

JPC

PPC-PC

JPC

JUIN

1985

Numéro-25

Le numéro 35FF

A PROPOS DU CLUB

Pierre DAVID	1	EDITORIAL
Pierre DAVID	2	PPC PARIS SE REUNIT
Franck LEBASTARD	3	COMPTE-RENDU DE LA REUNION DE LYON
	8	NOUS EN AVONS
Pierre DAVID	10	IL ETAIT UNE FOIS LE SICOB
	12	COURRIER DU COEUR
	13	COURRIER DES LECTEURS

HP-41

Christian JEGOUZO	18	HP41 ET RECURSIVITE (TOURS DE HANOI)
Philippe TENAND	24	EPROM ET MICROCODE
Gilbert TISSERAND	25	DES DRAPEAUX A LA CHAINE
Denis CASTELAIN	26	ORDINATEUR DE BORD
Steen PETERSEN	29	PROPOSED HP41 STANDARD MACHINE LANGUAGE BARCOD
Philippe GUEZ	30	TRADUCTION
Steen PETERSEN	32	ADDENDUM
Philippe GUEZ	33	TRADUCTION

HP-71

Olivier ARBEY	42	DEBUTER EN ASSEMBLEUR
Michel MARTINET	43	CHOC EN RETOUR
Jean-Jacques MOREAU	43	END UP !
Jean-Jacques MOREAU	46	DES REGISTRES D'ENTREE/SORTIE
Jean-Pierre BONDU	47	PILE A 16 NIVEAUX
Michel MARTINET	48	LES VARIABLES ET LA MATH STACK
Laurent ISTRIA	50	A PROPOS DE COMBINAISONS ET D'ARRANGEMENTS
Michel MARTINET	52	HEURES-MINUTES-SECONDES
Jean-Pierre BONDU	56	ENCORE UNE ERREUR !
Michel MARTINET	57	PROGRAMMATION SYNTHETIQUE
Michel MARTINET	59	LA PILE DE COMMANDE EN ASSEMBLEUR
Jean-Pierre BONDU	59	ADRESSE D'UNE COLONNE GRAPHIQUE (FORTH)
René OUVRAY	60	ANOMALIES DANS LE CALENDRIER
Serge VAUDENAY (the King)	61	REMORDS
Jean-Jacques DHENIN	61	POUR LA RUBRIQUE DEBUTANTS
Xavier BILLE	63	MASTER-MIND AMELIORE
Pierre DAVID	64	D.A.O. (SUITE...)
Serge VAUDENAY (The King)	65	HANTISE SUR SATURNE

Les illustrations sont de Filousan



Et de THINKJET

EDITORIAL

Chers HF-istes,

Voici enfin arrivé votre JPC ! Rien que des informations, des programmes, rien que pour votre ordinateur préféré...

Le Spécial-Sicob est maintenant terminé. Je profite de l'occasion pour remercier tous ceux qui ont travaillé d'arrache-pied pour que notre présence soit possible.

Et entre nous, cela en valait la peine: beaucoup d'échanges, de rencontres, de contacts fructueux (nous en reparlerons...), et... nous sommes maintenant près de 250 au Club !

Et les vacances arrivent à point pour nous reposer. Je vous en souhaite de très bonnes, très fertiles en nouveaux programmes que vous nous ferez découvrir dès la rentrée...

A bientôt,
Pierre

DIVERS

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 Eld Magenta, coté impairs.

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

La date de la prochaine réunion est: (16h-19h)
- Le Samedi 7 septembre 1985
Le centre est ferme durant les vacances scolaires.

A Bientôt...
Pierre DAVID

Franck LEBASTARD
9672T35P43
Appt. 1260
1, rue des Verchères
69120 VAULX EN VELIN
tél. (7) 880 77 34

COMPT E - RENDU DE LA
REUNION PPC DE LYON
des 27 & 28 avril 1985

AVEZ-VOUS JAMAIS ENTENDU PARLER DU CHANTER
UNE HP-41 OU UN HP-71?
N O U S SI !!!!!!!!!

Voici donc le compte-rendu de la dernière réunion PPC, réunion s'étant déroulée à Lyon les 27 & 28 avril derniers. Contrairement à mon espérance, cette réunion n'a pas suscité d'enthousiasme délirant de la part des locaux puisque seulement 50% des personnes présentes étaient de la région.

Étaient présents Yves Alajouanine (T144P15) de Lyon (69), Jean-Claude Bornes (9250T175) d'Evry (91), François Colet (P172) de Saint-Etienne (42), Pierre David (P37CHHU617) de Paris (75), Jean-Jacques Dhénin (T212P177) de Paris (75), Alain Farge (P132) de Paris (75), Jean-François Garnier (T371) de Grenoble (38), Laurent Istria (P3) de Paris (75), Patrick Le Guillou (T472) de Lyon (69), André Mémier de Meylan (38), José Pidival (T72P23) de Bivors (69), Cyril Reigner de Toulon (83), Robert Schwartz (T178) de Bondy (93), et moi-même (9672T35P43) de Vaulx en Velin (69), Soit 14 membres PPC.

Comme d'habitude, toutes les nouveautés HP étaient présentées, ainsi que le ZENROM, le PANAME pour la 41, le module FORTH/assembleur pour le 71, la table traçante 7470A et de nombreuses autres choses que je vais avoir la joie de vous présenter.

Les ordinateurs de poche HP-41, HP-71 et HP-75

étaient bien entendu en démonstration, ainsi que tous les nouveaux périphériques HP fonctionnant sur HP-IL, notamment la nouvelle unité de disquette IL ainsi que la Thinkjet.

De nombreuses démonstrations de matériels ont été faites, certaines provoquant d'ailleurs l'enthousiasme de l'assemblée; à noter la démonstration par Jean-Jacques Dhénin du module PANAME, démonstration qui se termina par une commande ferme de 5 modules, chèques à l'appui. Il faut dire que cette ROM est extraordinaire; si vous ne l'avez pas encore, n'hésitez pas à la commander au siège de votre Club, ces fonctions ne tarderont pas à envahir tous les programmes.

Un des moments forts du week-end a sans contestation possible été la présentation par Jean-François Garnier de ses prototypes, notamment son synthétiseur vocal qui a fait un tabac. La question posée en titre de ce compte rendu est à ce sujet tout à fait évocatrice, surtout notre réponse. Entendre la 41 ou le 71 chanter "A la claire fontaine" est je vous l'avoue un bien beau spectacle. De même, entendre la 41 vous annoncer l'heure qu'il est à intervalle régulier est tout à fait extraordinaire. Jean-François nous a également montré une interface centronics, précisons le, entièrement compatible avec le module paname, ainsi qu'un prototype de carte E/S sur HPIL:

Jean-François GARNIER:

"- Synthétiseur vocal HPIL. Ce synthétiseur est basé sur une réalisation décrite dans Micro-Système il y a quelques temps et interfacée HPIL par le convertisseur HPIL. La commande du synthétiseur s'effectue par l'envoi de chaînes d'octets (1 à 34). La vitesse et la hauteur du son sont programmables, ce qui permet au synthétiseur de chanter. ce matériel est une réalisation personnelle; devant l'intérêt suscité parmi les membres présents, j'envisage d'en fabriquer une version plus compacte avec interface HPIL intégrée.

- Interface centronics compatible graphique HP-6L (Conception JF GARNIER, fabrication KRISTAL).

Cette interface travaille en deux modes sélectionnés par logiciel:

- mode imprimante: compatible avec toutes imprimantes Centronics. Avantages: mémoire tampon 1,5Ko (possibilité de 15,5Ko), reconnue comme imprimante.

- mode graphique HP-GL: destiné à la CGP115 de Tandy (versions possibles pour d'autres tables tranchantes). Reconnue comme PLOTTER, entièrement compatible avec le module PLOTTER. Avantage: simplicité de programmation pour les tracés de courbes, histogrammes, etc... Prix de l'ordre de 4000FF.

- Prototype de carte E/S HPIL (Entrées-Sorties). Cette carte possède jusqu'à 62 lignes d'E/S programmées individuellement en entrées ou en sorties. Des commandes permettent de lire/écrire les lignes par port de 8 bits ou de les manipuler individuellement. Le but est de mettre à la disposition du programmeur le contrôle complet de PIO (Peripheral Input Output) avec possibilités d'exploiter les modes d'"handshake" automatique ou de timers internes. Applications: interfaçage de convertisseur numérique/analogique et réciproquement, commande de transistors de puissance, triacs, etc... Notice d'application en préparation (sera fournie avec la carte)."

Jean-François GARNIER (T371)

A également été montré par Jean-Claude BORNES la nouvelle interface HPIL pour IBM PC. Cette interface, de l'avis même de Jean-Claude, est bien décevante, surtout au niveau du logiciel qui est fourni avec. Telle qu'elle est fournie actuellement par HP, elle ne permet à l'IBM que de fonctionner en mode automatique, pas de mode manuel donc comparable au module IL de l'HP41, et donc peu de possibilité d'utilisation; quand l'IBM se comporte en périphérique, même chose, guère de possibilités, peu d'intérêts... pour l'instant, car, n'en doutez pas, Jean-claude travaille dessus !

Autrement plus intéressant, est l'autre appareil dont il nous a fait la démonstration et dont il est le concepteur, autrement dit l'interface permettant de relier l'HP41 ou l'HP71 à un

minitel via la boucle HPIL, et de s'en servir comme vidéo, ou mieux comme modem, mais laissons le plutôt lui-même nous parler de son oeuvre:

Jean-Claude BORNES: "Interface HPIL-MINITEL

Cette interface permet de connecter un minitel sur une boucle HPIL, permettant d'utiliser les constituants du minitel par le contrôleur de la boucle.

L'écran peut être utilisé, par exemple par une HP41 en tant qu'interface vidéo, pour un budget bien inférieur à celui nécessaire avec les solutions HP.

Un autre constituant important du minitel, est son modem. Ce constituant du minitel permet d'envoyer des informations sur la ligne téléphonique. Il est donc possible, par exemple de consulter de façon automatique une banque de données. Vu la vitesse de la 41, il ne faut pas compter enregistrer les pages reçues. Avec un calculateur plus rapide, 71 par exemple, cette application est envisageable.

Une autre application très intéressante pour nous membres PPC, est la communication entre deux calculateurs pour la transmission de programmes ou de données.

Au cours de la réunion de Bretagne, la première application (moniteur vidéo pour la 41 ou visualisation pour une 71 par PRINTER IS) a été montrée. En ce qui concerne la 41, tous les ordres non graphiques sont valables, la 41 étant en mode MANIO.

Au cours de cette réunion de Lyon, une deuxième interface était disponible, permettant de montrer les autres applications, deux minitels étant reliés ensemble. Après retournement du premier minitel par envoi d'un code spécifique, la pression des touches de connexion permet d'établir la communication. A partir de cet instant, les chaînes de caractères envoyées par ACA, PRA, OUTA ou PRINT sont transmises par l'intermédiaire de la ligne téléphonique à l'autre minitel et à l'autre calculateur. Un listing de programme envoyé par un calculateur

sera visualisé sur l'autre minitel, pouvant être récupéré par l'autre ordinateur. En ce qui concerne les affichages, les minitels sont reconfigurables à volonté. Du mode défaut écho vrai (est affiché à l'écran ce qui vient de la ligne téléphonique) on peut configurer en écho local (ce qui vient du clavier ou du ordinateur "local" est affiché à l'écran).

L'autre application qui a été utilisée au cours de cette réunion, est la transmission de programmes entre deux 41 munies du module Extended IO. La 41 réceptrice (minitel non retourné) exécute la fonction INP et la 41 émettrice exécute OUTP. La transmission de lots importants de données ne peut cependant se faire que depuis le minitel retourné vers le minitel non retourné, des données étant perdues dans l'autre direction du fait que la vitesse n'est que de 75 bauds.

Dans l'état actuel, cette interface permet donc de disposer, en plus d'une interface vidéo (non graphique du fait des caractéristiques du minitel) économique, de la possibilité de communication."

Jean-Claude BORNES (T175)

Un autre périphérique IL était présent à cette réunion, il s'agit de l'interface vidéo PAC SCREEN, commercialisée par les Editions du Cagire, interface se comportant dans la boucle autant comme vidéo que comme table traçante. Cette merveilleuse petite chose répond aussi bien aux ordres du plotter qu'aux ordres vidéo habituels, en particulier aux commandes du module Paname, ce qui assure à son utilisateur un contrôle extrêmement puissant de son moniteur. C'était Robert SCHWARTZ qui nous en faisait la démonstration:

Robert SCHWARTZ: "Vidéo PAC SCREEN

L'interface PAC SCREEN est un boîtier 160x90x30 alimenté par une alimentation 9,5V, 800mA qui se relie à un moniteur vidéo.

Il contient en fait deux appareils utilisant le

même moniteur VIDEO et la même entrée HPIL: une interface pour le texte offrant 24 lignes de 80 (ou 40) colonnes, et une interface graphique traçant des dessins de la même façon qu'une table traçante, à l'aide d'un sous-ensemble du langage graphique HP (HP-GL).

L'interface "texte" offre deux formes de curseur et les caractères soulignés sur trois pages accessibles par déroulement. L'interface graphique offre une résolution de 640x250 pixels.

De plus, l'interface possède une sortie Centronics pour imprimante qui permet une copie d'écran "texte" ou "dessin".

Une partie des fonctions du module Plotter est reconnue par l'interface en mode "dessin" et toutes les fonctions vidéo du module Paname sont utilisables."

Robert SCHWARTZ (T178)

Les autres interventions intéressantes étaient des exposés. Yves ALAJOUANINE nous présenta par exemple son dernier logiciel pour HP41 "Saisie des observations de Topométrie Moderne" (STOM), logiciel professionnel destiné à saisir et traiter sur le terrain les informations topométriques:

Yves ALAJOUANINE:"

Le programme de "Saisie des Observations de Topométrie moderne" a fait l'objet d'un exposé par Yves Alajouanine, le 27 avril 1985 pendant les deux jours de la réunion régionale PPC organisée par Franck Lebastard, au "Centre International de Séjour de Lyon".

Le ordinateur HP41CV ou CX est utilisé de préférence, compte tenu de l'existence dans le commerce de l'interface KERN DIF 41 transférant les mesures d'un théodolite informatisé directement dans les mémoires HP41 pour faire un enregistrement avec ou sans traitement en temps réel. de même, les Ets AGA GEOTRONICS vendent le GEODAT 126 pour la saisie et les calculs sur le terrain: c'est le boîtier contenant un HP41CX

pouvant enregistrer sur 2000 mémoires avec sortie HPIL et RS-232C.

Le logiciel STOM utilise 71 mémoires de données et 1730 octets, et une routine secondaire finale "M". Celle-ci reste fixe pour la saisie sans calcul; mais s'il y a calcul des XYZ, elle remplace une routine d'application qui est appelée en milieu de traitement grâce à un codage de 3 chiffres, CDU accolé à droite du numéro matricule du point levé. L'élaboration des routines d'application utilise les fonctions d'édition de la routine finale "M" terminent ensuite le traitement.

Le programme de "saisie des Observations de Topométrie Moderne" pour "Logiciel de Tracé Automatique Direct" est composé sur un calculateur de poche HP41CX (ou par défaut un modèle HP41CV avec X-FUNCTIONS) en utilisant une imprimante thermique HPB2162A sur une boucle HPIL pour éditer le listage du programme. Celui-ci est accompagné de commentaires édités en traitement de texte ST36 sur ordinateur Sanyo MBC-1250.

L'équipement minimum du calculateur HP41CV demande l'adjonction dans le port 1 d'un module HPB2180A de 127 registres d'extension de mémoire pour l'étude scolaire du logiciel SOTM. De préférence, il faut placer en plus 2 modules HPB2181A de 238 registres chacun dans les ports 2 et 3 des HP41CV ou CX.

L'utilisation professionnelle a besoin de la boucle HPIL avec l'unité de cassettes HPB2161A. L'alimentation modifiée de celle-ci est donnée par la batterie CaNi 6 ou 12 volts du distancemètre. Les 603 registres contiennent les routines d'application du logiciel SOTM. La saisie avec ou sans calcul des XYZ dure 25 ou 5 secondes respectivement et la capacité de 131072 octets permet l'enregistrement de 1800 points.

Si le géomètre souhaite enregistrer sans avoir de cordon de liaison, il peut installer un calculateur HP41CX connecté par son port 4 dans l'appareil GEODAT 12 AGA qui peut enregistrer 500 à 1000 points.

Le matériel professionnel le plus compact utilise

le calculateur HP41CX 1525 registres, accéléré x1,5 par M. J. Vaucelle, avec 4 batteries CaNi de 150 mA (interchangeables avec 4 éléments présentés avec un chargeur indépendant). Réalisé pour HP41CX 1525 reg., le module CX603C-JV sert de réserve de fichiers, données ou logiciel en le plaçant en port 1, et pour l'enregistrement fiable d'environ 60 en port 3 (avant de l'échanger avec un autre module disponible), pour obtenir une saisie illimitée et sûre: ces modules gardent l'information 10h, avant de recharger en 10s leur condensateur dans le calculateur et ne sont pas affectés en port 3 par l'affichage MEMORY LOST du HP41CX. Réalisés et diffusés par M. Jacques Vaucelle, La Chasserie, 35133 Romagné, ces modules regroupent un module HPB2180A et deux modules HPB2181A. Le transfert sur cassette des données se fait avec une unité HPB2161A modifiée pour être alimentée par l'allume-cigare d'un véhicule."

Yves ALAJOUANINE (T144P15)

Deux autres exposés parmi les plus appréciés ont été l'oeuvre de Jean-Jacques Dhénin et de Pierre David.

Jean-Jacques nous a expliqué le pourquoi et le comment des Poll Handlers sur l'HP41 et l'HP71. Il est toujours très intéressant quand on fait de l'assembleur (du microcode) de pouvoir reprendre la main tandis que le pointeur est dans les routines internes de la machine.

Par exemple, sur la HP41, de l'adresse FF4 à l'adresse FFA se trouvent ces 7 adresses très importantes que sont les Poll Handlers. Ces adresses sont interrogées par un Fetch dans de très nombreux points des routines internes de la machine et qui permettent de reprendre la main tandis que le calculateur travaille. En effet, si ces adresses contiennent un 0, le programme interrogateur continue sans s'en préoccuper, sinon il se branche à celle qui est non nulle; elle contient toujours un JNC 0-., ce qui est un saut relatif et permet de placer le pointeur au début d'un programme déterminé.

L'adresse est interrogée lors de:

FF4 Pause,
FF5 Après chaque ligne de programme,
FF6 Au réveil de sommeil profond,
FF7 Lors de la fonction OFF,
FF8 Si le flag E/S système est levé,
FF9 Allumage par pression sur ON,
FFA Départ à froid (MEMORY LOST)

(cf "Au fond de la HP41" de J.D. DODIN)

Pierre nous a fait un exposé sur le Forth en général qu'il a ensuite appliqué au HP71. Il a fait comprendre à ceux qui ne connaissaient pas ce langage toute sa puissance, sa vitesse, ses atouts, son concept. Il nous a montré, à nous, "dingues" de la 41, que le Forth était bien un langage RPN, comme celui de notre machine, mais en beaucoup plus puissant: comme sur la 41, une pile, mais infinie celle-là; une pile de retour de sous-programme, également infinie ! comme sur la 41, des XEQ, mais là, pas de distinction entre les routines utilisateur et les fonctions, les routines utilisateur étant elle-même compilées et devenant ainsi de nouvelles fonctions. Bref, il nous a fait envie, et si personne en l'écoutant n'a songé à lâcher sa 41, tout le monde s'est tout de même mis à rêver d'une super 41 qui peu à peu s'était mise à prendre sérieusement des allures de HP71... Gageons que beaucoup achèteront bientôt la grande soeur...

Enfin a été annoncée à cette réunion la création par moi-même ainsi que par Jean-Claude Bornes, Georges-Noël Nicolas et Bruno Trédez du club GUIC, Groupe des Utilisateurs de l'IBM PC et Compatibles. Ce club, destiné aux utilisateurs de l'IBM PC, du Sanyo 550 et autres compatibles, a la volonté d'être du type de PPC dont il a d'ailleurs adopté les statuts à quelques différences près. Comme lui, nous allons éditer un journal qui paraîtra tous les deux mois, et qui sera l'oeuvre de ses adhérents. Je peux vous annoncer que nous avons déjà obtenu des prix intéressants sur du logiciel et sur du matériel, notamment sur la mémoire. Me contacter pour renseignements.

Voilà, je vous ai raconté l'essentiel; l'essentiel à mes yeux naturellement, car je n'étais pas de toutes les conversations. Je n'ai pas pu être au courant de toutes les astuces qui se sont échangées, je n'ai pas pu voir toutes les démonstrations, je n'ai pas pu voir tous les programmes tourner. Mais rassurez vous pour moi, il aurait fallu à chacun beaucoup plus de deux jours pour montrer tout ce qu'il avait à montrer, pour raconter ses difficultés, la manière avec laquelle il les avait résolus et surtout recevoir des autres tout ce qu'ils avaient à donner.

En plus de tout ce que j'ai vu, je retiendrai l'ambiance extraordinaire qui y régna comme à l'accoutumée d'ailleurs, mélange de joie de la découverte, d'étonnements parfois, d'amitié surtout. Car c'était bien sûr une sorte de réunion de famille, une réunion de membres de PPC.

Remercions pour finir le CISL (Centre International de Séjour de Lyon) qui nous a permis de nous réunir dans de bonnes conditions, très bonnes même, puisqu'il nous a offert, sur place ou à proximité, une excellente salle de réunion, le couvert et l'hébergement. Il ne serait guère étonnant que la prochaine rencontre sur Lyon s'effectue sur les lieux de cette première... prochaine qui, cela dit entre nous, serait prévue pour le premier week-end d'octobre... Mais, motus, hein! Ce n'est qu'une confidence de vous à moi...

A la prochaine, donc

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;
- Cassettes digitales pour HPB2161A: 105 FF pièce, plus 3,10 FF de port par cassette.
- 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 en anglais des 3 premières pages de la 41) au prix de 170FF + 13,50FF de port;
- Le manuel de service de la 41 (en anglais) au prix de 50FF + 6,50FF de port;
- Le manuel technique du convertisseur (en anglais) au prix de 20FF + 3,20FF de port;
- HPIL Interface Specifications (en anglais) au prix de 80 FF + 9,50 FF de port
- The HPIL Integrated Circuit User's manual (en anglais): 40 FF + 6,50 FF de port
- Interface Kit Technical Guide (en anglais): 15 FF + 3,50 FF de port
- Manuel technique du Minitel R (Ref. 250 Alcatel): 40 FF + 6,50 FF 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 GROUPEES

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.

Sat page
73

Préserver cette page



Toujours en retard
ce facteur!!!

Pour toujours
H. P.

mini.

IL ETAIT UNE FOIS LE SICOB

Depuis le succès de notre présence au Sicob l'année dernière, nous nous étions promis de recommencer cette année. C'est maintenant chose faite. Et nous pensons déjà à l'année prochaine !

