JPC

PPC-PC
JPC
MARS
1985
NUMERO-22
Le numero 35 FF

.A PROPOS DU CLUB

Philippe GUEZ 1 EDITORIAL
Pierre DAVID 2 PPC PARIS SE REUNIT
Franck LEBASTARD 3 REUNION PPC A LYON
4 NOUS EN AVONS
5 COURRIER DU COEUR
Laurent ISTRIA 5 SOS
Philippe GUEZ 6 CONCOURS PROGRAMMES
6 COURRIER DES LECTEURS

.DIVERS

Laurent ISTRIA 9 PROJET "POP-CORN"
Pierre LANGLOIS 10 LA RAPIDITE DES BASICs

.HP41

Philippe DAVASE 16 ACCES DIRECT EN ROM
Aimé PIERARD 19 AFFICHAGE DES DONNEES ASCII
Aimé PIERARD 19 CARACTERE SPECIAL SUR 41 CX
Philippe TENAND 20 POINTEUR FANTOME
Philippe TENAND 20 LES CLUBS HP41 DANS LE MONDE
Philippe TENAND 21 MEMOIRE VIDE PLUS RAPIDEMENT
Philippe TENAND 21 ABRACADABRA

.HPIL

Lionel ANCELET 23 VARIATIONS SUR 41 ET HPIL
MARTINET/DAVID 24 ANALYSEUR DE BOUCLE POUR HP75

.HP71

Laurent ISTRIA 30 CONTRASTE ?
Jean-Jacques MOREAU 30 ASSEMBLER SANS ASSEMBLEUR
Michel MARTINET 31 LE DESASSEMBLEUR
DAVID/WEIL 44 FORTH RECURSIF ?
Serge VAUDENAY 46 L'EVENEMENT (Compilateur musical)
Jean-Jacques MOREAU 48 INTEGRALES
Jean-Pierre BONDU 49 MATRICES A EDITER
Olivier ARBEY 50 PAS PROPRES, MES VALEURS ?

EDITORIAL

CHERS AMIS

IL ARRIVE !!!

Nous avons pris un engagement ferme pour le Spécial-Sicob 85, qui se déroulera du 6 au 11 mai 1985.

Nous cherchons d'ores et déjà des bonnes volontés: il faut des bras pour aménager le stand, des idées pour le meubler, des têtes pour le tenir, des âmes généreuses pour nous prêter du matériel... Faites vous connaître le plus vite possible !

Nous aurons le plaisir de compter parmi nous Etienne Poupée, son matériel, et surtout son eprom "navigation" (Etienne parraine deux bateaux). Jean-Jacques Dhénin sera là pour nous présenter le module "Paname". Enfin, un représentant de Jacques Vaucelle pourra nous montrer la liaison HP41-MicroOrdinateur Sanyo.
A l'heure où j'écris ces lignes, PPC-Toulouse ne nous a pas encore donné une réponse définitive. Les Clubs étrangers non plus.

Il ne faut pas non plus oublier les contraintes bassement matérielles: un module de 9 m2 coûte 4000 FF HT, soit 14232 francs pour 3 modules, et ceci sans compter l'éléctricité, l'assurance, les meubles et éventuellement le téléphone pour présenter notre Carte Minitel.
Nous en appelons donc, comme l'an dernier, à votre bonté. Ceux qui veulent participer aux frais peuvent nous envoyer un petit chèque de 50, 100FF ou plus. Bien entendu, ce petit effort n'est pas imposé. Mais sachez que ce don nous sera très utile.

A bientôt pour une grande fête ...

SICOBEMENT VOTRE
Philippe

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 samedi de chaque mois, 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 19 h.

Pour l'accès en métro, trois possibilités s'offrent à vous à savoir:
Métro Strasbourg St Denis: sortie Porte St Martin / Bld St Denis, coté pairs.
Métro République: sortie Bld St Martin, coté pairs.
Métro J. Bonsergent: Sortie Bld Magenta, coté impairs.

Ah, j'oubliais: JPC est distribué en avant-première lors de ces réunions ... A bon entendeur Salut !

Les dates des prochaines réunions sont: (16h-19h)
- Le Samedi 2 Mars 1985
- Le Samedi 20 Avril 1985 (hors vacances scolaires)
- Le Samedi 1er Juin 1985

A Bientôt...
Pierre DAVID

Franck LEBASTARD
9672T35P43
Appt. 1085
15, chemin des Plates
69120 VAULX EN VELIN
Tél. (7) 880 77 34

Vaulx, le 19 Fevrier 1985

R E U N I O N P P C A L Y O N

Suite à la réunion PPC organisée en Bretagne les 9 & 10 février dernier, considérant l'important succès qu'elle a remporté comme à l'accoutumée d'ailleurs, voici pour nos amis du Sud-Est et pour la première fois dans leur région l'équivalente de la grande classique Bretonne.

Cette réunion se déroulera à Lyon même et sur deux jours, plus exactement les samedi & dimanche 27 & 28 avril 1985, à l'adresse suivante :

Centre International de Séjour de Lyon (CISL)
46, rue du Commandant Pegoud
69008 LYON

Cette rencontre permettra de réunir pour la première fois des membres PPC qui n'ont pas encore eu l'occasion de se connaitre. Comme lors des réunions Bretonnes, un certain nombre de spécialistes des clubs PPC seront présents (certains se sont déjà annoncés, notamment des Bretons et des Parisiens !), et ils pourront renseigner les personnes présentes sur les sujets les préoccupant.

Il est bien évident que toutes les nouveautés HP-41, HP-71, HP-75, HP-IL, etc seront présentées, comme par exemple les modules ZENROM, PANAME, CCDROM (FORTH sous réserve) de la 41, la THINKJET, la nouvelle unité de disquette IL, etc, et j'en passe parmi les meilleurs. Certains produits seront d'ailleurs proposé à la vente aux personnes présentes.

Des exposés et démonstrations seront proposés aux participants. Si vous préparez quelquechose, vous m'en indiquerez au préalable le sujet.

Comme d'habitude, une participation financière aux frais sera demandée aux participants, frais d'hébergement ainsi que frais de restauration. Cette participation n'excèdera pas 150.00 .

Comme indiqué implicitement ci-dessus, le gite et le couvert seront assurés pour tous. Comme pour toute réunion PPC, il faut prévenir si vous souhaitez être présent. Vous ne pouvez pas vous imaginer la difficulté qu'il y a à organiser une rencontre si, peu de temps avant, vous en ignorez encore le nombre des participants, sans parler d'acheter alors des vivres en quantités judicieuses ainsi que de retenir le bon nombre de chambres ! De grace, prévenez ! Mon adresse et mon numéro de téléphone sont en tète de cette page.

Il sera bien sùr possible d'aller chercher des gens aux gares de Lyon, convenons ensemble des heures d'arrivée au téléphone. Pour ceux qui ne connaissent pas la ville et qui arriveront par exemple en voiture, il sera toujours possible de fixer un point de ralliement. Je pourrai envoyer à ceux qui le souhaiteront un plan d'accès.

Il ne me reste plus qu'à souhaiter que cette réunion se passe dans les mèmes conditions que les réunions Bretonnes, dans la mème ambiance enthousiaste et sympathique, et bien sùr, avec un nombre au moins équivalent de participants. Ce n'est pas tous les jours qu'il y a une réunion PPC à proximité, peuchère, remuez-vous !

Donc, au 27 & 28 avril prochains,
à tous

Heureuse Programmation

Franck Lebastard

NOUS EN AVONS

Sont en vente au Club:

- Cartes magnétiques pour HP41 au prix de 150FF + 3,10FF de port les 50 cartes;
- Cartes magnétiques pour HP71B au prix de 71FF + 3,10FF de port les 10 cartes;
- Eproms (2716 & 2732) vierges au prix de 60FF + 3,10FF de port pièce. Pour les Eproms programmées, nous consulter;
- Module TIME (82182A) au prix de 400FF + 6,50FF de port. Livré avec manuel en français;
- Module QUADRAM (82170A) au prix de 400FF + 6,50FF de port;
- Module HPIL (82160A) au prix de 970FF + 6,50FF de port. Livré avec manuel en français;
- Pour les commandes des ZENROM, CCDROM et ROM PANAME nous envoyer 500FF minimum d'arrhes;
- VASM (listing des 3 premières pages de la 41) au prix de 170FF + 13,50FF de port;
- Le manuel de service de la 41 au prix de 50FF + 6,50FF de port;
- Le manuel technique du convertisseur au prix de 20FF + 3,20FF de port;
- Les anciens numéros au prix de:
  No 1 à 3 au prix de 15FF + 3,20FF de port par numéro,
  No 4 à 10 au prix de 20FF + 3,20FF de port par numéro,
  No 11 à 17 au prix de 25FF + 6,50FF de port par numéro,
  à partir du No 18 au prix de 35FF + 6,50FF de port par numéro.

Les ports sont les tarifs PTT actuels en non urgent.

ACHATS GROUPES

Nous assurons des commandes groupées pour ceux qui désirent profiter de notre réduction "spécial Club", soit 25% par rapport au prix HP. Pour ceux qui le désirent, nous assurons l'envoi postal en urgent ou en non urgent. Ces achats concernent aussi bien des modules que des calculatrices et autres.

NOTA: ces deux rubriques "nous en avons" et "achats groupés", sont des services "Club" que nous rendons exclusivement aux membres effectivement inscrits au club et possédant leur carte de membre PPC-PARIS.

INTERFACE VIDEO GRAPHIQUE HP-IL

Un grossiste envisage d'importer une nouvelle interface vidéo HPIL. Son originalité: elle est graphique. Son origine: RFA. Son prix: aux alentours de 5700 FF (TTC).
Sa résolution est environ de 600 par 300 pixels. Elle se programme en HPGL, qui est un "petit" langage spécialisé pour les graphiques (exemple: MD 0,0 , pour tracer un trait jusqu'en 0,0). Il s'agit là d'un sous-ensemble du HPGL, mais qui est largement suffisant.
Il ne faut cependant pas oublier la partie "texte". Cette interface permet d'afficher 25 lignes de 80 caractères. Cette partie texte se superpose au graphique.
Si nous groupons les commandes, il sera possible d'obtenir une réduction. Alors, plus vous serez nombreux, plus petit sera le prix...

COURRIER DU COEUR:

M. SEGALINI J. P., 19-23 rue du Docteur Finlay 75015 PARIS vends calculatrice programmable HP34C 290FF Tel: Dom (1) 577 02 49 ou Bur (1) 851 47 85.

Alain DULPHY (PPC119), 16 rue Pasteur, 92300 LEVALLOIS-PERRET, vends une HP75C (24k) à 5000FF une imprimante GP100 IL-A à 2500FF. Tel= Dom (1) 737 41 13 et Bur (1) 778 16 60.

Michel MARTINET (PPC-P#12), 17 rue eugène Hénaff - 93200 ST DENIS - Tel:8220149 - Vend HP75C + 8Ko + VISICALC + formateur de texte + très nombreux logiciels: 8000 Frs.

Daniel JACOB: 4,rue des blanchisseurs 92370 CHAVILLE Vends:Imprimante therm. HP 82143A tb état de marche + 3 rouleaux de papier noir: 1200FF; MLDL Hollandais (Ste ERAMCO) 8K RAM et 24K ROM + Eproms ERAMCO-MLDL + VASM:2500FF;Imprimante therm. HP 82143A en mauvais état (ne fonctionne pas, mais présence de toutes les fonctions au CAT 2; BLDSPEC fonctionne correctement) (problème: pont de diodes grillé..) Pour bricoleurs, cédée à 300FF.

S O S - S O S - S O S

Au sujet de PPC^PPC

Ce journal doit être un résumé d'un maximum de programmes parus dans les journaux PPC européens. Les journaux qui nous ont donnés l'autorisation d'utiliser leurs articles sont les suivants:

- PRISMA
- HP SCHWEIZ
- PPC LAUSANNNE
- PPC TN
- PPC NORVEGE

Vous avez sans doute remarqué que la diversité des langues employées est une caractéristique majeure de cette liste. Hélas, mes études ne m'ont permis que l'apprentissage de la langue de Shakespeare. C'est la raison pour laquelle je me permets de faire appel à vos connaissances pour m'aider. J'aurais donc un grand besoin de germanophones, de néerlandophones ...

D'avance, je vous remercie pour votre aide qui me sera très précieuse.

Laurent ISTRIA

CONCOURS PROGRAMMES

Eh oui! Comme nous l'avions annoncé dans le JPC No 21, notre Concours est lancé. Lors de la réunion du Bureau, les règles suivantes ont été établies:

ARTICLE 1 : - Sont acceptés pour ce concours tous les articles et programmes portant sur le matériel portable HP; séries 70, 40, 10, périphériques...
ARTICLE 2 : - Toutes les soumissions paraîtront dans le journal.
ARTICLE 3 : - Tous les articles ou programmes devront être inédits et pourront porter soit sur un sujet totalement nouveau, soit sur l'amélioration d'un article déjà paru (en citant les sources), soit sur un ensemble d'articles d'initiation...
ARTICLE 4 : - Les programmes et articles pourront provenir de tous les utilisateurs du matériel cité à l'article 1, qu'ils soient ou non membres PPC.
ARTICLE 5 : - Pour conserver le caractère impartial, les membres du Bureau ne pourront soumettre d'épreuve pour ce concours.
ARTICLE 6 : - La durée de ce concours a été fixée à un an, du vendredi 1er février 1985 au vendredi 31 janvier 1986.
ARTICLE 7 : - Le Bureau, à travers le Président, choisit un jury parmi les personnes qui se seront portées volontaires. Ces personnes ne peuvent pas concourir.

Nous attendons donc vos impressions, mais aussi et surtout vos articles.

Bon travail à tous,

Philippe.

COURRIER DES LECTEURS

François COLET
34, Rue de la Jomayère
42100 St Etienne

Le 16 janvier 1985

Monsieur,

Je trouve votre journal très bien et vous en félicite en espérant que celà durera encore longtemps.

Comme je vous l'ai déjà dit, je suis débutant dans la technique de programmation du calculateur HP41CV:

N'ayant que peu de temps pour écrire les programmes, j'aimerais pouvoir bénéficier de l'acquis des adhérents du Club et notamment pourriez vous s'il vous plait m'indiquer ce que vous avez en ce qui concerne les matrices, la physique, les méthodes de calcul d'intégrales. J'ai appris par la radio, aujourd'hui qu'Hewlett-Packard s'intéresse à la voile. Je m'y intéresse beaucoup également et je voudrais savoir ce que HP fait et propose dans ce domaine.
J'envisage de plus en plus sérieusement l'achat du lecteur de cartes magnétiques, pourriez-vous m'indiquer le prix et les délais de livraison ? Si vous pouviez me renseigner rapidement, cela me sortirait d'une situation critique: je vais bientôt ne plus avoir de place pour écrire mes programmes et je n'ose en éliminer car d'ici un mois, je vais de nouveau en avoir besoin.
D'autre part, à quoi servent les modules ZENROM, CCDROM et ROMPANAM, à quoi servent les fonctions synthétiques, n'existe-t-il pas d'ouvrage en français, mon anglais étant approximatif, ou en allemand relatif à leur utilisation. J'ai entendu dire qu'on pourrait créer des symboles et des notes de musique supplémentaires.
Existe-t-il des possibilités d'emprunt de matériel ?

En attendant votre réponse, veuillez agréer, monsieur, l'expression de mes sentiments respectueux.

François COLET.

Pascal LE ROY (PPCPC 69),
91, Rue du commerce
75015 PARIS - Tel: (1) 532 87 79

Paris, le 2 février 1985

Je possède une HP41C et j'ai beaucoup entendu parler du langage "synthétique" mais j'ignore ses avantages, aussi pourriez-vous me les indiquer ainsi que me conseiller pour l'achat d'un manuel pour l'apprendre ?

Merci beaucoup

Pascal LE ROY.

Chers amis,

Je répondrai à ces deux lettres d'un seul trait de plume.

Le gros morceau de l'acquis des Adhérents du Club se trouve dans notre journal de liaison (JPC). Il est difficile de vous envoyer un listing des articles parus, celui-ci n'ayant pas été fait. Mais il paraît régulièrement dans les numéros double de décembre-janvier (soit le No 10 pour l'année 84 et le No 20 pour l'année 85).

En ce qui concerne l'achat de matériel, nous pouvons avoir jusqu'à 25% de réduction chez certain de nos "fournisseurs Club". La réduction est faite sur le prix pratiqué par HP et dépend de l'augmentation possible des produits entre le moment où vous nous envoyez votre lettre et celui où nous payons votre commande. Le Club ne fait aucun bénéfice sur ce service.

Nous vous demandons donc de ne pas oublier de compter en supplément le port (urgent ou non urgent, mais nous vous conseillons le recommandé avec accusé de reception). D'autre part, la facture établie est celle du fournisseur.

- Les modules CCD et ZENROM vous permettent une meilleure utilisation de votre HP41. La ZENROM contient, en tout et pour tout, un éditeur de RAM (pour programmation synthétique ou non) et un éditeur de microcodes (vous permettant de décoder directement les listing microcodes ou de programmer un laboratoire de langage machine). La CCDROM, vous permet de programmer tout ce que vous désirez au sein de la machine en programmation synthétique ou non (un peu comme les ordres basic 'PEEK' & 'POKE', mais pour la programmation synthétique). La ROM PANAME a un objectif totalement différent. Elle contient surtout des utilitaires pour HP41 seule ou munie d'un périphérique (traitement de fichiers, traitement de tableau, calculs matriciels, utilisation d'une vidéo, d'une imprimante, d'une table traçante. Pour les ouvrages français, vous avez le JPC, la revue de PPC TOULOUSE 'MICRO-REVUE', des livres édités par les éditions du cagire (demandez à JEAN-DANIEL DODIN, 33 rue du Cagire, TOULOUSE), le plus connu étant "Au fond de la 41". En allemand, vous avez la revue PRISMA éditée par le club CCD (demandez à W & W, Dr W. BALTES und W. KOTZ, Im Aehlemaar 20, Postfach 800 133, 5060 Bergisch Gladbasch 2, ALLEMAGNE). Je ne sais pas ce que vous appelez des symboles, mais vous pouvez créer toutes sortes de caractères, des tones synthétiques (TONE M, TONE P, TONE 25...)

Amitiés à tous et Heureuse programmation.

Philippe.

DIVERS

LE PROJET POPCORN

La meilleure façon de parler des choses n'est-elle pas de commencer par le commencement ? Et bien soit. Voici contée pour vous l'histoire de la genèse de ce projet.

Tout a débuté au mois de septembre dernier. Olivier Arbey et moi même avions été dépéchés, par le Club, pour participer au tournoi d'Othello de l'OI. Le programme sous lequel tournaient nos HP71B avait été conçu par Fréderic Poupon, et nous permit de remporter 2 victoires sur 4 parties ainsi qu'une honorable quinzième place. Mais nous avions vu.

Oui, nous avions vu que certains programmes, tournant sur de petites machines étaient extrèmement performants. De là, naquit en nous l'idée de toutes les dépasser; de faire le meilleur programme, de fabriquer Le programme gagnant.

