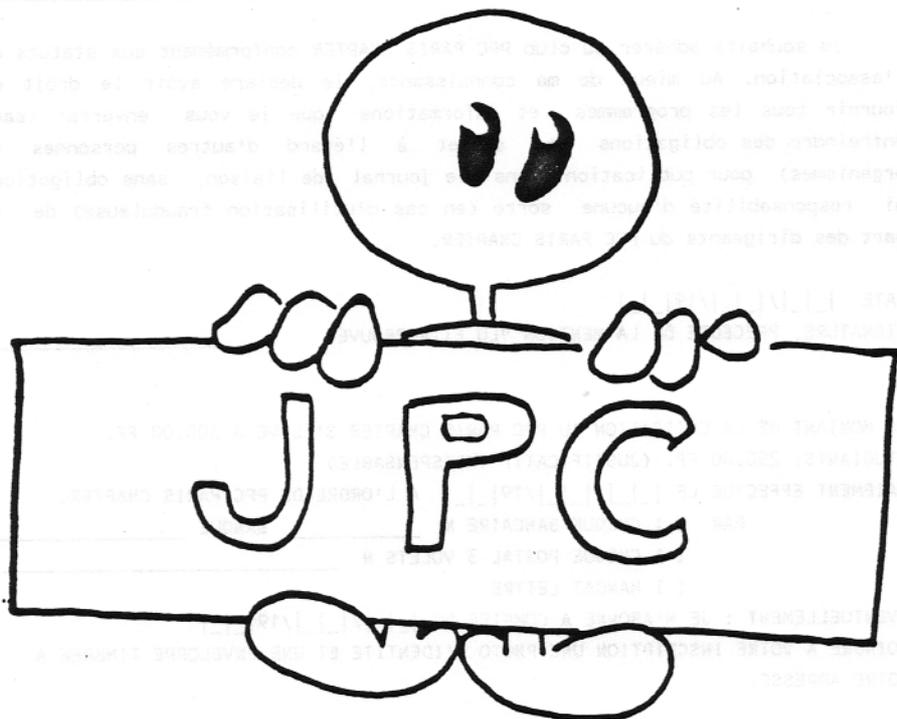
DATELEX ID#E1 545 octets

0123456789ABCDEF sm

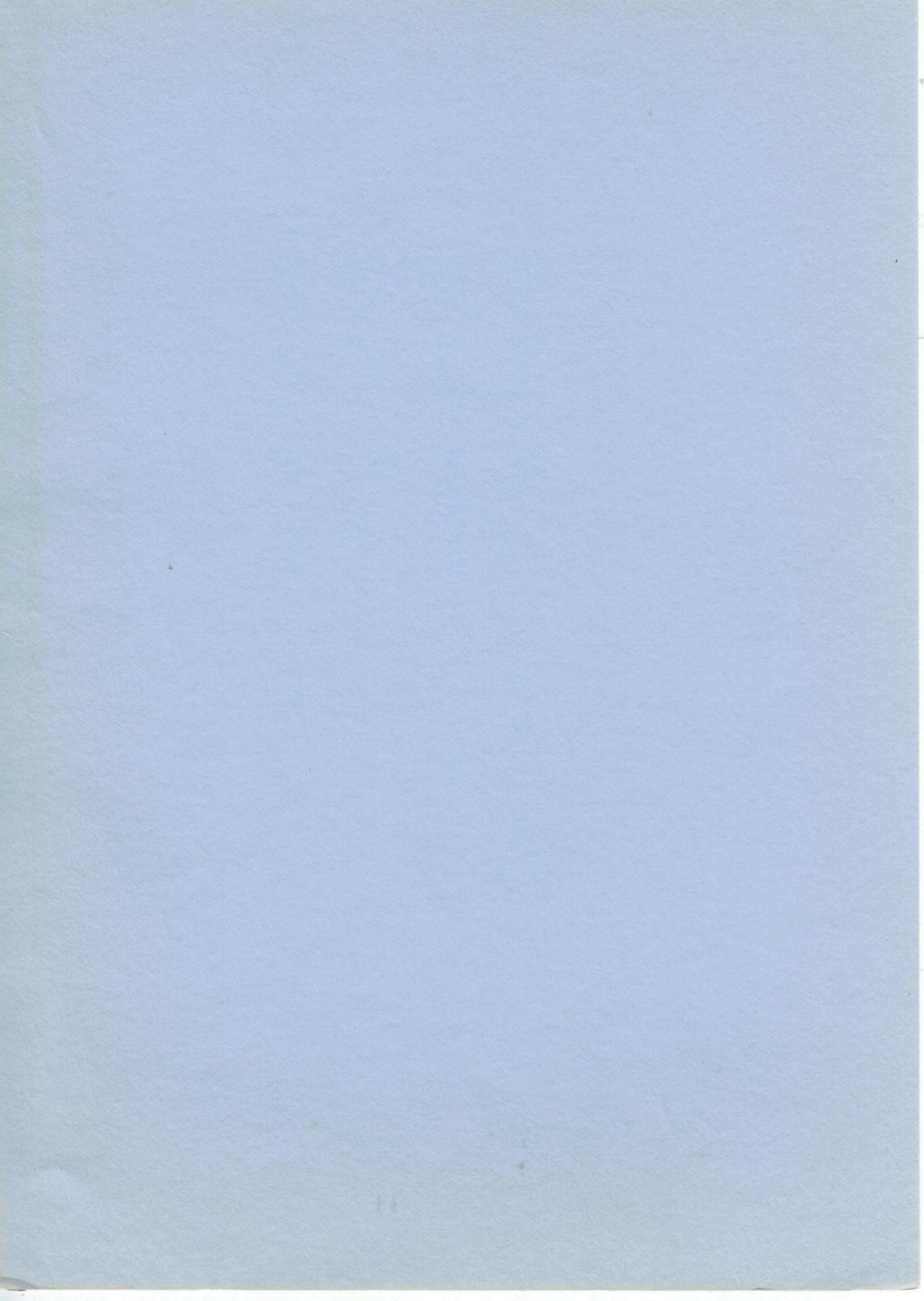
000: 44144554C4548502 35E  
 001: 802E001122512078 6AA  
 002: 744001E338300000 9F0  
 003: F440028000000000 D28  
 004: 0FA000FD00D4100F 0B8  
 005: A105E300D3200310 414  
 006: 0FE2037100F730CD 7A3  
 007: 300D944144554823 B09  
 008: 3944441495354354 E60  
 009: 4D49553744F47542 1DD  
 00A: 63544F475735D444 558  
 00B: 95831FF101001004 8B5  
 00C: 4414455402CF110B C23  
 00D: BF4574702F666022 FA4  
 00E: 516E67656CFF8822 346  
 00F: 8FC1DB08F322B184 6F2  
 010: 1401831606F92851 A4F  
 011: F810017F7DD17FB1 DFC  
 012: D9110861D08B6CDE 1AA  
 013: 26500C273C28F533 524  
 014: 3186080DDD FDD840 8D4