Nous avons décidé de préparer longtemps à l'avance une animation pour ce stand.

Bien évidemment, deux semaines avant, rien n'était prévu.

Ce n'est pas de la négligence de notre part, mais un excès de travail. Le journal que vous avez entre les mains a encore demandé beaucoup d'huile de coude !

HP nous a prêté trois interfaces vidéo, (avec leur moniteur cette fois-ci...), deux HP71B avec modules HPIL et Forth/Assembleur, et un lecteur de disquettes.

Ulérieurement, nous avons eu une HP41CX, un lecteur de cassettes, et une table traçante dont nous reparlerons bientôt.

La Règle à Calcul nous a prêté un HP71B avec module HPIL, et un module mémoire 4Ko.

Nous avons pris un stand de 18m2 (6x3m). PPC-Toulouse, en la personne de J.F. Sibille était représenté.

Nous étions quatre pour aménager le stand, le samedi précédant l'ouverture. Six heures de travail continu, et nous avons fini juste à temps pour être légèrement en retard à la réunion du 4 mai. Nous sommes revenus le lendemain pour figurer quelques détails.

Trois programmes tournaient en permanence sur le stand:

En premier lieu, un programme écrit par Philippe Guez, montrant les possibilités de l'interface minitel de Lionel Ancelet en tant que moniteur vidéo.

Ensuite, un programme collectif: Pierre Colignon, ainsi que Seniak Nitsan ont concocté un programme

de traduction d'expressions en notation traditionnelles (avec des parenthèses) vers la notation RPN. Eric Gengoux a fait un programme qui fait défiler un TGV sur l'écran du HP71B. Je suis intervenu pour réunir le tout, faire la présentation et l'introduction des formules au clavier. L'habillage, en quelque sorte...

Finalement, j'ai rédigé un petit programme qui contrôlait deux interfaces vidéo: il affichait 14 (8+6) pages d'informations sur les deux Clubs. En plus, il gérait le défilement d'un petit train sur l'afficheur du HP71B (l'idée vient du TGV de Eric...). Ce programme montrait les capacités "multi-tâches" du HP71, qui contrôlait le tout simultanément, sans oublier quelqu'un en cours de route...

Christian Baquet, pour sa part, présentait au public son interface analogique/digitale HPIL, en démonstration avec un HP71B.

Le stand était comme suit:

```
-----  
|   | ARM |   |  
|   | OI | TABLE DES JPC |  
|   | RE |   |  
|   |-----|  
|   |   |   |  
|   | BUR |   |  
|   | EAU |   |  
|   |-----|
```

```
-----  
| V 71 |   | |
|   | / \ |  
| V   | | |  
| 71 | \ / |  
| V   | - |  
|   |   |  
|-----|
```

"V" représente les Vidéos. En plus, sur le coin

de l'armoire était placé le minitel, ainsi qu'une magnifique affiche de Einstein, en carton rigide. Le HP71 controlant le minitel était à l'abri des regards (et surtout des mains) indiscrets...

Ce "plan" est à l'échelle dans la limite de la précision de la HPB2905B+HP71B qui imprime la maquette de ce journal...

Il y a eu, toute la semaine, une mobilisation des adhérents du Club pur venir tenir le stand.

Nous n'étions guère loin du stand HP. Il était juste à peine plus beau que le notre (!!!). Ce fut l'occasion de contacts fructueux avec les gens d'HPFrance. Après deux à trois années d'existence, on commence à s'intéresser à nous. Tant mieux. Peut-être un jour arriverons-nous aux 53,7% de réduction annoncés dans le numéro du 1er avril. Qui sait ? Plus sérieusement, ce genre de contacts peut déboucher sur des réalités encore inimaginables. Nous vous tiendrons bien sûr informés.

HP présentait sur son stand l'interface vidéo graphique dont je vous parlais dans le No22 (Mars85). Hélas, dès le deuxième jour, elle était volée. Alors, nous avons prêté à HP la table traçante qui nous avait été prêtée. Solidarité, solidarité...

Le déménagement s'est fait dans la précipitation, l'organisation du Sicob fermant toutes les issues afin d'éviter les vols. Attitude compréhensible, mais bien embêtante pour quelques pauvres individus qui prennent du temps sur leur week-end afin d'aider leur Club favori... Enfin, tout s'est bien passé, en moins de temps que pour l'aménagement.

C'est maintenant l'heure des comptes: nous sommes 24 de plus. Ceci peut paraître bien peu par rapport à l'investissement, mais il ne faut pas oublier les adhésions indirectes, les contacts, les rencontres, etc...

Cette année encore, un bilan largement positif, qui nous donne envie de recommencer. J'espère que ceux qui ont tenu le stand ont éprouvé comme moi du plaisir à voir, rencontrer,

discuter, échanger, communiquer.

Il me reste à remercier tous ceux qui ont donné de leur temps pour aider PPC-PARIS, tous ceux qui ont aidé financièrement, et tous ceux qui sont passé sur le stand.

Allez, on recommencera l'année prochaine...

Pierre DAVID



Et oui, voilà l'ambiance
du sicob

LE COURRIER DES LECTEURS

COURRIER DU COEUR

Jean REIBEL, 9 square Fleming, 92350 Le Plessis Robinson, (Tel.: (1) 631 46 11). Vends HP41CV 1100FF, Interface Vidéo 1200FF, module HPIL 500FF, Crayon Optique 700FF, Module Math/Stat/Jeux/STRD 200FF, Nx livres.

Philippe TENAND, 2bis rue De L'Egalité, 94300 Vincennes, (Tel. le soir: (1) 808 17 25). Vends un module HP41 REAL ESTATE 200FF (prix normal 407FF) et des livrets HP d'application complètement introuvables et utiles pour la programmation de n'importe quel ordinateur. Les organigrammes inclus font figure de référence (état impeccable). HP67/97 Astrology (11 programmes en anglais, très détaillés et expliqués) 100FF. HP67/97 Games (11 programmes en anglais, dont simulation de football et bridge) 100FF. HP67/97 Games of chance (10 programmes en anglais, livret retiré très tôt de la vente, même pendant l'époque des HP67) 100FF. HP19C/29C Finance (10 programmes en anglais avec détails et formules, comme tous les livrets de cette liste) 50FF. Deux paquets de feuillets de programmation, jamais utilisés. HP65 Programs (sous forme de , une vraie pièce de collection) 50FF. HP00097-90081 (bloc déjà plus récent, feuilles volantes, je fournis l'emballage) 35FF. Programmable pocket calculators (importation USA) 100FF, sur un historique des calculatrices HP et RPN, (prix normal environ 200FF). Le calculateur programmable de poche et ses jeux 50FF (prix normal inconnu, à débattre). Fascicule Micro-Revue (fait par PPCT) No 1, spécial codes-barres (avec programmes, tout pour HP) 40FF (c'est le prix normal, mais il paraît qu'il est épuisé, alors j'en profite!). Annonce faite pour Paris et sa région. Pour grande banlieue et ailleurs, en cas de colis, frais à votre charge (se renseigner et rajouter). Echange de livrets possible (contre livrets HP41, ou le tout contre module TI99/4A). D'autre part, j'achète tout module TI99/4A et journaux HP Key Notes.

Vincent HERLICO (PB1), 3 rue Berthaux Dumas, 92200 Neuilly, (Tel.: (1) 624 26 17). Vends HPB2161A (du 9/84) 3100FF et HPB2162A (du 6/83) 2900FF.

François ECKERT (PBB), 45 rue du docteur Babin, St Germain lès Arpajon, 91290 ARPAJON, (Tel.: (6) 016 48 54. Vends une HP41CV, une imprimante 82143A, un lecteur de cartes, les modules Time - XFunctions - XMemory - Math et Financial, cartes magnétiques, livres, doc... , le tout pour 4500FF.

Michel MARTINET (P#12 SIG#2), 17 rue Eugène Hénaff - 93200 ST DENIS (Tel: (1)8220149). Vend un HP75C + module 8k RAM + module Texte + VISICALC + de très nombreux logiciels, le tout pour 7000 Frs. (sept mille Francs).

dans le commerce, mais disponible auprès d'HP en pièces détachées), soit en achetant l'HPIL et l'interface HPIL <=> RS-232C et faire les branchements appropriés.

En ce qui concerne ta dernière question, je pense que la meilleure façon de connaître la loi et d'avoir une confirmation écrite, est de la demander au rectorat de ta circonscription. Quoiqu'il en soit, la HP41 entre dans la catégorie des "calculatrices" admises aux examens à la condition expresse qu'elle ne soit connectée à "rien", (ni chargeur, ni module, ni périphérique. Les modules peuvent être rendus invisibles soit par intégration dans la machine, soit par un gratage approprié de leur nom).

Amitiés et Heureuse Programmation aux examens.

Hugues OBERLE
1 Rue Beaulieu
68440 ZIMMERSHEIM

Monsieur,

Désirant faire un programme qui appelle certaines routines du module MATH, routines qui ne commencent pas sur un LBL apparaissant au catalog 2, j'ai désespérément cherché un moyen pour positionner (par Programme) le pointeur à un endroit précis en ROM.

Y-a-t'il un registre qui contienne une instruction qui permette à la machine de savoir si elle est en ROM ?

Cher Hugues

La réponse à ta question a déjà été apportée par P. DAVASE dans le numero 22 p. 16. Néanmoins, je préciserais le fait que le moyen utilisé consiste à placer le pointeur b en ROM à l'aide de la

programmation synthétique.

François LE GRAND
99 rue Josse Impens
1030 BRUXELLES
BELGIQUE

Réactions au JPC No 23

1. En bas de page 5: "Accord avec HP" avant dernière ligne: "Réduction de 53.7% sur le prix catalogue"

J'ai lu l'article au moins 2 fois avant de m'en remettre. D'ailleurs, je suis toujours sceptique.

3 solutions:

- Une faute de frappe (13.7%, 23.7%) ?...

- Un poisson d'avril

- C'est vrai ! Dans ce cas, bravo (on ne sait jamais). Si c'est vrai, comment faut-il s'y prendre ? Où faut-il s'adresser ? Pourriez vous m'envoyer les nouveaux prix du matériel qui nous interesse ?

2. La carte de membre offre 25% dans certains magasins, lesquels ? Et sur quels articles ?

3. Où peut on se procurer les I.D.S 71 au meilleur prix ? (club, HP, RAC)

4. Où peut on se procurer FRALEX ?

5. Qu'est ce que le S.I.G. ? Pourquoi: SATURN ?

6. Petit problème de programmation HP71.

Je veux arriver à un résultat semblable à l'exemple p 238 du manuel d'utilisation. Pour cela, j'utilise la formule: DISP USING...@ INPUT... au lieu de INPUT (INPUT USING indisponible) or, je n'arrive pas à faire entrer un CHR\$(27) dans un ordre IMAGE, pas plus que n'importe quel CHR\$(), ou n'importe quelle variable alphanumérique. Il faut que la chaîne à afficher soit en toutes lettres, entre guillemets

La seule solution que j'ai trouvée pour incorporer un CHR\$(27) dans un ordre IMAGE est d'utiliser un ordre PRINT USING au lieu de DISP USING, et d'utiliser le symbole image /(ENDLINE)

avec ENDLINE CHR\$(27). Tant qu'il n'y a pas d'imprimante sur la boucle, ça se passe bien. Autrement, je ne sais pas (pas d'imprimante directement sur la boucle) mais je pense que bonjour les dégats, ou du moins, bonjour la pagaille.

Beaucoup de questions
Pourriez vous répondre par retour de courrier aux questions 1 à 5, la sixième demandant sans doute quelque réflexion ?

Tous mes encouragements. Bien à vous

Réponse de S16 #3

Etant donné le nombre important de questions que tu nous poses, je pense qu'il serait logique de les reprendre une à une:

1. Il s'agit comme tu le propose d'ailleurs fort judicieusement d'un poisson d'avril.
2. La carte de membre offre des réductions sur tous les articles HP. Pour connaître les magasins qui nous font cette ristourne, contacte-nous directement au club.
3. Les I.D.S sont en vente à la Règle à calcul, et auprès d'HP.
4. Pour se procurer FRALEX, envoie-nous un support quelconque au club; nous nous ferons une joie de te permettre de rester Français.
5. Parce que TITAN est un des satellites de SATURNE.
6. Il est indiqué dans le manuel de référence, à la page 139 que le symbole image B a pour signification CHR\$. Il suffit donc d'entrer un ordre du genre: DISP USING "B,"P71";72 pour obtenir le message HP71.

Voilà; j'espère avoir éclairé ta lanterne sur ces quelques problèmes.

Bonne programmation.
Laurent ISTRIA

J. HORVILLE
31, rue Berthollet Arcueil, le 18 avril 1985
94110 ARCUEIL

#205

Monsieur,

M. Pierre DAVID, dans le No 23 de JPC, invite les lecteurs à signaler les erreurs trouvées dans dans le Journal.

Je pense qu'il s'en trouve une dans sa liste de rectificatifs, celle-ci: Dans le No 21:

"3160 FOR J=N TO N/2 STEP -1"

Cette ligne, où se trouve-t-elle ?... et que doit-on faire ?

Dans le programme VALPROP de M. Olivier Arbey, No22 page 53 tout est parfait pour les matrices 3 x 3 mais au-delà rien d'autre qu'un sifflement (méprisant pour ceux qui ne savent pas se dépanner tout seul).

J'ai d'autre part un problème qui doit-être commun à de nombreux HPiste. Est-il normal de ne pas pouvoir imprimer les quelques lettres grecques que la 71 affiche et que la 82162 imprime... mais sous la dictée de la 41 et non sous celle de la 71 (module 82401). Il me semble que ce n'est pas pinailler que de vouloir appeler un angle θ plutôt que T. Par contre, le CHR\$(96) s'imprime et m'intrigue.

J'ai par ailleurs remarqué que dans le programme MATRICES de M. Frédéric BARNAUD la 75 parlait exactement comme la 71. Si les différences sont si peu nombreuses, JPC ne pourrait-il pas publier un petit comparatif ?

J'espère ne pas vous avoir ennuyé(s) et j'adresse à M. le Président et à toute l'équipe JPC l'expression de ma plus cordiale sympathie.

REPONSE DU DOCTEUR JPC

Avant tout merci de lire JPC, et de le lire suffisamment pour y déceler les erreurs qui s'y sont glissées (de moins en moins nombreuses à l'avenir: nous l'espérons tous).

Pour ce qui est des caractères grecs, il faut d'abord savoir que la plupart des ordinateurs de nos jours utilisent l'ASCII (American Standard Code for Information Interchange) et que les portables HP eux aussi l'utilisent. Dans ce code, nous avons 128 caractères (CHR\$(0) à CHR\$(127)). Les codes 0 à 31 et le 127 sont des codes de contrôle, ils ne représentent généralement pas de lettres ou chiffres mais des commandes entre l'ordinateur et ses périphériques (Exemple: CHR\$(8), CHR\$(10), CHR\$(13) sont respectivement Back Space, Line Feed et Carriage Return). Les portables HP n'utilisent pas tous les caractères de contrôle et remplacent généralement les inutilisés par des caractères supplémentaires dans le genre des caractères grecs. Ces caractères sont "locaux" et ne sont pas implémentés sur les périphériques. Seuls les caractères (32 à 126) sont respectés et sont communs à tous les périphériques ASCII. Pour la compatibilité HP75 <=> HP71, à peu près tous les programmes HP75 sont compatibles sur le HP71 (Moyennant néanmoins des petites modifications). L'inverse est par contre quasiment impossible).

Pour ce qui est des programmes, ils ont été essayés et testés avant de paraître dans le journal. Il peut parfois arriver qu'une erreur se glisse dans un fichier texte mais pour ce qui est des programmes BASIC dont tu parles nous les utilisons assez souvent et pour notre part, ils fonctionnent à merveille. Tu peux venir aux réunions mensuelles (Le premier Samedi de chaque mois) et nous nous ferons un plaisir de te fournir tous les programmes dont tu as besoin.

Voilà, j'espère avoir été suffisamment clair.

Heureuse Programmation .



Housay
85

*Es tu compris mon
p'tit gars*

HP-41

Christian JEGOUZO	HP41 ET RECURSIVITE (TOURS DE HANOI)
Philippe TENAND	EPR0M ET MICROCODE
Gilbert TISSERAND	DES DRAPEAUX A LA CHAINE
Denis CASTELAIN	ORDINATEUR DE BORD
Steen PETERSEN	PROPOSED STANDARD HP41 MACHINE LANGUAGE BAR CODE
Philippe GUEZ	Traduction de l'article ci-dessus
Steen PETERSEN	ADDENDUM
Philippe GUEZ	Traduction de l'article ci-dessus

Lors de la définition d'une procédure ou d'une fonction, il est possible de faire référence, dans le bloc d'instructions, à cette procédure (ou à cette fonction) elle-même. C'est ce qu'on appelle une procédure (fonction) récursive.

Dans les langages évolués (PASCAL, MODULA...), il est possible d'écrire des procédures récursives puisqu'à chaque appel de la procédure, de nouvelles variables locales sont engendrées. La récursivité implique, en effet, l'empilement des résultats intermédiaires (segments de données).

Il est bien sûr, à priori, difficile de programmer sur une HP-41 des procédures récursives. L'objet de cet article est de vous montrer que l'on peut contourner le problème et j'utiliserai le célèbre exemple des TOURS de HANOI pour illustrer mon propos.

1 Différentes Méthodes d'Implémentation du langage PASCAL

Les différentes méthodes actuelles pour implémenter le langage PASCAL peuvent se regrouper en 3 familles :

- Traduction dans le langage de l'ordinateur cible.
- Traduction en langage intermédiaire et interprétation.
- Traduction en langage intermédiaire, puis traduction en langage de base de l'ordinateur cible.

Le langage intermédiaire est généralement appelé " P-CODE ".

Examinons la deuxième méthode :

L'exécution d'un programme PASCAL se fait en 2 temps :

- Le compilateur compile le programme (analyse lexicale, syntaxique, sémantique...) et génère un fichier p-code.
- Ce fichier p-code est interprété

La première opération n'est bien sûr pas réalisable avec une HP-41, mais il est possible d'écrire l'interpréteur de p-code.

Lors d'une étude du compilateur PL3 (proche de PASCAL) et de l'interpréteur associé (INTPL3), j'ai pu me procurer le fichier p-code obtenu après compilation du programme des tours de Hanoi.

Une bonne connaissance du compilateur permet d'ailleurs de générer le code " à la main ". (cas des tours de Hanoi où il y a peu de variables)

2 Architecture de la Machine PL3 (HP-41)

La possibilité d'utiliser la récursivité dans le langage PL3 suggère le choix d'une machine à pile. (cas de la HP-41)

Cette machine consiste en 2 mémoires, un registre d'instructions et 3 registres d'adresse, soit :

- Une mémoire de programmes chargée avec le code produit et qui reste inchangée durant l'interprétation de ce code. C'est une mémoire à lecture seule.
(cette mémoire sera pour la HP-41 un fichier de données HANOI. le p-code sera sous la forme de Nombres Non Normalisés et non sous forme alphanumérique)
- Une mémoire de données organisée comme une pile. Les instructions de calcul travailleront essentiellement sur le sommet de la pile et éventuellement sur le sous-sommet.
(les registres 01 à n de la HP-41 représenteront la pile)
- Un registre d'instruction I qui contient l'instruction en cours d'interprétation.
(une instruction est de la forme INS L,A . Elle est décodée dans le registre M. A l'appel du sous-programme INS, L se trouve dans le registre O et A en X dans la pile)
- Un registre programme P qui contient l'adresse de la prochaine instruction. à interpréter = compteur ordinal.
(registre N de la HP-41)
- un registre T qui contient l'adresse du sommet de la pile. L'élément du sommet de la pile est adressé par ce registre.
(registre M de la HP-41)
- Un registre de base B qui contient l'adresse du segment de données le plus récemment alloué.
(registre 00 de la HP-41)

3 Récursivité

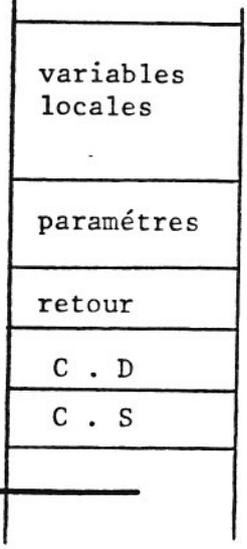
Chaque procédure dans PL3 contient des variables locales. Ces procédures pouvant être activées récursivement en cours d'exécution, la mémoire pour les variables locales ne peut être allouée avant l'appel effectif de ces procédures.

A chaque appel d'une procédure correspondra l'empilement d'un segment de données qui contient, outre l'emplacement des variables locales et des paramètres :

- L'adresse programme de l'appel
- L'adresse du dernier segment de données (chainage dynamique)
- La différence de niveau d'imbrication (chainage statique)

L'adresse origine du segment de données le plus récemment empilé est mémorisé dans le registre de base B.

Un segment de données peut donc être représenté ainsi :



le chainage dynamique permet de savoir qui a appelé la procédure. Le chainage statique assure la récursivité et l'accès aux variables.
Pour accéder à une variable, il faut donc connaître

- l'adresse relative de la variable dans le segment de données.
- l'adresse de base du segment approprié.

Accès à une variable :

- Si la variable est locale à la procédure, l'adresse de base du segment de données est donnée par le registre de base B.
- Sinon elle est obtenue en descendant les segments de données grâce au chaînage statique. (sous-pgm BASE)

4 HP-41 et TOURS de HANOI

Il s'agit d'un jeu et d'une légende, rapportés d'Extrême-Orient. Des moines bouddhistes jouent avec des disques empilés sur des pieux, et la fin de la partie est censée symboliser et annoncer la fin du monde. Ils disposent de 3 pieux, ou tours, susceptibles de recevoir des disques, empilés suivant des diamètres décroissants (il y en a 64). Le jeu consiste à faire passer les 64 disques de la tour n° 1 à la tour n° 2, en ne déplaçant qu'un seul disque à la fois, et en utilisant la tour intermédiaire n° 3 de telle sorte qu'à aucun moment un disque ne soit empilé sur un disque de plus petite dimension.

Le programme des Tours de Hanoi en langage PL3 peut s'écrire ainsi :

```
0  VAR X,Y;
1  PROCEDURE HANOI (N,D,B,I);
2  IF N > 0 THEN
6   BEGIN
7   HANOI (N-1,D,I,B);
15  Y:=Y+1;
19  WRITELN(Y:2,':',D:1,' > ',B:2);
23  HANOI(N-1,I,B,D);
31  END;
32  BEGIN (* corps du pgm *)
33  WRITE('N = ?');
35  READLN(X);
37  Y:=0;
39  HANOI (X,1,2,3);
44  END.
```

Les chiffres placés à gauche indique le numéro de la ligne p-code correspondante.