A la fin du tournoi, nous nous retrouvâmes donc, Olivier et moi, dans mon humble demeure. L'idée de base était dans notre tête, encore fallait-il la coucher sur du papier, afin de la rendre exploitable. A l'époque venait à peine de sortir la ROM Forth/Assembleur et, ne nous doutant pas le moins du monde des quelques difficultés qui allaient apparaître lors de la tentative d'apprentissage de ces langages, nous décidâmes joyeusement que le programme sus-nommé serait écrit en Basic, en Assembleur et en Forth (BAF, d'où le nom du projet).

A l'évidence, nous pensâmes que nos deux personnes ne seraient pas suffisantes à l'achèvement du problème. Les tentatives de programmation en Assembleur révèlerent que notre conception de la difficulté était bien au dessous de la vérité. Le fait était que nous avions besoin de 'suppléants'; il nous sont devenus indispensablés. C'est la raison pour laquelle nous faisons appel à vous. Nous désirerions entrer en contact avec:

- Des Logiciens
- Des Evaluateurs
- Des Programmeurs.

Les personnes qui seraient intéressées par ce genre de travail voudront bien se faire connaître auprès de Philippe Guez.

D'avance, je remercie tous ceux qui, je l'espère, nous aideront.

Laurent ISTRIA (#3)

LA RAPIDITE DES BASICS: 'le HP71B face à ses concurrents directs'.

Vous vous souvenez sans doute encore du JPC nr13 et de son épais dossier su le HP71B. Vous avez très certainement remarqué que ce dossier évoque brièvement le fait que le petit dernier de HP, n'est pas un sprinter de l'informatique.

Depuis, vous vous demandez avec angoisse ce que celà peut bien vouloir dire. Vous souvenant avec inquiétude du petit trot pépère de notre chère HP41. Si vous en possédez une, elle a peut être même subi une opération chirurgicale pour pallier l'inconvénient. Le fer à souder à la main, vous avez eu des minutes de suspens intense qui valent les meilleurs Hitchcock. Votre palpitant en a souffert. Faudra-t-il recommencer avec Titan???

Votre attente s'achève avec cet article. Votre serviteur a pu en effet tester maintes machines plus ou moins comparables avec le HP71B, grâce à l'aide on ne peut plus précieuse de la règle à calcul, 67, Blvd St germain. (RAC pour les intimes.)

MACHINE	1 à 1000	1000 SINUS	1000 PI/e	1000 G-R	1000 MID$	100 ?CHR$
HP75	3"23	52"91	13"50	5"51	11"56	4"53
HP71B	8"82	66"36	23"43	18"48	21"10	15"01
PC 1500	15"	3'...	45"	30"	37"	1'29"38
X07	3"50	2'15"	49"	5"83	10"22	15"65
PB700	11"85	2'22"48	51"25	33"82	1'37"97	5"43
PB100	7"71	2'01"97	30"57	31"42		
PC1251	42"71...					
M10	3"27	1'56	41"53	5"	8"25	2"
HX20	3"10	32"88	13"50	6"06	12"36	2"
HHC	1"45	xxxxxxxxxx	6"	3"16	7"39	17"60

Lorsque l'on regarde le tableau "Résultat des tests", on remarque dès l'abord la grande ligne blanche qui fait face au PC1251. De là l'oeil est attiré par le seul test effectué sur cette machine. 42"71 pour 1000 boucles vides. C'est au moins plus de trois fois plus lent que ce que font les autres machines testées. Il suffit de savoir qu'en toutes choses le PC1251 est plus lent que le PC1500 ou le PB700, vous verrez plus loin qu'il sagit d'un critère suffisant d'élimination car ces deux machines sont plutôt lentes.

L'instant d'après l'oeil est attiré par les deux blancs de la ligne qui fait face au PB100. Celui qui est dans la colonne '100 ?CHR$' (qui correspond à '100 fois la fonction PRINT CHR$'; voir le tableau n2: "liste des programmes utilisés".) s'explique par l'absence de l'instruction CHR$ du basic du PB100. le blanc de la colonne 1000 MID$ vient de ce que, lorsque j'ai fait les tests sur cette machine, j'ignorais la manière de programmer sa fonction MID. Maintenant je sais comment on fait, mais je n'ai pas le courage d'aller embêter à nouveau les gens de la RAC pour si peu.

Enfin, votre oeil s'est immobilisé, interloqué, sur le xxxxxxxxxx qui fait face au HHC de Panasonic dans la colonne des 1000 SINUS. Le modèle de cette machine que j'ai eu entre les mains à la RAC ignorait ce que veut dire SIN(I), c'est à dire que le module qui apporte les fonctions mathématiques n'était pas dedans. Plus brièvement, le basic était réduit à sa plus simple expression, c'est ce qui explique, à mon avis, les résultats ahurissant du HHC, qui écrase tous les autres par sa vitesse. On m'a même dit que cette machine compilerait peut-être les boucles FOR-NEXT. Dans ce climat de suspicion insupportable, j'ai disqualifié le HHC.

Je signale, pour ceux qui l'ignorent que:
- X07 désigne le dernier de chez CANON,
- PB700 et PB100 sont, chez CASIO les 2 successeurs haut et bas de gamme du FX702P,
- M10 désigne le dernier de chez OLIVETTI: format 21x29,7 (pour mallette, donc), écran à cristaux liquides orientable (8 lignes de 40 caractères) et clavier machine à écrire (très confortable).
- HX20 désigne la machine EPSON sortie il y a un ou deux ans au même format 21x29,7 avec en bas, un clavier machine à écrire, au milieu en haut un écran à cristaux liquides de 4 lignes de 20 caractères, à gauche de celui-ci une petite imprimante thermique et à droite un port d'extensions.
J'ajoute enfin que 1000 G-R est l'abréviation de 1000 GOSUB-RETURN. (J'espère que ces brèves explications, que je souhaite superflues, auront mis tout le monde sur la même longueur d'onde).

Après ces quelques mises au point un peu longues, venons-en à l'analyse des tests. Et d'abord pourquoi ceux-là plutôt que d'autres? Les mille boucles vides sont désormais la règle parmi ceux qui veulent comparer les rapidités des petits matériels, mais j'ai estimé que ce test n'apporte pas *suffisamment d'informations, sur les qualités d'un matériel. D'autant plus que dans un vrai programme, il* y a peu de chance d'avoir effectivement à implanter 1000 boucles vides, à moins de ne pas avoir l'ordre WAIT. Par contre MID$ est, en basic, un des ordres les plus fréquemment utilisés. De même les quatre opérations sont utilisées de manière intensive, j'ai choisi la division car c'est la plus difficile des quatres à programmer en langage machine.

Ce qui fait que la rapidité des 1000 PI/e dépend autant des caractéristiques physiques de la machine que du travail de ceux qui ont programmé le basic résident. J'ai rajouté le test des 1000 SINUS parce que je suis d'abord intéressé par les fonctions scientifiques, parce que la majorité des adhérents au PPC ont une HP41 qui est d'abord une machine scientifique et enfin parceque SINUS est peut-être la fonction la plus difficile à programmer en langage machine et le plus souvent la fonction la plus lente sur les ordinateurs qui la possèdent. Ce test représente bien ce que chaque machine est capable de faire de plus lent. Les 1000 GOSUB-RETURN viennent du traumatisme que j'ai eu face à la lenteur des branchements de ma chère 41, je lui ai pardonné depuis, mais je suis sûr que tout utilisateur de cette machine, donc adepte bon gré mal gré de la programmation structurée, apréciera comme importante la vitesse que met la machine à localiser une adresse.

Enfin le test 100 PRINT CHR$(65) qui doit afficher une série de A m'a été plus ou moins suggéré par un autre membre du PPC-PC. C'est à mon avis le test le plus contestable de la série car il dépend énormemment des caractéristiques physiques du matériel. Sans compter que les ordres d'affichage sont on ne peut moins standardisés, les résultats obtenus sur PC1500 sont par exemple aberrants, car l'ordre PRINT de cette machine interrompt le programme en cours (équivalent à PROMPT en RPN) et les 100 boucles ont été programmés avec l'ordre PAUSE qui ralentit l'exécution du programme. Et bien entendu, pour ce test, les machines à grand écran sont incroyablement avantagées, puisqu'elles ne font pas de gestion des caractères qui sortent de l'écran comme sur les machines à écran monoligne comme le 71. On notera cependant la contre performance surprenante du X07 qui semblerait donc moins adapté aux jeux graphiques qu'on le croit habituellement. Je laisse le soin de trancher cette question étonnante aux utilisateurs de ce matériel, chose qu'ils ne feront sans doute jamais si celà déprécie leur machine.

"LISTE DES PROGRAMMES UTILISES POUR LES TESTS"

a) 1000 boucles vides

```
10 FOR I=1 TO 1000
20 NEXT I
30 PRINT "Fini!"
```

Classement: HX20, HP75, M10, X07, PB100, HP71B, PB700, PC1500.

b) 1000 fois la fonction SINUS.

```
10 FOR I=1 TO 1000
20 A=SIN(I)
30 NEXT I
40 PRINT "Fini!"
```

Classement: HX20, HP75, HP71B, M10, PB100, X07, PB700, PC1500.

c) 1000 fois PI/e

```
10 A=PI : B=EXP(1)
20 FOR I=1 TO 1000
30 C=A/B
40 NEXT I
50 PRINT "Fini!"
```

Classement: HP75 = HX20, HP71B, PB100, M10, PC1500, X07, PB700.

d) 1000 fois GOSUB-RETURN.

```
10 FOR I=1 TO 1000
20 GOSUB 100
30 NEXT I
40 PRINT "Fini!"
50 END
100 RETURN
```

Classement: M10, HP75, X07, HX20, HP71B, PC1500, PB100, PB700.

e) 1000 fois la fonction MID$

```
10 A$= "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
20 FOR I=1 TO 1000
30 B$= MID$(A$,10,6)
40 NEXT I
50 PRINT "Fini!"
```

Classement: M10, X07, HP75, HX20, HP71B, PC1500, PB700.

f) 100 fois la fonction PRINT CHR$

```
10 FOR I=1 TO 100
20 PRINT CHR$(65)
30 NEXT I
40 PRINT "Fini!"
```

Classement: M10 = HX20, HP75, PB700, HP71B, X07, HHC, PC1500

Enfin pour essayer de donner une vision plus synthétique des tests réalisés, j'ai effectué un classement entre les machines pour chaque test (sauf pour PRINT CHR$) comme vous pouvez le voir ci-dessus. Et pour chaque machine j'ai effectué la moyenne des rangs où est classée la machine pour chaque test. Par exemple, le HP75 est 2 pour le test a), 2 pour le b), 1 (ex-aequo) pour le c), 2e pour le d) et 3e pour le e). Ce qui fait en tout: (2+2+1+2+3)/6 = 10/6 # 1,6. (J'ai dû adopter le système de la moyenne pour intégrer le PB100 dans ce classement car il ne participe qu'à 5 des tests utilisés contre 6 pour les autres machines).

Résultats:

1e HP75 avec 1,6; 2e HX20 avec 1,8; 3e M10 avec 2,1; 4e ex-aequo HP71B et X07 avec 3,5; 6e PB100 avec 4; 7e PC1500 avec 5,5 et 8e PB700 avec 6.

Cette manière de faire m'a paru contestable, aussi j'attire votre attention sur le fait que les temps du HX20 sont tous inférieurs à 35", que ceux du HP75 sont tous inférieurs à 55", ceux du HP71B à 1'10", ceux du M10 à 2', la suite des résultats n'étant pas caractéristique. Déjà, dire que les temps du M10 sont inférieurs à 2', c'est lui faire beaucoup de tord juste pour la lenteur de sa fonction SINUS.

Mais la vitesse n'est pas tout ce qui importe dans un ordinateur. Les utilisateurs de la 41 sont bien placés pour le savoir, qui continuent à vénérer une machine totalement dépassée sur ce plan là. Ainsi le HHC est très rapide mais je n'achèterais sûrement pas une machine dont le clavier est le plus illisible que

j'ai jamais vu (indépendemment du fait que l'absence de SINUS dans la version de base me paraît être un véritable scandale).

D'autant plus que la qualité des programmes utilisés pour les tests risque d'être contestée - à juste titre - par les spécialistes des matériels testés. Par exemple, certains ordinateurs sont beaucoup plus rapides lorsqu'on met les sous programmes en tête de programme plutôt qu'en fin, comme d'habitude. Mais c'est sciemment que j'ai adopté le point de vue du néophyte possédant juste quelques notions de basic et cherchant la machine qu'il va acheter. J'ai donc programmé simplement. A priori les programmes simples doivent être simple à entrer au clavier. Eh bien en fait pas du tout! Bien sûr mon néophyte est un sacré hypocrite car, il cherche surtout, en tant que membre du PPC, une machine qui lui permette de passer des moment agréables à programmer.

Dans cette optique, il est invraisemblable d'investir dans un M10, qui est, certe, une excellente machine mais à destination ouvertement professionnelle, plus orientée vers la fonction d'outil de bureau (le traitement de texte en particulier a l'air reussi quoique pas très évident au premier abord) plutôt que vers celle de babasse pour félé du langage machine. J'ajoute que l'éditeur basic du M10 n'est pas des plus faciles à saisir. Ce n'est pas un pleine page et l'ordre EDIT vous met instantanément en mode traitement de texte (semble-t-il). Mais j'imagine qu'avec le manuel d'utilisation de la machine tout doit devenir plus simple... C'est un peu dommage.

Le HX20 par son prix pourrait être intéressant, son basic est puissant, du même niveau que celui du HP75 (donc très puissant en fait). J'ai beaucoup aimé l'éditeur basic qui considère que le petit écran de la machine est une fenêtre sur un grand écran. Celà permet un véritable éditeur pleine page (les touches de déplacement sont mal foutues mais c'est toujours pareil, seul le X07 a vraiment inové sur ce plan). Vraiment, si le HX20 n'avait pas été aussi énorme (le clavier de machine à écrire est pratique mais je peux m'en passer), j'aurai fini par acheter ça (si le HP71B n'était pas sorti). Mais le HX2 a un défaut encore plus important à mes yeux: comme beaucoup de portables, c'est un matériel mal ciblé commercialement; trop gros pour être vraiment pratique à transporter et pas assez puissant pour concurrencer les ordinateurs de table. (le HP75 n'échappe pas à ce phénomène, même s'il semble rencontrer un bon succès auprès des laboratoires scientifiques et quoique pour lui c'est le prix qui joue le rôle de l'encombrement. Bref, ces matériels se vendent mal. Ce qui fait que peu de logiciels spécifiques sont créés pour eux. C'est bien joli d'avoir une machine puissante et portable, mais si on est seul à l'avoir, celà cesse vite d'être amusant. La floraison actuelle des clubs tournant autour de matériels spécifiques témoigne clairement de la nécessité de ne pas s'isoler en informatique. Plus on est de fous à s'éclater sur un système donné, plus il y a de chances que chacun profite au maximum de son système.

De même que le HX20, le HP75 est une excellente machine, très puissante, dont les logiciels sont faciles à utiliser, et le clavier des plus agréables. Mais à ce prix là autant acheter un ordinateur de table, qui, même pour un prix légèrement supérieur, ne pourrait être que plus puissant grâce à ses disquettes et à son grand écran. En plus de sa conception de système fermé fait que peu d'utilisateurs se sont penchés sur le problème délicat d'accroître ses possibilités et les seuls logiciels tournant sur HP75 actuellement sont tous ou presque d'origine HP. Cette erreur de conception semble avoir été largement ratrappée dans le HP71B, qui est un véritable système ouvert. (on devrait dire système "éventré" car il y a tellement de manières d'accroître ses possibilités qu'on aura du mal à choisir.)

Le PC1500 ne rencontre pas ce genre de difficulté, comme en témoigne l'activité fébrile des "Charpentiers". J'ai un peu hésité avant de l'introduire dans ces tests, car elle y fait piètre figure et celà fait du tord à une machine que je trouve excellente à tous points de vues. Les seules choses que je lui reproche sont: son clavier (j'aime les claviers HP, je n'y peut rien) et son sélecteur de mode qu'il faut utiliser toutes les dix secondes pour mettre au point un programme, sous prétexte qu'on ne peut pas le modifier en mode RUN et qu'on ne peut pas le lancer en mode WRITE. J'ajoute qu'à la moindre erreur de manipulation de ce sélecteur, on se retrouve en mode réserve, ce qui oblige à une concentration de tous les instants pour surveiller les indicateurs de mode à l'écran. (Je rappelle que sur le PC1500, contrairement à ce qui se passe pour la HP41, le contenu de l'écran ne permet pas le plus souvent de savoir dans quel mode on est.) Malgré ce défaut, auquel on doit pouvoir s'habituer cette machine reste, à mes yeux, une nécessité dans le genre, qui continuera à avoir du succès si son constructeur ne la laisse pas tomber. Il est regrettable, par exemple, que cette machine au prix modique ait des périphériques parfois aussi chers. (Au moins avec HP, on sait qu'on devra de toutes façons payer très cher...)

Le X07 m'a agréablement surpris par ses facilités d'utilisation et cette machine me semble promise à un bel avenir, comme en témoigne sa bibliothèque de programmes qui s'étend chaque jour. Le clavier est très bien conçu, il m'a fait l'impression d'une copie moyenne des claviers HP, avec touches bisautées et cliquantes. L'éditeur basic est très bien soutenu par les quatre grosses touches blanches à droite de l'écran qui servent à déplacer le curseur. Mais contrairement à l'écran du HX20, celui-ci n'est pas une fenêtre ouverte sur un plus grand écran. Ce qui fait qu'après un LIST, par exemple, les lignes se bousculent à l'écran et c'est plutôt difficile à suivre. Impossible bien entendu de rappeler avec le curseur les lignes qui ont disparu en haut de l'écran. Je lui reproche surtout de ne pas tenir dans ma mallette et d'être mathématiquement sous développée. Autre détail désagréable: les touches sont très sensibles, il est donc impossible d'oublier sa main sur le clavier, comme il m'arrive souvent de le faire sur ma 41, sous peine d'introduire des aberrations à l'écran. En fait Je reproche aussi au X07 une orientation gadget: grand écran, qui, on l'a vu, ne rend en fait pas beaucoup service pour l'édition, communicateur à infrarouge (ça peut servir un jour d'examen, mais sinon?), imprimante trop petite pour être vraiment intéressante. Je passerai rapidement sur le PB100 et le PB700 qui m'ont déçu au delà de toute expression. ils ont pratiquement le même éditeur sous basic et ne connait rien de plus mal pratique. Si vous avez vu fonctionner celui du FX702P, c'est exactement le même. C'est à dire qu'il est impossible de passer directement de la ligne 110 à la ligne 100 et il faut se farcir un LIST complet, passé à appuyer sur la touche EXE, pour simplement remonter d'une ligne. Indépendemment du fait que le PB100 est la machine la plus moche actuellement disponible sur le marché, elle m'a surpris par sa rapidité. Vu la petitesse de son basic, c'est normal. La lenteur de la fonction MID$ sur PB700 est dûe au fait qu'il est impossible d'avoir sur cette machine une chaîne de 26 caractères sans l'introduire dans un tableau, les chaines normales (non incluses dans un tableau) font au plus 16 caractères je crois. C'est tellement ridicule que je me suis soigneusement attaché à utiliser une chaine de 26 caractères. Je ne vous parlerai pas de la qualité des claviers CASIO, je tiens à rester poli. Je signale que sur le PB100, comme sur le PC1500 il faut à chaque instant changer de mode et que le sélecteur est encore plus mal pratique que sur le PC1500.