JPC PARIS CHAPTER

|                           |                           |                            |
|---------------------------|---------------------------|----------------------------|
| 015: AFB2190A50850812 C57 | 036: 9E1A130296123A0E 162 | 008: C6305F011A49D8D2 9D6  |
| 016: 812A798F834B1A72 FED | 037: 0E05213036B10101 4B0 | 00C: 71308DC5E208F871 D6A  |
| 017: 8FF24B187080812A 37D | 038: 301A8A0E05B0E111 832 | 00D: F032760AB58F155F 109  |
| 018: 3EB3668408118526 6F1 | 039: 213039E3115C5319 B98 | 00E: 041216A142136E21 461  |
| 019: B4088227E5117F10 A6C | 03A: 29EBC05058D91FB0 F45 | 00F: 34D015A3C4C4C416 7F2  |
| 01A: 874511109F661B72 DDE | 03B: 9E36FAE9109102D2 2E4 | 010: 30131358D3939057 B52  |
| 01B: AFA8FB13B1BCC6D0 1BF | 03C: 304058FE6CE08ADA 69F | 011: FFF77FFF7FAF1007 F40  |
| 01C: 0B7A8FB13B1AF68D 589 | 03D: D11296C71320048F A18 | 012: A4F1188AC70E4590 2DF  |
| 01D: 832F08118427A11A 8FF | 03E: E6CE08A9606EBF11 DD8 | 013: 8BA40DA74006D31C 67F  |
| 01E: FACCD23078FE6CE0 CDD | 03F: 9AE51128F40331AF 170 | 014: CD8AF0D481C81C81 A3E  |
| 01F: 87221816AD2AA281 064 | 040: EAF22034532D89F2 50E | 015: CD23070EF1ED31F4 DF4  |
| 020: 66CCF17F13310113 3E7 | 041: FOAF2CEAFE9FE000 8F5 | 016: E2C6132C2134AE2B 191  |
| 021: 38F6242029021012 737 | 042: 733101E8DA939003 C6E | 017: 666600A66CD5AFAE 54F  |
| 022: 0030540450960399 A80 | 043: 00023000315E6310 FAB | 018: 514E0E6D14C13001 8CA  |
| 023: 646E657C42969803 E03 | 044: E100002000315E21 2F4 | 019: 3FEFFCFEFFF7D2F10 CD2 |
| 024: 996462716D429677 17C | 045: 8F3F5318D84A808D 6A1 | 01A: 21361083051B178F 038  |
| 025: 03F9646562736275 4E9 | 046: 3035003 806          | 01B: 2631010971BE1401 393  |
| 026: 6D42F68503996465 86C |                           | 01C: 64119A0E5CE8AC40 73B  |
| 027: 756A4296D403F964 BF8 | GRAPHLEX ID#E1 302 octets | 01D: E4118134101706F1 A9C  |
| 028: 6562746E656652F6 F7B |                           | 01E: F088F21111478AE7 E2E  |
| 029: E203B964656D6163 2FF | 0123456789ABCDEF sm       | 01F: 0E6145CACC11A8BA 1EB  |
| 02A: 52B1CB61203F5686 689 |                           | 020: F0111E2E61451121 555  |
| 02B: 36E616D696442F1C A21 | 000: 7425140584C45485 365 | 021: 00792F1C4147111C 8C7  |
| 02C: F66001C915518F06 DA0 | 001: 802E002122512078 6B2 | 022: A1011188BEF41C41 C56  |
| 02D: 4A1840208D7B1812 115 | 002: 062001EE4F400000 A13 | 023: 4710A1C4D2E610B1 FE0  |
| 02E: 0315E048408FC463 491 | 003: F020000000000000 D3B | 024: 1179FE111E410111 34D  |
| 02F: 1550850018FC1D80 812 | 004: 095100DD00EC0000 0BE | 025: 88BE52113E410311 68F  |
| 030: AF878DFAF990A508 BE1 | 005: 974C494E454E4974 455 | 026: A8B2BD111147CA10 A56  |
| 031: 12812812AF721A92 F57 | 006: 05355445F41FF8FB 7FB | 027: 11188B21C8D84A80 DE2  |
| 032: B73812812AF5A92B 2F3 | 007: C63157017F018D91 B79 | 028: F E29                 |
| 033: 718F834B1AFA23A9 6A2 | 008: FB08D074508DE6A2 F27 |                            |
| 034: 2B7A86080DDDFDDD A82 | 009: 0305A0E10A8FD963 2BB |                            |
| 035: 22031219E1F43170 DE1 | 00A: 0781011A90A218FD 63D |                            |