Le code généré est le suivant :

0	JMP 0,32	1	JMP 0, 2
2	INT 0, 7	3	LOD 0, 3
4	LIT 0, 0	5	OPR 0;12 [†]
6	JPC 0,31	7	MST 1, 0
8	LOD 0, 3	9	LIT 0, 1
10	OPR 0, 3	11	LOD 0, 4
12	LOD 0, 6	13	LOD 0, 5
14	CAL 4, 2	15	LOD 1, 4
16	LIT 0, 1	17	OPR 0, 2
18	STO 1, 4	19	LOD 1, 4
20	LOD 0, 4	21	LOD 0, 5
22	OPR 0,15	23	MST 1, 0
24	LOD 0, 3	25	LIT 0, 1
26	OPR 0, 3	27	LOD 0, 6
28	LOD 0, 5	29	LOD 0, 4
30	CAL 4, 2	31	OPR 0, 0
32	INT 0, 5	33	LCA 0,34

34 N = ?	35 OPR 0,17
36 STO 0, 3	37 LIT 0, 0
38 STO 0, 4	39 MST 0, 0
40 LOD 0, 3	41 LIT 0, 1
42 LIT 0, 2	43 LIT 0, 3
44 CAL 4, 2	45 OPR 0, 0

Vous remarquerez que je n'ai pas tenu compte du champ d'affichage des variables (ex : Y:2) afin d'afficher le résultat sur la HP-41 en une seule fois (OPR 0,15).

J'ai donc modifié le code original et les instructions d'affichage pour être compatible avec la HP-41.

5 L'Interpréteur INTPL3 :

Il suffit de le lancer par XEQ^T INTPL3. Lorsque l'affichage montre " N = ?" Entrer le nombre de tours considérées et R/S.

L'affichage indiquera le numéro du déplacement, le numéro du socle sur lequel on prend le disque et le numéro du socle sur lequel on le pose : ex : 6: 2 > 3 6^{ème} déplacement - socle n° 2 vers n° 3

Auparavant, il faut créer le fichier de données HANOI (Nombres Non Normalisés). Procéder comme suit :

- Créer un fichier HANOI de 46 registres.
- Se positionner en début de fichier.
- Ecrire en Alpha l'instruction p-code sur 6 caractères.
ex : LOD 0, 3 s'écrira LOD0,03
puis RCL M et SAVEX etc...

Description sommaire de l'interpréteur :

- Pas 01 à 40 décodage de l'instruction p-code.
- Les labels alphanumériques correspondent à l'interprétation de chaque type d'instruction.

Description sommaire des instructions p-code :

- LIT : chargement immédiat sur le sommet de la pile S(T).
- LOD : chargement du contenu d'une variable.
- STO : rangement du sommet de la pile dans une mémoire.
- MST : construction des chainages dynamique et statique.
- CAL : activation de procédure.
- INT : gestion du pointeur de pile
- JMP : saut incondtionnel.
- JPC : saut conditionnel.
- LCA : prépare affichage d'un message.

- OPR : instructions arithmétiques et logiques

Champ A = 00 : retour procedure

01 : addition

02 : soustraction

12 : comparaison $S(T-1) > S(T)$?

15 : affichage de $S(T)$, $S(T-1)$, $S(T-2)$

17 : affichage d'un message

L'interpréteur sur la HP-41 fonctionne rigoureusement de la même façon que sur l'ordinateur. Seules les fonctions d'affichage ont été adaptées et transformées.

6 Exemple d'exécution :

Pour n tours, il faut $2^n - 1$ déplacements ($2^{64} - 1$ est un nombre très grand)
Je vous donne les 63 déplacements nécessaires si n = 6, ainsi que le chronométrage sur une machine non accélérée.

1: 1 > 3	0h03mn42s	33: 3 > 2	0h46mn28s
2: 1 > 2	4 35	34: 3 > 1	47 22
3: 3 > 2	5 54	35: 2 > 1	48 41
4: 1 > 3	6 49	36: 3 > 2	49 35
5: 2 > 1	8 35	37: 1 > 3	51 23
6: 2 > 3	9 29	38: 1 > 2	52 15
7: 1 > 3	10 48	39: 3 > 2	53 35
8: 1 > 2	11 45	40: 3 > 1	54 33
9: 3 > 2	14 00	41: 2 > 1	56 48
10: 3 > 1	14 53	42: 2 > 3	57 40
11: 2 > 1	16 12	43: 1 > 3	59 00
12: 3 > 2	17 08	44: 2 > 1	59 54
13: 1 > 3	18 55	45: 3 > 2	1h01mn42s
14: 1 > 2	19 48	46: 3 > 1	2 35
15: 3 > 2	21 07	47: 2 > 1	3 54
16: 1 > 3	22 07	48: 3 > 2	4 35
17: 2 > 1	24 50	49: 1 > 3	7 37
18: 2 > 3	25 43	50: 1 > 2	8 30
19: 1 > 3	27 02	51: 3 > 2	9 49
20: 2 > 1	27 57	52: 1 > 3	10 44
21: 3 > 2	29 44	53: 2 > 1	12 33
22: 3 > 1	30 37	54: 2 > 3	13 27
24: 2 > 3	31 57	55: 1 > 3	14 43
25: 1 > 3	32 54	56: 1 > 2	15 42
26: 1 > 2	36 04	57: 3 > 2	17 56
27: 3 > 2	37 20	58: 3 > 1	18 49
28: 1 > 3	38 16	59: 2 > 1	20 09
29: 2 > 1	40 03	60: 3 > 2	21 04
30: 2 > 3	40 56	61: 1 > 3	22 50
31: 1 > 3	42 15	62: 1 > 2	23 44
32: 1 > 2	0h43mn16s	63: 3 > 2	25 02
		N = ?	1h26mn00s

Pour n = 9, j'ai obtenu les 511 déplacements en 11 heures et 32 minutes.

Pour n = 6, la pile d'exécution monte jusqu'au registre 54.

Si vous voulez regarder de très près le fonctionnement récursif, il faut :

- rajouter AVIEW après STO M du pas 14 afin de visualiser l'instruction en cours.

- recopier la pile d'exécution sur papier RCL 01, RCL 02

BON COURAGE...

L'INTERPRETEUR INTPL3

1 LBL INTPL3	51 RCL IND X	101 -	151 CLA	201 RTN
CLRG	X <> Y	STO IND Y	STO M	LBL [†] CAL
1	RDN	RDN	R [†]	RCL M
STO 00	DSE Y	STO M	R [†]	RCL 0
CLST	GTO 10	RTN	PROMPT	2
[†] HANOI	+	LBL 12	STO IND Z	+
SEEKPTA	RTN	RCL M	RDN	-
LBL 20	LBL [†] IIT	1	STO N	STO 00
CLA	1	ST- M	RDN	2
10 SEEKPT	60 ST+ M	110 RCL IND Y	160 STO M	210 +
1	X <> Y	RCL IND M	RTN	RCL N
+	STO IND M	X > Y ?	LBL [†] LCA	STO IND Y
GETX	RTN	SF 00	RCL M	X <> Y
STO M	LBL [†] OPR	FS? 00	1	3
RDN	GTO IND X	1	ST+ N	-
CF 28	LBL 00	FC?C 00	+	STO M
ANUM	RCL 00	0	STO M	RCL Z
[†] ABC	1	STO IND M	X <> Y	STO N
0	-	RTN	STO IND Y	RTN
20 STO M	70 STO M	120 LBL 15	170 RTN	220 LBL [†] INT
RDN	3	RCL N	LBL [†] LOD	ST+ M
3	+	RCL M	1	RTN
AROT	RCL IND X	2	ST+ M	LBL [†] JMP
ASTO X	STO N	-	RDN	STO N
R [†]	X <> Y	CLA	XEQ [†] BASE	RTN
STO M	1	FIX 0	+	LBL [†] JPC
R [†]	-	CF 29	RCL IND X	RCL IND M
STO N	RCL IND X	ARCL IND X	STO IND M	1
X <> T	STO 00	[†] †:‡	RTN	ST- M
30 INT	80 RTN	130 1	180 LBL [†] STO	230 RDN
STO 0	LBL 02	+	XEQ [†] BASE	X ≠ 0 ?
RDN	RCL M	ARCL IND X	+	RTN
LAST X	RCL IND X	[†] †‡‡	RCL IND M	RDN
FRC	X <> Y	1	STO IND Y	STO N
E2	1	+	1	1
*	-	ARCL IND X	ST- M	ST- M
XEQ IND Z	X <> Y	AVIEW	RTN	END
RCL M	RCL IND Y	TONE P	LBL [†] MST	
RCL N	+	3	RCL M	
40 GTO 20	90 STO IND Y	140 -	190 1	
LBL [†] BASE	RDN	STO M	+	
RCL 0	STO M	X <> Y	XEQ [†] BASE	
X ≠ 0 ?	RTN	STO N	STO IND Y	
GTO 11	LBL 03	RTN	X <> Y	
RDN	RCL IND M	LBL 17	1	
RCL 00	RCL M	RCL M	+	
RTN	1	RCL N	RCL 00	
LBL 11	-	RCL IND Y	STO IND Y	
RCL 00	RCL IND X	SEEKPT	3	
50 LBL 10	100 RCL Z	150 GETX	200 ST+ M	

EPRM ET MICROCODE
POUR HP41

Ces matériels sont importés et vendus par des clubs d'utilisateurs à leur membres ou par des importateurs spécialisés. A savoir, le PPC PARIS-CHAPTER, les éditions du Cagire, ou par LCI. Les différents types de matériels sont:

- Les eproms,
- Les lecteurs d'epron
- Les programmeurs et duplicateurs d'epron,
- Les MLDL (pour le microcode) et le protocoder,
- L'union d'un MLDL et d'un lecteur d'epron, abusivement appelé parfois MLDL.

Tout d'abord, qu'est-ce qu'une epron? On sait que la ROM est une mémoire morte en circuit intégré, obtenu par procédé photographique et ineffaçable. L'epron est programmable, mais est effaçable seulement par ultraviolets et en laboratoire. Elle revient beaucoup moins cher qu'une ROM. Il faut séparer le système des eproms du système MLDL, dont le but est la programmation en microcode sur HP41. Un MLDL est aussi appelé un simulateur de ROM. Pourquoi? Parce que la programmation en microcode ne peut pas se faire directement dans la HP41, mais dans des RAM et des ROM externes (contenus dans le MLDL). Ceci, parce que les instructions du microprocesseur sont codées sur 10bits, alors que le langage utilisateur l'est sur 8bits.

Il existe en effet 3 sortes de programmation dans une HP41; l'affichage normal (ou assembleur évolué); '113 AVIEW', ce qui est dans la mémoire; 'F4 42 49 4C 4C F3 53', le microcode ou langage machine (au niveau du microprocesseur); '00 0100 1101'. pour éviter une programmation fastidieuse, ils se programment par codes ou s'écrivent en signes appelés mnémoniques (ex: C=0 ALL).

L'avantage de tout ceci, un programme en microcode tourne 350 fois plus rapidement.

Des démonstrations avaient eu lieu pendant le Spécial - Sicob, au stand du club PPC PARIS-CHAPTER. Les prix sont extrêmement variables suivant les nombreux modèles et les

spécificités en fonction des capacités de chacun. Se renseigner auprès des clubs et des organismes dont les adresses ont déjà été données. Voyons les logiciels, très variés (4, 8 ou 16k); navigation, mathématique, programmation synthétique, comptabilité, traitement de texte, gestion, astronomie, construction navale, etc... certains sont en microcodes, d'autres non (de 350 à 600FF). Par comparaison, une ROM fabriqué en série limitée coûte environ 1200FF. Un lecteur, le moins performant (4k, juste pour une epron de 4k maximum), coûte à partir de 800FF, un bon modèle pour 2500FF et un modèle ressemblant à un lecteur de cartes pour 5000FF. A choisir selon le but recherché.

Philippe TENAND.

Notes de l'éditeur:

Quelques précisions quant à ta prose. Les éproms peuvent être non seulement effacés par les ultraviolet (utilisable par n'importe qui s'il possède la lampe appropriée), mais aussi selon l'epron (EEPROM) doivent être effacés électriquement (demande un système électronique beaucoup plus complexe).

Il existe effectivement plusieurs façons de programmer la HP41, que nous appellerons 'standard, synthétique et microcode'. les deux premières étant très proches l'une de l'autre, beaucoup de programmeurs les mêlent l'une et l'autre et utilisent des principes de programmation synthétique pour améliorer, raccourcir et un programme. La première forme de programmation, est celle préconisée par HP et développée par tous les utilisateurs de calculettes HP (séries 50, 90, 60, 30, 20, 10 et 40). C'est du langage symbolique élémentaire ('113 AVIEW') aussi appelée RPN (reverse polish notation). Pour les besoins internes de la HP41, ont été créés à l'intérieur de cette machine, un certain nombre de registres (M, N, ...), qui associés à des octets (144 = RCL, ...) ont permis aux programmeurs PPC de développer la programmation synthétique, qui ne fut acceptée par HP qu'environ 2 ans et demi après la sortie du HP41. D'autre part, sans parler de microcode, seule la programmation synthétique vous permet de "visiter" toute la mémoire de votre chère HP41,

et de pouvoir soit de lire, soit écrire dans les 64k (mémoire ROM + RAM), et cela par le nom des fonctions ('RCL M'), soit par leurs codes hexadécimaux ou décimaux ('90 75'). Enfin, le microcodes vous permet toutes les possibilités de résolution précédemment décrites, mais plus encore, car elle permet d'une part d'avoir une vitesse de 250 fois supérieure à celle du même programme en langage utilisateur mais aussi de pouvoir créer tout ce dont vous avez besoin, entre autres de nouvelles fonctions (par ex, PSIZE ne peut être fait qu'en microcode) et enfin, toutes les routines créées, n'utilisent aucune place dans les 319 registres de mémoire vive de la HP41. Selon les utilitaires de programmation, on peut programmer le MLDL soit en envoyant le code décimal codé binaire, soit envoyer les codes hexadécimaux, soit à l'aide des mnémoniques. Suite à ces quelques précisions, je pense qu'il faut ajouter, qu'on appelle microcode, tous les codes binaire (01 1101 1001), le langage machine est une programmation par l'envoi des codes binaires et langage assembleur est un traducteur du langage machine en mnémoniques.

Il faut enfin signaler que la démonstration de ces matériels, a été faite durant le Spécial Sicob 84 et non le Sicob 85 ou d'autres logiciels ont été présentés.

Philippe.

 DES DRAPEAUX A LA CHAINE

Il arrive souvent que, dans un programme, on ait à mettre une série de drapeaux dans un état donné.

Une solution primitive consiste à faire SF et CF autant de fois qu'il y a de drapeaux. Mais voilà: deux problèmes se posent. D'une part, cette méthode est très lourde, très longue et très gourmande en place mémoire. D'autre part, si l'on veut agir sur les drapeaux 30 à 55, on se heurte

à un message d'erreur bien senti !

La solution consiste à utiliser un peu de programmation synthétique: il faudra utiliser une séquence du type

Chaîne synthétique
 RCL M
 STD d

Ce qui aura pour effet de mettre d'un coup tous les flags dans une configuration donnée. Mais de quoi est composée cette chaîne synthétique ?

Le programme que soumetts à votre perspicacité aujourd'hui a pour but de vous donner les codes correspondant à une configuration de drapeaux donnée.

Vous entrez les numéros des drapeaux que vous désirez voir armer, puis vous finissez en appuyant simplement sur [R/S]. La HP41 vous affiche alors les codes décimaux de la chaîne synthétique à rentrer.

Heureuse Programmation,

Gilbert TISSERAND

01#LBL "SFd"
 02 CLR6
 03 FIX 0
 04#LBL d
 05 "SF?"
 06 PROMPT
 07 FC?C 22
 08 GTO A
 09 ENTER^
 10 ENTER^
 11 B
 12 /
 13 GTO IND X
 14#LBL 00
 15 XEQ 07
 16 ST+ 00
 17 GTO d
 18#LBL 01
 19 XEQ 07
 20 ST+ 01

21 GTO d
22*LBL 02
23 XEQ 07
24 ST+ 02
25 GTO d
26*LBL 03
27 XEQ 07
28 ST+ 03
29 GTO d
30*LBL 04
31 XEQ 07
32 ST+ 04
33 GTO d
34*LBL 05
35 XEQ 07
36 ST+ 05
37 GTO d
38*LBL 06
39 XEQ 07
40 ST+ 06
41 GTO d
42*LBL 07
43 RDN
44 B
45 MOD
46 E
47 +
48 256
49*LBL 08
50 2
51 /
52 DSE Y
53 GTO 08
54 RTN
55*LBL A
56 "247:"
57 ARCL 00
58 ARCL 01
59 ARCL 02
60 ARCL 03
61 ARCL 04
62 ARCL 05
63 ARCL 06
64 AVIEW
65 END

ORDINATEUR DE BORD

Le programme "OB" transforme votre HP41 préférée en ordinateur de bord pour votre automobile.

Il nécessite la présence en XMemory d'un fichier de données "FOB" de 8 registres.

L'utilisation est on ne peut plus simple: il suffit de choisir une touche parmi les suivantes.

[A]: départ de la première étape

[B]: fin d'étape

[C]: départ de l'étape suivante

[D]: moyenne cumulée des différentes étapes, par rapport au temps roulé.

[E]: moyenne cumulée par rapport au temps total.

[a]: consommation sur un plein de carburant, avec rappel de la quantité d'essence consommée depuis l'origine, et rappel du kilométrage au plein précédent.

[b]: prochaine vidange

[e]: sauvegarde des informations en XMemory, et en cassette le cas échéant.

Cette dernière routine vous permet de conserver l'usage de votre HP41 entre deux calculs, les registres étant restaurés par XEQ "OB"

[a] et [b] permettent une simple consultation, sans mise à jour.

P représente la consommation moyenne sur le plein, T la consommation moyenne totale.

A chaque demande d'heure, faites R/S pour obtenir l'heure courante, ou indiquez l'heure souhaitée.

Un petit exemple pour mettre tout ceci en pratique:

[R/S]	-->	"CHOIX"	:	24	RCL 03	
[A]	-->	"COMPTEUR"	:	25	X(Y?)	
9696,00	[R/S]	-->	"HEURE DEPART"	:	26	XEQ 02
10,05	[R/S]	-->	"HEURE ARRET"	:	27	RDN
12,15	[R/S]	-->	"COMPTEUR"	:	28	-
9905,00	[R/S]	-->	"T=02:10/209KM"	:	29	CHS
		-->	"M ETAPE=96KM/H"	:	30	CLA
[R/S]	-->	"HEURE REDEMARRAGE"	:	31	ARCL X	
13,30	[R/S]	-->	"ARRET=01:15"	:	32	"├─KM "
[R/S]	-->	"HEURE ARRET"	:	33	ARCL 01	
18,25	[R/S]	-->	"COMPTEUR"	:	34	"├─L"
10600,00	[R/S]	-->	"T=04:55/695KM/H"	:	35	AVIEW
		-->	"M ETAPE=141KM/H"	:	36	XEQ 01
[D]	-->	"904KM/07:05"	:	37	"P="	
		-->	"MDY. RDUL.=128KM/H"	:	38	ARCL X
[E]	-->	"904KM/08:20"	:	39	"├─"	
		-->	"MDY. TOT.=108KM/H"	:	40	RCL 01
[a]	-->	"Q=(870,2)"	:	41	RCL 02	
44,0	[R/S]	-->	"KM?(9696,0)"	:	42	XEQ 01
10224,0	[R/S]	-->	"528,0KM 914,2L"	:	43	"├─T="
		-->	"P=8,3 T=8,9"	:	44	ARCL X
			:	45	PROMPT	
			:	46	GTO a	
		Denis CASTELAIN (P56)	:	47	*LBL 01	
			:	48	E2	
			:	49	/	
			:	50	/	
			:	51	RTN	
01*LBL "DB"			:	52	*LBL 02	
02 "FOB"			:	53	"FAIRE"	
03 GETR			:	54	XEQ 00	
04 SF 21			:	55	BEEP	
05 "CHOIX"			:	56	AVIEW	
06 PROMPT			:	57	RTN	
07*LBL a			:	58	*LBL b	
08 FIX 1			:	59	"PROCHAINE"	
09 "Q=("			:	60	XEQ 00	
10 ARCL 01			:	61	FIX 0	
11 "├─)"			:	62	ARCL 03	
12 CF 22			:	63	CF 22	
13 PROMPT			:	64	PROMPT	
14 FS?C 22			:	65	FS? 22	
15 ST+ 01			:	66	STO 03	
16 RCL 02			:	67	RTN	
17 "KM?("			:	68	*LBL 00	
18 ARCL 02			:	69	"├─ VIDANGE "	
19 "├─)"			:	70	RTN	
20 PROMPT			:	71	*LBL A	
21 FC? 22			:	72	CF 29	
22 GTO a			:	73	FIX 2	
23 STO 02			:			

74 0	:	124 PROMPT
75 STO 05	:	125 GTD B
76 STO 06	:	126*LBL 04
77 STO 07	:	127 "HEURE "
78 XEQ 03	:	128 RTN
79 XEQ 04	:	129*LBL 01
80 "└-DEPART"	:	130 RCL 04
81 XEQ 00	:	131 CHS
82 STO 04	:	132 XEQ 00
83 CLA	:	133 STO 04
84 ATIME	:	134 HMS+
85 PROMPT	:	135 RND
86*LBL B	:	136 RTN
87 XEQ 04	:	137*LBL 00
88 "└-ARRET"	:	138 CF 22
89 XEQ 01	:	139 PROMPT
90 HR	:	140 FC? 22
91 ST+ 05	:	141 TIME
92 HMS	:	142 RTN
93 RCL 00	:	143*LBL E
94 CHS	:	144 SF 00
95 XEQ 03	:	145*LBL D
96 +	:	146 RCL 07
97 "T="	:	147 FIX 0
98 X<>Y	:	148 CLA
99 ATIME	:	149 ARCL X
100 X<>Y	:	150 "└-KM/"
101 "└-/"	:	151 RCL 05
102 FIX 0	:	152 FS? 00
103 ARCL X	:	153 RCL 06
104 "└-KM"	:	154 FS? 00
105 AVIEW	:	155 +
106 ST+ 07	:	156 HMS
107 X<>Y	:	157 FIX 2
108 "M ETAPE="	:	158 ATIME
109 GTD 02	:	159 AVIEW
110*LBL 03	:	160 "MDY. "
111 "COMPTEUR"	:	161 FC? 00
112 PROMPT	:	162 "└-ROUL.="
113 STO 00	:	163 FS?C 00
114 RTN	:	164 "└-TOT.="
115*LBL C	:	165*LBL 02
116 XEQ 04	:	166 HR
117 "└-REDEMARRAGE"	:	167 /
118 XEQ 01	:	168 FIX 0
119 "ARRET="	:	169 ARCL X
120 FIX 2	:	170 FIX 2
121 ATIME	:	171 "└-KM/H"
122 HR	:	172 PROMPT
123 ST+ 06	:	173 GTD C

174#LBL e
 175 CLX
 176 SEEKPT
 177 7 E-3
 178 SAVERX
 179 "K7?"
 180 PROMPT
 181 "FOB"
 182 SEEKR
 183 WRTRX
 184 END

 Proposed Standard HP41 Machine Language Bar Code

The following two HP-41 machine language routines will enable you to make bar code from HP-41 machine language (M-code). These bar codes will be type 0, which unused by the normal HP-41 system.