Enfin le HP71B. A part le HP75 et le HX20, c'est la machine la plus facile à utiliser. J'ai envie moi aussi de vous en faire l'éloge mais cette machine a tout de même quelques défauts: les touches DEL et INS sont shiftées et leur présentation non standard les rend difficiles à localiser. (Il s'agit de -CHAR et de I/R). Mais je m'aperçois qu'il est totalement idiot de chercher des défauts à ce clavier, puisque vous pouvez le redéfinir à votre guise. Dommage l'écran monoligne? Je préfère personnellement celà à la bousculade qui se produit à chaque LIST sur les écrans des X07 et PB700. Les temps du HP71B sont excellents, vous avez remarqué? Si je vous dit qu'en plus pour 1000 SINUS et 1000 PI/e, les calculs sont effectués sur 12 chiffres (+ 3 de garde) alors que les X07, M10 et PB700 les font seulement sur 6 chiffres, celà laisse rêveur. Bien sûr j'aurai pu faire calculer ces machines, qui ignorent ce qu'est PI, en double précision mais elles auraient été encore ralenties. D'autre part un néophyte n'est pas censé savoir ces subtilités.

Non, vraiment je n'ai pas trouvé de défaut au HP71B qui ne soit rectifiable par l'utilisateur. Et c'est là la puissance d'un véritable système ouvert: quel que soit le défaut rencontré, il existe une solution. Par exemple j'ai beaucoup regretté l'absence sur le HP71B des fonctions HR, HMS, HMS+ et HMS- de la HP41. En astronomie, où les calculs d'angle se font en degré, minute, secondes, ces fonctions sont très utiles. Il me suffira lorsque j'aurai mon 71 de creer ces fonctions de toutes pièces, probablement en langage machine ou en FORTH, puis de les intégrer au basic. Les acrobaties rendues nécessaires sur la HP41 parce qu'elle n'était pas au départ un système ouvert ne seront même pas nécessaires. Les seules choses attaquables sur le 71 sont ces caractéristiques physiques: son écran monoligne (j'ai donné mon avis la dessus plus haut) et la petitesse de son clavier, qui ne devrait pas déranger les utilisateurs de HP41, à mon avis. Face au HP75, que je connais malheureusement mal, il semble que le seul désavantage du HP71 soit sa vitesse. Si vous y tenez vraiment, il reste le fer à souder...

En espérant vous avoir aidé à choisir entre le HP71B...

Pierre LANGLOIS (PPC.PC n°87)

PS: N'oubliez pas que plus nombreux nous seront à l'avoir, plus il sera puissant!!!

HP 41

Philippe DAVASE	ACCES DIRECT EN ROM Pour profiter davantage de vos modules ROM.
Aimé PIERARD	AFFICHAGE DES DONNEES ASCII Petit programme astucieux pour visualiser un fichier.
Aimé PIERARD	CARACTERE SPECIAL SUR HP41CX Comment entrer des caracteres speciaux avec ED ?
Philippe TENAND	POINTEUR FANTOME Une bogue alphanumérique de la HP41.
Philippe TENAND	LES CLUBS HP41 DANS LE MONDE Récapitulatif (presque) exhaustif des Clubs.
Philippe TENAND	MEMOIRE VIDE PLUS RAPIDEMENT Eviter l'attente lors d'un CLP.
Philippe TENAND	ABRACADABRA Code nul en mode Alpha

ACCES DIRECT EN ROM PAR STO b POUR HP41

On peut affirmer sans trop s'avancer, que chaque possesseur de HP41 possède au minimum un exemplairede ces petits boitiers de plastique noir que les initiés appellent une "RRRDMM", ou encore un "module ROM".

Pourquoi ROM ? Encore un héritage outre-atlantique, mon brave monsieur, et oui est l'abréviation de Read Only Memory; soit en un français approximatif une mémoire que l'on ne peut que lire; donc, dans laquelle on ne peut pas écrire (cela parait évident, mais...); dont au sujet de laquelle certains (les érudits!) l'appellent mémoire morte, ou MEM. Bref, un bout de plastique vous dis-je, mais qui recelle bien des trésors quand on apprend à l'utiliser.
Précisons de suite que je ne suis certes pas un maître es-ROM, mais que j'ai réussi à comprendre et utiliser certaines possibilités de ces ROM, grâce aux documents existants sur le sujet. Voilà donc le résultat de mes cogitations. Il appartient à chacun de compléter éventuellement ces quelques astuces dans les numéros suivants de la revue (qui n'est alimentée que par chacun de nous), donc à vos claviers !

Notre brave machine possède une suite de registres particuliers, qui ne sont pas laissés à la disposition de l'utilisateur (normalement bien sûr!). Ce sont des registres que le système se réserve pour stocker des informations qui lui permettent de ne pas perdre complètement la boule face à nos chères élucubrations. Donc, donc disais-je, le processeur se met de côté les meilleurs morceaux, et ne nous laisse que les registres courants, adressables par les ordres connus (RCL, STO, RCL IND, STO IND, etc...), ordres que l'on compose facilement au clavier.

Il faut cependant savoir que le processeur n'est pas aussi fûté qu'il n'y parait, car il reconnait et exécute aussi les ordres permettant d'accéder à ces fameux registres cachés au fond de ses entrailles, pour peu que ces ordres soient exécutés à l'intérieur d'un programme, ou encore par assignation de touches. Ces instructions spéciales, n'étant pas prévues à l'origine par HP, ont reçu le doux surnom "d'instructions synthétiques", (voir la littérature sur le sujet).

Créer et utiliser toute la panoplie de ces instructions ne relève pas de ces quelques lignes, et nous allons donc considerer que nous avons déjà sous la main un magnifique 'STO b' bien frais, et son frère siamois 'RCL b'. Ces deux instructions vont nous permettre d'accéder au registre de stockage du pointeur de programme. En effet le processeur, toujours malin, utilise un pointeur de programme pour savoir ou il se trouve dans la mémoire de la machine, et ce pointeur se déplace en fonction des opérations qu'effectue la machine. Comme notre processeur a la vue défaillante, il note dans le registre [b] la dernière adresse valide du pointeur de programme, ce qui lui évite de devoir parcourir toute la mémoire quand, par exemple, il a besoin de retrouver le pas de programme auquel il doit se brancher.

Si nous modifions perfidement les instructions contenues dans le registre [b], nous allons illico provoquer un saut du processeur, qui va aller se positionner dans la mémoire, à l'adresse que nous venons de stocker dans [b]. Nous sommes donc en mesure de nous positionner sur un pas de programme quelconque de la RAM et aussi (de plus en plus dur...), de la ROM. Tout cela en jouant avec le seul registre [b], mais avec quelques précautions quand même.

La procédure est donc la suivante:

1) Se positionner "manuellement sur le pas de programme précis de la ROM, sur lequel on veut effectuer le branchement.
2) Rappeler en X la valeur du pointeur de programme par un 'RCL b' bien dosé.
3) Stocker précieusement cette valeur dans le registre alpha (pour ne pas normaliser le contenu de X ???????? SOS, SOS, SOS).
4) Transférer le contenu du registre alpha dans un registre quelconque R().
5) Pour se brancher par la suite sur le pas de programme choisi, il suffit de rappeler la valeur contenue dans R(), et de faire courageusement un 'STO b' sans retour. Et généralement cela fonctionne parfaitement, sauf erreur de manipulation.

Nous allons maintenant en venir au fait de ces lignes, à savoir 'LA ROUTE DE LA ROM', mais via l'utilisation d'un programme contenu dans la PPCROM, et qui répond au petit nom de "XE".

Pour illustrer les facilitées apportées par ce programme, nous allons faire une petite application, par un branchement sur le label "T1" du PPCROM, qui produira un BEEP très particulier, (si tout marche bien!!!). Il faut assigner 'STO b', 'RCL b', 'STO M' à trois touches de la machine, (cette condition est impérative pour la suite de la manoeuvre).

* Tout d'abord, nettoyer le registre alpha de son contenu.
* Se positionner sur "T1" par un GTO "T1", et vérifier en passant en mode PRGM, puis revenir en mode CALCUL.
* Passer en mode USER et effectuer le rappel du registre [b] par 'RCL b'.
* Stocker le contenu bizarroïde de X dans le registre ALPHA à l'aide de 'STO M'.
* Recopier le contenu du registre ALPHA dans un autre registre (par exemple: R01), en effectuant au clavier (en mode ALPHA): 'shift', 'ASTO 01'.
* Effectuer maintenant un 'CAT 1', histoire de sortir le pointeur de la ROM, puis purger le contenu du registre ALPHA, pour être sûr qu'il n'y a pas de triche...
* Rappeler le contenu de R01 en ALPHA, par un 'shift', 'ARCL 01' en mode ALPHA.
* Effectuer maintenant au clavier la dernière opération par XEQ "XE". Si tout a été fait dans les règles, un son modulé doit se faire entendre. Sinon.....

AVANTAGES:

- Rapidité d'exécution du sous programme appelé.
- Gain de place important en RAM (c'est à mon avis l'argument principal).
- Disponibilité permanente du sous programme pour peu que le module soit là.
- Accès a une multitude de sous-programmes existants et disponibles qu'il suffit de retrouver à l'intérieur des modules (bon courage!).

INCONVENIENTS:

- Obligation d'avoir le module appelé dans la machine.
- Impossibilité de remodeler le sous-programme utilisé, comme cela est possible après un COPY RAM.
- Obligation de laisser disponibles tous les registres que va utiliser le sous programme appelé en ROM.
- Nécessité de mettre toujours le module dans le même port, pour ne pas avoir a changer l'adresse du pointeur à chaque nouvelle utilisation du programme.

Voici le listing du programme "XE" pour ceux qui n'ont pas le PPCROM.

```
01 LBL "XE"        13 RTN
02 XEQ 14          14 ⊢###
03 LBL 14          15 X<>M
04 ⊢##             16 X<>N
05 STO N           17 X<>M
06 RDN             18 ⊢##
07 SF 14           19 STO N
08 RCL b           20 X<>D
09 X<>N            21 X<>N
10 FC? 14          22 ⊢##
11 CLA             23 X<>N
12 FC?C 14         24 STO b
```

.....Bonne programmation à tous.

Philippe DAVASE (PPCPC77-PPCT404).

A. PIERARD PC 55

AFFICHAGE DES DONNEES ASCII

```
01♦LBL "GET"
02 "DEBUT"
03 PROMPT
04 SEEKPT
05♦LBL 01
06 GETREC
07 "+ "
08 RCLPT
09 INT
10 ARCL X
11 AVIEW
12 FS? 55
13 GTO 01
14 ALENG
15♦LBL 14
16 DSE X
17 GTO 14
18 GTO 01
19 .END.
```

Lors de l'apparition du texte celuieei reste à l'affichage un laps de temps proportionnel à la longueur de la ligne. Cette séquence est ignorée lors de l'impression.
Le programme que j'ai conçu met le n° de la ligne à la fin.
Si ce n° n'est pas nécessaire on peut supprimer les pas 7 à 10.

CARACTERE SPECIAL SUR HP 41 CX

```
01♦LBL "CaSP"
02 FIX 3
03 "CODE,XXX"
04 PROMPT
05 GETREC
06 PROMPT
07 ENTER↑
08 INT
09 X<>Y
10 FRC
11 1 E3
12 *
13 ENTER↑
14 ENTER↑
15 ALENG
16 -
17 ARCT
18 ATOX
19 R↑
20 XTOA
21 R↑
22 1
23 +
24 CHS
25 ARCT
26 DELREC
27 INSREC
28 CLA
29 XROM 25,51 (ED)
30 .END.
```

MÄRKLIN
MÄRKLIN
RÖSSLER
RÖSSLER
RUTER
RUTER

La fonction ED permet une meilleure gestion du fichier ASCII.
Ayant dû introduire la liste de mes fournisseurs j'ai fait la petite routine ci-contre.
Faire défiler le texte soit par GET soit en mode ED.
La ligne étant trouvée procéder comme suit :
mettre le CODE en DEC, . . .
nombre de lettres ou espaces avant la lettre à changer. R/S
Apparaît alors la ligne changée.
Sans une HP 41 CX remplacer les lignes 28 et 29 par AVIEW

J'espère que ces petits tours de main auront intéressé quelques amis.

Que ceux qui estiment avoir été aidés persévèrent et que les autres qui lisent ces lignes en haussant les épaules prennent leur plume et nous fassent profiter de leurs astuces.

HP philement vôtre

POINTEUR FANTOME SUR HP41

Une anomalie m'est apparue dans le registre alpha de la HP41. Remplissez ce registre avec une succession de virgules et d'espaces, alternativement. Quand le bip vous indiquera qu'il n'y a plus de place, effacez tout. Au moment d'effacer les derniers signes alphanumériques, ralentissez la cadence. Après avoir effacé le dernier, continuez à appuyer sur la touche d'effacement (ayez confiance!) et que remarquez-vous à la gauche de l'écran ? Le pointeur est revenu...

Etrange, n'est il pas?

Philippe TENAND (PPCPC 44)
2 bis, Rue de l'égalité
94300 VINCENNES.

LES CLUBS HP41 DANS LE MONDE.

Chaque club d'utilisateur de HP41 éditant son journal, vous avez peut-être envie de vous abonner a un journal étranger. Voici les plus représentatifs d'Europe et d'ailleurs.

- Club PPC PARIS-CHAPTER 56, rue Jean-Jacques Rousseau 75001 PARIS-FRANCE,
- HP41 CLUB (Schweiz) Plattenstrasse 44 8 152 GLATTBRUG-SUISSE,
- CLUB de Toulouse 77, rue du Cagire 31100 TOULOUSE-FRANCE,
- PPC JOURNAL 2541 W. Camden Place, SANTA-ANA, CALIFORNIA 92704 U.S.A.,
- CCD club 143, Guardinistrasse D.8 MUNCHEN 70 R.F.A.
- PPC UK Astage, Rectory lane, windlesham, SURREY GU 206BW, GRANDE-BRETAGNE,
- PPC Hollande Valentijnkade 2 1094/SR AMSTERDAM-HOLLANDE
- CCD Kempener str. 156, 5060 BERZISH-GLADBACH 2 R.F.A.
- PPC Lausanne case postale 118, 1000 LAUSANNE 20-SUISSE,
- CCD COMPUTERCLUB DEUTSCHLAND Postfach 129, 6242 KRONBERG-ALLEMAGNE

Sachez qu'il existe des clubs PPC en: Australie, Canada Danemark, Norvège, Espagne, Suède, Angleterre, Etats-Unis (ceux qui se sont manifestés à l'ancêtre PPC Journal) et rien qu'aux Etats-Unis il y a 29 clubs (avec journaux), dans 21 états d'Amérique. NOUS NE SOMMES PAS SEULS !

Philippe TENAND (PPCPC 44)

Cher ami,

Quelques précisions:

Les trois club CCD sont un seul et même club, dont l'adresse est: COMPUTERCLUB DEUTSCHLAND e.V Postfach 2129 - 6242 KRONBERG 2 - ALLEMAGNE.
Le club PPC US initialement créé par Richard NELSON, après une sacré polémique, s'est arrangé de façon que Richard donne sa démission et les nouvelles adresses sont: pour le club PPC, P.O.

Box 9599 - FOUNTAIN VALLEY - CA 92728 - 9599 - U.S.A. Ce club édite deux journaux, l'un réservé au calculatrices programmables, l'autre aux ordinateurs. Et Richard a créé un nouveau club avec son propre journal, totalement indépendant de PPC. L'adresse est: CHHU, Richard NELSON - 2545 W. Camden Place - SANTA ANA - CA 92704 - U.S.A.
Enfin un autre club dont tu n'a pas mentionné l'adresse, est le club PPC TN (club australien) et dont le journal est très intéressant. Son adresse est: PPC Technical Notes - Box 512 - Ringwood - VICTORIA - AUSTRALIE.

Philippe

MEMOIRE VIDE PLUS RAPIDEMENT SUR HP41.

Vous avez un lecteur de codes barres, et dans votre HP41 vous voulez effacer un long programme. Vous ne voulez pas faire MEMORY LOST, mais l'attente après un CLP, ALPHA, ALPHA, vous semble interminable. Si vous avez à portée de la main un quelconque programme imprimé en codes barres, vous avez la solution. Prenez votre lecteur de codes barre et entrez dans la machine la première ligne de n'importe quel programme en codes barres, puis appuyez sur la touche d'effacement. La ligne a automatiquement remplacé votre long programme précédent. Faites alors CLP, ALPHA, ALPHA, et le tour est joué. Toute cette suite d'opération est plus rapide que le temps d'attente pour qu'un long programme s'efface. La ligne lue efface juste le programme ou le pointeur était situé, ce qu'un MEMORY LOST n'aurait pas fait. La seconde solution, si vous possédez un lecteur de cartes magnétiques, est d'avoir préparé une carte ou il n'y a rien d'enregistré. Cette carte ne doit pas être neuve, elle doit avoir déjà au moins une fois enregistré quelque chose. Quand vous voulez effacer un long programme, passez cette carte magnétique, c'est tout. De plus, après avoir rempli une HP41 de '+', il est impossible de faire GTO 2000 ou plus, le maximum étant 1999. A part aller à la fin de la mémoire et revenir en arrière pas à pas, ou d'aller au pas 1999 et avancer manuellement, ce qui est très long quand la mémoire est pleine, Hewlett-Packard n'avait pas pensé à ça. On ne pense pas toujours à créer un label là ou on voudra aller plus tard modifier quelquechose, et y aller avec un GTO ALPHAnumérique.

Philippe TENAND (PPCPC n44).

ABRACADABRA

Sur une HP41 en mode alpha, faire (:) ou (SPACE, puis :) semble visuellement la même chose. Pourtant si on actionne la touche d'effacement, dans le premier cas on efface qu'un seul signe, mais dans le second cas deux signes d'un seul coup.

C'est ce que l'on pense en faisant: A, B, C, Virgule, Space, Virgule, puis en appuyant 6 fois

sur la touche d'effacement: la machine efface 8 caractères, puisqu'un 'Space' est un signe en lui même. C'est deux de trop, que le pointeur en fait n'efface pas! La HP41 conserve donc les deux premiers (A et B). La preuve? Vous ne voyez plus rien en alpha, mais sortez-en et remettez vous-y: A et B sont revenus. Le même phénomène que ce soit avec le point, la virgule, ou (:). L'espace blanc est bien confondu avec la fonction nulle (ou 'NULL' codée zéro), ce qui permet l'erreur, car sans mettre de 'Space' il n'y a plus ce phénomène. Les erreurs du pointeur sur le code nul sont connues: avec un code barre, le lecteur confond code nul (barre en haut en programmation synthétique) avec l'effacement (<-). En mode affichage, la HP41 refuse d'afficher un code nul s'il n'est pas précédé d'un code non nul. Seul la programation synthétique permet de générer a volonté un code nul avec la barre en haut de l'affichage. Le code nul (NULL) est une des hantises du programmeur de HP41, surtout du 'NNU' ou 'NON-NORMALIZED-USER' (utilisateur non normalisé), comme dirait William C. Wickes !