*Le Journal JPC est le bulletin de liaison entre les membres de l'Association "PPC-PC", régie par la loi de 1901. Le Club est éditeur du JPC, et son siège social est au 56, rue Jean-Jacques Rousseau, 75001 Paris.*

*La maquette de ce numéro a été préparée et réalisée par Pierre David, Jean-Jacques Dhénin et Janick Taillandier, grâce à un système comprenant un HP71B, deux lecteurs de disquettes HP9114A, une imprimante HP2225B et une imprimante LaserJet.*

*Directeur de la publication : Philippe Guez  
Numéro ISSN : 0762 - 381X*

## ENGLISH SUMMARY

### JPC 42 - MARCH 1987

This issue of JPC begins with the report of Richard Nelson's decision to stop CHHU. This is a very sad news for long time HP handheld users who had many opportunities in the past to appreciate his invaluable contribution.

Our "JPC Eprom" seems to have a nice success : more informations in the next JPC.

The Journal is made of the following articles :

The column "A propos du Club" is made of various articles or member letters about the Club. The article on page 5 "Choc en retour : interface Minitel - HPIL" describe a correction to article published in JPC 34. This interface allows you to use the "Minitel". It is a low cost terminal distributed by the French postal administration ; more than 2.5 millions Minitel are used to day. It can be used as a display / keyboard / modem (75 / 1200 bauds).

The HP-41 program on page 10 is used to compute the profil of streets according to french regulations.

On page 12, you will find an introduction by Eric Gengoux about the translation of HP-75 Basic programs for the HP-71. This throw light on compatibility problems, especially when you are using the HP-75 I/O Rom, with file manipulation and HP-IL statements.

The first program (page 16) in Assembler, Forth and Basic draws functions on a 132x128 pixels plotting area. The LCD is a 132x8 window in this area ; you can scroll the complete area using [^] and [v]. The function is defined by a word ( x -- x y ) : this allows you to use polar coordinates.

After that, you have a new implementation of the classical "Life game" in Forth and Basic. The display handling is very sophisticated.

FRAC\$ (page 21) is a new Basic keyword which approximates a real number by a fraction of two integers. The optional second parameter allows you to specify the precision. The Basic program (page 27) allows you to test the precision of FRAC\$.

COMPTEUR (page 28) is an example, for beginners, of loops and tests. It counts the occurences of a letter in a sentence.

On the same page, you will find a little time manager which allows you to store two weeks of appointments.

BARGRAF2 prints bar graph from data stored in HPAF files (sample outputs on page 31). It uses a sub-program and Lex file by Pierre David described in JPC 35.

Happy JPC reading !