MCP - M-Code Bar Code Pattern

MCP works like BCP in the HP-41 Plotter module, and creates a pattern in the ALPHA register to make bar codes from. You will need the Plotter Module to print the bar codes on a HP7470 or on a HPB2162.

The format of the bar codes is as follows:

Bit 0 -> 7 contains the checksum, which is the sum of all the bytes with wrap-around carry, not a running checksum. ---

Bit 8 -> 11 are all set to 0, the type number of M-Code bar code.

Bit 12 -> 15 contains the row number minus one, mod 16 (example: row 1 = 0).

Bit 16 -> 15+10*N contains the information of N M-code instructions, each occupying 10 bits.

Bit 16+10*N is the end indicator. If it is one, this row is the last row; if it is zero, you will be prompted for more rows. After this row there will be between 1 and 7 bits to fill up the last byte in the row.

The final bar code looks as follow:

```

||          Type Seq.   N M-code End      ||
|| Checksum =0 NO. instructions bitFiller ||

```

Instructions for use:

Input:

Y-reg.: 000000aaaabbbb Where aaaa-bbbb indicate the interval from which to make bar code.

X-reg.: ccc,dd Where ccc is the row number, and dd is the number of instructions in each row. If dd=00 then 11 will be used (maximum length).

Note that it is not possible to have a row number higher than 999, giving a maximum of 8k 10-bits words of bar code in one sequence.

Output:

Z-reg.: -ee,00 Where ee is the number of bytes in ALPHA register. Together with the minus sign, this is necessary to make bar codes on a HPB2162 (LBL 01, MCP, RCL Z, BCD, RDN, X#0?, GTD 01, RTN).

Y-reg.: 000000aaaabbbb Unchanged.
 X-reg.: ccc+1,dd or 0 Increment to the next row if not done, else X-reg. is cleared.

L-reg.: 000000ffffgggg ffff is the number of the first instruction in the row (if the instructions are numbered sequentially from 0000 and onward), and gggg is the number of the last instruction.

If used together with a HP7470 Plotter, the MCP function must be followed by the BC function in the Plotter Module.

Error messages: DATA ERROR if aaaa>bbbb,
aaaa+2000<bbbb or dd>11.
OUT OF RANGE if ccc>999.

MCSCAN - M-Code Bar Code SCAN

MCSCAN reads the bar codes made by MCP. You will of course need a wand. It works as follows:

Input:

X-reg.: 0000000000aaaa Where aaaa is the first address you want to read bar codes to.

Output:

X-reg.: 000000aaaabbbb Where aaaa is the start address and bbbb is the address of the last instruction (see SST) plus one.

L-reg.: 0000000000aaaa True LASTX.

If you want to read several programs immediately after each other, you can just repeat using MCSCAN without thinking of input expect for the first program.

During the execution of MCSCAN the following keys are active:

ON : Turns the HP41 off,
<- : Terminates the MCSCAN routine,
R/S : As above
SST : Asks for the next row (skips one row) if it is not possible to read the current row. The words skipped will not be cleared as it may be possible to read the missing rows later. the number of instructions skipped will be the number of instructions in the last read row. If the first row is skipped, the jump will be calculated after the first possible row has been read. The X-reg. will be updated after each read row and after SST (if row 1 not skipped).

Once you have entered the MCSCAN routine, some routines in the wand-ROM will be used, setting the time-out period to approx. 7 min. at normal speed. You will be prompted as usual: 'W: RDY nnn' to scan row nnn. If the MCSCAN routine meets an end-bit set to one, the routine will terminate.

Error Messages: NO WAND if wand no present,
W: CHKSUM ERR if error in checksum, try again.
W: TYPE ERR if you are trying to read non M-Code bar code.
W: SEQ ERR if you are trying to read in a wrong sequence (this message will not occur if you read exactly 16 rows wrong).

Both MCP and MCSCAN will be in the BOOT ROM EPROM set, expected to be completed soon.

MCP and MCSCAN are released for non-commercial use only, by the type zero bar code described under MCP is of course free to anyone.

(c) Copyright 1985 Steen Petersen, PPC-Denmark.

Proposition d'un standard langage machine 41 en CB

Les deux routines suivantes, vous permettent de tracer les code-barres de vos routines écrites en langage machine (M-code). Ces code-barres sont de type 0, inutilisé par le système normal de la HP41.

MCP - Modèle de code-barre pour M-Code.

MCP s'utilise comme BCP du module Plotter de la 41, et crée le modèle des code-barres dans le registre ALPHA. Ensuite, pour les imprimer sur

une table traçante HP7470 ou imprimante HP82162, vous devez utiliser le module Plotter.

Format des code-barres:

Bit 0 -> 7 contient la somme de contrôle. Somme de tous les octets carry cumulée, mais pas un checksum.

Bit 8 -> 11, tous à 0, donne le numéro de type des code-barres (M-Code).

Bit 12 -> 15 contient le numéro de la ligne moins 1, modulo 16 (exemple: ligne 1=0).

Bit 16 -> 15+10*N contient les renseignements de N instructions M-Code, chacune occupant 10 bits.

Bit 16+10*N, bit de fin. S'il est à 1, la rangée courante est la dernière; à zéro, on recherche la suivante. Après cette rangée, il y a entre 1 et 7 bits pour remplir le dernier octet de la ligne.

Finalement, les code-barres ressemblent à:

```
!!      Type Seq.   N M-Code End      !!
!! Checksum =0 NO. instructions bitFiller !!
```

Mode d'emploi:

Entrée:

Reg. Y: 000000aaaabbbb Où aaaa-bbbb indique l'intervalle dans lequel on veut faire les code-barres.

Reg. X: ccc,dd Où ccc est le numéro de la rangée et dd le nombre d'instructions par rangée. Si dd=00, alors, la valeur 11 est utilisée par défaut. (longueur maximale).

Notez qu'il est impossible d'avoir un numéro de rangée supérieur à 999, ce qui donne un maximum de 8k mots de 10-bits de code-barres dans un séquence.

Sortie:

Reg. Z: -ee,00 Ou ee est le nombre

d'octets en ALPHA, qui associé au signe moins, trace les code-barres sur imprimante HP82162A (LBL 01, MCP, RCL Z, BCO, RDN, X#0?, GTD 01, RTN).

Reg. Y: 000000aaaabbbb Inchangé,
Reg. X: ccc+1,dd ou 0 Incrémente pour la rangée suivante si le tracé n'est pas terminé. Sinon, le registre X est effacé.

Reg. L: 000000ffffgggg ffff est le numéro de la première instruction dans la rangée (si les instructions sont numérotées séquentiellement de 0000 à), gggg est le numéro de la dernière instruction.

Si vous utilisez la table traçante HP7470, la fonction MCP doit précéder la fonction BC du module Plotter.

Messages d'erreur: DATA ERROR si aaaa>bbbb, aaaa+2000<bbbb ou dd>11
OUT OF RANGE si ccc>999.

MCSCAN - Analyseur de code-barres pour M-Code.

MCSCAN lit les code-barres créés par MCP. Vous avez bien entendu besoin d'un lecteur optique. Elle fonctionne comme suit:

Entrée:

Reg. X: 0000000000aaaa Où aaaa est la première adresse dont vous voulez lire les code-barres.

Sortie:

Reg. X: 000000aaaabbbb Où aaaa est la première adresse et bbbb est l'adresse de la dernière instruction (voir SST) plus 1.

Reg. L: 0000000000aaaa LAST X réel.

Si vous désirez lire plusieurs programmes les uns après les autres, vous devez seulement réutiliser MCSCAN sans vous soucier de l'entrée du 1er

programme.

Durant l'exécution de MCSCAN, les touches suivantes restent actives:

DN : Eteint la HP41,
<- : Sort de la routine MCSCAN,
R/S : comme ci-dessus,
SST : Demande la rangée suivante (saute une rangée) s'il n'est pas possible de lire la rangée courante. Les mots sautés ne sont pas effacés s'il est possible de lire les rangées oubliées plus tard. Le nombre d'instructions sautées sera le nombre d'instructions dans la dernière rangée lue. Si la première rangée est sautée, le saut sera calculé après la première rangée qu'il aura été possible de lire. Le Registre X est remis à jour après chaque lecture et après SST (si la première rangée est lue).

Vous entrez d'abord la routine MCSCAN, quelques routines de la ROM du lecteur optique sont utilisées. Elles mettent la période d'attente à 7mn à vitesse normale. Vous devrez attendre l'habituel: 'W: RDY nnn' pour analyser la rangée nnn. Si la routine MCSCAN rencontre un bit de fin armé à 1, la routine s'arrête.

Messages d'erreur: NO WAND si le lecteur optique est inexistant.
W: CKSUM ERR s'il y a une erreur dans la somme de contrôle, recommencez.
W: TYPE ERR si vous essayez de lire des code-barres de type différent.
W: SEQ ERR si vous tentez de lire une mauvaise séquence (ce message ne s'affichera pas si vous lisez exactement 16 mauvaises lignes).

MCP et MCSCAN seront dans l'EPROM "BOOT ROM" bientôt disponible.

MCP et MCSCAN sont communiqués seulement pour une utilisation non commerciale. Les code-barres de type 0 décrit par MCP, sont bien entendu libres

d'édition pour tous.

(c) Copyright 1985 Steen Petersen, PPC-DENMARK

Traduction de P. Guez

Albertslund, 13. 5. 1985

DEAR FRIENDS,

Following are two routines which makes and reads m-code bar-codes.

In the last minute I discovered that ROM's in wand 1E and 1F are equal at a very few places, so I had to rewrite MCSCAN. It now only uses the error-messages in the wand, and should now work with all wand revisions.

Remember to change the port dependent jumps if you place the routines at different addresses. It is only necessary to enter the MCSCAN routine, because the m-code bar-codes for MCP are included.

Also included is a program MCBAR which makes m-code bar-codes on an HP-7470 plotter, with the help of a plotter module and the following m-codes routines:

CODE : The normal code routine,

RXR : Rotates the X register one digit to the
--- right,
4-D : Decodes the last four digits in the X
--- register and appends them to the alpha
register,
MCP : Makes a pattern for m-code bar-codes,

The program prompts for a routine name in ALPHA, a start address and an end address which both are

entered in the ALPHA register as hex digits. It then prompts for paper when necessary, and makes bar-codes as those made for the MCP routine.

Any questions may be directed to: Steen PETERSEN
PPC-Denmark
61. Landevej 19
DENMARK.

Yours sincerely,

Steen Petersen.

CHERS AMIS,

Voici deux routines permettant de créer et lire vos routines microcode en codes barres.

En dernière minute, je découvre que les lecteurs optique de version 1E ou 1F, sont très peu équivalents. Aussi, j'ai du réécrire MCSCAN. Il utilise les messages d'erreur du lecteur optique et peu fonctionner avec n'importe quelle version de lecteur optique.

N'oubliez pas de changer le "port dependant jump" si vous placez les routines à une adresse différente. Il est seulement nécessaire d'entrer la routine MCSAN. MCP est donné en codes-barres.

Vous trouverez d'autre part le programme MCBAR qui trace les codes-barres sur la table traçante HP7470, votre 41 munie du module PLOTTER et vous aurez d'autre part besoin des routines microcode suivantes:

CODE : Code en BCD (ou NNN) un HEX en ALPHA,

RXR : Rotation du registre X de 1 digit vers la droite,

4-D : Décode les 4 derniers digits de X et les ajoute (APPEND) au registre ALPHA,

MCP : Donne vos routines en CB.

Le programme s'arrête pour vous demander le nom

de la routine en ALPHA, puis, les adresses de début et de fin. Il s'arrête d'autre part pour remplacer le papier quand c'est nécessaire.

Pour toutes questions, écrire à Steen PETERSEN.

Philippe.



EN QUOI
PARLE-T-IL
CE PHILIPPE?
QUEL
CHARACTÈRE!!!

Delaware
81

9600 090 *P*	9633 158 M=C ALL	9666 07C RCR 4	9699 23A C=C+1 M
9601 003 *C*	9634 008 READ 2(Y)	9667 10E A=C ALL	969A 0FA C<>B M
9602 00D *M*	9635 10E A=C ALL	9668 008 READ 2(Y)	969B 1E6 C=C+C S&X
9603 0F8 READ 3(X)	9636 07C RCR 4	9669 07C RCR 4	969C 1E6 C=C+C S&X
9604 004 CLRF 5	9637 302 ?A<C ER	966A 14A A=A+C R<	969D 1EE C=C+C ALL
9605 200 SETDEC	9638 037 JC 963E +06	966B 3EE LSHFA ALL	969E 1EE C=C+C ALL
9606 0ED	9639 362 ?A+C ER	966C 3EE LSHFA ALL	969F 23C RCR 2
9607 064 ?NCXQ 193B	963A 03B JMC 9641 +07	966D 3EE LSHFA ALL	96A0 070 N=C ALL
9608 226 C=C+1 S&X	963B 102 A=A-1 ER	966E 066 A<>B S&X	96A1 36D
9609 226 C=C+1 S&X	963C 362 ?A+C ER	966F 106 A=A-1 S&X	96A2 08C PORT DEP:
960A 260 SETHEX	963D 01B JMC 9640 +03	9670 345	96A3 344 XQ 9744
960B 38D	963E 005	9671 040 ?NCXQ 10D1	96A4 05B JMC 96AF +0B
960C 008 ?NCXQ 02E3	963F 002 ?NCGO 282D	9672 00E A<>C ALL	96A5 2FC RCR 13
960D 006 A<>C S&X	9640 162 A=A+1 ER	9673 0F0 C<>N ALL	96A6 1EE C=C+C ALL
960E 130 LDI S&X	9641 1CA A=A-C R<	9674 33C RCR 1	96A7 38C ?FSET 0
960F 00B	9642 3E7 JC 963E -04	9675 05A C=0 M	96A8 013 JMC 96AA +02
9610 006 A<>C S&X	9643 384 CLRF 0	9676 046 C=0 S&X	96A9 22E C=C+1 ALL
9611 2E6 ?C#0 S&X	9644 198 C=M ALL	9677 27E C=C-1 MS	96AA 1EE C=C+C ALL
9612 01F JC 9615 +03	9645 24A C=A-C R<	9678 33C RCR 1	96AB 1EE C=C+C ALL
9613 006 A<>C S&X	9646 01F JC 9649 +03	9679 130 LDI S&X	96AC 1EE C=C+C ALL
9614 106 A=C S&X	9647 2EA ?C#0 R<	967A 0FF	96AD 003 JMC 96C1 +14
9615 39C R= 0	9648 06F JC 9655 +0D	967B 37C RCR 12	96AE 09B JMC 96C1 +13
9616 058 G=C ER;+	9649 28A C=0-C R<	967C 168 WRIT 5(M)	96AF 0FC RCR 10
9617 1C6 A=A-C S&X	964A 00A B=A R<	967D 106 A=C S&X	96B0 0FA C<>B M
9618 01B JMC 961B +03	964B 10A A=C R<	967E 0B0 C=M ALL	96B1 330 FETCH S&X
9619 0B5	964C 04A C=0 R<	967F 0EE C<>B ALL	96B2 23A C=C+1 M
961A 002 ?NCGO 282D	964D 39C R= 0	9680 0FA C<>B M	96B3 0FA C<>B M
961B 0F8 READ 3(X)	964E 098 C=G ER;+	9681 330 FETCH S&X	96B4 1E6 C=C+C S&X
961C 38D	964F 01C R= 3	9682 23A C=C+1 M	96B5 1E6 C=C+C S&X
961D 008 ?NCXQ 02E3	9650 00A A<>C R<	9683 0FA C<>B M	96B6 23C RCR 2
961E 070 N=C ALL	9651 1CA A=A-C R<	9684 1EE C=C+C ALL	96B7 070 N=C ALL
961F 266 C=C-1 S&X	9652 06A A<>B R<	9685 1EE C=C+C ALL	96B8 36D
9620 3CF JC 9619 -07	9653 04A C=0 R<	9686 33C RCR 1	96B9 08C PORT DEP:
9621 02E B=0 ALL	9654 388 SETF 0	9687 070 N=C ALL	96BA 344 XQ 9744
9622 0E6 C<>B S&X	9655 07C RCR 4	9688 36D	96BB 053 JMC 96C5 +0A
9623 098 C=G ER;+	9656 24A C=A-C R<	9689 08C PORT DEP:	96BC 37C RCR 12
9624 00E A=0 ALL	9657 02B JMC 965C +05	968A 344 XQ 9744	96BD 38C ?FSET 0
9625 266 C=C-1 S&X	9658 345	968B 04B JMC 9694 +09	96BE 01B JMC 96C1 +03
9626 01F JC 9629 +03	9659 040 ?NCXQ 10D1	968C 2FC RCR 13	96BF 226 C=C+1 S&X
9627 12E A=A+B ALL	965A 0E8 WRIT 3(X)	968D 38C ?FSET 0	96C0 226 C=C+1 S&X
9628 3EB JMC 9625 -03	965B 3E0 RTN	968E 01B JMC 9691 +03	96C1 070 N=C ALL
9629 00E A<>C ALL	965C 198 C=M ALL	968F 226 C=C+1 S&X	96C2 365
962A 10E A=C ALL	965D 38C ?FSET 0	9690 226 C=C+1 S&X	96C3 08C PORT DEP:
962B 0FC RCR 10	965E 03B JMC 9665 +07	9691 05E C=0 MS	96C4 342 GO 9742
962C 098 C=G ER;+	965F 07C RCR 4	9692 2FC RCR 13	96C5 0FC RCR 10
962D 01C R= 3	9660 10A A=C R<	9693 173 JMC 96C1 +2E	96C6 1EE C=C+C ALL
962E 0EA C<>B R<	9661 12A A=A+B R<	9694 1BC RCR 11	96C7 1EE C=C+C ALL
962F 0CA C=B R<	9662 10A A=A-1 R<	9695 1EE C=C+C ALL	96C8 0FA C<>B M
9630 14A A=A+C R<	9663 0FC RCR 10	9696 1EE C=C+C ALL	96C9 330 FETCH S&X
9631 10A A=A-1 R<	9664 00A A<>C R<	9697 0FA C<>B M	96CA 23A C=C+1 M
9632 00A A<>C R<	9665 12B WRIT 4(L)	9698 330 FETCH S&X	96CB 0FA C<>B M

96CC 1E6 C=C+C S&X	96FF 11A A=C M	9732 260 SETHEX	9765 1F8 READ 7(O)
96CD 1E6 C=C+C S&X	9700 1B8 READ 6(N)	9733 0E8 WRIT 3(X)	9766 37C RCR 12
96CE 1EE C=C+C ALL	9701 2EE ?C*0 ALL	9734 3E0 RTH	9767 0D8 C<>G OR;+
96CF 1EE C=C+C ALL	9702 038 JNC 9709 +07	9735 0B0 C=M ALL	9768 1E8 WRIT 7(O)
96D0 03C RCR 3	9703 070 M=C ALL	9736 2FE ?C*0 MS	9769 238 READ 8(P)
96D1 070 M=C ALL	9704 36D	9737 02F JC 973C +05	976A 37C RCR 12
96D2 058 G=C OR;+	9705 08C PORT DEP:	9738 17E A=A+1 MS	976B 098 C=G OR;+
96D3 31C R= 1	9706 335 XQ 9735	9739 1BA A=A-1 M	976C 228 WRIT 8(P)
96D4 14A A=A+C R<	9707 1A8 WRIT 6(N)	973A 37C RCR 12	976D 3E0 RTH
96D5 013 JNC 96D7 +02	9708 058 JNC 9713 +0B	973B 3B8 JNC 9736 -05	
96D6 16A A=A+1 R<	9709 01C R= 3	973C 37C RCR 12	
96D7 39C R= 0	970A 1D0 LDR 7	973D 0AA A<>C R<	
96D8 36D	970B 31C R= 1	973E 23C RCR 2	
96D9 08C PORT DEP:	970C 11A A=C M	973F 1BE A=A-1 MS	
96DA 35D XQ 975D	970D 178 READ 5(M)	9740 3F3 JNC 973E -02	
96DB 0B0 C=M ALL	970E 070 M=C ALL	9741 3E0 RTH	
96DC 37C RCR 12	970F 36D	9742 308 SETF 1	
96DD 070 M=C ALL	9710 08C PORT DEP:	9743 013 JNC 9745 +02	
96DE 36D	9711 335 XQ 9735	9744 304 CLRF 1	
96DF 08C PORT DEP:	9712 168 WRIT 5(M)	9745 0B0 C=M ALL	
96E0 344 XQ 9744	9713 04E C=0 ALL	9746 31C R= 1	
96E1 013 JNC 96E3 +02	9714 138 LDI S&X	9747 14A A=A+C R<	
96E2 023 JNC 96E6 +04	9715 00A	9748 013 JNC 974A +02	
96E3 365	9716 38E RSHFA ALL	9749 16A A=A+1 R<	
96E4 08C PORT DEP:	9717 38E RSHFA ALL	974A 39C R= 0	
96E5 280 GO 9680	9718 38E RSHFA ALL	974B 058 G=C OR;+	
96E6 046 C=0 S&X	9719 246 C=A-C S&X	974C 36D	
96E7 38C ?FSET 0	971A 037 JC 9720 +06	974D 08C PORT DEP:	
96E8 2C8 JNC 96C1 -27	971B 31C R= 1	974E 35D XQ 975D	
96E9 130 LDI S&X	971C 050 LDR 1	974F 30C ?FSET 1	
96EA 080	971D 03C RCR 3	9750 023 JNC 9754 +04	
96EB 2B3 JNC 96C1 -2A	971E 226 C=C+1 S&X	9751 365	
96EC 01E A=0 MS	971F 01B JNC 9722 +03	9752 08C PORT DEP:	
96ED 05A C=0 M	9720 0A6 A<>C S&X	9753 2EC GO 96EC	
96EE 05C R= 4	9721 23C RCR 2	9754 0E6 C<>B S&X	
96EF 050 LDR 1	9722 2BC R= 13	9755 266 C=C-1 S&X	
96F0 150 LDR 5	9723 250 LDR 9	9756 023 JNC 975A +04	
96F1 31C R= 1	9724 068 WRIT 1(Z)	9757 1B0 POP ADR	
96F2 11A A=C M	9725 04E C=0 ALL	9758 23A C=C+1 M	
96F3 1F8 READ 7(O)	9726 38C ?FSET 0	9759 170 PUSH ADR	
96F4 2EE ?C*0 ALL	9727 01B JNC 972A +03	975A 0E6 C<>B S&X	
96F5 038 JNC 96FC +07	9728 0E8 WRIT 3(X)	975B 0B0 C=M ALL	
96F6 070 M=C ALL	9729 3E0 RTH	975C 3E0 RTH	
96F7 36D	972A 35C R= 12	975D 178 READ 5(M)	
96F8 08C PORT DEP:	972B 050 LDR 1	975E 37C RCR 12	
96F9 335 XQ 9735	972C 10E A=C ALL	975F 0D8 C<>G OR;+	
96FA 1E8 WRIT 7(O)	972D 0F8 READ 3(X)	9760 168 WRIT 5(M)	
96FB 0C3 JNC 9713 +18	972E 11E A=C MS	9761 1B8 READ 6(N)	
96FC 01C R= 3	972F 2A0 SETDEC	9762 37C RCR 12	
96FD 390 LDR E	9730 01D	9763 0D8 C<>G OR;+	
96FE 31C R= 1	9731 060 ?NCXQ 1087	9764 1A8 WRIT 6(N)	