Philippe TENAND (PPCPC 44)

HP IL

Lionel ANCELET HP41 VARIATIONS SUR L'HPIL
Les messages du module I/O.

MARTINET/DAVID HP75 ANALYSEUR DE BOUCLE
Le mode "Scope" du HP75.

VARIATIONS SUR L'HPIL

J'ai fait des expériences sur le HPIL pour déterminer quels messages élémentaires sont envoyés sur la boucle par les instructions du module "Extended I/O".

La configuration étudiée correspond à une boucle ne comportant que le contrôleur HPIL, et l'analyseur (qui ne se comporte pas comme un périphérique, puisqu'il retransmet TOUS les messages sans modification). Le contrôleur est une HP41 + module HPIL + module X.I/O, le commutateur du HPIL étant en position "Printer Disable". J'ai supposé que le lecteur était familier avec la terminologie HPIL. Il peut lire "Autour de la Boucle" ou "The HP-IL System" dont le premier est une traduction déguisée.

Les fonctions étudiées sont les fonctions de contrôle, et non celles d'impression ou de stockage de masse (pour l'instant)...

Pour faciliter la lecture, appelons la séquence :
IDY00 AAU RFC AAD01 = Init

```
FINDID.............Init TAD01 RFC SDI UNT RFC IDY00
                        AAU RFC AAD01
FINDAID............Init TAD01 RFC SAI UNT RFC
POLL...............Init IDY00
POLLD..............Init LAD01 RFC PPD RFC UNL RFC
POLLE x............Init LAD01 RFC PPEx RFC UNL RFC
POLLUNC............Init PPU RFC
SRQ?...............Init IDY00
7 DEVL.............Init LAD01 RFC DDL07 RFC UNL RFC
```

```
7 DDL.............DDL07 RFC
INXB (mode ADRON).Init TAD01 RFC SDA UNT RFC
 "  (mode ADROFF).SDA
CLRLOOP...........Init DCL RFC
CLRDEV............Init LAD01 RFC SDC RFC UNL RFC
OUTXB (ADRON).....Init LAD01 RFC DABx UNL RFC
  "   (ADROFF)....DABx
OUTAE (ADRON).....Init LAD01 RFC DABx1  ... DABxn-1
                  ENDxn  UNL RFC
LOCK..............Init LLO RFC
NOTREM............Init NRE RFC
AID...............Init TAD01 RFC SAI UNT RFC
ID................Init TAD01 RFC SDI UNT RFC
NLOOP.............Init
STOPID............Init IFC RFC
TRIGGER...........Init LAD01 RFC GET RFC UNL RFC
INSTAT............Init TAD01 RFC SST UNT RFC
DIR...............Init TAD01 RFC SAI (41 demande
 (sans Drive,      le type d'accessoire pour savoir
  mode AUTOID)     s'il s'agit d'un drive...) UNT
                   RFC IFC RFC
OUTA (CF 17)......Init LAD01 RFC DABx1  ... DABxn
                   DAB13 DAB10 UNL RFC
INA et IND........Init TAD01 RFC SDA UNT RFC IFC
                   RFC
PWRDN.............Init LPD RFC
REMOTE............Init REN RFC Init LAD01 RFC
                   CMD00 RFC UNL RFC
LOCAL.............Init LAD01 RFC GTL RFC UNL RFC
Mise sous tension.IDY35 IFC RFC Init TAD01 RFC SAI
                   UNT RFC
AUTOID............Init TAD01 RFC SAI UNT RFC
MANIO.............Rien sur la boucle
LISTEN............Init LAD01 RFC
STAT..............Init TAD01 RFC SST UNT RFC
XFER (1->2).......Init TAD01 RFC LAD02 RFC SDA UNL
    (ADRON)        RFC UNT RFC
"   (ADROFF)......SDA
SEND x............CMDx RFC
```

Les autres fonctions d'Entrée/Sortie comme INAC, OUTAC, INP, OUTP, etc... marchent comme INXB, OUTXB, et sont affectées par le mode ADRON ou ADROFF.

Je ferai ensuite des expériences avec un ou plusieurs périphériques:

A Suivre !

Lionel ANCELET

ANALYSEUR DE BOUCLE, POUR HP75

En complément du dossier sur l'HPIL paru dans JPC No 20 (décembre 84/janvier 85), nous vous proposons ce mois-ci un petit programme qui saura vous intéresser, pour peu que les messages aient excité votre curiosité.

En effet: la HP41 dispose du module HPIL Development, vendu dans le "Kit convertisseur" référencé HP82166C (4000 à 5000 Frs environ). Pour ce prix là, vous disposez, entre autres, de la fonction "SCOPE". Pour ceux qui n'ont pas eu la chance de connaître la HP41, "SCOPE" permet de suivre en direct le grand feuilleton palpitant de la boucle (autrement dit, de suivre les messages HPIL)...

Le 75 serait-il donc moins puissant que la 41, puisqu'il souffre de la tare de ne point disposer d' "analyseur de boucle" ?
Eh bien NON ! Nous l'avons créé grâce au LEX "RIOWIO" se trouvant lui aussi dans le "Kit convertisseur" et conçu pour le HP75.

PRINCIPE:

Le HP75 comunique avec l'extérieur par l'intermédiaire d'un circuit intégré: le Circuit HPIL 1LB3-0003. C'est lui qui se charge de toutes les opérations d'Entrée/Sortie.

Ce circuit est un processeur spécialisé, qui est commandé par l'intermédiaire de registres internes décrits ci-dessous:

Ils sont au nombre de 8 (0 à 7), chaque registre faisant 8 bits (0 à 7).

R0: Status register
R1: Interrupt Register
R2: Data Register

R3: Parallel Poll Register
R4: Loop Adress Register
R5: Scratchpad Register
R6: Scratchpad Register
R7: Auxiliary input Register

Les registres qui nous intéressent sont les registres R0, R1 et R2. Certains d'entre eux ont une signification différente suivant l'action exercée sur eux (lecture ou écriture).

Le registre 0 est particulier: il contient les informations sur l'état du circuit HPIL (donc du HP75): Etat Controleur actif (Bit 6=1); Etat émetteur actif (Bit 5=1) ou Récepteur actif (Bit 4=1). En mode normal, le bit R0:6 (bit 6 dans R0) est à 1. D'autre part, on ne peut avoir simultanément les états récepteur et émetteur actifs. Le mode "Loop Analyzer" ("SCOPE") s'obtient en mettant à 1 les bits 4 et 5 (48=32+16 en décimal). C'est ce qui est fait en ligne 2000 (WIO 0,48 : écrire dans le registre 0 la valeur 48). Les bits 0 à 3 sont fonction des messages reçus (IFC, RFC, etc).

Quand une instruction HPIL arrive, sa partie "Contrôle" est rangée dans les 3 bits de poids fort de R1, et le reste est placé dans R2.
Le circuit intégré envoie automatiquement un message dès qu'il y a écriture dans R1. Le programme lit donc R1 pour conaître le type de l'instruction (Groupes Données, Commande, Prêt et identification), et ensuite les 8 bits de R2 pour avoir la partie "information" du message.
Cette instruction, une fois décodée est enregistrée dans un fichier texte (édité), et renvoyée, par réécriture de sa partie contrôle dans R1:5-7.

UTILISATION:

Prenez une boucle normale, mettez un contrôleur, 500 g de farine et ses périphériques. Préparez-la en prenant soin de garder le HP75 en dehors.
Préparez le HP75 comme suit: s'il n'était pas déjà déclaré OFFIO, faites-le. Lancez le programme (RUN "SCOPE"). SCOPE édite le fichier texte "HPIL".
Ouvrez la boucle, insérez le HP75, et refermez le tout. Exécutez un ordre HPIL sur le contrôleur, et regardez ce qui se passe sur le HP75 !
Une fois votre séquence terminée, déconnectez le HP75, et arretez-le par [ATTN]. Le fichier HPIL étant édité, vous pouvez visualiser tous les messages.

RESTRICTIONS:

Trois séquences du HP71 ne sont toutefois pas comprises par "SCOPE": il s'agit de "CONTROL ON", "RESTORE IO" en état contrôleur. Si vous éteignez votre HP71, ne le rallumez pas en présence du HP75.

Une seule explication nous paraît plausible:
Durant ces séquences, le HP71 envoie une série de NOP toutes les 47 ms environ, et attend le retour du premier d'entre eux, afin de vérifier la continuité de la boucle. Et le HP75 attend un RFC après NOP..... (?)

REMARQUES:

Vous trouverez, plus loin, quelques séquences type, générées par un HP71. Afin de bien comprendre ces séquences, quelques renseignements supplémentaires ne seront pas superflus:

Configuration de la boucle:
Position 0: HP71 contrôleur
  position A
Position 1: HP82163A (interface vidéo)
Position 2: HP82905B (imprimante 80 colonnes)
Position 3: HP9114A (lecteur de disquettes)
Position 4: HP82161A (lecteur de cassettes)
  position B
Et enfin retour au HP71 contrôleur.

De par le principe du programme "SCOPE", le HP75 est totalement invisible sur la boucle. Il n'est ni périphérique, ni contrôleur. Il a un statut assimilable à celui d'un cable HPIL (Un peu cher le cable, tout de même). De ce fait il n'a pas d'adresse. Nous l'avons mis, pour chaque séquence, en position A (entre HP71 et Vidéo), puis en position B (entre lecteur de cassettes et HP71), ceci pour bien montrer les différences.

Comme vous l'avez peut-être déjà compris, ce programme nécessite la présence du fichier LEX "RIOWIO" pour fonctionner. Il n'est pas très long (117 octets). A bientôt aux réunions...

Pour de basses raisons de place dans JPC, nous n'avons malheureusement pas pu vous expliquer en détail le fonctionnement de ce circuit intégré. Si cela vous intéresse, référez-vous au "Kit convertisseur". Si vous êtes nombreux à écrire, nous irons un peu plus loin dans son étude.

A bientôt, et...
Heureuse programmation !

Michel MARTINET et Pierre DAVID

EXEMPLE 1 : PRINTER IS HP82905B

POSITION A	POSITION B
UNL	UNL
RFC	RFC
TAD 1	TAD 1
RFC	RFC
SDI	SDI
TAD 2	TAD 2
RFC	RFC
SDI	DAB 72 H
DAB 72 H	DAB 80 P
DAB 80 P	DAB 56 8
DAB 56 8	DAB 50 2
DAB 50 2	DAB 57 9
DAB 57 9	DAB 48 0
DAB 48 0	DAB 53 5
DAB 53 5	DAB 66 B
NRD	NRD
DAB 66 B	ETO

EXEMPLE 2 : DEVID$(":TAPE")

POSITION A	POSITION B
UNL	UNL
RFC	RFC
TAD 1	TAD 1
RFC	RFC
SAI	DAB 48 0
DAB 48 0	ETO
TAD 2	TAD 2
RFC	RFC
SAI	DAB 33 !
DAB 33 !	ETO
TAD 3	TAD 3
RFC	RFC
SAI	DAB 16
DAB 16	ETO
UNL	UNL
RFC	RFC
TAD 3	TAD 3
RFC	RFC
SDI	DAB 72 H
DAB 72 H	DAB 80 P
DAB 80 P	DAB 57 9
DAB 57 9	DAB 49 1
DAB 49 1	DAB 49 1
DAB 49 1	DAB 52 4
DAB 52 4	DAB 65 A
DAB 65 A	DAB 13
NRD	NRD
DAB 13	ETO
UNT	UNT
RFC	RFC
UNL	UNL
RFC	RFC

EXEMPLE 3 : SPOLL(2)

POSITION A	POSITION B
UNL	UNL
RFC	RFC
TAD 2	TAD 2
RFC	RFC
SST	DAB 161
DAB 161	DAB 4
DAB 4	ETO
UNT	UNT
RFC	RFC
UNL	UNL
RFC	RFC

PROGRAMME SCOPE

- Cette fonction utilisateur teste le bit B dans l'octet N. Elle renvoie 1 s'il est vrai.

```
================================================================
 500 DEF FNT(N,B) = INT(2*FP(N/2^(B+1)))

================================================================
   - Programme principal.
1000 DIM D$[24]
   - Les messages sont stockés dans un fichier texte appelé HPIL. Le programme SCOPE l'édite
     automatiquement dès son lancement.
1010 EDIT "HPIL",TEXT
   - Les éventuelles lignes sont supprimées pour le nouveau "scope".
1020 DELETE 1,9999
1030 ASSIGN # 1 TO "HPIL",TEXT
1040 INTEGER R1,R2
   - Mnémoniques des groupes données et identification.
1050 C$='DABDSRENDESRCMDRDYIDYISR'
1060 DELAY 0
   - Permet au HP75 d'attendre indéfiniment une réponse d'un appareil de la boucle.
1070 STANDBY ON
   - Boucle d'attente.
2000 WID 0,48
2010 R1=RID(1)
2020 IF NOT FNT(R1,2) THEN 2010
2030 R2=RID(2)
2040 R1=32*INT(R1/32)
2050 WID 1,R1
2060 R1=R1/32
2070 D$=C$[R1*3+1,R1*3+3]&' '&STR$(R2)
2080 ON R1+1 GOSUB 3000,3000,3000,3000,4000,5000,3000,3000
2090 PRINT # 1 ; D$
2100 PRINT D$
2110 WID 2,R2
2120 GOTO 2000
   - Groupe données et groupe identification.
3000 IF R2<32 OR R2>126 THEN RETURN
   - Affichage si le caractère est représentable.
3010 D$=D$&' '&CHR$(R2)
3020 RETURN
   - Groupe commande.
4000 IF R2=0 THEN D$='NUL'
4010 IF R2=1 THEN D$='GTL'
4020 IF R2=4 THEN D$='SDC'
4030 IF R2=5 THEN D$='PPD'
4040 IF R2=8 THEN D$='GET'
4050 IF R2=15 THEN D$='ELN'
4060 IF R2=16 THEN D$='NOP'
4070 IF R2=17 THEN D$='LLO'
4080 IF R2=20 THEN D$='DCL'
4090 IF R2=21 THEN D$='PPU'
4100 IF R2=24 THEN D$='EAR'
4110 IF R2>31 AND R2<63 THEN D$='LAD '&STR$(R2-32)
4120 IF R2=63 THEN D$='UNL'
4130 IF R2>63 AND R2<95 THEN D$='TAD '&STR$(R2-64)
4140 IF R2=95 THEN D$='UNT'
4150 IF R2>95 AND R2<127 THEN D$='SAD '&STR$(R2-96)
4160 IF R2=127 THEN D$='ISA'
```

```
4170 IF R2>127 AND R2<144 THEN D$='PPE '&STR$(R2-128)
4180 IF R2=144 THEN D$='IFC'
4190 IF R2=146 THEN D$='REN'
4200 IF R2=147 THEN D$='NRE'
4210 IF R2=154 THEN D$='AAU'
4220 IF R2=155 THEN D$='LPD'
4230 IF R2>159 AND R2<192 THEN D$='DDL '&STR$(R2-160)
4240 IF R2>191 AND R2<224 THEN D$='DDT '&STR$(R2-192)
4250 RETURN
  - Groupe prêt.
5000 IF R2=0 THEN D$='RFC'
5010 IF R2=64 THEN D$='ETO'
5020 IF R2=65 THEN D$='ETE'
5030 IF R2=66 THEN D$='NRD'
5040 IF R2=96 THEN D$='SDA'
5050 IF R2=97 THEN D$='SST'
5060 IF R2=98 THEN D$='SDI'
5070 IF R2=99 THEN D$='SAI'
5080 IF R2=100 THEN D$='TCT'
5090 IF R2>127 AND R2<159 THEN D$='AAD '&STR$(R2-128)
5100 IF R2=159 THEN D$='IAA'
5110 IF R2>159 AND R2<191 THEN D$='AEP '&STR$(R2-160)
5120 IF R2=191 THEN D$='IEP'
5130 IF R2>191 AND R2<223 THEN D$='AES '&STR$(R2-192)
5140 IF R2=223 THEN D$='IES'
5150 IF R2>223 AND R2<255 THEN D$='AMP '&STR$(R2-224)
5160 IF R2=255 THEN D$='IMP'
5170 RETURN
```

HP 71B

HP71 ASSEMBLEUR

Laurent ISTRIA	CONTRASTE ? La fonction CNTRST.
Jean-Jacques MOREAU	ASSEMBLER SANS ASSEMBLEUR Méthode d'assemblage sans module Forth/Assembleur.
Michel MARTINET	LE DESASSEMBLEUR Désassembleur HP71.

HP71 FORTH

DAVID/WEIL	FORTH RECURSIF ? Récursivité du Forth: les bonnes adresses.

HP71 BASIC

Serge VAUDENAY	L'EVENEMENT Compilateur musical très performant.
Jean-Jacques MOREAU	INTEGRALES Propres ou généralisées sans module maths.
Jean-Pierre BONDU	MATRICES A EDITER Editeur de matrices avec module Maths.
Olivier ARBEY	PAS PROPRES, MES VALEURS ? Polynôme caractéristique avec module Maths.

QUEL CONTRASTE AVEZ VOUS ?

Comme chacun le sait, le HP71B est un appareil tourné vers l'avenir. En effet, Hewlett Packard a eu la gentillesse de nous fournir toutes sortes de spécifications qui lui sont relatives.

Grâce aux IDS, il nous est possible de trouver un grand nombre d'adresses dont le contenu est bien intéressant. Ainsi, a l'adresse 2E3FE se trouve la valeur du contraste courant. Un simple fichier LEX nous permet alors de récupérer cette valeur.

Laurent ISTRIA (#3)

ASSEMBLAGE SANS ASSEMBLEUR

Le but de cet article est de vous permettre de faire de l'assembleur sans module assembleur.

La méthode en elle-même est fort simple (encore fallait-il y penser): il faut d'abord traduire les mnémoniques du programme à assembler en codes hexadécimaux, puis les placer dans une variable alphanumérique de taille convenable. Nous nommerons cette variable A$. Il est alors nécessaire de disposer d'un port indépendant absolument vide (par exemple le port 0.01). Pour cela, exécuter CLAIM PORT(.01), puis FREE PORT(.01). A l'adresse hexadécimale '40000' se trouvent les 8 codes : B3DDDDDE. Ceci est vrai si votre système est configuré sans aucun module supplémentaire. Autrement l'adresse est différente à x(4096) octets près.
Ils signalent à la machine l'existence du port (0.01). - A propos, pourquoi ne tenterait-on pas de créer des ports indépendants de 8, voire 16 Ko? Si maintenant l'on copiait un programme dans le port (0.01), il viendrait se placer derrière le E, donc à l'adresse '40008'.

Il nous suffit donc de placer A$ à cet endroit (A$ contient, rappelons-le, le programme assembleur et l'en-tête du fichier LEX correspondant), pour disposer d'un fichier LEX.

Prenons comme exemple le fichier KEYWAIT paru dans JPC No 20 (Décembre 84/Janvier 85). Entrez dans A$ l'ensemble des codes se trouvant dans la 3ème colonne de la page 52 (un programme à la fin de cet article vous aidera dans cette manipulation). Videz le port (0.01) comme décrit précédemment. Tapez alors: POKE '40008',A$.
Un catalogue vous assurera de la présence d'un fichier LEX de 55 octets dénommé KEYWAIT. La fonction KEYWAIT$ n'est pas encore exécutable, car le HP71 ignore partiellement son existence. Pour y remédier, copier le fichier en mémoire centrale (COPY KEYWAIT).
Désormais, vous pourrez utiliser la fonction KEYWAIT$ comme n'importe laquelle des autres fonctions d'origine du HP71.

REMARQUES:
Il est encore indispensable de travailler en mémoire indépendante à cause de l'organisation de la mémoire principale (présence de la pile de commande...), et de l'impossibilité d'utiliser la fonction POKE sur la totalité d'un fichier. Cette dernière contre-indication de posera plus aucun problème lorsque nous disposerons des fonctions SEEK$ et SOKE, encore en préparation à l'heure à laquelle j'écris ces lignes. Il suffira alors de créer en mémoire (indépendante ou principale) un fichier de type quelconque (BASIC...) et de longueur la longueur du fichier LEX à créer, puis de faire SOKE ADDR$('NOM'),A$.

Nous tenons à mettre en garde le lecteur en garde contre le danger permanent de Memory Lost, qui détruire tous les ports indépendants.

Programme d'aide à la frappe:
Pierre m'a demandé d'écrire ce programme pour faciliter la création des fichiers LEX. Il utilise des entrées compatibles avec la nouvelle présentation dans JPC des fichiers LEX, c'est à dire: un ensemble de dix codes hexadécimaux, suivis d'un code de contrôle, somme des octets modulo 256, code qui permettra une vérification sommaire des entrées.

Le programme commence par vous demander les 5 premiers octets du fichier, puis les 5 suivants, etc...
Il affiche en permanence:
A gauche, le No de la ligne en cours de traitement
A droite, le No de la ligne précédente et son code de contrôle.

Une fois le programme mémorisé, vous pouvez ou le stocker immédiatement (c'est la "phase d'assemblage"), ou visualiser chacune des lignes entrées, le No de la ligne et le code de contrôle de celle-ci étant affichés (c'est l'option "Montre").
Dans ce dernier cas, une pression sur la touche [M] permet de modifier la ligne courante. Une pression sur [END LINE] ramène à la ligne suivante.

Jean-Jacques MOREAU

CODES DE KEYWAIT CODE DE CONTROLE

B4 54 95 75 14	188
94 45 02 80 2E	91
00 61 10 80 90	19
48 37 00 02 51	61
01 00 00 00 F7	18
10 00 00 00 00	10
00 00 0E 10 00	10
FF B4 54 95 75	124
14 94 45 42 10	117
1F F0 01 36 10	122
B1 37 10 98 F2	73
C6 00 45 11 19	210
13 51 18 13 48	56
D8 AC A1 8F 12	56
70 06 BD F0 00	82

DESASSEMBLEUR

Voici comme annoncé le mois dernier dans JPC une version courte de mon désassembleur HP71B. Cette version fonctionne sans l'HPIL, ce qui est plus compatible avec un article paraissant dans notre journal préféré. En effet, une version HPIL etc. aurait considérablement augmentée la place occupée par l'article et surtout la place prise par les fichiers dans la mémoire centrale de Titan. La version suivante est à mon sens la plus courte tout en gardant une bonne souplesse pour le désassemblage. J'ai choisi le basic comme langage pour deux raisons: la première, tout le monde parle "Basic" et la seconde est que la moindre déclaration Forth dans la mémoire de Titan occupe beaucoup d'octets (FORTHRAM).

Le désassembleur tient dans trois fichiers: un fichier basic comportant deux routines (SUB DESAS(n$) et MNEMO(m$,n$)), un fichier texte DESAT contenant toutes les mnémoniques de l'assembleur du HP71B (lignes 0 à 384) et enfin un petit LEX d'"ID" 92 contenant 7 fonctions alphanumériques. Il existe une table contenant les mnémoniques dans la ROM Forth/Assembleur qui sera bientôt décodée ce qui permettra de remplacer le fichier DESAT et la SUB MNEMO par quelques octets et permettra de faire tenir l'ensemble du désassembleur dans moins de 4 ko (un port).

DESAS(n$): c'est la routine principale, n$ peut être une adresse ou un nom de fichier. Si le fichier est privatisé, DESAS lit malgré tout. On a donc accès à toute la carte mémoire.

MNEMO(m$,n$): décode les codes HEXA se trouvant dans m$ et renvoie la mnémonique dans n$. Si le code (m$) ne correspond à rien, MNEMO renvoie m$ dans n$ suivi d'un point d'interrogation. Les mnémoniques sont stockées dans le fichier texte DESAT. Il est très important que ces mnémoniques soient classées par ordre (de [00 RTNSXM] à [FF D=-D-1 A] respectivement des lignes 0 à 384 ou 1 à 385 si vous utilisez l'éditeur de texte.

DESAL: LEX092 comprenant les sept fonctions alphanumériques suivantes:

ADBUF$(n$): cette fonction est de Pierre David, elle renvoie l'adresse d'un BUFFER. Exemple: ADBUF$("BFC"). "BFC" est le buffer "bLEX" où se trouve toutes les adresses des LEX présents dans la mémoire du HP71B. Voir article de Pierre dans JPC-AVRIL.

ASC$(n$): transforme une chaîne de caractères en ASCII, c'est-à-dire en ne gardant que les caractères dont les codes sont compris entre 32 et 126. Les autres sont remplacés par des points (caractère 46).

HTA$: transforme une chaîne alphanumérique représentant des caractères codés HEXA en ASCII. HTA$ ne remplace pas les codes non ASCII (voir ASC$) par des points. Il faudra donc utiliser la fonction ASC$ en complément.

KEYWAIT$: cette fonction est de Hewlett-Parkard, voir JPC décembre-janvier, où elle est désasssemblée en long en large et en travers.

RED$(n$) : supprime les blancs en début et en fin de chaîne. (ADDR$(" 12345 ")) donne une erreur alors que ADDR$(RED$(" 12345 ")) fonctionne correctement.

REV$(n$) : inverse tout simplement le contenu d'une chaîne alphanumérique (REV$("aReUh")="hUeRa").

SEEK$(n$,m) : équivalent à PEEK$(n$,m) sans tenir compte des zones privatisées. SEEK$ est plus rapide que PEEK$ (15%).

Enfin voici quelques bonnes adresses pour utiliser DESAS:
LEX000
Main Table : 1E59F
LEX001
Main Table : 1DD6C
LEX002 (MATHROM) lorsque MATHROM est en 48008
Main Table : 48095
LEX047 (FORTHROM) lorsque FORTHROM est en 58008
Main Table : 58046
LEX255 (HPILROM) lorsque HPILROM est en 40008
Main Table: 40095

Voilà, il faut environ 8300 octets de libre dans la mémoire et environ 200 de plus pour faire tourner DESAS, et surtout beaucoup de courage pour l'entrer en mémoire. Il vous faudra avant d'écrire le BASIC assembler le fichier texte DESOURCE listé, plus loin pour obtenir les sept fonctions alphanumériques complémentaires.
A la ligne 1080 ('W': DISP.....), si vous désirez afficher les op-codes des mémoniques, ils se trouvent dans B$[1,P-1]. Je n'ai pas programmé leur affichage pour éviter les chaînes trop longues et les déroulements d'écran répétés.
Enfin une dernière chose: n'écrivez pas les apostrophes en pupitrant les labels alpha, vous gagnerez quelques octets. Bon courage et à tout de suite ! Les feignants pourront se procurer tout cela aux réunions mensuelles.

MODE D'EMPLOI DU DESASSEMBLEUR

MODE HEXA

{An} (n de 1 à 8) NIBASC
Interprète les quartets à partir de l'adresse courante comme de l'ASCII. Seuls les caractères ASCII sont affichés (caractères de 32 à 126), les autres sont remplacés par un point (caractère 46).
```
CALL DESAS("1F0A9")
1F0A9 [94D41474]
{A8}
1F0A9 [NIBASC 'IMAGE...']
{A5}
1F0A9 [NIBASC 'IMAGE']
```

{Cn} (n de 1 à 4) CON(n)
Interprète les n quartets de l'adresse comme une constante décimale si n est plus petit que 5 et comme une constante HEXA si n égale 5.
```
1F0A9 [NIBASC 'IMAGE']
{C1}
1F0A9 [CON(1) 9]
{C5}
1F0A9 [CON(5) #14D49]
{C3}
1F0A9 [CON(3) 3401]
```

{CHn} (n de 1 à 5) CON(n)
Interprète les n quartets de l'adresse courante comme une constante HEXA.
```
1F0A9 [CON(3) 3401]
```

{CH3}
1F009 [CON(3) #D49]

{H} HEXA
Passe en mode HEXA sans changer l'adresse.
1F0A9 [CON(3) #D49]
{H}
1F0A9 [94D41474]

{Nn} (n de 1 à 16) NIBHEX n
Interprète les quartets comme de l'HEXA tout simplement, la pression de "." est considérée comme valeur 10.
1F0A9 [94D41474]
{N9}
1F0A9 [NIBHEX 94D414745]
{N.0} {N10}
1F0A9 [NIBHEX 94D41474544]
{N.1} {N11}
1F0A9 [NIBHEX 94D4147454F]

{Rn} (n de 1 à 5) REL(n)
Interprète les quartets comme une adresse relative
1F0A9 [NIBHEX 94D4147454F]
{R2}
1F0A9 [REL(2) #49]
{R3}
1F0A9 [REL(3) #D49]

{T}
Transforme l'"offset" affiché par l'instruction {Rn} en adresse absolue.
1F0A9 [REL(3) #D49]
{T}
1F0A9 [REL(3) #1EDF2] (0)
L'indicateur binaire 0 indique que l'on est en présence d'une adresse absolue.

{>}
Incrémente l'adresse d'un quartet, le programme revient en mode HEXA. A noter que si la chaîne affichée est d'une longueur supérieure à 22 caractères, la touche {>} sert à déplacer l'affichage. Il faut alors utiliser la commande {H} pour accéder à la commande {>}.

{<}
Décrémente l'adresse d'un quartet, le programme revient en mode HEXA. Idem {>} pour l'affichage s'il est supérieur à 22 caractères.

{v}
Incrémente l'adresse de 8 quartets, le programme revient en mode HEXA.

{^}
Décrémente l'adresse de 8 quartets, le programme revient en mode HEXA.

{g{v}}
Incrémente l'adresse de 128 quartets, le programme revient en mode HEXA.

{g{^}}
Décrémente l'adresse de 128 quartets, le programme revient en mode HEXA.

{(} GOSUB
Branchement à une adresse absolue donnée par {RnT} ou par le désassembleur avec possibilité de retour. Noter que l'on peut opérer un branchement que lorsque l'indicateur binaire 0 est affiché. On est alors en présence d'une adresse absolue et non d'un "offset".
1F0A9 [94D41474]
{R3T}
1F0A9 [REL(3) #1EDF2] (0)
{(}
1EDF2 [EDF437C9] (1)
L'indicateur binaire 1 indique un niveau de sous-programme (GOSUB). Le niveau de sous-programme est représenté par les indicateurs binaires 1 à 4.

{)} RETURN
Retour à l'adresse précédente (équivalent à RETURN). Noter que l'adresse est alors incrémentée d'une valeur égale au nombre de quartets de l'instruction d'appel (REL(3)). L'indicateur binaire 1 s'efface.
1EDF2 [EDF437C9] (1)
{)}
1F0AC [4147454F]

{=} GOTO
Identique à {(} (GOSUB) mais sans possibilité de retour (GOTO). Là aussi le branchement ne peut se faire qu'en présence d'une adresse absolue (indicateur binaire 0).

{ATTN}
Quitte le désassembleur.
1F0AC [4147454F]
{ATTN}
>

{END LINE}
Incrémente l'adresse courante d'une valeur égale au nombre de quartets de l'instruction affichée. Le programme revient en mode HEXA. Si aucune instruction n'est affichée, {END LINE} n'a pas d'effet.

CALL DESAS("1F246")
1F246 [90555257]
{C1}
1F246 [CON(1) 9]
{END LINE}
1F247 [05552574]
{A5}
1F247 [NIBASC 'PURGE']
{END LINE}
1F251 [BED25144]
{C2}
1F251 [CON(2) 235]
{END LINE}
1F253 [D2514449]
{C1}
1F253 [CON(1) 13]
{END LINE}
1F254 [25144494]
{A7}
1F254 [NIBASC 'RADIANS']
{END LINE}
1F262 [4D525144]
{C2}
1F262 [CON(2) 212]
{END LINE}
1F264 [5251444E]
Etc. C1 A(CON(1)(n+1)/2) C2 ..... C'est la table texte du LEX001 de la ROM basic du 71B.

MODE DESASSEMBLEUR

{D} DESASSEMBLE
Dans l'exemple ci-desous, la fonction LEN(N$) est désassemblée, son point d'entrée se trouve dans le LEX001 du système d'exploitation à l'adresse "0BF28". La fonction n'accepte qu'un seul paramètre alphanumérique non optionnel, ceci est codé devant le point d'entrée:
0BF27 [NIBHEX 1] (nombre minimum de paramètres),
0BF26 [NIBHEX 1] (nombre maximum de paramètres) et
0BF25 [NIBHEX 4] (paramètre alphanumérique).
CALL DESAS("0BF28")
0BF28 [7C0E137C]
{D}
0BF28 [GOSUB #E0C]
{T}
0BF28 [GOSUB #0BD38] (0)
Avant l'exécution d'une fonction, ses paramètres sont logés dans un endroit de la mémoire appelé "Math Stack". "0BD38" est le point d'entrée d'une routine appelée POP1S (POP 1 String argument off math stack). En entrée POP1S n'a besoin que de D1 pointant le sommet de la "Math Stack" (ceci est déjà fait avant le branchement au point d'entrée). En sortie: D1 pointe la fin de l'en-tête de la chaîne alpha (D1=D1+ 16), P= 0 et A(A) contient la longueur de la chaîne en quartets (champs A du registre A).
{END LINE}
0BF2C [CD1EX]
{END LINE}
0BF2F [C=C+A A]
{END LINE}
0BF31 [D1=C]
{END LINE}
0BF34 [D1=D1- 16]
{END LINE}
0BF37 [P= 5]
{END,LINE}
0BF39 [A=0 P]
{END LINE}
0BF3C [ASRB]
{END LINE}
0BF3F [GOSUB #F84]
{T}
0BF3F [GOSUB #0BEC7] (0)
Point d'entrée de la routine HDFLT (convert Hex integer to Decimal FLoaTing-point). En entrée HDFLT n'a besoin que du nombre hexa dans A(A). En sortie, A(W) contient le nombre décimal, P=0 et "Carry" est à 1. HDFLT fait appel à une autre routine: HEXDEC (0ECAF).
{END LINE}
0BF43 [GOTO #9CF]
{T}
0BF43 [GOTO #0B913] (0)

{=}
0B913 [DAT1=A W]
{END LINE}
0B917 [GOLONG #3923]
{T}
0B917 [GOLONG #0F23C]
Point d'entrée de la routine EXPR. Cette routine passe le contrôle des opérations au "Expression Execution Controller" après l'évaluation d'une fonction. En entrée: D0=PC (Program Counter) (LEN ne touche pas à D0), et D1 pointe la première expression se trouvant sur la "Math Stack".

{L}
Transforme toutes les expressions [LC(n) H..H] ou [LCHEX H..H] en [LCASC 'A..A'] si la longueur de la chaine hexa est paire.

Voilà, j'espère que vous aurez le courage de rentrer les trois fichiers "DESAS", "DESAT" et "DESAL" dans la merveilleuse mémoire de votre HP71B et que vous passerez vos longues journées pluvieuses de cette fin d'hiver à désassembler Titan. Quelle superbe machine ! Non ?

Heureuse Programmation.

Michel MARTINET (PPC-P12)