MCP: 366 INSTRUCTIONS

9D50 08E *M*
9D51 001 *A*
9D52 003 *C*
9D53 013 *S*
9D54 003 *C*
9D55 00D *M*
9D56 244 CLRF 9
9D57 130 LDI S&X
9D58 01B
9D59 106 A=C S&X
9D5A 04E C=0 ALL
9D5B 15C R= 6
9D5C 110 LDR 4
9D5D 15C R= 6
9D5E 222 C=C+1 OR
9D5F 03F JC 9D66 +07
9D60 330 FETCH S&X
9D61 366 ?A#C S&X
9D62 3E7 JC 9D5E -04
9D63 046 C=0 S&X
9D64 03C RCR 3
9D65 0A8 JNC 9D7A +15
9D66 3A1
9D67 088 ?NCXQ 22E8
9D68 3C1
9D69 0B0 ?NCXQ 2CF0
9D6A 3BD
9D6B 01C ?NCXQ 07EF
9D6C 00E *H*
9D6D 00F *O*
9D6E 020 " "
9D6F 017 *M*
9D70 001 *A*
9D71 00E *H*
9D72 204 *D*
9D73 3DD
9D74 0AC ?NCXQ 2BF7

9D75 108 SETF 8	9D88 05C R= 4	9DDB 0F7 JC 9DF9 +1E	9E8E 017 JC 9E10 +02
9D76 201	9D89 250 LDR 9	9DDC 38C ?FSET 0	9E8F 33C RCR 1
9D77 070 ?NCKQ 1C80	9D8A 05C R= 4	9DDD 047 JC 9DE5 +08	9E10 104 CLR 8
9D78 3ED	9D8B 22A C=C+1 R<	9DDE 38C ?FSET 1	9E11 3D8 C<>ST XP
9D79 08A ?MCGO 22FB	9D8C 023 JNC 9DB0 +04	9DDF 017 JC 9DE1 +02	9E12 08C ?FSET 5
9D7A 130 LDI S&X	9D8D 3AD	9DE0 18E A=A-1 MS	9E13 013 JNC 9E15 +02
9D7B 001	9D8E 08C PORT DEP:	9DE1 37C RCR 12	9E14 108 SETF 8
9D7C 268 WRIT 9(C)	9D8F 292 GO 9E92	9DE2 0EE C<>B ALL	9E15 3D8 C<>ST XP
9D7D 0F8 READ 3(X)	9D90 3CC ?KEY	9DE3 388 SETF 0	9E16 10C ?FSET 8
9D7E 128 WRIT 4(L)	9D91 3E7 JC 9DAD -04	9DE4 368 JNC 9DD1 -13	9E17 013 JNC 9E19 +02
9D7F 01C R= 3	9D92 22C ?FI 2	9DE5 1EE C=C+C ALL	9E18 288 SETF 7
9D80 10A A=C R<	9D93 3C3 JNC 9DAB -08	9DE6 1EE C=C+C ALL	9E19 0CE C=B ALL
9D81 07C RCR 4	9D94 130 LDI S&X	9DE7 30C ?FSET 1	9E1A 20C ?FSET 2
9D82 0AA A<>C R<	9D95 010	9DE8 017 JC 9DEA +02	9E1B 02F JC 9E20 +05
9D83 0FC RCR 10	9D96 270 RAM SLCT	9DE9 33C RCR 1	9E1C 37C RCR 12
9D84 0E8 WRIT 3(X)	9D97 130 LDI S&X	9DEA 37C RCR 12	9E1D 38C ?FSET 1
9D85 278 READ 9(C)	9D98 0FE	9DEB 3C6 RSHFC S&X	9E1E 013 JNC 9E20 +02
9D86 070 M=C ALL	9D99 3F0 PRPH SLCT	9DEC 1E6 C=C+C S&X	9E1F 18C RCR 11
9D87 160 ?LOWBAT	9D9A 038 READ 0(T)	9DED 1E6 C=C+C S&X	9E20 00C ?FSET 3
9D88 043 JNC 9D90 +08	9D9B 37C RCR 12	9DEE 38C ?FSET 1	9E21 028 JNC 9E26 +05
9D89 388 READ 14(d)	9D9C 10E A=C ALL	9DEF 02B JNC 9DF4 +05	9E22 04C ?FSET 4
9D8A 358 ST=C XP	9D9D 22C ?FI 2	9DF0 03C RCR 3	9E23 017 JC 9E25 +02
9D8B 14C ?FSET 6	9D9E 02F JC 9DC3 +05	9DF1 0DA C=B M	9E24 158 M=C ALL
9D8C 027 JC 9D90 +04	9D9F 01E A=0 MS	9DF2 37C RCR 12	9E25 0F0 C<>M ALL
9D8D 398 C=ST XP	9DA0 3AD	9DF3 208 SETF 2	9E26 0EE C<>B ALL
9D8E 16D	9DA1 08C PORT DEP:	9DF4 2FC RCR 13	9E27 149
9D8F 01C ?NCKQ 075B	9DA2 227 GO 9E27	9DF5 0EE C<>B ALL	9E28 024 ?NCKQ 0952
9D90 265	9DA3 038 READ 0(T)	9DF6 388 SETF 1	9E29 35E ?A=0 MS
9D91 020 ?NCKQ 0899	9DA4 31C R= 1	9DF7 384 CLR 0	9E2A 05F JC 9E35 +08
9D92 3C1	9DA5 10A A=C R<	9DF8 2CB JNC 9DD1 -27	9E2B 385
9D93 080 ?NCKQ 2CF0	9DA6 3EE LSHFA ALL	9DF9 00C ?FSET 3	9E2C 08C PORT DEP:
9D94 38D	9DA7 3EE LSHFA ALL	9DFA 023 JNC 9DFE +04	9E2D 2C5 XQ 9EC5
9D95 01C ?NCKQ 07EF	9DA8 10A A=C R<	9DFB 158 M=C ALL	9E2E 014 ?R= 3
9D96 097 "M"	9DA9 22C ?FI 2	9DFC 048 SETF 4	9E2F 385
9D97 020 " "	9DAA 3AB JNC 9DBF -08	9DFD 288 JNC 9DCE -2F	9E30 08C PORT DEP:
9D98 012 "R"	9DAB 31C R= 1	9DFE 070 M=C ALL	9E31 2D5 XQ 9ED5
9D99 004 "D"	9DAC 3C4 ST=0	9DFF 088 SETF 3	9E32 3AD
9D9A 019 "Y"	9DAD 01E A=0 MS	9E00 273 JNC 9DCE -32	9E33 08C PORT DEP:
9D9B 220 " "	9DAE 384 CLR 0	9E01 20C ?FSET 2	9E34 185 GO 9D85
9D9C 00E A=0 ALL	9DAF 304 CLR 1	9E02 037 JC 9E88 +06	9E35 31C R= 1
9D9D 080 C=M ALL	9DB0 204 CLR 2	9E03 08C ?FSET 5	9E36 0AA A<>C R<
9D9E 106 A=C S&X	9DB1 22C ?FI 2	9E04 027 JC 9E88 +04	9E37 10A A=C R<
9D9F 3A1	9DB2 17B JNC 9E81 +2F	9E05 088 SETF 5	9E38 0AE A<>C ALL
9DA0 014 ?NCKQ 05E8	9DB3 038 READ 0(T)	9E06 04A C=0 R<	9E39 07C RCR 4
9DA1 3D9	9DB4 14A A=A+C R<	9E07 288 JNC 9DD8 -2F	9E3A 0AE A<>C ALL
9DA2 01C ?NCKQ 07F6	9DB5 013 JNC 9DD7 +02	9E08 0CE C=B ALL	9E3B 36A ?A=C R<
9DA3 3DD	9DB6 16A A=A+1 R<	9E09 33C RCR 1	9E3C 033 JNC 9E42 +06
9DA4 0AC ?NCKQ 2BF7	9DB7 17E A=A+1 MS	9E0A 20C ?FSET 2	9E3D 385
9DA5 149	9DB8 0EA C<>B R<	9E0B 02F JC 9E10 +05	9E3E 08C PORT DEP:
9DA6 024 ?NCKQ 0952	9DB9 0CE C=B ALL	9E0C 03C RCR 3	9E3F 2C5 XQ 9EC5
9DA7 04E C=0 ALL	9DBA 20C ?FSET 2	9E0D 38C ?FSET 1	9E40 00A A=0 R<

9E41 373 JNC 9E2F -12	9E74 18C RCR 11	9E97 381	9EDA 037 JC 9EE0 +06
9E42 35E ?A=0 MS	9E75 0EE C(>)B ALL	9E98 042 ?NCGO 10E0	9EDB 308 C(>)ST XP
9E43 347 JC 9E2B -18	9E76 27C RCR 9	9E99 130 LDI S&X	9EDC 10C ?FSET 8
9E44 3EE LSHFA ALL	9E77 0EE C(>)B ALL	9E9A 0C2	9EDD 3A0 ?NC RTH
9E45 278 READ 9(Q)	9E78 20C R= 13	9E9B 366 ?A=C S&X	9EDE 265
9E46 33C RCR 1	9E79 0D0 LDR 3	9E9C 023 JNC 9EB0 +04	9EDF 022 ?NCGO 0899
9E47 27E C=C-1 MS	9E7A 18E A=A-1 MS	9E9D 261	9EE0 106 A=C S&X
9E48 37E ?A=C MS	9E7B 077 JC 9EB9 +0E	9E9E 000 ?NCKQ 0098	9EE1 3C4 ST=0
9E49 033 JNC 9E4F +06	9E7C 0C6 C=B S&X	9E9F 303 JNC 9E8F -20	9EE2 208 ST(>T
9E4A 385	9E7D 040 WROM	9E90 0F8 READ 3(X)	9EE3 130 LDI S&X
9E4B 08C PORT DEP:	9E7E 23A C=C+1 M	9E91 18C RCR 11	9EE4 0FF
9E4C 2C5 XQ 9EC5	9E7F 0EE C(>)B ALL	9E92 11A A=C M	9EE5 358 ST=C XP
9E4D 01D	9E80 18C RCR 11	9E93 278 READ 9(Q)	9EE6 10C ?FSET 8
9E4E 308 ?CGO C207	9E81 0EE C(>)B ALL	9E94 266 C=C-1 S&X	9EE7 057 JC 9EF1 +0A
9E4F 23E C=C+1 MS	9E82 27E C=C-1 MS	9E95 2E6 ?C=0 S&X	9EE8 130 LDI S&X
9E50 2FC RCR 13	9E83 388 JNC 9E7A -09	9E96 017 JC 9EB8 +02	9EE9 065
9E51 226 C=C+1 S&X	9E84 0EE C(>)B ALL	9E97 248 SETF 9	9EEA 208 ST(>T
9E52 268 WRIT 9(Q)	9E85 0F0 C(>)M ALL	9E98 226 C=C+1 S&X	9EEB 000 NOP
9E53 104 CLRF 8	9E86 108 C(>)M ALL	9E99 226 C=C+1 S&X	9EEC 266 C=C-1 S&X
9E54 385	9E87 0EE C(>)B ALL	9E9A 268 WRIT 9(Q)	9EED 3EB JNC 9EEA -03
9E55 08C PORT DEP:	9E88 368 JNC 9E75 -13	9E9B 24C ?FSET 9	9EEE 046 A(>)C S&X
9E56 206 XQ 9ED6	9E89 05E C=0 MS	9E9C 38F JC 9EAD -0F	9EEF 358 ST=C XP
9E57 0AE A(>)C ALL	9E8A 046 C=0 S&X	9E9D 046 C=0 S&X	9EF0 3E0 RTH
9E58 18C RCR 11	9E8B 03C RCR 3	9E9E 2FC RCR 13	9EF1 130 LDI S&X
9E59 0AE A(>)C ALL	9E8C 0E8 WRIT 3(X)	9E9F 10E A=C ALL	9EF2 191
9E5A 278 READ 9(Q)	9E8D 28C ?FSET 7	9EA0 0F8 READ 3(X)	9EF3 208 ST(>T
9E5B 08E A(>)C MS	9E8E 0CF JC 9E97 +19	9EC1 01C R= 3	9EF4 000 NOP
9E5C 11E A=C MS	9E8F 3AD	9EC2 20A C=C+A R<	9EF5 000 NOP
9E5D 268 WRIT 9(Q)	9E90 08C PORT DEP:	9EC3 0E8 WRIT 3(X)	9EF6 000 NOP
9E5E 24C ?FSET 9	9E91 185 GO 9D85	9EC4 34B JNC 9EAD -17	9EF7 000 NOP
9E5F 0A3 JNC 9E73 +14	9E92 3CC ?KEY	9EC5 180 POP ADR	9EF8 266 C=C-1 S&X
9E60 244 CLRF 9	9E93 321	9EC6 104 CLRF 8	9EF9 303 JNC 9EF3 -06
9E61 01C R= 3	9E94 046 ?NCGO 11C8	9EC7 330 FETCH S&X	9EFA 3A3 JNC 9EEE -0C
9E62 05A C=0 M	9E95 04E C=0 ALL	9EC8 23A C=C+1 M	
9E63 18C RCR 11	9E96 220 C=KEY KY	9EC9 170 PUSH ADR	
9E64 11A A=C M	9E97 03C RCR 3	9ECA 10E A=C ALL	
9E65 03C RCR 3	9E98 106 A=C S&X	9ECB 278 READ 9(Q)	
9E66 046 C=0 S&X	9E99 39C R= 0	9ECC 18C RCR 11	
9E67 2FC RCR 13	9E9A 1E2 C=C+C 0R	9ECD 09C R= 5	
9E68 106 A=C S&X	9E9B 321	9ECE 3EA LSHFA R<	
9E69 0F8 READ 3(X)	9E9C 047 ?CGO 11C8	9ECF 3EA LSHFA R<	
9E6A 18A A=A-1 M	9E9D 130 LDI S&X	9ED0 3EA LSHFA R<	
9E6B 18A A=A-1 M	9E9E 0C3	9ED1 00A A(>)C R<	
9E6C 18A A=A-1 M	9E9F 366 ?A=C S&X	9ED2 210 LDR 8	
9E6D 02F JC 9E72 +05	9EA0 02B JNC 9EA5 +05	9ED3 105	
9E6E 206 C=C+A S&X	9EA1 130 LDI S&X	9ED4 072 ?NCGO 1C75	
9E6F 3EB JNC 9E6C -03	9EA2 087	9ED5 108 SETF 8	
9E70 222 C=C+1 0R	9EA3 366 ?A=C S&X	9ED6 388 READ 14(d)	
9E71 30B JNC 9E6C -05	9EA4 02F JC 9EA9 +05	9ED7 17C RCR 6	
9E72 0E8 WRIT 3(X)	9EA5 261	9ED8 308 C(>)ST XP	
9E73 0F8 READ 3(X)	9EA6 000 ?NCKQ 0098	9ED9 08C ?FSET 5	

MCSCAN: 427 INSTRUCTIONS

ROUTINE: MCP
ADR: 9600-976D

ROW 1 (0000-000A)



ROW 2 (000B-0015)



ROW 3 (0016-0020)



ROW 4 (0021-002B)



ROW 5 (002C-0036)



ROW 6 (0037-0041)



ROW 7 (0042-004C)



ROW 8 (004D-0057)



ROW 9 (0058-0062)



ROW 10 (0063-006D)



ROW 11 (006E-0078)



ROW 12 (0079-0083)



ROW 13 (0084-008E)



ROW 14 (008F-0099)



ROW 15 (009A-00A4)



ROW 16 (00A5-00AF)



ROUTINE: MCP
ADR: 9600-976D

ROW 17 (00B0-00BA)



ROW 18 (00BB-00C5)



ROW 19 (00C6-00D0)



ROW 20 (00D1-00DB)



ROW 21 (00DC-00E6)



ROW 22 (00E7-00F1)



ROW 23 (00F2-00FC)



ROW 24 (00FD-0107)



ROW 25 (0108-0112)



ROW 26 (0113-011D)



ROW 27 (011E-0128)



ROW 28 (0129-0133)



ROW 29 (0134-013E)



ROW 30 (013F-0149)



ROW 31 (014A-0154)



ROW 32 (0155-015F)



ROUTINE: MCP
ADR: 9600-976D

ROW 33 (0160-016A)



ROW 34 (016B-016D)



01♦LBL "MCB	37 BC	77 MOVE
AR"	38 X<> 05	78 4
02 PINIT	39 "ROW "	79 CSIZE
03 CF 29	40 ARCL X	80 CLA
04 FIX 0	41 LASTX	81 LABEL
05 .	42 ENTER↑	82 "ROUTINE
06 ACOS	43 RXR	83 ARCL 00
07 LDIR	44 RXR	84 ARCL 01
08♦LBL 00	45 RXR	85 LABEL
09 "ROUTINE	46 RXR	86 "ADR: "
NAME: "	47 "F (<"	87 ARCL 02
10 AON	48 4-D	88 "F--"
11 BEEP	49 "F--"	89 ARCL 03
12 STOP	50 RDN	90 LABEL
13 ASTO 00	51 4-D	91 5
14 ASHF	52 "F)"	92 ST+ 04
15 ASTO 01	53 3	93♦LBL 05
16 "START A	54 RCL 04	94 3
DR: "	55 MOVE	95 CSIZE
17 STOP	56 LABEL	96♦LBL 02
18 ASTO 02	57 XEQ 02	97 RCL 04
19 "END ADR	58 RCL 05	98 7.6
:"	59 X=0?	99 +
20 STOP	60 GTO 00	100 STO 04
21 ASTO 03	61 GTO 01	101 131
22 AOFF	62♦LBL 04	102 X<=Y?
23 XEQ 04	63 0	103 GTO 04
24 1	64 ENTER↑	104 RDN
25 STO 05	65 PEN	105 3
26♦LBL 01	66 MOVE	106 X<>Y
27 .	67 "PAPER"	107 MOVE
28 ENTER↑	68 BEEP	108 END
29 .7	69 PROMPT	
30 IMOVE	70 RDN	
31 CLA	71 1	
32 ARCL 02	72 PEN	
33 ARCL 03	73 3	
34 CODE	74 ENTER↑	LBL "MCBAR
35 RCL 05	75 0	END
36 MCP	76 STO 04	232 BYTES

HP-71B

HP71 ASSEMBLEUR

Olivier ARBEY	DEBUTER EN ASSEMBLEUR
Michel MARTINET	CHOC EN RETOUR
Jean-Jacques MOREAU	END-UP !
Jean-Jacques MOREAU	DES REGISTRES D'ENTREE/SORTIE
Jean-Pierre BONDU	PILE A 16 NIVEAUX
Michel MARTINET	LES VARIABLES ET LA MATH STACK
Laurent ISTRIA	A PROPOS DE COMBINAISONS ET D'ARRANGEMENTS
Michel MARTINET	HEURES-MINUTES-SECONDES
Jean-Pierre BONDU	ENCORE UNE ERREUR
Michel MARTINET	PROGRAMMATION SYNTHETIQUE
Michel MARTINET	LA PILE DE COMMANDE EN ASSEMBLEUR
Jean-Pierre BONDU	ADRESSE D'UNE COLONNE GRAPHIQUE (FORTH)

HP71 BASIC

René OUVRAY	ANOMALIE DU CALENDRIER (VERSION HP71:1BBBB)
Serge VAUDENAY	REMORDS (CHOC EN RETOUR)
Jean-Jacques DHENIN	POUR LA RUBRIQUE DEBUTANTS
Xavier BILLE	MASTER-MIND AMELIORE
Pierre DAVID	D.A.O. (SUITE...)
Serge VAUDENAY	HANTISE SUR SATURNE

Débuter en Assembleur : les primitives FORTH

Vous désirez vous lancer dans l'Assembleur du HP-71 mais vous reculez devant l'ampleur de la tâche ? Je vous comprends ! Aussi, cet article a-t-il pour vocation d'essayer de vous guider dans votre démarche de débutant...

Tout d'abord, il faut savoir qu'en l'état actuel de nos connaissances, l'écriture de routines en Assembleur sur 71 ne peut se faire que sous 3 formes : LEX, primitive FORTH ou BIN. Vous avez néanmoins dû vous rendre compte (cf. les précédents JPC) que le LEX est de loin la forme actuellement la plus usitée, ce qui est somme toute logique, vu la puissance de cette formule d'extension de langage. Cependant, et d'après la Loi de Murphy, c'est aussi la routine la plus délicate à mettre au point, en grande partie à cause du problème posé par l'écriture des routines de Parse et Decompile; celles-ci ont pour fonction de "voir" vos instructions Basic telles que Saturn les comprend et les affiche, plutôt que sous la forme d'une série de chiffres. Il a fallu attendre la livraison des IDS III pour que cette hypothèque soit levée, mais ceci est une autre histoire... Un LEX parfaitement au point dans sa partie Assembleur sera totalement inutilisable si vous ne pouvez pas écrire ces fichues routines, un peu comme si, n'ayant que 2 roues pour votre voiture, vous soyez contraint de construire une patinette pour pouvoir vous déplacer... Heureusement, le Seigneur, dans Son infinie bonté, nous a donné le moyen de faire de l'Assembleur sans être d'abord devenus d'infâmes scribes païens en Parse et Decompile. Ce moyen, ce sont les Primitives FORTH.

Pourquoi des Primitives et non des BIN, me direz-vous ? Bonne question, dont la réponse réside dans le fait que tout programme, quel qu'il soit, a souvent besoin d'effectuer des entrées/sorties (saisie de données, restitution des résultats d'un calcul, etc...); pour cette tâche, il s'avère que la pile de données du FORTH offre une souplesse maximale, contrairement au BIN qui nécessite une déclaration de paramètres (cf. IDS I.11). Bien entendu, et c'est là le principal avantage, pas de routines d'analyse à

rédigier. Seules contraintes, il y a 3 (petits) pointeurs à maintenir: D0, D1 et B[A]. Mais quelle puissance en contrepartie ! Un simple $D1=D1+5$ ou $D1=D1-5$ vide ou rempli d'un nombre la pile de données; la même opération dans un LEX oblige à rechercher les routines ad-hoc, à sauvegarder tout un tas de registres, à jongler avec la pile math... Le chaînage de vos routines est grandement facilité: il suffit d'appeler vos mots les uns après les autres, tandis qu'un LEX ne peut en appeler un autre, tout devant être dans le LEX principal; en FORTH, on peut bricoler avec les CFA et autres PFA pour qu'une primitive puisse appeler des portions d'une autre...

Une primitive FORTH résoud aussi le cas très grave connu sous le nom de Frustration du Programmeur. Combien d'idées géniales ont été perdues par peur de l'Inconnu au moment de passer à l'Acte ? Ici, pas de crainte à avoir car vous n'êtes pas devant un LEX: vous écrivez et mettez au point votre routine sous forme d'une primitive FORTH, quitte à l'adapter dans un LEX ultérieurement. Pour terminer, deux règles pratiques pour le jour où vous aurez décidé de franchir le pas:

- Attention aux sauvegardes ! De nombreuses routines assemblées "plantent" car elles font appel à des routines de la MEM sans que l'on ait fait suffisamment attention aux registres qu'elles utilisent; toujours soigneusement comparer les registres à sauvegarder et ceux employés dans les routines.

- Utiliser la liste-objet ! L'Assembleur du 71 vous offre la possibilité d'éditer une liste-objet commentée après l'assemblage de vos routines. Ne négligez pas ce moyen inestimable pour éviter les erreurs de syntaxe, malgré la place mémoire "grappillée".

Avec le module FORTH/Assembleur, nous avons un choix de possibilités extrêmement variées pour une initiation graduée à l'Assembleur de Saturn, en démarrant avec l'écriture de primitives FORTH et jusqu'à, pourquoi pas, l'élaboration de votre propre langage évolué de programmation, vouant aux Gémonies les Basic, Pascal et autre Prolog.

Olivier ARBEY (SIG7-P118-T164)

CHOC EN RETOUR

A propos de REPLACE\$:

J'ai refait le Lex REPLACE il est un peu plus grand que la première version. Je le tiens à votre disposition. La première version ne remplace pas correctement la chaîne à retourner à BASIC sur la Math Stack et utilise la partie basse du registre R4 (R4(A)).

A propos de CLEX:

bien lire à la ligne 5 du programme:
if o then a0=a0+o+6

A propos de HTA\$:

Mon deuxième Lex est un mangeur de Math Stack, une nouvelle version vous sera proposée d'ici la fin de l'année. Avec HTA\$ et ATH\$.

M. MARTINET #12

END-UP !

Le fichier LEX que je vous propose est une excellente illustration des propos tenus par MM. David et Istria dans le JPC du mois d'avril, propos tenant à l'utilisation des POLLS et des BUFFERS. Mais il est aussi, et surtout, le résultat de mon acquisition du tome III des IDS, ouvrage qui s'avère indispensable au programmeur averti.

Le LEX contient une fonction ENDUP qui "agit" de

la même manière que la fonction STARTUP du HP71, mais lorsqu'on l'éteint. Le principe en est simple: l'ordre ENDUP place dans un buffer qui lui est réservé (Provisoirement bENDUP a pour ID: '80B') les mots-clés basic qui le suivent. Le buffer est créé s'il n'existait pas déjà. Ainsi: ENDUP 'DISP "123" @ WAIT 2'
Place la chaîne: DISP "123" @ WAIT 2 dans bENDUP, puis y ajoute le caractère de code ASCII 13 (Il sera utilisé plus tard par LINEP+). Le HP71 détruisant les buffers à chaque reconfiguration (Cf JPC 23 d'avril 85 art. P. David p.30), il est nécessaire d'intercepter le poll pCONFIG et de spécifier au 71 de ne pas détruire notre buffer.

Mais comment faire, me direz-vous, pour que le 71 affiche: 123 et attende 2" avant de s'endormir ? C'est fort simple: il suffit d'intercepter le poll pPWROF qui est envoyé par la routine DSLEEP (Deep Sleep), d'appeler I/OFND qui renverra l'adresse du début de la chaîne contenue dans bENDUP (C'est à dire l'adresse du D de DISP), puis de se brancher sur LINEP+ qui traitera DISP "123" @ WAIT 2 comme n'importe quelle ligne Basic (Cette routine est utilisée par RUN...), puis se branchera sur MAINLoop, à savoir la boucle principale (C'est elle qui détermine toute action à entreprendre). Il faut prendre soin d'allumer le drapeau fITNOF (TurN Off) avant d'appeler LINEP+, car alors MAINLP se branchera sur DSLEEP. Le poll pPWROF sera donc de nouveau envoyé, ce qui nous oblige à tester l'état du drapeau fITNOF au début de notre routine (xeq) pour savoir ce que l'on doit faire.

Principe de xeq:

OFF ou BYE ou [f] [ON]

PWROFF

GOSUB DSLEEP ---

LC(2) bSTARTUP

GOSUB I/OFND

GONC out

GOTO LINEP+


```

polhnd
LC(2) pCONFIG Est-ce le moment de
      $
      sauver le buffer ?
?B=C B
GOYES savebf -Oui
LC(2) pPWROFF -Non; le 71 veut-il
      $
      dormir ?
?B=C B
GOYES xeq -Oui
GONC rtnc -Non; c'est un BET.

out
C=R0 Ces lignes font partie
      $
      d'un point de sortie.
      $
      Cette place
D=C A est inhabituelle, mais
      $
      évite un GOTO
      $
      supplémentaire.
savebf LC(3) bENDUP Trouve ce buffer, s'il
      $
      existe, et empêche sa
      $
      destruction.
GOSBVL I/DRES
rtnc A=0 A Il est indispensable de
      $
      désarmer Carry, que le
      $
      poll ait
      $
      ou non été intercepté.
RTNSXM Rend la main à l'appelant.

xeq
LC(3) bENDUP Il n'y a rien à faire si
      $
      le buffer n'existe pas.
GOSBVL I/OFND
GONC rtnc
C=D A Le poll intercepté étant
      $
      un poll rapide, il faut
      $
      impérativement
      $
      préserver D(A).
LC(2) f1PWN Les 3 lignes suivantes
      $
      GOSBVL SFLAG? peuvent être supprimées si
      $
      GONC out l'on désire que le HP71
      $
      exécute le buffer même en
      $
      mode programme. Mieux vaut
      $
      les garder, en vue d'une
      $
      utilisation future par HP,
      $
      à moins que ce ne soit
      $
      indispensable.
LC(2) f1TNOF La démoniaque bête
      $
      s'était-elle endormie ?
GOSBVL SFLAG?

GDC out Alors il ne faut pas la
      $
      déranger.
LC(2) f1TNOF Amènera TITAN à se
      $
      rendormir après un cours
      $
      passage par
      $
      MAINLP.
GOSBVL SFLAGS
CD1EX D1 est inchangé depuis
      $
      GOSBVL I/OFND. Il pointe
      $
      sur la
      $
      chaîne dans le buffer.
GOVLNG LINEP+ Le buffer est copié,
      $
      "tokenize", puis exécuté.

strngp GOVLNG STRNGP
delayd GOVLNG DELAYD

REL(5) delayd Décompile ENDUP.
REL(5) strngp Compile ENDUP: ENDUP
      $
      [chaîne].
endup GOSBVL EXPEXC Prépare l'évaluation
      $
      d'une expression (?).
      $
      GOSBVL XXHEAD Retire l'en-tête de la
      $
      chaîne sur la pile.
LCHEX #0D Signale du BASIC en
      $
      ASCII: la chaîne devra
      $
      donc être
      $
      "tokenize".
P= 1

$ STKWP n'est pas un point d'entrée supporté par
$ HP. Les possesseurs d'un HP-71 version 1BBBB
$ peuvent éventuellement remplacer les lignes qui
$ suivent par: GOSBVL #1A55D.

ST=0 0 $ Desarmé pour que
CD1EX $ S ADHEAD rende la main.
C=-C A $
C+P+1 $ T
C=-C A $
?C<D A $ K
GOYES charer $ Erreur en cas de
CD1EX $ W mémoire insuffisante.
DAT1=C WP $
P= 0 $ P
GOSBVL ADHEAD $ Rajoute un en-tête à
      $
      la chaîne.

C=0 A
LC(1) 2
RO=C
LC(3) bENDUP Sauvegarde l'ID du buffer

```

```

      †          dans R2.
† ENDUP ne diffère de STARTUP que par l'usage d'un
† buffer d'ID différent. Les possesseurs de HP71B
† version 1BBBB peuvent donc remplacer ce qui suit
† par: GOVLNG #1A4AF (C'est la suite de STARTUP).

R2=C
GOSBVL REVPOP
CD1EX
CROEX
?A>C A
GOYES char10 † Sort en cas de chaîne
C=R2 † nulle.
GOSBVL I/DDAL † Détruit l'ancien buffer
† si nécessaire.
GOVLNG NXTSTM † Continue l'exécution
charer GOVLNG MEMERR † d'une ligne multi-ordres
char10 B=A A † Copie la longueur
C=0 A † actuelle du buffer.
LC(2) 192 † 95 caractères + CR
?A<=C A
GOYES char40
GOVLNG STROVF † Chaîne est trop longue.
char40 R1=A
C=R2 † C(A)= buffer ID.
GOSBVL I/DALL † Alloue au buffer l'esp.
† occupé par la chaîne.
GONC charer † Erreur en cas de mémoire
† insuffisante.
A=R0 † A= pointeur de chaîne.
DO=A
C=R1 † C= longueur de la chaîne
GOSBVL MOVEU3 † Copie la chaîne dans le
GOVLNG NXTSTM † buffer

NIBHEX 00
start LC(3) bSTART
GONC run (B.E.T.)

NIBHEX 00
endst LC(3) bENDUP
run AD1EX Sauvegarde D1
R1=A
GOSBVL I/OFND Recherche le buffer. Arme
† Carry s'il existe.
CD1EX DO= adresse du début de
† la chaîne.
CDOEX

```

```

B=C A Sauvegarde l'ancien DO =PC
GOSBVL D=AVMS
A=R1 Réactualise D1.
D1=A
GONC bf2s20 Carry est désarmée si le
† buffer n'existe pas.
† Renvoie
† alors la chaîne nulle.
bf2s10 A=DAT0 B Recopie un caractère du
† buffer.
LCHEX 0D Peut-on faire autre chose?
?A=C B
GOYES bf2s20 -Oui
C=A B -Pas encore; il faut
† placer le caractère sur la
† pile
GOSBVL STKCHR
DO=DO+ 2
GONC bf2s10 et passer au suivant
† (B.E.T.).
bf2s20 C=B A Restaure DO.
DO=C
GOVLNG ADHEAD Renvoie enfin la chaîne à
† l'utilisateur impatient
† et...

END

```

Des registres d'entrée et de sortie.

Les registres IN et OUT se présentent ainsi:

```

IN          15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Soit 16 bits

OUT                12 11 10 9 8 7 6 5 4 3 2 1 0
Soit 12 bits

```

Ce qui suit ne concerne que l'interaction clavier-CPU.
 Les 4 bits de poids faible du registre OUT (3-0) correspondent aux 4 rangées du clavier:

```

      3      2      1      0
Rangées: 0.../  A...#  Z...-  on...+

```

Si l'on y place 'F', les 4 rangées sont actives; 8 la seule rangée supérieure
 ...Supposons qu'on y place '1'. Si l'on appuie sur la touche: + , IN(0) sera mis à '1'. La correspondance se fait de manière fort logique:

```

      14 13 12 ..... 1 0
      on f g ..... , +

```

C'est ainsi qu'on peut savoir si une touche est ou n'est pas enfoncée. Il est à remarquer que la touche ON a un double traitement: celui des autres touches (OUT(0)-IN(14)); mais aussi IN(15). Cette dernière interruption n'est donc pas masquable.

Ceux qui ont lu mon article (dans ce même numéro) sur ENDUP comprennent désormais pourquoi la touche ON n'est pas la seule qui réveille la machine (même si c'est la seule qui soit prise en considération). En effet, l'instruction SHUTDN est précédée de:

```

LCHEX 001
OUT=CS

```

Jean-Jacques MOREAU

PILE A 16 NIVEAUX

La pile de Commandes est une invention si pratique qu'elle me manque horriblement dès que je me sers d'un ordinateur autre que le 71. Mais on en veut toujours plus, le refrain est connu ...

Alors 5 niveaux c'est bien, mais 16 serait tellement mieux ! (16 représente en effet un maximum). L'ordre CMD16 y pourvoit, pour le prix modique de 77 octets tout compris. A titre de comparaison le programme BASIC joint effectue exactement le même travail et occupe 204 octets, plus une bonne centaine pour stocker les chaînes alphanumériques intermédiaires: méditez sur les bienfaits du Langage Machine. Ceux qui n'ont pas

le module FORTH arriveront au même résultat en exécutant la routine CMDST. Reportez-vous au programme pour les explications, et au plaisir de vous lire dans JPC.

Jean-Pierre Bondu (#33-SIG#4)

```

LEX 'CMD16'
ID #5D
MSG 0
POLL 0

ENTRY CMD16e
CHAR #D

KEY 'CMD16'
TOKEN 130
ENDTXT

```

```

IOBFEN EQU #2F576 fin des buffers et début
          # de la Pile de Commandes
RAWBFR EQU #2F580 CALC Mode Raw Buffer (fin
          # de Cmd Stack)
MAXCMD EQU #2F976 nombre de niveaux-1 de la
          # pile de commandes
NXTSTM EQU #08A48 NeXT StatMent
STDp EQU #03FB2 analyse de STD
STDd EQU #05303 décompilation de STD

```

```

REL(5) CMD16d
REL(5) CMD16p
CMD16e D1=(5) IOBFEN
          A=DAT1 A
          D1=A D1= ADR: adresse début
          # Cmd Stack
          C=0 W
          C=C-1 S C[S]= 15 : nombre de
          # boucles-1
          P= 3
          C=P 3 C[S-0]= 003000
          CMD10 DAT1=C 6 on remplit l'espace
          # réservé à la Command Stack
          D1=D1+ 6 par 16 fois la chaîne
          # 003000 (qui code une
          # entrée)
          C=C-1 S C[S]= compteur de boucle
          GONC CMD10
          CD1EX C[A]= ADR+96

```

```

D1=(5) RAWBFR
P=P-1      P= 2 : nombre de boucles-1
CMD20 DAT1=C A      on poke 3 fois la
      *            longueur de la CMDS, mais
      *            pourquoi??
D1=D1+ 5
P=P-1      3 boucles exécutées?
GONC  CMD20      non
D1=(5) MAXCMD
DAT1=C S      MAXCMD= C[S]= 15
GOVLNG NXTSTH

CMD16d GOVLNG STDd
CMD16p GOVLNG STDp
      END

```

* Programme BASIC équivalent

```

10 SUB CMDST
20 DEF FNR$(A$)=A$[5]&A$[4,4]&A$[3,3]&A$[2,2]&
  A$[1,1]
30 DIM S$(96) @ A=HTD(FNR$(PEEK$("2F576",5)))
40 FOR I=1 TO 16 @ S$=S$%"000300" @ NEXT I
50 E$=FNR$(DTH$(A+96)) @ POKE "2F580",E$&E$&E$
60 POKE DTH$(A),S$ @ POKE "2F976","F"
70 END SUB

```

LES VARIABLES ET LA MATH STACK
La naissance des IDS VI

Au mois d'Avril, je vous proposais un article sur les manipulations de chaînes alphanumériques sur la Math Stack. Cette fois-ci, je vous propose une vue plus générale sur toutes les représentations de variables supportées par Hewlett-Packard.

C'est en désassemblant la fonction TYPE de la ROM Maths que je me suis penché sur le problème. Vous trouverez ci-joint le Lex "MATH48" qui n'est autre que la fonction TYPE (H.P. id:002, tk:048).

Nous ne recensons pas moins de 13 types différents de variables dans le système HP71B mais

lorsqu'elles sont déposées sur la Math Stack, il n'en reste plus que 5.

```

REAL
  !E x pi. . M a n t i s s e . . !S!
  !. . . . . ! =>Ad.haute
SHORT
  !E x pi. . M a n t i s s e . . !S!
  !. . . . . ! =>Ad.haute
INTEGER
  !E x pi. . M a n t i s s e . . !S!
  !. . . . . ! =>Ad.haute
COMPLEX
  !E 0!
  !E x pi. Mantissee imaginaire .!S!
  !E x pi. . Mantissee réelle .!S!
  !. . . . . ! =>Ad.haute
COMPLEX SHORT
  !E 0!
  !E x pi. Mantissee imaginaire .!S!
  !E x pi. . Mantissee réelle .!S!
  !. . . . . ! =>Ad.haute
Chaîne
  !F 0!Longueur !Adresse .!. DIM .!. Chaîne....
  !. . . . . ! =>Ad.haute
INTEGER tableau
  !A!#!b! Dim./Longueur ! Adresse !
  !. . . . . ! =>Ad.haute
SHORT tableau
  !B!#!b! Dim./Longueur ! Adresse !
  !. . . . . ! =>Ad.haute
REAL tableau
  !C!#!b! Dim./Longueur ! Adresse !
  !. . . . . ! =>Ad.haute
COMPLEX SHORT tableau
  !D!#!b! Dim./Longueur ! Adresse !
  !. . . . . ! =>Ad.haute
COMPLEX tableau
  !E!#!b! Dim./Longueur ! Adresse !
  !. . . . . ! =>Ad.haute
Tableau de chaînes
  !F!#!b! Dim./Longueur ! Adresse !
  !. . . . . ! =>Ad.haute
Chaîne de tableau non-existant
  !F 8!Longueur !Nom .!0 0!Nr.Ele.!
  !. . . . . ! =>Ad.haute

```

b = option base (1 ou 0)
= Nombre de dimensions (1 ou 2 et 0 pour les variables scalaires.

Vous remarquerez que les nombres simples REAL, SHORT et INTEGER sont déposés sur la Math Stack sous la même forme. En effet, les formes SHORT et INTEGER ne sont retenues par le système que pour des questions de place en mémoire. Durant les calculs ou les affichages de résultats, nous sommes en présence de nombres du type REAL.

La fonction TYPE:

SYNTAXE: TYPE(X) où X est REEL, COURT, COMPLEX, CHAINE ou TABLEAU.

La fonction retourne un nombre entre 0 et 8 correspondant au type de variable.

TYPE 0: INTEGER, SHORT et REAL
 TYPE 1: COMPLEX et COMPLEX SHORT
 TYPE 2: Chaîne et Chaîne de tableau
 TYPE 3: INTEGER tableau
 TYPE 4: SHORT tableau
 TYPE 5: REAL tableau
 TYPE 6: COMPLEX SHORT tableau
 TYPE 7: COMPLEX tableau
 TYPE 8: tableau de chaînes (sans indices)

Voilà, Heureuse Programmation à tous et à plus tard.

M.MARTINET #12 (S.I.G. #2)

LEX 'MATHAB'
 ID #02
 MSG 0
 PDLL 0
 ENTRY type
 CHAR #F
 KEY 'TYPE'
 TOKEN 48
 ENDTXT

FNRTN4 EQU #0F238

NIBHEX C11 La fonction TYPE admet n'importe quel type de paramètre.

type A=DAT1 7 On charge dans A(6-0) le début du spécificateur du paramètre se trouvant

ASRC
 *
 *
 ASRC
 *
 *
 C=0 W On charge dans
 P= 14 C(14) la valeur hexa
 LCHEX 9 9.
 ?C<A P Si A(14) est supérieur à
 * 9 nous ne sommes pas en
 * présence d'un nombre
 GOYES NN REAL, SHORT ou INTEGER.
 C=0 P Alors on branche en NN.
 * La variable est du type
 GOND Fnrtn4 0, on charge 0 dans C(W)
 A=A+A S avant le retour à BASIC.
 * Si la variable est un
 * tableau: A(S) est
 * différent de 0. D'où A=A+A
 * S.
 ?A#0 S Est-on en présence d'un
 * tableau ?
 GOYES TY01 Oui: on branche alors en
 * TY01.
 LCHEX D
 C=A-C P
 A=A+1 P Si A(15-14)= 00 nous
 * sommes en présence d'une
 * chaîne. Autrement C(14)= 1
 * et c'est soit un COMPLEX
 * ou COMPLEX SHORT: il n'y a
 * pas de Carry au test
 * suivant.
 GOC TY00 Branchement en TY00 pour
 * retour à BASIC avec 2 pour
 * une chaîne.
 D1=D1+ 2 Nous avons affaire à un
 * nombre complex, on
 * décrémente le pointeur D1
 * pour sauter la signature
 D1=D1+ 16 et la partie imaginaire du
 * nombre: D1=D1+ 16
 *
 GOND Fnrtn4 Retour à Basic.
 TY00 CD1EX C'est une chaîne: on
 * place D1 sur la fin de
 * la chaîne.
 C=C+A A A(A) contient la longueur
 * de la chaîne en quartets.
 CD1EX D1=D1+ A(A)

```

GDNC  Fnrtn4  Retour à BASIC.
TY01  LCHEX  7   C'est un tableau: A(P)-7
      *       varie de 3 à 7 (voir
      *       article).
      C=A-C P
Fnrtn4 GOVLNG FNRTN4  Retour à BASIC (voir JPC
      *       précédents).
      END

```

```

: C(N,P)=C(N,N-P)
: Max(P)C(N,P)=C(N,N/2)
: C(N,N)=C(N,0)=1
: C(N,1)=C(N,N-1)=N
: A(N,N)=A(N,N-1)=N!
: A(N,1)=N
: A(N,0)=1

```

Mais quelles sont donc les utilisations précises de ces 2 fonctions aux définitions pour le moins rébarbatives ? Une des branches des mathématiques qui fait cas de ces 2 abominations est le calcul des probabilités. Voici 2 petits problèmes qui vous ouvriront la voie vers la connaissance suprême et divine de la statistique. Soit une urne contenant 20 boules(N) numérotées afin d'être reconnaissables.

A PROPOS DE COMBINAISONS ET D'ARRANGEMENTS

Dans tout l'arsenal des fonctions mathématiques du HP71B, il est 2 fonctions que l'on cherche souvent et qui manque à toute personne désirant effectuer du calcul de probabilités: ce sont les fonctions traitant les combinaisons et les arrangements.

Muni des I.D.S. 2 et de bonne volonté, je décidai donc dernièrement de lever le voile sur les probabilités avec le HP71B. Peut-être est-il nécessaire de rappeler quelles sont les définitions des arrangements et des combinaisons ?

La combinaison de N éléments pris P à P (et notée: C(N,P)), est le nombre de manières qui permettent de disposer P éléments pris dans un ensemble de N éléments, sans tenir compte de l'ordre.

L'arrangement de N éléments pris P à P (notée: A(N,P)), est le nombre de manière d'agencer ces P éléments en tenant compte, cette fois de l'ordre dans laquelle ils interviennent.

Les formules donnant ces 2 valeurs sont les suivantes:

$$C(N,P) = \frac{N!}{P!(N-P)!}$$

$$A(N,P) = \frac{N!}{(N-P)!}$$

Soit en fait: $A(N,P) = C(N,P) * P!$

Où N! représente Factorielle N, définie par: $N! = N * (N-1) * (N-2) * \dots * 2 * 1$. Et pour laquelle on a par définition: $0! = 1$.

On déduira alors les résultats évidents suivants:

Soit également une personne d'intelligence moyenne et qui désire tirer 4 boules parmi ces 20. Sa question sera probablement la suivante: combien d'ensembles différents puis-je former en agissant ainsi ?