FICHIER TEXTE DESOURCE

```
      LEX    'DESAL'
      ID     #5C
      MSG    0
      POLL   0

ADHEAD EQU   #181B7
ARGERR EQU   #0BF19
CKSREQ EQU   #00721
D=AVMS EQU   #1A460
EXPR   EQU   #0F23C
FLTDH  EQU   #1B223
IOFNDO EQU   #118C1
KEY$   EQU   #1ACAB
POP1R  EQU   #0E8FD
POP1S  EQU   #0BD38
REV$   EQU   #1B38E
REVPOP EQU   #0BD31
SLEEP  EQU   #006C2
STKCHR EQU   #18504

       ENTRY ADBUF
       CHAR  #F
       ENTRY ASCII
       CHAR  #F
       ENTRY HTA
       CHAR  #F
       ENTRY KWAIT
       CHAR  #F
       ENTRY REDUC
       CHAR  #F
       ENTRY REVER
       CHAR  #F
       ENTRY SEEK
       CHAR  #F

       KEY   'ADBUF$'
       TOKEN 1
       KEY   'ASC$'
       TOKEN 2
       KEY   'HTA$'
       TOKEN 3
       KEY   'KEYWAIT$'
       TOKEN 4
       KEY   'RED$'
       TOKEN 5
       KEY   'REV$'
       TOKEN 6
       KEY   'SEEK$'
       TOKEN 7

       ENDTXT

       NIBHEX 411
REVER  GOSBVL REV$

Expr   GOVLNG EXPR

       NIBHEX 411
REDUC  D=0    S
```

```
REDUC1 GOSBVL REVPOP
       LCASC  ' '
       CDEX   B

REDUC2 ?A=0   A
       GOYES  REDUC3
       C=DAT1 B
       ?C#D   B
       GOYES  REDUC3
       GOSUB  D1ACHR
       GONC   REDUC2

REDUC3 GOSUB  Adhd
       D=D+1  S
       GONC   REDUC1
       GOTO   Expr

       NIBHEX 411
ASCII  CD1EX
       D1=C
       CDEX   A
       GOSBVL POP1S

ASCII1 ?A=0   A
       GOYES  ASCII4
       C=DAT1 B
       B=C    B
       LCHEX  7E20
       ?B<C   B
       GOYES  ASCII2
       CSR    W
       CSR    W
       ?B<=C  B
       GOYES  ASCII3

ASCII2 LCASC  '.'
       DAT1=C B

ASCII3 GOSUB  D1ACHR
       GONC   ASCII1

ASCII4 CDEX   A
       D1=C
       GOTO   Expr

D1ACHR D1=D1+ 2
       A=A-1  A
       A=A-1  A
       RTN
```

```
ERRA   GOVLNG ARGERR

       NIBHEX 411
HTA    GOSBVL POP1S
       ?A=0   A
       GOYES  HTA2
       C=A    B
       ST=C
       ?ST#0  1
       GOYES  ERRA
       LC(5)  4
       R0=C

HTA1   R1=A
       ACEX   A
       GOSUB  ASCHEX
       A=R0
       R0=C
       CD1EX
       C=C-A  A
       D1=C
       CR0EX
       DAT1=C B
       C=R0
       C=C+A  A
       D1=C
       A=A+1  A
       A=A+1  A
       R0=A
       A=R1
       LC(5)  4
       B=C    A
       A=A-B  A
       ?A#0   A
       GOYES  HTA1
       A=R0
       A=A-B  A
       CD1EX
       C=C-A  A
       C=C-A  A
       D1=C

HTA2   GOSUB  Adhd
       GOTO   Expr

Adhd   C=A    A
       CSL    W
       CSL    W
       LCHEX  F
```

```
        D1=D1- 16
        DAT1=C 7
        RTN

        NIBHEX 411
ADBUF   GOSBVL REVPOP
        ?A=0   A
        GOYES  ERR
        LCHEX  00006
        ?A>C   A
        GOYES  ERR
        GOSUB  ASCHEX
        CD1EX
        D1=C
        CR1EX
        C=D    X
        GOSBVL IOFNDO
        GONC   ADB5
        D1=D1- 7
        C=R1
        CD1EX
        A=0    W
        A=C    A
        GOSBVL D=AVMS

ADB1    B=0    B
        B=A    P
        LCHEX  9
        ?B>C   P
        GOYES  ADB2
        LCHEX  30
        GOTO   ADB3

ADB2    LCHEX  37

ADB3    C=C+B  B
        GOSBVL STKCHR
        ASR    W
        ?A#0   A
        GOYES  ADB1
        GOTO   ADB4

ERR     GOVLNG ARGERR

ADB5    C=R1
        D1=C

ADB4    ST=1   0
        GOSBVL ADHEAD
        GOSBVL REV$
        GOTO   Expr

ASCHEX  D=0    W

AH1     C=DAT1 B
        B=C    B
        LCASC  '0'
        ?B<C   B
        GOYES  ERR
        LCASC  '9'
        ?B<=C  B
        GOYES  AH3
        LCASC  'A'
        ?B<C   B
        GOYES  ERR
        LCASC  'F'
        ?B<=C  B
        GOYES  AH2
        LCASC  'a'
        ?B<C   B
        GOYES  ERR
        LCASC  'f'
        ?B>C   B
        GOYES  ERR
        LCHEX  57
        GOTO   AH4

AH2     LCHEX  37
        GOTO   AH4

AH3     LCHEX  30

AH4     BCEX   B
        C=C-B  B
        DSL    W
        D=D+C  B
        GOSUB  D1ACHR
        ?A#0   A
        GOYES  AH1
        C=D    A
        RTN

Argerr  GOVLNG ARGERR

        NIBHEX 8422
SEEK    GOSBVL POP1R
        GOSBVL FLTDH
        GONC   Argerr
        D1=D1+ 16
        AROEX
```

```
        GOSBVL REVPOP
        ?A=0   A
        GOYES  Argerr
        LCHEX  0000A
        ?A>C   A
        GOYES  Argerr
        GOSUB  ASCHEX
        A=C    A
        GOSBVL D=AVMS
        CD1EX
        D1=C
        R1=C

SEEK1   C=R0
        ?C=0   A
        GOYES  SEEK4
        C=C-1  A
        R0=C
        AD0EX
        C=0    B
        C=DAT0 P
        AD0EX
        A=A+1  A
        B=C    B
        LCHEX  9
        ?B>C   B
        GOYES  SEEK2
        LCHEX  30
        GOTO   SEEK3

SEEK2   LCHEX  37

SEEK3   BCEX   B
        C=C+B  B
        GOSBVL STKCHR
        GOTO   SEEK1

SEEK4   ST=0   0
        GOVLNG ADHEAD

        NIBHEX 00
KWAIT   CD0EX
        R0=C
        CD1EX
        R1=C

KWAIT1  GOSBVL SLEEP
        GOC    KWAIT2
        C=R1
        D1=C
        C=R0
        D0=C
        GOVLNG KEY$

KWAIT2  GOSBVL CKSREQ
        GOTO   KWAIT1

        END
```

FICHIER TEXTE DESAT

```
00 RTNSXM       10B R3=C        151a DAT1=A     813 DSLC        8BF ?D<=C A     Ab2 C=0         C9 C=C+B A
01 RTN          10C R4=C        152a A=DAT0     814 ASRC        8Chhhh GOLONG   Ab3 D=0         CA A=A+C A
02 RTNSC        110 A=R0        153a A=DAT1     815 BSRC        8Dhhhhh GOVLNG  Ab4 A=B         CB C=C+D A
03 RTNCC        111 A=R1        154a DAT0=C     816 CSRC        8Ehhhh GOSUBL   Ab5 B=C         CC A=A-1 A
04 SETHEX       112 A=R2        155a DAT1=C     817 DSRC        8Fhhhhh GOSBVL  Ab6 C=A         CD B=B-1 A
05 SETDEC       113 A=R3        156a C=DAT0     81C ASRB        9a0 ?A=B        Ab7 D=C         CE C=C-1 A
06 RSTK=C       114 A=R4        157a C=DAT1     81D BSRB        9a1 ?B=C        Ab8 B=A         CF D=D-1 A
07 C=RSTK       118 C=R0        158x DAT0=A     81E CSRB        9a2 ?A=C        Ab9 C=B         D0 A=0 A
08 CLRST        119 C=R1        159x DAT1=A     81F DSRB        9a3 ?C=D        AbA A=C         D1 B=0 A
09 C=ST         11A C=R2        15Ax A=DAT0     821 XM=0        9a4 ?A#B        AbB C=D         D2 C=0 A
0A ST=C         11B C=R3        15Bx A=DAT1     822 SB=0        9a5 ?B#C        AbC ABEX        D3 D=0 A
0B CSTEX        11C C=R4        15Cx DAT0=C     824 SR=0        9a6 ?A#C        AbD BCEX        D4 A=B A
0C P=P+1        120 AR0EX       15Dx DAT1=C     828 MP=0        9a7 ?C#D        AbE ACEX        D5 B=C A
0D P=P-1        121 AR1EX       15Ex C=DAT0     82F CLRHST      9a8 ?A=0        AbF CDEX        D6 C=A A
0Ea0 A=A&B      122 AR2EX       15Fx C=DAT1     831 ?XM=0       9a9 ?B=0        Ba0 A=A-B       D7 D=C A
0Ea1 B=B&C      123 AR3EX       16x D0=D0+      832 ?SB=0       9aA ?C=0        Ba1 B=B-C       D8 B=A A
0Ea2 C=C&A      124 AR4EX       17x D1=D1+      834 ?SR=0       9aB ?D=0        Ba2 C=C-A       D9 C=B A
0Ea3 D=D&C      128 CR0EX       18x D0=D0-      838 ?MP=0       9aC ?A#0        Ba3 D=D-C       DA A=C A
0Ea4 B=B&A      129 CR1EX       19hh D0=(2)     84n ST=0        9aD ?B#0        Ba4 A=A+1       DB C=D A
0Ea5 C=C&B      12A CR2EX       1Ahhhh D0=(4)   85n ST=1        9aE ?C#0        Ba5 B=B+1       DC ABEX A
0Ea6 A=A&C      12B CR3EX       1Bhhhhh D0=(5)  86n ?ST=0       9aF ?D#0        Ba6 C=C+1       DD BCEX A
0Ea7 C=C&D      12C CR4EX       1Cx D1=D1-      87n ?ST=1       9b0 ?A>B        Ba7 D=D+1       DE ACEX A
0Ea8 A=A!B      130 D0=A        1Dhh D1=(2)     88n ?P#         9b1 ?B>C        Ba8 B=B-A       DF CDEX A
0Ea9 B=B!C      131 D1=A        1Ehhhh D1=(4)   89n ?P=         9b2 ?C>A        Ba9 C=C-B       E0 A=A-B A
0EaA C=C!A      132 AD0EX       1Fhhhhh D1=(5)  8A0 ?A=B A      9b3 ?D>C        BaA A=A-C       E1 B=B-C A
0EaB D=D!C      133 AD1EX       2n P=           8A1 ?B=C A      9b4 ?A<B        BaB C=C-D       E2 C=C-A A
0EaC B=B!A      134 D0=C        3ih..h LCHEX    8A2 ?A=C A      9b5 ?B<C        BaC A=B-A       E3 D=D-C A
0EaD C=C!B      135 D1=C        4hh GOC         8A3 ?C=D A      9b6 ?C<A        BaD B=C-B       E4 A=A+1 A
0EaE A=A!C      136 CD0EX       400 RTNC        8A4 ?A#B A      9b7 ?D<C        BaE C=A-C       E5 B=B+1 A
0EaF C=C!D      137 CD1EX       420 NOP3        8A5 ?B#C A      9b8 ?A>=B       BaF D=C-D       E6 C=C+1 A
0EF0 A=A&B A    138 D0=AS       5hh GONC        8A6 ?A#C A      9b9 ?B>=C       Bb0 ASL         E7 D=D+1 A
0EF1 B=B&C A    139 D1=AS       500 RTNNC       8A7 ?C#D A      9bA ?C>=A       Bb1 BSL         E8 B=B-A A
0EF2 C=C&A A    13A AD0XS       6hhh GOTO       8A8 ?A=0 A      9bB ?D>=C       Bb2 CSL         E9 C=C-B A
0EF3 D=D&C A    13B AD1XS       6300 NOP4       8A9 ?B=0 A      9bC ?A<=B       Bb3 DSL         EA A=A-C A
0EF4 B=B&A A    13C D0=CS       64000 NOP5      8AA ?C=0 A      9bD ?B<=C       Bb4 ASR         EB C=C-D A
0EF5 C=C&B A    13D D1=CS       7hhh GOSUB      8AB ?D=0 A      9bE ?C<=A       Bb5 BSR         EC A=B-A A
0EF6 A=A&C A    13E CD0XS       800 OUT=CS      8AC ?A#0 A      9bF ?D<=C       Bb6 CSR         ED B=C-B A
0EF7 C=C&D A    13F CD1XS       801 OUT=C       8AD ?B#0 A      Aa0 A=A+B       Bb7 DSR         EE C=A-C A
0EF8 A=A!B A    140 DAT0=A A    802 A=IN        8AE ?C#0 A      Aa1 B=B+C       Bb8 A=-A        EF D=C-D A
0EF9 B=B!C A    141 DAT1=A A    803 C=IN        8AF ?D#0 A      Aa2 C=C+A       Bb9 B=-B        F0 ASL A
0EFA C=C!A A    142 A=DAT0 A    804 UNCNFG      8B0 ?A>B A      Aa3 D=D+C       BbA C=-C        F1 BSL A
0EFB D=D!C A    143 A=DAT1 A    805 CONFIG      8B1 ?B>C A      Aa4 A=A+A       BbB D=-D        F2 CSL A
0EFC B=B!A A    144 DAT0=C A    806 C=ID        8B2 ?C>A A      Aa5 B=B+B       BbC A=-A-1      F3 DSL A
0EFD C=C!B A    145 DAT1=C A    807 SHUTDN      8B3 ?D>C A      Aa6 C=C+C       BbD B=-B-1      F4 ASR A
0EFE A=A!C A    146 C=DAT0 A    8080 INTON      8B4 ?A<B A      Aa7 D=D+D       BbE C=-C-1      F5 BSR A
0EFF C=C!D A    147 C=DAT1 A    808F INTOFF     8B5 ?B<C A      AaB B=B+A       BbF D=-D-1      F6 CSR A
0F RTI          148 DAT0=A B    809 C+P+1       8B6 ?C<A A      Aa9 C=C+B       C0 A=A+B A      F7 DSR A
100 R0=A        149 DAT1=A B    80A RESET       8B7 ?D<C A      AaA A=A+C       C1 B=B+C A      F8 A=-A A
101 R1=A        14A A=DAT0 B    80B BUSCC       8B8 ?A>=B A     AaB C=C+D       C2 C=C+A A      F9 B=-B A
102 R2=A        14B A=DAT1 B    80Cn C=P        8B9 ?B>=C A     AaC A=A-1       C3 D=D+C A      FA C=-C A
103 R3=A        14C DAT0=C B    80Dn P=C        8BA ?C>=A A     AaD B=B-1       C4 A=A+A A      FB D=-D A
104 R4=A        14D DAT1=C B    80Fn CPEX       8BB ?D>=C A     AaE C=C-1       C5 B=B+B A      FC A=-A-1 A
108 R0=C        14E C=DAT0 B    810 ASLC        8BC ?A<=B A     AaF D=D-1       C6 C=C+C A      FD B=-B-1 A
109 R1=C        14F C=DAT1 B    811 BSLC        8BD ?B<=C A     Ab0 A=0         C7 D=D+D A      FE C=-C-1 A
10A R2=C        150a DAT0=A     812 CSLC        8BE ?C<=A A     Ab1 B=0         C8 B=B+A A      FF D=-D-1 A
```

```
1000 SUB DESAS(C$) @ POKE "2F441","F" @ DIM B$[41] @ RESET @ ON ERROR GOTO 1050 @ C$=ADDR$(C$)
1010 DEF FNO$(B$,C$,D$) @ SFLAG 0 @ J=16^LEN(D$)/2 @ INTEGER Q @ IF NOT FLAG(5) THEN 1030
1020 IF B$[7,7]="?" THEN Q=2 ELSE Q=POS(" 8C  8E"," "&B$[1,2]) @ IF B$[1,1]="7" THEN Q=3
1030 FNO$=DTH$(MOD(Q+HTD(C$)+(MOD(HTD(D$)+J,2*J)-J),16^5)) @ END DEF
1040 DEF FNA$(P)=ASC$(HTA$(A$[1,P]))&"'"
1050 DELAY 0,0 @ WIDTH 96 @ C$=RED$(C$) @ ON ERROR GOTO 1540 @ 'V': A$=SEEK$(C$,18)
1060 IF FLAG(5) THEN CALL MNEMO(A$,B$) @ P=POS(B$," ") ELSE B$=A$[1,8] @ P=0 @ CFLAG 7
1070 'U': CFLAG 0 @ IF NUM(B$[LEN(B$)-5])=35 AND NOT FLAG(8) THEN SFLAG 0
1080 'W': DISP C$&" ["&B$[P+1]&"]" @ IF LEN(B$[P])>15 THEN SCROLL LEN(B$[P])-14
1090 'Y': D$=KEYWAIT$ @ 'X': IF POS("#012345678 ",D$) THEN 'Y'
1100 Q=POS("DHCRTNA=()L",D$) @ IF NOT Q THEN 1120
1110 CFLAG 12 @ ON Q GOTO 1220,1210,1260,1300,1290,1340,1370,1500,1480,1510,1350
1120 CFLAG 8 @ Q=POS("#38 #51 #50 #43 #163#162#47 #48",D$) @ IF NOT Q THEN 'Y'
1130 ON (Q+3)/4 GOTO 1140,1160,1190,'F',1170,1200,1230,1180
1140 IF NOT (FLAG(7,0) OR FLAG(5)) THEN 'Y' ELSE 1250
1150 IF FLAG(5) THEN 'Y' ELSE 'V'
1160 P=9 @ GOTO 'Z'
1170 P=129 @ GOTO 'Z'
1180 P=2 @ GOTO 'Z'
1190 P=-7 @ GOTO 'Z'
1200 P=-127 @ GOTO 'Z'
1210 CFLAG 5,7 @ GOTO 'V'
1220 P=1 @ SFLAG 5 @ GOTO 1250
1230 INTEGER P
1240 'Z': CFLAG 5
1250 C$=DTH$(MOD(HTD(C$)+MOD(P-1,16^5),16^5)) @ IF FLAG(10,0) THEN RETURN ELSE 'V'
1260 SFLAG 12 @ GOSUB 'T' @ IF P<1 OR P>5 THEN 'Y' ELSE D$=REV$(A$[1,P])
1270 IF P<5 AND NOT FLAG(12) THEN D$=STR$(HTD(D$)) ELSE D$="#"&D$
1280 B$=A$[1,P]&" CON("&STR$(P)&") "&D$ @ P=P+1 @ CFLAG 0,8 @ GOTO 'W'
1290 IF FLAG(5) THEN 1390 ELSE IF FLAG(8) THEN P=P-1 @ GOTO 1310 ELSE 'Y'
1300 GOSUB 'T' @ IF P<1 OR P>5 THEN 'Y' ELSE CFLAG 0,8
1310 D$=REV$(A$[1,P]) @ IF NOT FLAG(8,1) THEN 1330
1320 CFLAG 8 @ D$=FNO$("",C$,D$)
1330 B$=A$[1,P]&" REL("&STR$(P)&") #"&D$ @ P=P+1 @ GOTO 'U'
1340 SFLAG 6 @ GOSUB 'T' @ B$=A$[1,P]&" NIBHEX "&A$[1,P] @ P=P+1 @ GOTO 'W'
1350 IF NOT (FLAG(5) AND B$[1,1]="3" AND MOD(HTD(B$[2,2]),2)) THEN 'Y' ELSE B$=B$[1,P] @ P=P-3
1360 A$=SEEK$(C$,18)[3] @ B$=B$&"LCASC "&REV$(FNA$(P))&"'" @ P=P+3 @ GOTO 'W'
1370 GOSUB 'T' @ IF P<1 OR P>8 THEN 'Y' ELSE P=P+P
1380 B$=A$[1,P]&" NIBASC '"&FNA$(P) @ P=P+1 @ GOTO 'U'
1390 I=POS(B$," #") @ IF NOT I OR LEN(B$[I+2])>4 THEN 'Y' ELSE D$=B$[I+2]
1400 D$=FNO$(B$,DTH$(HTD(C$)+1),D$)
1410 B$[I+2]=D$ @ GOTO 'U'
1420 'T': INTEGER P @ 'S': D$=KEYWAIT$ @ IF D$="#43" THEN 'F'
1430 IF D$="H" AND FLAG(12,NOT FLAG(12)) THEN 'S'
1440 IF D$="." AND FLAG(6,0) THEN SFLAG 9 @ P=10 @ GOTO 'S'
1450 IF (D$<"1" OR D$>"9") AND NOT FLAG(9) OR (D$<"0" OR D$>"6") AND FLAG(9,0) THEN 1470
1460 P=P+VAL(D$) @ SFLAG 7 @ CFLAG 5 @ A$=SEEK$(C$,18) @ RETURN
1470 P=POS(B$," ") @ POP @ GOTO 'X'
1480 IF NOT FLAG(0) THEN 'Y'
1490 CFLAG MOD(T-1,4)+1 @ T=T+1 @ SFLAG MOD(T-1,4)+1,10 @ DIM T$(T)[5] @ GOSUB 1250 @ T$(T)=C$
```

```
1500 Q=POS(B$," #") @ IF LEN(B$[Q+2])=5 AND FLAG(0) THEN C$=B$[Q+2.Q+6] @ GOTO 'V' ELSE 'Y'
1510 IF NOT T THEN 'Y' ELSE C$=T$(T) @ T=T-1
1520 IF T THEN CFLAG MOD(T,4)+1 @ SFLAG MOD(T-1,4)+1 ELSE CFLAG 1 @ DESTROY T$
1530 GOTO 'V'
1540 DISP ERRM$ @ D$="#4B"
1550 'F': POP @ POKE "2F441","0" @ RESET @ PUT D$ @ END SUB



2000 SUB MNEMO(A$,B$) @ ASSIGN #1 TO DESAT
2010 FOR I=1 TO 4 @ K=I+1 @ E$=A$[K,K]
2020 A=SEARCH("\^\"&A$[1,I],1,A,385,1)
2030 IF NOT A THEN 2080
2040 READ #1,A;B$
2050 P=POS(B$," ")-1
2060 IF POS("nxaih",B$[K,K]) THEN GOSUB 2140
2070 NEXT I
2080 IF POS(B$,"?",POS(B$," ")+1) EXOR NOT POS(B$," ") THEN GOSUB 2110
2090 IF A$[1,P]#B$[1,P] THEN B$=A$&" ?" @ GOTO 2320
2100 A$=A$[POS(B$," ")] @ GOTO 2320
2110 B$=A$[1,5]&B$[4]
2120 IF NOT HTD(A$[4,5]) THEN B$=B$&" RTNYES" ELSE B$=B$&" GOYES #"&REV$(A$[4,5])
2130 RETURN
2140 ON POS("nxaih",B$[K,K]) GOSUB 2150,2160,2270,2230,2170 @ RETURN
2150 B$[K,K]=E$ @ B$=B$&" "&STR$(HTD(E$)) @ RETURN
2160 B$[K,K]=E$ @ B$=B$&" "&STR$(HTD(E$)+1) @ RETURN
2170 B$[K,P]=A$[K,P] @ B$=B$&" #"
2180 B$=B$&REV$(A$[K,P]) @ RESTORE @ DATA 400,420,500,6300
2190 FOR J=1 TO 4 @ READ D$
2200 IF A$[1,LEN(D$)]=D$ THEN A=SEARCH("\^\"&D$,1,138,143,1) @ READ #1,A;B$ @ GOTO 2220
2210 NEXT J @ IF A$[1,5]="64000" THEN READ #1,144;B$
2220 I=5 @ RETURN
2230 I=K @ K=K+1 @ P=HTD(A$[2,2])+3 @ B$=A$[1,P]&" "
2240 IF P<8 THEN B$=B$&"LC("&STR$(P-2)&") " ELSE B$=B$&"LCHEX "
2250 IF P=7 THEN B$=B$&"#"
2260 GOTO 2180
2270 IF E$<"B" THEN 2290
2280 A=SEARCH("\^\"&A$[1,1]&"b",1,A,385,1) @ IF NOT A THEN POP @ RETURN
2290 A=SEARCH("\^\"&A$[1,I]&"\.\"&A$[I+2,P]&" ",1,A,385,1) @ READ #1,A;B$ @ B$[K,K]=E$
2300 C$=RED$("P WPXSX S M B W "[MOD(HTD(E$),8)*2+1][1,2])
2310 B$=B$&" "&C$ @ POP @ GOTO 2220
2320 END SUB
```

LISTAGE SOURCE DE LA FONCTION CNTRST

```
       LEX    'CONSTANT'
       ID     #5D
       MSG    0
       POLL   0
FNRTN2 EQU    #0F219        Renvoie le nombre codé dans C dans la Mathstack et retourne à Basic.
DCONTR EQU    #2E3FE        Emplacement du contraste (cf JPC No 13)
       ENTRY  FNCT
       CHAR   #F
       KEY    'CNTRST'
       TOKEN   1
       ENDTXT
       NIBHEX  00
FNCT
       ADOEX                Dans D0 se trouvait l'adresse du pointeur programme.
       R0=A
       D0=(5) DCONTR        On y charge l'adresse du contraste.
       P=     15            Tout le monde comprend.
       LCHEX  9             On charge 9.
       A=C    S             Que l'on passe dans A
       C=0    W
       C=DAT0 S             La valeur du contraste est écrite dans C(S).
       ?C>A   S
       GOYES  SAUT          Si elle est supérieure à 9 on saute.
       CSRC                 Sinon, on fait tourner C d'un quartet à droite.
       A=R0
       GOVLNG FNRTN2        Et on affiche le nombre.
SAUT                        Conversion en décimal des nombres supérieurs à 9.
       C=C-1  S
       C=C-1  S
       C=C-1  S
       C=C-1  S
       C=C-1  S
       C=C-1  S
       C=C-1  S
       C=C-1  S
       C=C-1  S
       C=C-1  S             On retire 10 à la valeur du contraste.
       CSRC
       CSRC                 2 rotations à droite pour placer le nombre au quartet 13.
       P=     0
       LCHEX  1             La valeur est supérieure à 9, donc il faut le convertir en décimal.
       P=     14
       LCHEX  1             Il faut bien entendu ajouter un 1 devant la valeur du contraste.
       A=R0
       GOVLNG FNRTN2        cf plus haut.
       END
```

PROGRAMME ASSEMBLE

```
1000 INTEGER A,I,J,K,L
1010 INPUT "Longueur du LEX :";L
1020 DIM A$[L],B$[10],N$[8]
1030 A$=""
1040 J=1
1050 B$=""
1060 GOSUB 1420
1070 WINDOW 4,14
1080 INPUT "";B$
1090 IF NOT LEN(B$) THEN 1140 ELSE A$=A$&B$ @ GOSUB 1330
1100 WINDOW 1
1110 J=J+1
1120 GOSUB 1420
1130 GOTO 1050
1140 WINDOW 1
1150 DISP "(A)ssemble    (M)ontre"
1160 ON POS("AM",KEYWAIT$)+1 GOTO 1140,1170,1240
1170 INPUT "Nom du LEX :";N$
1180 INPUT "ADR.:PORT(.01)? ";B$
1190 POKE B$,A$
1200 COPY N$
1210 PURGE N$&":PORT(.01)"
1220 END

================================================================
1230 IMAGE 2Z,X,K,5X,3Z
1240 J=1
1250 FOR K=1 TO IP(LEN(A$)/10)$10 STEP 10
1260 B$=A$[K,K+9]
1270 GOSUB 1330
1280 DISP USING 1230;J;B$;A
1290 ON POS("M",KEYWAIT$)+1 GOTO 1300,1390
1300 J=J+1
1310 NEXT K
1320 GOTO 1140
1330 A=0
1340 FOR I=1 TO 10
1350 A=A+VAL(B$[I,I+1])
1360 NEXT I
1370 A=MOD(A,256)
1380 RETURN
1390 INPUT "",CHR$(27)&"("&STR$(J)&" "&CHR$(27)&")";B$
1400 A$[K,K+9]=B$
1410 GOTO 1300
1420 DISP USING 1440;J;J-1;A
1430 RETURN
1440 IMAGE 2Z,13X,2Z,"!",3Z
```

FORTH RECURSIF ?

J'ai eu, au début de cette année 1985, un échange de courrier très intéressant avec Michel WEIL, de Besançon. Nous avons jugé opportun de vous en présenter quelques extraits, à propos de la récursivité de FORTH.

"Besançon, le 19/12/84

Cher Monsieur David,

Si vous n'avez pas le temps, vous jetez cette lettre au panier; sinon, voici quelques petites questions:
Toujours à propos de SMUDGE (ou plutôt de la récursivité):
- Bien entendu, ce n'est pas à propos du calcul de n! que j'ai besoin de récursivité (il y a d'ailleurs tout de suite dépassement de capacité, en simple ou double longueur). Alors, prenons un exemple plus simple:
SOM(n) = 1 + 2 + ... + (n-1) + n (=n(n+1)/2)
dont le mot FORTH serait:

```
: SOM [ SMUDGE ] DUP IF DUP 1- SOM + ELSE DROP 0
  THEN [ SMUDGE ] ;
```

Voici le problème: pour n > 72 ou 73, cela ne marche pas, l'engin affichant "ERR: Data type" !
L'ennui ne vient pas de la pile de données (puisqu'on peut l'agrandir), ni de la capacité de mémoire vive. Mais alors d'où ? (Pile de retour ?)

Tout ce que je sais, c'est: (en utilisant un module mémoire de 4 Ko)
- en FORTH itératif:

```
: SOMM 1+ 0 ROT SWAP DO I + LOOP ;
```

le "n" maximum (simple longueur) est n=1400, et c'est normal (car le résultat est 1400*1401/2=980700 (< 2^20)
- en BASIC récursif:

Le "n" maximum vaut n=282, et c'est normal (pour n>282, "Insuficient memory")

Bref, à cause de tout cela, le FORTH récursif me laisse insatisfait.

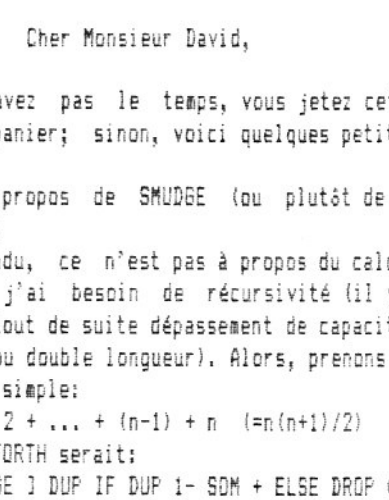
Bien amicalement et bonne année
Michel WEIL"

"Paris, le 13/1/85

Cher Monsieur Weil,

Ci-joint quelques éléments de réponse à vos questions:

"A propos de SMUDGE", la réponse figure page 27 du manuel du module FORTH/Assembleur:



A chaque appel récursif, une adresse (5 quartets) est empilée dans la pile de retour (ainsi que n dans la pile de données, mais ça ne nous intéresse pas ici). Il arrive un moment où le TIB rencontre la susdite Pile de Retour.
Le TIB (Terminal Input Buffer) stocke la commande entrée au clavier. Ex: si vous tapez 5 SOM [END LINE], vous aurez 53 02 35 F4 D4 00 00 (c'est à dire 5, espace, S, O, M, puis les deux 00 servent de marqueur de fin.
Imaginez que la pile de retour rencontre le TIB. La commande devient sans signification aucune, d'où le message d'erreur, car FORTH n'arrive plus à analyser correctement la commande, après avoir effectivement calculé SOM(72) !

Si vous voulez vous amuser, et si vous n'avez pas peur du Memory Lost, essayez SOM(500). Là, la pile de données et la pile de retour écrivent un peu sur tout ce qui existe dans FORTHRAM et ailleurs ! FORTH est sans protection...

Pour éviter votre insatisfaction, la solution est de faire GROW, pour élargir le FORTHRAM.

Note: FORTH, avec 3 Ko environ calcule SOM jusqu'à n=70 environ. BASIC, avec à peu près la même place, ne dépasse pas n=45 !

Amicalement,
Pierre DAVID"

Comme je l'avais dit à Michel, ce ne sont que des indications. Un point n'était pas précis: le fonctionnement de GROW. En effet, ce mot ne fait qu'étendre la place allouée au dictionnaire (et donc à la pile de données) mais pas la place allouée à la pile de retour. Le problème restait entier.

Là encore, la réponse se trouve dans le manuel du FORTH/Assembleur. Il faut "déplacer manuellement" le bas de la pile de données et le haut du TIB. Par exemple, après avoir fait 2048 GROW, il suffit de remettre les pointeurs 2FB11 et 2FB1B aux bonnes valeurs. Si on veut laisser à la pile de données le même espace qu'auparavant, on fera 30D7C 2FB11 !, puis 30D7C 2FB1B ! .
FORTH proteste contre cette manière de procéder par le biais d'un message d'erreur bien senti, mais il faut passer outre pour voir devant soi les horizons éclatants de la récursivité Forthienne !

Je laisserai la conclusion à Michel:

"Besançon le 23/1/85

Cher Pierre,

MERCI beaucoup pour vos indications: en combinant GROW, et en remettant les adresses "Bas de data stack" et "haut de TIB", la récursivité fonctionne très bien (et est même très rapide: 10 fois plus qu'en BASIC). Elle est seulement limitée par la mémoire (mais beaucoup moins qu'en BASIC).

Avec 20 Ko de FORTHRAM, on va jusqu'à n=830 pour SOM.
Avec 20 Ko de BASIC, on va jusqu'à n=270.

Quand on a fini, avant le SHRINK, il faut remettre les bonnes adresses.

Bien amicalement
Michel WEIL"

L'EVENEMENT

(c) Serge Vaudenay 1985, NOMAS (Not MAnufactured Suported), RC No XXX, etc...

Bonjour à tous, me revoilà: je vous avais présenté, le mois dernier, un programme de calcul de déterminant, et je récidive, cette fois, avec un programme encore mieux, encore plus puissant, encore plus utile, encore plus sérieux... J'ai même du faire un nouveau password encore plus stupide que les autres.

Pour plus de sureté, j'ai privatisé le listing que vous avez entre les mains, pour éviter les copies pirates (le journal s'auto-détruit dans ce cas-là).
Pour toutes réclamations, adressez-les au journal qui me les transmettra... mais sachez qu'elles seront très mal reçues.

Bref, voici LE programme qui permet de transformer votre HP-71 en orgue musical !!!

J'entends déjà les rires de ceux qui affirment que l'HP-71 n'a pas des vocations de musicien. SI! La fonction BEEP permet au 71 de rivaliser avec le meilleur des diapasons. Sa puissance vient du fait que le paramètre de fréquence qu'il demande doit être en Hertz (à prononcer le nez bouché), et donc en paramètres standards. Pour une fois que Hewlett Packard fait un effort de standardisation, Ça vaut la peine d'être souligné.

Vous trouverez ci-joint (ci-contre, ci-dessus, ou ci-dessous, je ne sais pas) un tableau des fréquences des notes de musique.

Le premier concert sur 71 qui se déroulera le 29/02/85 affiche déjà complet.

Le programme que je vous propose aujourd'hui permet donc d'enregistrer un morceau de musique sous forme de fichier, puis de le jouer.
Vous pourrez utiliser EDTEXT (module traitement de texte ou module FORTH) pour mettre votre morceau sous forme de fichier TEXT.
Il vous faut simplement entrer une note dans les enregistrements, et la base de temps (en secondes), dans le premier (ce sera donc la longueur d'une noire).

Les notes sont codées de la manière suivante:
- longueur (en unités de temps) suivie de "," mais ceci est optionnel si c'est 1,
- nom de la note (en français, s.v.p.), ou "S" pour un silence,
- No de l'octave, ci celle-ci est différente de 3.

Par exemple, "1/2,MIb4" sera le MI bémol de l'octave 4 en croche, "FA#" sera le FA dièse 3 en noire, "4,S" sera un silence valant une blanche (un soupir, un demi-soupir, ou un quart de murmure, vous appellerez Ça comme vous voudrez car j'avoue mon incompétence dans ce domaine).
Si vous commencez votre enregistrement par un moins ("-"), la ligne sera compilée telle quelle. Ceci peut être utile pour reprendre des parties du morceau (voir SOURIS avant et après compilation). Les lignes portent les mêmes numéros que les enregistrements. Dans ce cas, vous vous référerez à votre mode d'emploi sur le BASIC.
L'utilisation de GOSUB ou de CALL est fortement conseillée (voir exemples).

Vous devrez ensuite lancer le programme (RUN ou CALL), entrer le nom du fichier source virgule le nom du fichier destination.

Le programme fabriquera à partir de ce fichier source un programme BASIC utilisable directement.
Vous remarquerez que j'ai utilisé des MSG$ dans mon programme, et donc qu'il ne sera en français que si vous avez FRALEX.
Ce programme utilise donc EDLEX, par trois de ses messages, et par la fonction SEARCH (à prononcer en se débouchant le nez).

Vous trouverez également ci-joint (ci-dessus, ci-dessous ou ci-contre) des exemples de morceau (de musique, évidemment).

Le programme nécessite un fichier 'NOTES', qui contient la fréquence des notes.

Joyeuses compositions, et à bientôt.

Serge VAUDENAY (c'est moi!).

Frequence des notes (Hz).

Note	0	1	2	3	4	5	6
- SI	62	124	247	494	984	1984	3906
- SIb	58	116	233	466	933	1866	3676
- LA	55	110	220	440	880	1761	3472
- LAb	52	104	208	415	833	1667	3289
- SOL	49	98	196	392	786	1563	3125
- FA#	46	93	185	370	740	1488	2976
- FA	44	87	175	349	698	1404	2778
- MI	41	82	165	330	658	1316	2660
- MIb	39	78	155	311	622	1250	2500
- RE	37	73	147	293	587	1179	2358
- DO#	35	69	139	277	556	1106	2232
- DO	33	65	131	262	523	1050	2083

LISTAGE DU FICHIER TEXTE NOTES

```
0     S
494   SI
466   SIB
440   LA
415   LAB
392   SOL
349   FA
370   FA#
330   MI
311   MIB
293   RE
262   DO
277   DO#
124   SI1
116   SIB1
110   LA1
104   LAB1
98    SOL1
93    FA#1
87    FA1
82    MI1
78    MIB1
73    RE1
69    DO#1
65    DO1
247   SI2
233   SIB2
220   LA2
208   LAB2
196   SOL2
185   FA#2
175   FA2
165   MI2
155   MIB2
147   RE2
139   DO#2
131   DO2
984   SI4
933   SIB4
880   LA4
833   LAB4
786   SOL4
740   FA#4
698   FA4
658   MI4
622   MIB4
587   RE4
556   DO#4
523   DO4
1050  DO5
```

FICHIERS TEXTE:

CHEVAL	SUSANA	BERCEUSE
.3	.2	.4
.5,DO	.5,DO	.5,MI
.5,FA	.5,RE	.5,MI
FA	MI	1.5,SOL
.5,FA	SOL	.5,MI
.5,LA	1.5,SOL	MI
.5,LA	.5,LA	2,SOL
.5,DO4	SOL	.5,MI
DO4	MI	.5,SOL
LA	1.5,DO	DO4
.5,LA	.5,RE	SI
.5,LA	MI	LA
SOL	MI	LA
.5,DO	RE	SOL
.5,SOL	DO	.5,RE
.5,SOL	3,RE	.5,MI
.5,SOL	.5,DO	FA
2,FA	.5,RE	RE
.5,DO	MI	.5,RE
.5,FA	SOL	.5,MI

```
FA
.5,FA
.5,LA
.5,LA
.5,DO4
DO4
LA
.5,LA
.5,LA
SOL
.5,DO
.5,SOL
.5,SOL
2,FA
.5,DO4
.5,DO4
RE4
.5,SIB
.5,SIB
SIB
.5,RE4
.5,RE4
DO4
.5,LA
.5,LA
LA
.5,DO4
.5,DO4
SIB
.5,SOL
.5,DO4
.5,DO4
.5,DO4
2,LA
.5,DO4
.5,DO4
RE4
.5,SIB
.5,SIB
SIB
.5,RE4
.5,RE4
DO4
.5,LA
.5,LA
LA
.5,DO4
.5,DO4
SIB
```

```
1.5,SOL
.5,LA
SOL
MI
1.5,DO
.5,RE
MI
MI
RE
RE
3,DO
DO
2,FA
2,FA
LA
2,LA
LA
SOL
SOL
MI
DO
3,RE
.5,DO
.5,RE
MI
SOL
1.5,SOL
.5,LA
SOL
MI
1.5,DO
.5,RE
MI
MI
RE
RE
3,DO
```

```
2,FA
.5,RE
.5,FA
.5,SI
.5,LA
SOL
SI
2,DO4
.5,DO
.5,DO
2,DO4
.5,LA
.5,FA
2,SOL
.5,MI
.5,DO
FA
SOL
LA
2,SOL
.5,DO
.5,DO
2,DO4
.5,LA
.5,FA
2,SOL
.5,MI
.5,DO
.5,FA
.25,SOL
.25,FA
MI
RE
2,DO
-GOTO 2
```

```
.5,SOL
.5,DO4
.5,DO4
.5,DO4
2,FA

FICHIER TEXTE SOURIS (Voir souris2):

.15
-GOSUB PART1@GOSUB PART1@GOSUB PART2
.5,DO4
-GOSUB PART2@GOSUB PART3@GOSUB PART3@GOSUB PART3
.5,FA
-GOSUB PART1@END
-'PART1':
1.5,MI
1.5,RE
3,DO
-RETURN
-'PART2':
1.5,SOL
FA
.5,FA
3,MI
-RETURN
-'PART3':
.5,SOL
DO4
.5,DO4
.5,SI
.5,LA
.5,SI
DO4
.5,SOL
SOL
-RETURN
```

INTEGRALES

Le programme proposé peut-être utilisé en complément ou en palliatif du module maths du HP71 (non seulement parce qu'il est plus rapide,

mais aussi parce qu'il calcule des intégrales généralisées).