Et bien le problème que se pose cette personne peut être résolu par le raisonnement suivant: le résultat sera équivalent si l'on tire d'abord une première boule, puis une seconde, sans avoir remis la première, et ainsi de suite. Au premier tirage, il y a donc 20(N) possibilités, au deuxième, il n'y en a plus que 19(N-1), au troisième, 18, et au quatrième, 17. Le nombre total de possibilités, est donc: $NP = 20 * 19 * 18 * 17$.

Ce nombre est également le nombre de combinaisons possibles de 20 éléments pris 4 à 4, soit: $NP = C(20,4) = 4845$.

Prenons maintenant l'exemple du tiercé. Combien de bulletins faut-il valider si l'on veut être sûr de gagner dans l'ordre, si 15 chevaux sont au départ ?

Le raisonnement du début est strictement le même, et l'on trouvera le nombre de bulletins égal à: $NB = 15 * 14 * 13$. Mais, en y regardant de plus près, on s'apercevra que ce nombre NB ne tient pas compte de l'ordre d'arrivée des chevaux; en effet, les combinaisons 8,5,6, 6,5,8, 5,8,6... n'ont été comptées qu'une fois, or, on veut toucher le tiercé dans l'ordre. Il faut donc multiplier NB par le nombre de permutations de 3 éléments, soit 3!. Le bon résultat est donc: $NB = 15 * 14 * 13 * 3!$. Soit encore: $NB = A(15,3)$.

On comprendra donc l'immense champ d'applications de 2 fonctions telles que l'arrangement et la

combinaison. Grâce au petit LEX qui suit, pour calculer A(N,P), il vous suffira de taper: ARR(N,P); quant au calcul de C(N,P) vous frapperez: COMB(N,P).

Maintenant qu'on été rappelés quelques notions d'analyse combinatoire, revenons à notre bon vieux HP71B, pour voir comment tournent ces 2 routines.

L'algorithme de calcul est très simple, puisqu'il suit les définitions des fonctions données plus haut. En revanche, il est judicieux de se demander comment le HP 71 code les nombres de manière interne, et comment il effectue les opérations sur eux.

Il faut tout d'abord savoir que tout nombre entré dans une fonction (MOD(20,4),INT(5,1), ANGLE(43,15)...) est codé sur 16 quartets: 1 pour le signe, 12 pour la mantisse, et trois pour l'exposant et son signe. La précision de ces nombres est donc de 12 chiffres significatifs. Mais lorsque le HP 71 se met à calculer, il commence par traduire les nombres dans 2 registres de 16 quartets, permettant alors des opérations avec 15 chiffres significatifs. Une fois les opérations terminées, il suffit de laisser tomber les trois quartets supplémentaires pour retrouver un nombre de 12 chiffres représentable par la machine. Les opérations qui permettent de passer de 12 à 15 chiffres sont SPLITA, SPLTAC, celle qui permet de passer de 15 à 12, est: uRES12.

Les fonctions AD2-15, DV2-15, MP2-15, FCSTRT, effectuent respectivement: addition, division, multiplication, et factorielle sur des nombres de 15 chiffres.

Voilà, j'espère qu'avec ces quelques précisions, vous nous submergerez bientôt avec des tas de fonctions mathématiques, toutes plus intéressantes les unes que les autres.

L. ISTRIA (PPC#3,SIG#3)

LEX 'COMBARR'
ID #5D
MSG 0

```

:          POLL 0
: FNRTN1 EQU #0F216 * Donne un résultat en
:          * rendant la main à BASIC.
: POP2N EQU #0BC8C * Prend les 2 nombres
:          * au sommet de la Math Stack
:          * et les passe dans A et C.
: ARGERR EQU #0BF19 * Renvoi: Arg. non valide.
: SPLITA EQU #0C6BF * Traduit un 12 digits
:          * dans A en un 15 dans A/B.
: SPLTAC EQU #0C934 * Traduit 2 12 digits dans
:          * A et C dans A/B et C/D.
: AD2-15 EQU #0C363 * AB = AB + CD.
: DV2-15 EQU #0C4AC * AB = AB/CD.
: FCSTRT EQU #0E757 * AB = FACT(AB).
: uRES12 EQU #0C994 * Transforme un 15 digits
:          * dans AB en 12 digits
:          * dans C.
: TST12A EQU #0D476 * Effectue des Tests sur
:          * A et C.
: STAB2 EQU #0D400 * Stocke A/B dans R2/R3.
: RCCD2 EQU #0D41C * Rappelle dans C/D R2/R3.
: CLRFR3 EQU #0C6F4 * AB = INT(AB).

:          ENTRY ARGMT
:          CHAR #F
:          ENTRY CMBS
:          CHAR #F
:          KEY 'ARR'
:          TOKEN 18
:          KEY 'COMB'
:          TOKEN 19
:          ENDTXT
:          NIBHEX 8822

: ARGMT
:          GOSUB STOCK
:          GOSBVL uRES12
:          GOVLNG FNRTN1
:          NIBHEX 8822

: CMBS
:          * Calcul de la combinaison
:          * de N et P.
:          GOSUB STOCK
:          GOSBVL STAB2
:          A=R1
:          GOSBVL SPLITA
:          GOSBVL FCSTRT
:          C=B W
:          D=C W
:          C=A W
:          A=R3
:          B=A W
:          A=R2

```

```

GOSBVL DV2-15
GOSBVL uRES12
GOVLNG FNRTN1
STOCK
GOSBVL POP2N
D1=D1+ 16 * Actualise D1
GOSUB STKTST
A=R0
C=R1
GOSUB iARR
RTN
iARR * Calcul de l'arrangement
* de N et P
C=-C-1 S
GOSBVL SPLTAC
GOSBVL AD2-15 * N-P
GOSBVL FCSTRT * (N-P)!
GOSBVL STAB2
A=R0
GOSBVL SPLITA
GOSBVL FCSTRT * N!
GOSBVL RCCD2
GOSBVL DV2-15 * N!/(N-P)!
RTN
STKTST
R0=A
R1=C
GOSUB TEST
C=R0
GOSUB TEST
A=R0
C=R1
P= 6
GOSBVL TST12A * N est-il inférieur à P ?
RTNC
GOTO ERR
TEST * routine de test de la
* validité des entrées.
P=C 15 * Nombre négatif ?
?P= 9 *
GOYES ERR
A=C W *
P= 0 * Le nombre est-il > 999 ?
LCHEX 002 *
?A>C X *
GOYES ERR
GOSBVL SPLITA
GOSBVL CLRFR * Le nombre a-t-il un
* partie fractionnaire ?
RTNC
ERR GOVLNG ARGERR

```

END

HEURES-MINUTES-SECONDES OU LES NOMBRES REELS

Le HP41 possède 4 petites fonctions numériques permettant des conversions et des calculs entre les heures décimales et les heures sexagésimales. Ces fonctions ne faisaient pas partie du système d'exploitation de titan.

Lorsqu'un nombre réel est introduit dans un des registres du microprocesseur ou dans la mémoire centrale, il est toujours codé sur 16 quartets (1 registre interne).

Le champ S (1 quartet) du registre contient le signe du nombre (0 pour + et 9 pour -).

Le champ M (12 quartets) contient la mantisse du nombre.

Le champ X (3 quartets) contient l'exposant.

```

-----
|S|. . M A N T I S S E . . |E X P|
-----
| F E D C B A 9 8 7 6 5 4 3 2 1 0

```

La mantisse ne contient pas de virgule, la position de la partie décimale est uniquement fonction de la valeur du champ X.

Exemple:

```

+10      |0|1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0|0 0 1|
-0.067   |9|6 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0|9 9 8|
PI        |3|3 1 4 1 5 9 2 6 5 3 5 9|0 0 0|
Inf       |1|? ? ? ? ? ? ? ? ? ? ? ? ? ?|F 0 0|
-Inf      |9|? ? ? ? ? ? ? ? ? ? ? ? ? ?|F 0 0|
NaN       |1|? ? ? ? ? ? ? ? ? ? ? ? ? ?|F 0 1|
-0        |1|1 2 3 4 5 6 7 8 9 0 1 2|5 0 0|
(Underflow)

```


(A,B).

La transformation 15 => 12 est effectuée grâce à la routine "uRES12". Elle transforme le nombre se trouvant dans X en un nombre codé sur 12 chiffres dans le registre C.

Les échanges X,Y (<=> Sc1,Sc2) sont réalisés grâce aux routines suivantes: STAB1, EXAB1, RCCD1, STAB2, EXAB2, RCCD2 et STCD2.

LE PROGRAMME

Les fonctions:

HR(N) donne N en hh,d...d

HMS(N) donne N en hh,m:ssd...d

HMS+(N,M) donne N+M en hh,m:ssd...d

HMS-(N,M) donne N-M en hh,m:ssd...d

Les routines utilisées:

AD2-15	X=X+Y
DV2-15	X=X/Y
EXAB2	X(<>Sc2)
POP1R	Voir JPC précédents
SPLITA	X=A
STAB1	Sc1=X
CLRFRC	X=INT(X)
STAB2	Sc2=X
RCCD1	Y=Sc1
STCD2	Sc2=Y
RCCD2	Y=Sc2
uRES12	C=X
SB159	X=X-Y
FNRTN4	Retour à BASIC.
MP2-15	X=X*Y
FUNCR0	Scratch (RAM réservée)
FUNCR1	Scratch (RAM réservée)

Voilà, la meilleure précision sera obtenue via: OPTION ROUND NEAR. J'espère que ces 4 petites fonctions vous seront très utiles.

Heureuse Programmation.

Michel MARTINET #12

LEX 'HLEX'

TITLE HMS *** UTILITIES *** 01/05/1985

CON(2) 92

CON(2) 60

CON(2) 63

CON(5) 0

NIBHEX F

REL(4) 1+(TXTB)

CON(4) 0

CON(5) 0

CON(3) (TXT1)-(TXTB)

REL(5) HMS+e

CON(1) 15

CON(3) (TXT2)-(TXTB)

REL(5) HMS-e

CON(1) 15

CON(3) (TXT3)-(TXTB)

REL(5) HMSe

CON(1) 15

CON(3) (TXT4)-(TXTB)

REL(5) HRe

CON(1) 15

TXTB

TXT1

CON(1) 7

NIBASC 'HMS+'

CON(2) 60

TXT2

CON(1) 7

NIBASC 'HMS-'

CON(2) 61

TXT3

CON(1) 5

NIBASC 'HMS'

CON(2) 62

TXT4

CON(1) 3

NIBASC 'HR'

CON(2) 63

TXTE

NIBHEX 1FF

AD2-15 EQU #0C363

DV2-15 EQU #0C4AC

EXAB2 EQU #0D40E

POP1R EQU #0E8FD

SPLITA	EQU	#0C6BF	:		NIBHEX	811
STAB1	EQU	#0D3D9	:	HRe	ST=0	0
CLRFRC	EQU	#0C6F4	:	HRs	GOSUB	INTX
STAB2	EQU	#0D400	:		GOSUB	Y=100
RCCD1	EQU	#0D3F5	:		GOSUB	MM.SS
STCD2	EQU	#0D427	:		C=0	W
RCCD2	EQU	#0D41C	:		P=	13
uRES12	EQU	#0C994	:		LCHEX	36
SB15S	EQU	#0E19A	:		D=C	W
FNRTN4	EQU	#0F238	:		C=C+1	A
MP2-15	EQU	#0C43A	:		GOSBVL	DV2-15
FUNCR0	EQU	#2F89B	:		GOSBVL	EXAB2
FUNCR1	EQU	#2F8AB	:		GOSUB	Y=60
			:		GOSBVL	DV2-15
			:		GOSBVL	RCCD2
HMS+e	NIBHEX	8822	:		GOSBVL	AD2-15
	ST=1	0	:	HD	GOSBVL	RCCD1
	GOSUB	HRs	:		GOSBVL	AD2-15
	GOSUB	SAVEX	:		?ST=1	0
	GOSUB	HRs	:		RTNYES	
	GOSUB	RESTRY	:		XM=0	
	GOSBVL	AD2-15	:		GOSBVL	uRES12
HJ	GOSUB	INTXs	:		GOVLNG	FNRTN4
	ST=0	0	:			
	GOTO	HMSs	:			
			:	INTX	GOSBVL	POP1R
			:		GOSBVL	SPLITA
HMS-e	NIBHEX	8822	:	INTXs	GOSBVL	STAB1
	ST=1	0	:		GOSBVL	CLRFRC
	GOSUB	HRs	:		GOSBVL	RCCD1
	GOSUB	SAVEX	:		GOSBVL	STAB1
	GOSUB	HRs	:	ADCHSX	?A=0	S
	GOSUB	RESTRY	:		BOYES	SA-
	GOSBVL	SB15S	:		A=0	S
	GONC	HJ	:		GONC	AD
			:		SA-	A=A-1 S
HMSe	NIBHEX	811	:	AD	GOSBVL	AD2-15
	ST=0	0	:		RTNCC	
	GOSUB	INTX	:			
HMSs	GOSUB	Y=60	:			
	GOSUB	MM.SS	:	Y=100	C=0	W
	C=0	W	:		P=	14
	P=	14	:		LC(1)	1
	LC(1)	6	:		D=C	W
	D=C	W	:		C=C+1	A
	C=C-1	A	:		C=C+1	A
	GOSBVL	MP2-15	:		RTN	
	GOSBVL	RCCD2	:			
	GOSBVL	AD2-15	:	MM.SS	GOSBVL	MP2-15
	GOSUB	Y=100	:		GOSBVL	STAB2
	GOSBVL	DV2-15	:		GOSBVL	CLRFRC
	GOTO	HD	:		GOSBVL	RCCD2
			:		GOSBVL	STAB2

```

GOSUB ADCHSX
RTN
Y=60 C=0 W
      P= 14
      LC(1) 6
      D=C W
      C=C+1 A
      RTN

```

```

RESTRY CD1EX
        RO=C
        D1=(5) FUNCRI
        C=DAT1 W
        D=C W
        D1=D1- 16
        C=DAT1 W
        CROEX
        D1=C
        C=RO
        RTN

```

```

SAVEX CD1EX
        D1=(5) FUNCRO
        DAT1=A W
        D1=D1+ 16
        A=B W
        DAT1=A W
        D1=C
        D1=D1+ 16
        RTN

        END

```

ENCORE UNE ERREUR !

Lorsque l'on possède un LEX traducteur de messages (FRALEX...), les tests d'erreur sont plus délicats à réaliser. Ceux qui n'ont pas le module FORTH se contenteront pour l'instant de la fonction utilisateur indiquée. La fonction EXTERR admet pour argument un numéro d'erreur, et renvoie soit 0 (bon numéro), soit 1 (vous avez tout faux...).

Par exemple tapez: CAT 125. Le numéro de cette erreur ('Invalid Specifier') est 58, ou 1058 si FRALEX est en machine. Tapez maintenant EXTERR(58), le résultat est 1 même si ERRN est 1058. En pratique il suffit donc de proposer le code "normal", ce qui est plus simple à la rédaction, plus clair à la relecture et enfin plus court dans un programme !

J'espère que ces explications sont assez claires car il est tard et mes pensées ne le sont plus, et je suis trop fatigué pour me relire. Et puis après tout, un peu à vous de travailler !

Sans rancune, et à la prochaine réunion !

Jean-Pierre Bondu (#33-SIG#4)

```

LEX 'XERRLEX'
ID #5D
MSG 0
POLL 0

```

```

RNDHX EQU #136CB
R3=D1C EQU #0352C
D1C=R3 EQU #03047
A-MULT EQU #1B349
ERRN EQU #2F7E6
FNRTN1 EQU #0F216

```

```

ENTRY XTERR
CHAR #F
KEY 'XERR'
TOKEN 128
ENDTXT

```

```

XTERR NIBHEX 811 un argument numérique: X
      GOSBVL RNDHX A[A]= X en hexa
      C=A A
      GOSBVL R3=D1C R3[A]= C[A] ; R3[9-5]= D1
      LC(5) ERRN
      CD1EX
      A=0 W
      C=0 W
      A=DAT1 B A[A]= No de LEX de ERRN
      LC(3) 1000
      D=C A D[A]= 1000
      GOSBVL A-MULT A[A]= (No LEX) #1000

```

```

C=0 A
D1=D1- 2
C=DAT1 B CIA]= numéro d'erreur
A=A+C A A[AI]= No d'erreur formaté
B=A A B[AI]= ERRN
GGSBVL D1C=R3 CIA]= X ; D1= sommet MSTK
A=C A A[AI]= X
A=0 S A[CS]= valeur du test
?A=B A X=ERRN ?
GOYES Un oui: on renvoie 1
?C>D A X>1000 ?
GOYES 1b11 oui: ajouter 128
LC(5) 872 sinon ajouter 1000=872+128
A=A+C A
1b11 C=0 W
LC(2) 128 plus court que LC(5) 128
A=A+C A A[AI]=X modifié ; B[AI]=ERRN
?A#B A X#ERRN ?
GOYES Zero oui: on renvoie 0
Un A=A+1 S
Zero C=0 W
C=A S
CSR W C[AW]= X flottant
D1=D1+ 16
GOVLNG FNRTN1
END

REM: R3 est utilisé.

```

(SB). Lorsque le bit perdu (bit de poids faible du champ "0") est égal à 1.

Exemple: supposons que nous ayons dans le registre D les codes suivants:

```

5 1 2 3 4 5 6 7 8 9 0 1 2 3 4 3
(0101)1 2 3 4 5 6 7 8 9 0 1 2 3 4(0011)

```

Après exécution de NIBHEX 81B nous obtenons:

```

2 1 2 3 4 5 6 7 8 9 0 1 2 3 4 9
(0010)1 2 3 4 5 6 7 8 9 0 1 2 3 4(1001)
(0010) = = = > (1001)

```

Voilà, il y en a d'autres encore mais elles ne sont pas encore très claires... à la prochaine.

Michel MARTINET (#12 SIG#2)

LA PILE DE COMMANDE EN ASSEMBLEUR

La pile de commande peut-être programmée de 1 à 16 niveaux. C'est un outil merveilleux qui à ma connaissance n'existe nulle part ailleurs que chez Hewlett-Packard.

Depuis la venue du HP71B, nous connaissons la routine basic SETCMDST qui faisait la joie de tous avant le module Forth/Assembleur.

Après quelques mois de mise en route, nous sommes aujourd'hui bien avancé en matière d'assembleur.

Je vous propose le Lex STKLEX qui est en tout point semblable à SETCMDST mais qui a le mérite de ne faire que 114 au lieu de 278 octets et qui est incomparable au point de vue rapidité.

La syntaxe: STACK N , et le tour est joué. La commande est programmable et utilisable après THEN/ELSE.

Michel Martinet #12 (SIG#2)

PROGRAMMATION SYNTHETIQUE

Encore quatre petites instructions que Hewlett-Packard nous avait caché:

```

NIBHEX 818 (ASRBS0) Pourquoi pas ?
NIBHEX 819 (BSRBS0)
NIBHEX 81A (CSRBS0)
NIBHEX 81B (DSRBS0)

```

Ces quatre instructions opèrent un déplacement de 1 bit vers la droite entre le champ "S" et le champ "0" des registres respectifs A, B, C et D. Contrairement aux instructions rSRB ou rSRC (fs), ces quatre opérations n'arment pas le Sticky Bit

					* du programme BASIC.
				C=DAT1 A	**
				D=C A	**
				LC(5) 6	**
				GOSBVL A-MULT	**
				C=D A	**
				C=C+A A	**
				D1=(5) RAWBFR	Identique à la ligne 120.
				DAT1=C A	**
				D1=D1+ 5	**
				DAT1=C A	**
				D1=D1+ 5	**
				DAT1=C A	**
				C=D A	**
				D1=C	**
				A=R0	**
				LCHEX 003000	Boucle FOR/NEXT (Lignes
				*	80 à 100 du programme.)
				Boucle DAT1=C 6	**
				A=A-1 A	**
				D1=D1+ 6	**
				?A#0 A	**
				GOYES Boucle	**
				D1=(5) MAXCMD	Ligne 140
				A=R0	Restitution du nombre de
				*	niveaux.
				A=A-1 A	**
				DAT1=A 1	**
				GOVLNG NXTSTM	Fin de programme.
				STKp GOVLNG FIXP	**
				STKd GOVLNG FIXDC	**
				END	**
				10 CALL SETCMDST(16)	**
				20 SUB SETCMDST(X)	**
				30 DEF FNR\$(A\$)=A\$[5]&A\$[4,4]&A\$[3,3]&A\$[2,2]	**
				&A\$[1,1]	**
				40 IF X<1 THEN X=1	**
				50 IF X>16 THEN X=16	**
				60 DIM S\$(X*6)	**
				70 A=HTD(FNR\$(PEEK('2F576',5)))	**
				80 FOR I=1 TO X	**
				90 S\$=S\$&'000300'	**
				100 NEXT I	**
				110 E\$=FNR\$(DTH\$(A+X*6))	**
				120 POKE '2F580',E\$&E\$&E\$	**
LEX	'STKLEX'				
ID	#5C				
MSG	0				
POLL	0				
ENTRY	STKe				
CHAR	#D				
KEY	'STACK'				
TOKEN	50				
ENDTXT					
A-MULT	EQU	#1B349			
CLCBFR	EQU	#2F576			
EXPEXC	EQU	#0F186			
FIXDC	EQU	#05493			
FIXP	EQU	#02A6E			
FLTDH	EQU	#1B223			
MAXCMD	EQU	#2F976			
NXTSTM	EQU	#08A48			
RAWBFR	EQU	#2F580			
	REL(5) STKd				
	REL(5) STKp				
STKe	SETHEX				
	GOSBVL EXPEXC	Routine qui place le			
	*	paramètre de STACK dans			
	*	A(W).			
	?A=0 S	Le nombre est-il positif ?			
	GOYES fltdh	Oui, on branche en fltdh		STKp	
	*	pour le transformer en			
	*	Hexa.		STKd	
	A=0 W	S'il est négatif, on met			
	*	0 pas défaut.			
fltdh	GOSBVL FLTDH	Dec.(A(W)) devient			
	*	Hex.(A(A)).			
	?A#0 A	Le nombre est-il égal à			
	*	zéro ?			
	GOYES STEXQ	Ok on branche pour			
	*	l'exécution de l'ordre.			
	A=A+1 A	Autrement on remplace 0			
	*	par 1.			
STEXQ	LC(5) 16	C(A)= 00010			
	?A<C A	Le paramètre est-il < 17?			
	GOYES STK10				
	A=C A	A(A)= C(A)= 00010 par			
	*	défaut si le paramètre est			
	*	supérieur à 16.			
STK10	R0=A	Sauvegarde			
	D1=(5) CLCBFR	Exécution de la ligne 70			

```

130 POKE DTH$(A),S$
140 POKE '2F976',DTH$(X-1)[5,5]
150 END SUB

```

ADRESSE D'UNE COLONNE GRAPHIQUE

Le mot ADR-AFF simplifie la gestion de l'affichage en FORTH. C'est une primitive, car destinée à servir de support à une animation graphique: il fallait favoriser la vitesse d'exécution.