Pour plus de détails sur la méthode employée (méthode de Gauss-Legendre en 24 points), référez-vous au programme "INTG" paru pour HP41 dans le No 18 de JPC (octobre 84).

Exemples:

Calculez $\int_1^4 \frac{dx}{x^2} = \frac{3}{4}$

```
CALL INTEGRAT (1,4,"1/X^2")
```

Calculez $\int_1^{+\infty} \frac{dx}{x^2} = 1$

```
CALL INTEGRAT (1,Inf,"1/X^2")
```

Remarque: vous pouvez supprimer des chiffres aux DATA.

Amicalement,
Jean-Jacques MOREAU (#149)

MATRICES A EDITER !

Soucieux d'exploiter l'excellent module MATHS du 71, mais peu désireux d'affronter sur son terrain le superbe programme de Laurent ISTRIA (JPC No17 sept84), je vous propose ce mois-ci un éditeur de matrice. Baissez les doigts, j'explique !

Après avoir créé une matrice X (assimilée ici à un tableau) vous désirez modifier certains de ses éléments. Faites CALL MATEDIT(X).
Placez-vous sur l'élément concerné à l'aide des touches [<], [>], [^], [v] (qui peuvent être préfixées par [g] pour accéder aux éléments extrêmes des lignes et des colonnes). Vous pouvez alors entrer la nouvelle donnée sous la forme soit d'un nombre, soit d'une expression comme vous le feriez en mode CALC. Dès lors vous êtes en mode 'Introduction' et vous n'en sortirez que lorsque vous changerez de case, repassant ainsi en mode 'Visualisation'.
Confère exemple joint pour de plus amples informations, et prendre garde à ne pas taper trop vite (nous en reparlerons). En plus de cela trois commandes vous sont offertes:
- changer de signe: touche [=].
- effacer la dernière entrée: touche [ ] (espace).
- résultat précédent: touche [g] [END LINE].
Cette dernière fonction est bien pratique en cas d'erreur. Elle permet de rappeler la valeur initiale de la cellule du tableau, 'effaçant' en quelque sorte les modifications effectuées. Elle autorise également le réemploi dans un calcul de la valeur trouvée en vous placant en mode 'Introduction'.

Un exemple éclaircira les idées:
Soit X une matrice identité de dimension (5,5), de plus petit indice 0. On se propose d'assigner à X(2,2) la valeur PI.
Créons la matrice: DIM X(5,5) et appelons la routine: CALL MATEDIT(X).
S'affiche: 0,0: 0.
Plaçons nous sur X(2,2): deux fois [v], puis deux fois [>]. Tapons: PI. Il est inutile de valider par [END LINE], à moins que vous ne désiriez connaître le résultat, ce qui ne change rien à la manière d'agir du programme.
Mais au fait, ce n'est pas en X(2,2), mais en X(3,2) qu'il fallait aller !...
Qu'à cela ne tienne. Faites [CMDS] (soit [g][END LINE]) et le 1 initial réapparaît. Comme aurait réapparu tout autre chiffre d'ailleurs. Allez donc en (3,2) et introduisez PI. Changez de case et revenez en (3,2) (vous suivez ?). Multipliez ce résultat par l'inverse de PI. Comment faire ? Très simple: appuyez sur [CMDS] * 1/PI puis [END LINE].
Je vous laisse vous familiariser avec le maniement de [SPC] pour en venir à la structure de ce programme.

Il n'en a pas, l'affaire est entendue et honte à moi ! Tout a été sacrifié à la vitesse d'éxécution. A ce sujet, le sous-programme

'ENTREE' sera prochainement remplacé par un mot Forth. Mais n'allez pas le répéter, cela ferait des envieux ...

Jean-Pierre BONDU (33)

Le programme 'VALPROPR' a pour finalité la détermination du polynôme caractéristique d'une matrice carrée réelle A d'ordre n, et ainsi le calcul de ses valeurs propres, réelles ou complexes, par résolution du polynôme. Rappelons qu'une valeur propre λ est telle que, pour toute matrice colonne X non nulle associée à λ, on ait: $AX=\lambda X$.

Pour réaliser tout cela, deux atouts: les formules de récurrence de Bôcher et le module de maths du HP71B.

- Les formules:
Soit $P(\lambda)=\lambda^n+\alpha_1\lambda^{n-1}+\cdots+\alpha_{n-1}\lambda+\alpha_n$
le polynôme caractéristique de A.
Soit $S_1=tr(A)$
$S_2=tr(A^2)$
$\vdots$
$S_n=tr(A^n)$.
Alors,
$\alpha_1=-S_1$
$\alpha_2=-\frac{1}{2}(\alpha_1S_1+S_2)$
$\alpha_3=-\frac{1}{3}(\alpha_2S_1+\alpha_1S_2+S_3)$
$\vdots$
$\alpha_n=-\frac{1}{n}(\alpha_nS_1+\alpha_{n-1}S_2+\ldots+\alpha_1S_{n-1}+S_n)$

- Le module math: ce sont les fonctions de produits de matrices et de résolution de polynômes (PROOT) qui ont été principalement utilisées.

Mode d'emploi:
RUN VALPROPR, entrer l'ordre de la matrice, puis celle-ci; en sortie: P(λ), puis CONT pour avoir les valeurs propres.

Variables:

Tableau A: Matrice A
Tableau B: Matrice A
Tableau S: Les Si
Tableau X: Valeurs propres
Tableau P: Les α_i
P$: P(λ)
N: ordre de A
I,J: indices de calcul

A vous de polynômer maintenant...
Olivier ARBEY (P118 - T164)

PROGRAMME MUSIQUE

```
   - Compilateur musical.
  20 DESTROY E$,F$,G$ @ DISP MSG$(240004); @ INPUT '';F$,G$
  30 DISP MSG$(240010)
  40 CALL COMP(F$,G$,E$)
  50 DISP E$ @ BEEP

================================================================
  60 SUB COMP(F$,G$,E$) @ DIM A$[96],B$[96]
  70 ON ERROR GOTO 190 @ SFLAG -1 @ PURGE G$
  80 CREATE TEXT G$ @ ASSIGN #2 TO G$ @ ASSIGN #1 TO F$
  90 ASSIGN #3 TO NOTES
 100 READ #1;A$ @ T0=VAL(A$) @ L=1
 110 ON ERROR GOTO 160
 120 READ #1;A$ @ A$=UPRC$(A$) @ L=L+1
 130 B$=('0000'&STR$(L))[LEN(STR$(L))+1]
 140 IF A$[1,1]#'-' THEN CALL VAL(A$,B$,#3,T0) ELSE B$=B$&A$[2]
 150 PRINT #2;B$ @ GOTO 120
 160 ASSIGN #2 TO *
 170 TRANSFORM G$ INTO BASIC
 180 E$=MSG$(240011)&': '&G$ @ GOTO 200
 190 E$=ERRM$
 200 END SUB

================================================================
 210 SUB VAL(A$,B$,#3,T0)
 220 T=1 @ IF POS(A$,',')=0 THEN 250
 230 ON ERROR GOTO 240 @ T=VAL(A$)
 240 OFF ERROR @ A$[1,POS(A$,',')]=''
 250 RESTORE #3,SEARCH(A$,7,0,1000,3)
 260 READ #3;A$
 270 F=VAL(A$)
 280 IF F#0 THEN B$=B$&' BEEP '&STR$(F)&','&STR$(T*T0) ELSE B$=B$&' WAIT '&STR$(T)
```

PROGRAMME SOURIS2

```
   2 GOSUB 'PART1' @ GOSUB 'PART1' @ GOSUB 'PART2'
   3 BEEP 523,.075
   4 GOSUB 'PART2' @ GOSUB 'PART3' @ GOSUB 'PART3' @ GOSUB 'PART3'
   5 BEEP 349,.075
   6 GOSUB 'PART1' @ END

================================================================
   7 'PART1':
   8 BEEP 330,.225
   9 BEEP 293,.225
  10 BEEP 262,.45
  11 RETURN

================================================================
```

```
 12 'PART2':
 13 BEEP 392,.225
 14 BEEP 349,.15
 15 BEEP 349,.075
 16 BEEP 330,.45
 17 RETURN
```

```
================================================================
```

```
 18 'PART3':
 19 BEEP 392,.075
 20 BEEP 523,.15
 21 BEEP 523,.075
 22 BEEP 494,.075
 23 BEEP 440,.075
 24 BEEP 494,.075
 25 BEEP 523,.15
 26 BEEP 392,.075
 27 BEEP 392,.15
 28 RETURN
```

PROGRAMME INTG

```
================================================================
```

```
  1 SUB INTEGRAT(C,B,F$) @ DISP @ DESTROY S @ DIM T(24) @ OPTION BASE 1 @ DEF FNF(X)=VAL(F$)
  2 RESTORE @ READ T() @ IF B=INF THEN 10 ELSE A=(B+C)/2 @ B=(B-C)/2 @ GOTO 8
```

```
================================================================
```

```
  3 DATA .064056892862605,.127938195346752,.191118867473616,.125837456346828,.315042679696163
  4 DATA .121670472927803,.433793507626045,.115505668053725,.545421471388839,.107444270115965
  5 DATA .648093651936975,.097618652104113,.740124191578554,.086190161531953,.820001985973902
  6 DATA .073346481411080,.886415527004401,.059298584915436,.938274552002732,.044277438817419
  7 DATA .974728555971309,.028531388628933,.995187219997021,.012341229799987
```

```
================================================================
```

```
  8 FOR J=0 TO 1 @ B=(J*2-1)*ABS(B) @ FOR I=1 TO 24 STEP 2 @ S=S+T(I+1)*FNF(A+B*T(I)) @ NEXT I
  9 NEXT J @ DISP B*S @ END
 10 A=C-1 @ FOR J=0 TO 1 @ X=J*2-1
 11 FOR I=1 TO 24 STEP 2 @ D=1/(1+X*T(I)) @ S=S+T(I+1)*D^2*FNF(A+2*D) @ NEXT I @ NEXT J
 12 DISP 2*S @ END SUB
```

PROGRAMME MATEDIT

```
================================================================
```

```
1000 SUB MATEDIT(A(,))
1010 INTEGER X,Y,S @ DIM L,X$,K$[4],T$[1],X$[40] @ DELAY 0,0
1020 T$=PEEK$('2F441',1) @ POKE '2F441','F'
   - Touche 'ON' inhibée
1030 X=LBOUND(A,1) @ Y=LBOUND(A,2)
1040 ON ERROR GOTO 1080
```

```
1050 S=0 @ L=A(X,Y)
1060 X$=STR$(A(X,Y))
1070 BEEP 2000,.01
1080 Z$=STR$(X)&","&STR$(Y)&": "&X$
1090 IF S THEN DISP Z$&CHR$(95) ELSE DISP Z$
1100 K$=KEYWAIT$
1110 IF K$='=' THEN A(X,Y)=-VAL(X$) @ GOTO 1060
1120 IF K$=' ' THEN GOSUB 'CORR' @ GOTO 1070
1130 IF K$[1,1]#'#' THEN GOSUB 'ENTREE' @ GOTO 1070
1140 IF K$='#38' THEN A(X,Y)=VAL(X$) @ GOTO 1060
1150 IF K$='#47' AND Y>LBOUND(A,2) THEN A(X,Y)=VAL(X$) @ Y=Y-1 @ GOTO 1050
1160 IF K$='#48' AND Y<UBOUND(A,2) THEN A(X,Y)=VAL(X$) @ Y=Y+1 @ GOTO 1050
1170 IF K$='#50' AND X>LBOUND(A,1) THEN A(X,Y)=VAL(X$) @ X=X-1 @ GOTO 1050
1180 IF K$='#51' AND X<UBOUND(A,1) THEN A(X,Y)=VAL(X$) @ X=X+1 @ GOTO 1050
1190 IF K$='#150' THEN A(X,Y)=L @ S=1 @ GOTO 1060
1200 IF K$='#159' THEN A(X,Y)=VAL(X$) @ Y=LBND(A,2) @ GOTO 1050
1210 IF K$='#160' THEN A(X,Y)=VAL(X$) @ Y=UBND(A,2) @ GOTO 1050
1220 IF K$='#162' THEN A(X,Y)=VAL(X$) @ X=LBND(A,1) @ GOTO 1050
1230 IF K$='#163' THEN A(X,Y)=VAL(X$) @ X=UBND(A,1) @ GOTO 1050
1240 IF K$='#43' THEN 1300 ELSE 1100

================================================================
1250 'CORR': IF S THEN X$=X$[1,LEN(X$)-1] ELSE X$='0'
1260 RETURN

================================================================
1270 'ENTREE':
1280 IF S THEN X$=X$&K$ ELSE X$=K$ @ S=1
   - S=1: Introduction de la donnée en cours
1290 RETURN
1300 A(X,Y)=VAL(X$)
1310 POKE '2F441',T$
1320 END SUB


PROGRAMME VALPROP


   - Initialisations:
  10 DESTROY ALL @ STD @ DELAY .5,.5
  20 INPUT "Ordre ? ";N
  30 ON ERROR GOTO 410
  40 OPTION BASE 1 @ DIM A(N,N),B(N,N),S(N) @ COMPLEX SHORT X(N)
  50 OPTION BASE 0 @ DIM P(N)
  60 DIM P$[96]
   - Entrée de la matrice A:
  70 MAT INPUT A
   - Calcul des Si:
  80 MAT B=A @ P(0)=1
  90 FOR I=1 TO N
 100 FOR J=1 TO N
 110 S(I)=S(I)+A(J,J)
 120 NEXT J
 130 MAT A=B*A
 140 NEXT I
```

```
  - Calcul des αi:
150 FOR I=1 TO N
160 FOR J=1 TO I
170 P(I)=P(I)+P(I-J)*S(J)
180 NEXT J
190 P(I)=-P(I)/I
200 NEXT I
  - Résolution du polynôme caractéristique, et racines dans le tableau X:
210 MAT X=PROOT(P)
  - Mise en forme de P(λ):
220 P$="P="
230 FOR I=0 TO N
240 IF P(I)=0 THEN 320
250 IF ABS(P(I))#1 THEN 270 ELSE IF P(I)=1 THEN P$=P$&"+" ELSE P$=P$&"-"
260 IF ABS(P(I))=1 AND I=N THEN P$=P$&"1" @ GOTO 320 ELSE 300
270 IF P(I)>0 THEN P$=P$&"+"
280 P$=P$&STR$(P(I))
290 IF I=N THEN 320
300 IF I=N-1 THEN P$=P$&CHR$(11) @ GOTO 320
310 P$=P$&CHR$(11)&"^"&STR$(N-I)
320 NEXT I
330 IF P$[3,3]="+" THEN P$=P$[1,2]&P$[4]
  - Affichage des résultats:
340 BEEP @ BEEP @ DISP P$ @ PAUSE
350 FOR I=1 TO N
360 DISP CHR$(11);I;"=";
370 IF IMPT(X(I)) THEN DISP X(I) ELSE DISP REPT(X(I))
380 PAUSE
390 NEXT I
400 GOTO 340
  - Traitement d'erreur:
410 BEEP INF @ DISP MSG$(24)
420 END
```

PPC PARIS CHAPTER

ASSOCIATION REGIE PAR LA LOI DE 1901, ENREGISTREE
A PARIS LE 2 DECEMBRE 1982 SOUS LE NUMERO 82/3240

BULLETIN D'ADHESION

NOM
PRENOM
ADRESSE
COMMUNE
CODE POSTAL
PAYS
TELEPHONE DOMICILE
BUREAU
DATE DE NAISSANCE

PROFESSION
INTERETS

MATERIEL HP EN VOTRE POSSESSION

AUTRE MATERIEL MICRO-INFORMATIQUE

COMMENT AVEZ-VOUS CONNU PPC PARIS CHAPTER ?
PUBLICITE
MAGAZINE
AUTRE CLUB
HP
RELATIONS, MEMBRES DU CLUB, AUTRES

QUE RECHERCHEZ-VOUS AU SEIN DU PPC PARIS CHAPTER ?

Je souhaite adhérer au club PPC PARIS CHAPTER conformément aux statuts de l'association. Au mieux de ma connaissance, je déclare avoir le droit de fournir tous les programmes et informations que je vous enverrai (sans enfreindre des obligations de secret à l'égard d'autres personnes ou organismes) pour publication dans le journal de liaison, sans obligations ni responsabilité d'aucune sorte (en cas d'utilisation frauduleuse) de la part des dirigeants du PPC PARIS CHAPTER.

DATE !_!_!/!_!_!/19!_!_!
SIGNATURE, PRECEDEE DE LA MENTION "LU ET APPROUVE"

LE MONTANT DE LA COTISATION AU PPC PARIS CHAPTER S'ELEVE A 300.00 FF.
ETUDIANTS: 250.00 FF. (JUSTIFICATIF INDISPENSABLE)
PAIEMENT EFFECTUE LE !_!_!/!_!_!/19!_!_! A L'ORDRE DE PPC PARIS CHAPTER.
PAR [ ] CHEQUE BANCAIRE N° BANQUE
[ ] CHEQUE POSTAL 3 VOLETS N°
[ ] MANDAT LETTRE
EVENTUELLEMENT: JE M'ABONNE A COMPTER DU !_!_!/!_!_!/19!_!_!
JOINDRE A VOTRE INSCRIPTION UNE PHOTO D'IDENTITE ET UNE ENVELOPPE TIMBREE A VOTRE ADRESSE.

VEUILLEZ ENVOYER TOUTE CORRESPONDANCE A:
MR PHILIPPE GUEZ, 56 RUE J.J. ROUSSEAU, 75001 PARI