Son rôle est le suivant: vous fournissez le numéro de la colonne que vous désirez adresser, ADR-AFF vous renvoie son adresse mémoire. Une colonne fait 1 octet, mais l'adressage se faisant par quartet, c'est l'adresse du quartet de poids faible qui vous est rendue. Autrement dit l'adresse du quartet le plus bas de la colonne.

Un exemple: DECIMAL 20 ADR-AFF H. --> 2E12A adresse de la colonne 20. Essayez 255 20 ADR-AFF C! ABORT pour saturer la colonne 20. Convaincu ? L'argument doit être compris entre 1 et 132. Si tel n'est pas le cas, 0 est renvoyé (ex: 140 ADR-AFF -> 0).

Pour remplir l'affichage, définissez le mot :
DECIMAL : FULL-AFF 255 132 I DO DUP I C! LOOP ;
Vous trouverez le mot ADR-AFF défini en FORTH; je vous suggère de rentrer les 2 et de comparer la vitesse d'exécution ... A la prochaine !

Jean-Pierre Bondu (#33)

FORTH
WORD 'ADR-AFF'

```

RANGE EQU #1B07C teste si octet A[B] entre
* C[B] et C[3,2]
DD1ST EQU #2E104 début Driver 1
DD2ST EQU #2E200 " " 2
DD3ST EQU #2E300 " " 3

```

```

A=DAT1 A A[A] = colonne
A=A-1 A 0<=colonne<=131
LC(5) 131
?A>C A colonne>131 ou <0 ?
GOYES Null oui: renvoyer 0

LC(5) #2D00 intervalle = [45,0]
GOSBVL RANGE colonne dans l'intervalle?
GOC DD2 non: voir Driver 2
A=A+A A 1 colonne = 2 quartets
LC(5) DD1ST que l'on ajoute à
* l'adresse de la colonne 1
A=A+C A pour obtenir l'adresse de
* la colonne en A[A]
GONC Adr renvoie le résultat
LC(5) #5D2E` intervalle = [93,46]
D=C B D[C] = 46
GOSBVL RANGE dans l'intervalle ?
GOC DD3 non: voir Driver 3
C=D B
A=A-C A adresse relative de la
* colonne dans le Driver 2
A=A+A A quartets -> octets
LC(5) DD2ST
A=A+C A A[A] = adresse absolue de
* la colonne
GONC Adr que l'on renvoie
DD3 LCHEX 5E 5E = 94
A=A-C B A[B] = adresse relative
* de la colonne dans le
* Driver 3
A=A+A A quartets -> octets
LC(5) DD3ST
A=A+C A A[A] = adresse absolue de
* colonne
Adr DAT1=A A on renvoie A[A]
RTNCC Retour avec Carry Clear
Null A=0 A Argument non valide: on
* renvoie 0
GOC Adr
END

```

René OUVRAY,
50 rue de Courbevoie,
92000 NANTERRE,
Tel.: (1) 721 19 93

Nanterre, le 6 mai 1985

Hewlett-Packard,
Division Informatique Personnelle
S.A.V. Calculateurs de Poche
F - 91947 Les Ulis cedex.

Objet: Anomalie constatée sur des fonctions du
HP71B. ((Fonctions CALENDRIER - SETDATE - DATE -
DATE\$ -)).

Messieurs,

Faisant suite à une conversation
téléphonique du 3 mai dernier au No 907 78 25 au
cours de laquelle un technicien de vos services
(dont j'ignore les coordonnées) a été mis à même
de me confirmer le bien fondé d'une anomalie que
j'ai constaté sur le fonctionnement du HP71B ...;
je vous adresse ci-joint, une analyse détaillée
de ce "hiatus" ainsi que le remède que je propose
en l'absence de tout autre que vous pourriez me
suggérer.

Je vous signale qu'étant un usager
enthousiaste des produits HP, je n'ai communiqué
à personne d'autre, la découverte de cet incident
dont l'importance pratique est d'ailleurs assez
limitée, bien que regrettable.

Veillez agréer, Messieurs, l'assurance de
mes sentiments les meilleurs.

HP71B - Fonctions SETDATE - DATE - DATE\$

Erreurs concernant un ensemble de résultats

relatifs à un sous-groupe d'années bissextiles.

1.- Rappel des principes fondamentaux.

SETDATE: Qui opère sous le contrôle de l'horloge
du système, permet l'introduction d'une date
quelconque sous l'une des deux formes suivantes:
"AA/MM/JJ" en style alphanumérique: ("85/05/05"),
ou: AAJJJ en style numérique: (85125).

DATE: Affiche le jour réel en fonction du jour
introduit à l'origine et actualisé par l'horloge.
(125ème jour de l'an 85).

DATE\$: Affiche ce même jour sous la forme
AA/MM/JJ (sans "...")

La séquence: a) SETDATE(...) ... ENDLINE
permet d'introduire une date quelconque dont la
validité relative est contrôlée par l'indicateur
binaire IVL(-8) qui provoque le message d'erreur
Nr11 "Invalid Arg", si l'argument n'est pas
représentatif d'une date "plausible"; notamment,
29 février pour une année non bissextile... AA366
pour un 31 décembre d'une année non bissextile.
b) DATE ou DATE\$ affiche et rend disponible sous
la forme numérique ou alphanumérique, l'argument
en cours dans l'horloge.

...d'où l'intérêt d'éviter travailler sur des
dates aux environs de minuit... Le risque de
fausser les résultats attendus du fait de
l'horloge...

EN PRINCIPE: l'horloge du HP71B est sensée
tenir compte des années bissextiles.

2.- Vérification et contrôle du fonctionnement du
système.

Le traquenard de l'an 1900 (non bissextile) est
respecté: (à la condition de l'introduire sous la
forme: "1900/12/31" et non pas: "00/12/31" qui
est correctement traitée pour l'an 2000
(bissextile).

Résultats: 1900/02/29 = DATE Invalid Arg.
1900/02/29 = DATE\$ Invalid Arg.
1900/12/31 = DATE 365

1900/12/31 = DATE\$ 1900/12/31
 2000 ou 00/02/29 = DATE 60
 2000 ou 00/02/29 = DATE\$ 2000/12/31
 00/12/31 = DATE 366
 00/12/31 = DATE\$ 00/12/31

Les années AA= 00; 04; 08; 20; 24; 28; 40; 44;
 48; 60; 64; 68; 80; 84; 88. Sont traitées
 correctement.

Exemples: 44/02/29 = DATE 44060
 44/02/29 = DATE\$ 44/02/29
 Bissextils 44/12/31 = DATE 44366
 " 44/12/31 = DATE\$ 44/12/31
 non 53/02/29 = DATE Invalid Arg.
 " 53/02/29 = DATE\$ Invalid Arg.
 Bissextils 53/12/31 = DATE 53365
 " 53/12/31 = DATE\$ 53/12/31
 " 53/02/28 = DATE 53059
 " 53/02/28 = DATE\$ 53/02/28

3.- Détection de la divergence de principe sur
 l'ensemble des années bissextiles: 12 - 16; 32 -
 36; 52 - 56; 72 - 76; 92 -96. Les années de ce
 sous-ensemble sont traitées comme non bissextiles.

- a) Elles ne provoquent pas le message "Invalid
 Arg." pour le 29/02.
- b) Le 29/02 et le 01/03 donnent la même valeur:
 60 jours ((31+29)).
 32/02/29 = 32/03/01 = 60 c/ 61 pour le 1er mars,
 32/12/31 = 365 c/ 366 réel.
- c) Tous les jours intermédiaires sont minorés
 d'une unité.

4.- CONCLUSION: Si "AA" est divisible par 4 et
 non par 2 et si "MM" est plus grand que 2, JJJ
 réel est égal à "..JJJ" (HP71) +1, soit j j +1.

pourraient choquer les âmes les plus fragiles.
 Les plus endurant devront tout de même s'armer
 d'un bon tube d'aspirine et de calmants...
 Je prends donc de ma main droite ma plume qui se
 trouve à vingt centimètres de ma chaise pour
 étaler un peu d'encre sur un papier de 165
 centimètres carrés de manière à vous faire
 comprendre la honte et la déprime qui me tenaille.
 J'espère que mon écriture tremblante ne se fera
 pas trop ressentir sur le listing que vous
 parcourrez de vos yeux compatissant.
 Ce n'est pas sans une certaine gêne que je doit
 vous avouer tous mes remords, les yeux fixés sur
 mes orteils et la main gauche me grattant
 machinalement la nuque, au sujet d'une erreur de
 ma part. (sourir!)"
 J'ai, moi, Serge Vaudenay, fait une erreur de
 programmation. (sanglot!)
 Je vous permets de me mépriser, de me maudir, de
 m'envoyer vos insultes (que vous adresserez au
 journal, bien entendu), je suis impardonnable.
 Le temps de me moucher bruillament et de jeter
 mon mouchoir liquéfié à la poubelle, et
 j'explique mon erreur, ce qui me soulagera sans
 doute. (snif!)
 Les pauvres compositeurs en herbe que vous êtes
 auront remarqué dans le JPC numéro 22 à la page
 51 une erreur monstrueuse qui saute aux yeux du
 premier imbécile venu: à la ligne 280, j'ai
 oublié de multiplier le T du STR\$(T) par T0
 (STR\$(T*T0)), ce qui eu pour conséquence de faire
 des silences monstrueux. (ouff!)
 Cet article fut donc un événement à deux titres:
 il traite d'un programme musical sur 71, et
 comprend une erreur de ma part.
 Je réclame donc péniblement votre clémence, et
 vous donne rendez-vous au prochain article, si je
 ne me suis pas fait hara-kiri entre temps.
 Après tout, errare humanum est !

Serge VAUDENAY

RECORDS

POUR LA RUBRIQUE DEBUTANTS

Attention: certains passages de cet article

Voici un tout petit programme destiné à afficher l'heure. Pour les chevronnés du HP71 ce programme contient une erreur ! Donc cet exercice offre un double intérêt : une initiation au magnifique BASIC HP et un test pour les vétérans.

```

5 DELAY 0,0
10 ON TIMER #1,1 GOSUB 30
20 A$=KEY$ @ IF A$="" THEN 20 ELSE PUT A$ @ END
30 DISP CHR$(27)&CHR$(13)&TIME$[1,2]&" h "
    &TIME$[4,5]&" mn "&TIME$[7]&" s" @ RETURN

```

5 DELAY 0,0 Cela permet de ne pas provoquer de "ralenti" à l'affichage. Lire DELAY dans le manuel de référence.

10 ON TIMER #1,1 Il existe 3 compteurs qui permettent d'interrompre un programme à intervalles régulier. La valeur qui suit la virgule indique le nombre de seconde de l'intervalle. Essayer ON TIMER #1,10.

GOSUB 30 Toutes les secondes le programme se branche à la ligne 30 et effectue les instructions jusqu'au RETURN.

20 A\$=KEY\$ A\$ est une variable destinée à recevoir du texte. C'est le signe \$ (dollar) qui est la marque d'une variable alpha. KEY\$ contient une "valeur" différente de "" (chaîne vide) lorsque l'on vient d'appuyer sur une touche. La valeur est alors celle correspondant à la touche. Voir le programme 'TOUCHE' plus loin.

@ IF A\$="" @ (arobasque) sert à mettre plusieurs instructions sur une même ligne.

THEN 20

ELSE PUT A\$

@ END

30 DISP

CHR\$(27)

CHR\$(27)&CHR\$(13)

If xxx THEN yyy ELSE zzz si l'expression xxx est vraie alors exécuter l'ordre yyy sinon exécuter zzz.

A\$="": A\$ est une chaîne vide = on a pas appuyé sur une touche (KEY) c.f. plus haut.

alors (aller à la ligne) 20 ; l'ordre de branchement est implicite.

Sinon PUT (placer) A\$ dans le tampon de clavier. Du fait que le programme tourne, seule la touche [ON] est active.

Cependant le programme interne de l'ordinateur continue à scruter le clavier, mais les touches ne sont pas exécutées sauf si le programme BASIC en donne l'ordre explicitement comme ici.

END = fin du programme. Il peut cependant comporter encore des lignes. Il peut y avoir plusieurs END.

Ordre d'afficher ce qui suit.

lire C H R \$ de 27. Dans le manuel de référence P 323 et suivantes se trouve la liste des caractères et de leur code. Le caractère de code 27 est particulier: on le nomme caractère d'échappement. Il commence une séquence spéciale dite d'échappement destinée à envoyer un ordre à un périphérique. (affichage, imprimante ...)

Page 235 du manuel de l'utilisateur sont décrites les séquences d'échappement pour l'écran (affichage). A

partir du clavier on peut soit effectuer [g] [CTRL] [g] [[] [H] ou DISP CHR\$(27)& "H" ; dans un programme seule la 2ème solution est possible. & lire "et" c'est un "et commercial". Il indique que la chaîne se poursuit. Eh! oui CHR\$(nn) est une chaîne de caractères (un seul : Ec). Ec 13 (c.F.p 236) replace le curseur à gauche de l'écran ce qui va provoquer l'écriture de la suite par dessus (en écrasant) ce qui est déjà à l'écran. En l'occurrence l'heure passée.

TIME\$ exécuté au clavier: donne l'heure du moment sur 8 caractères. 6 pour les chiffres et deux pour les séparateurs ':'. Il s'agit encore d'une chaîne alpha (\$).

TIME\$(1,2) prélève les 2 premiers caractères de cette chaîne & "et commercial" ! " h " continue la chaîne avec "espace,h,espace" etc... @ RETURN fin du sous-programme.

J'espère ne pas être trop confus !
EDIT TOUCHES

```
10 A$=KEY$ @ IF A$="" THEN 10
20 IF A$="#43" THEN END
30 DISP A$ @ GOTO 10
```

Pour les débutants regardez de près le traitement des chaînes de caractères: d'une part les manuels ne sont pas très locaces sur ce sujet, et d'autre part nous sommes loin des fonctions LEFT\$, RIGHT\$ etc. Il faut quelques minutes pour saisir le mécanisme, après quoi vous vous demanderez pourquoi cette syntaxe n'est pas implantée sur

tous les BASIC.

Pour les vétérans la réponse n'est ni DELAY,0 ni l'inutilité du RETURN. Il s'agit d'une erreur qui provoque un disfonctionnement dans un cas particulier !

Nous attendons votre courrier.

J.J DHENIN #177 (SIG#5)

Master-Mind amélioré.

Nouveau au Club PPC-PARIS, j'ignore si quelqu'un a déjà pensé à un programme de Master-Mind. Ci-joint ma version de ce jeu connu. L'originalité vient du fait que ce programme propose des combinaisons de lettres ou de chiffres; un même élément pouvant être généré plusieurs fois dans une combinaison. Les instructions ne présentent rien de particulier, rien qui ne soit pas utilisé dans les programmes de JPC No22 ou 23.

884 pas de programme.

Après [RUN], le HP demande la taille de la combinaison, 3 à 9 caractères: "Array (3-9)". Une pression sur [ENDLINE] produit une combinaison de 5 caractères.

Il est demandé ensuite le type de ces caractères. N: numériques (chiffres de 0 à 9); entrer 1 [Drapeau 1].

N: alphanumériques (lettres de A à Z); entrer 2 [Drapeau 2].

N+N: Lettres et chiffres à la fois; entrer 3 [Drapeaux 1 et 2].

Les drapeaux sont là pour rappeler le type de combinaison présentée par le HP71.

Le jeu:

L'affichage:

```
..... 0 & 0 1
  ^     ^   ^ ^ ^
  |     | | | No du coup joué
  |     | | [g] [CTRL] [G]
  |     | | Nb de chiffres mal placés
  |     | | Nb de chiffres bien placés
Combinaison précédemment entrée (remplacée par
des [.] au premier coup).
```

On entre la combinaison (majuscules ou minuscules indifféremment), puis [ENDLINE]. Si l'entrée excède en taille (la taille choisie au début du programme), le HP sonne et tronque à partir de la droite).

A l'affichage, une fois la combinaison trouvée: "Game Over (Y/N)"

Vous avez 5 secondes pour appuyer sur Y ou N. Toute pression sur une touche autre que [N] est prise pour [Y]. Une pression sur [N] initialise une autre combinaison.

Ce programme utilisant WINDOW, voici un "truc" qui permet d'annuler les effets de cet ordre:

```
Faire WINDOW 1,1 [ENDLINE]
Passer en mode CALC: [f] [CALC]
puis revenir en Basic: [f] [CALC]
L'écran a retrouvé ses 22 colonnes...
```

D'autre part, pour ceux qui veulent tricher, R\$ contient la combinaison à découvrir...

Remarque: le principe des tests entre l'essai du joueur et la combinaison de la machine est basée sur la reconnaissance, puis l'élimination (dans l'essai, le caractère est remplacé par \$, dans la combinaison de référence par +), d'où la nécessité de deux variables auxiliaires W\$ et T\$ pour ces tests.

Amicalement,

Xavier BILLE

DAD (Suite...)

Lors de la préparation du Stand du Sicob, un des objectifs était de présenter des programmes de démonstration attrayants, et "implantables". Mon but n'est pas aujourd'hui de vous les révéler. Ils ne sont pas tous très passionnants à étudier. Néanmoins, il en est un dont nous reparlerons certainement...

Juste un mot quand même sur la manière dont ces programmes fonctionnaient: ils consistaient à afficher des pages de texte sur la vidéo (sous une forme plus ou moins perfectionnée).

Seul problème: l'afficheur restait vide. Il fallait alors animer nos LCD chéris. Un des programmes faisait défiler à l'écran un TGV, dessiné par Eric GENGOUX. Je me suis inspiré de son TGV pour faire un petit train rétro.

Seulement voilà: aligner une suite de codes Hexadécimaux n'est pas pour me plaire. On est feignant, ou on ne l'est pas !

Alors, je me suis fabriqué un petit éditeur de dessin, afin de faire ce train dans les meilleures conditions possibles.

Brièvement, je vous expose ses caractéristiques: Introduction d'un dessin à partir d'un fichier, et recopie d'un dessin dans un fichier.

Déplacement du curseur en mode trace, ou gomme. Clignotement du curseur.

Effacement ou insertion d'une série de colonnes. Positionnement du curseur à partir de ses coordonnées (X,Y).

Forçage d'un point à 1 ou à 0.

De plus, une fonction assembleur permet d'accélérer notablement l'exécution. Après vous avoir alléché avec ses caractéristiques, nous allons voir son fonctionnement...

Tout d'abord, après avoir fait [RUN], le programme vous demande le nom d'un fichier:

"G-fichier". Vous répondez par [ENDLINE] si vous

n'en avez pas.

Ensuite, vous avez le curseur clignotant. Essayez donc les touches de curseur.

Maintenant, appuyez sur [/]. L'indicateur [I] s'allume, pour vous signaler que vous êtes en mode "gomme". Déplacez vous avec les touches de curseur, et admirez.

Si vous voulez sauvegarder ce dessin dans un fichier, faites [E]. Le programme vous demande alors dans quel fichier vous désirez mettre votre oeuvre d'art.

Si vous faites [ENDLINE], vous pourrez déplacer directement le curseur à une adresse (X,Y) donnée. [I] inverse l'affichage.

[I] ou [O] force le point à l'état allumé ou éteint, indépendamment du mode choisi.

[-] vous autorise à détruire un certain nombre de colonnes à partir de la position du curseur. Si vous n'entrez rien, et faites [ENDLINE], aucune colonne n'est détruite.

[+] insère des colonnes à partir de la position du curseur.

Voilà pour l'utilisation. Le programme lui-même n'est pas un modèle de clarté. Les opérations sur les fichiers ne sont pas protégées contre les fausses manœuvres.

Voilà, dessinez bien...

A Bientôt,
Pierre DAVID

~~XXXXXXXXXXXXXXXXXXXX~~

HANTISE SUR SATURNE

Amis spectrivores, bonjour.

: Ne passez surtout pas à l'article suivant, vous
: risqueriez, comme les mois précédents d'ailleurs,
: de rater le programme du siècle.

: Après cette arrivée en fanfare, entretenons un
: peu le suspens. Avouez que vous vous demandez
: bien ce qui va vous tomber sur le tête après le
: compilateur musical et le débilomètre. Non, ce
: n'est pas un Visicalc, ni même un Word star, tout
: ceci ferait bien piètre figure face à ce qui vous
: attend...

: Quelle géniale idée a encore germé de mon non
: moins génial cerveau ?

: Vous êtes bien assis?

: Tenez-vous bien, c'est un jeu!!!

: A cheval entre le mastermind et le Hi-Lo, tout le
: monde peut y jouer, que ce soit ceux qui n'ont
: pas compris comment répondre à la première
: question du débilomètre, ou ceux qui ont fait un
: INIT 1 à la deuxième, agacés.

: Le tarif de ce mois-ci sera de 9,5Ko.

: Ne vous découragez pas trop vite: sur ces 9,5K,
: plus de 5K se rentreront automatiquement (je le
: précise pour ceux à qui le débilomètre a laissé
: un mauvais souvenir).

: Je m'expliquerai plus clairement plus tard, car
: il faut que je parle du jeu en question, sinon
: ceux qui ne seront pas tombés en syncope
: passeront à l'article suivant. De plus, je ne
: tiens pas à me faire égorger par Pierre David,
: car je devine que vous êtes déjà assez nombreux à
: adresser vos réclamations au Journal.

: Imaginez-vous donc chasseur de fantômes dans un
: labyrinthe hanté...

: Encore un Pac-man, me direz-vous? Non!

: Tout d'abord, vous n'avez pas de calories à
: avaler, mais des fantômes à désintégrer avant
: qu'ils ne vous tuent, et ensuite, le labyrinthe
: dans lequel vous vous déplacez vous est visualisé
: en perspective...

: C'est comme si vous y étiez.

: Après vous avoir mis l'eau à la bouche, voici les
: explications: le programme principal "FANTOME"

qui pèse 3K nécessite un fichier TEXT "DISPLAY" qui pèse 5K. Afin de créer ce fichier TEXT dont le listing est horrible à voir, j'ai fait un programme "CRDIS" de 887 octets qui se charge de tout. Entrez donc CRDIS, lancez-le, et vous aurez, au bout de 6 minutes votre fichier DISPLAY. Vous pourrez alors vous amuser avec FANTOME.

Appuyez donc sur la fatale touche RUN, et entrons dans ce labyrinthe maudit.

Le programme commence à vous rassurer et à vous décontracter en vous montrant des fantômes ricanant.

Quelques secondes plus tard, vous voyez le couloir dans lequel vous êtes tombé. Ce couloir peut avoir des portes à gauche ou à droite. Je vous laisse donc comprendre l'affichage tout seul.

Pour le clavier, commençons par la touche la plus inutile: ATTN qui sert à abandonner.

La touche END LINE sert à actionner le pistolaser à rayon cosmique et à variateur bêta que j'ai eu la gentillesse de vous donner. En général, les fantômes n'y résistent pas.

La touche M vous servira beaucoup, j'en suis sûr, pour consulter le plan du labyrinthe (vous y serez représenté par un point clignotant).

Pour vous déplacer, vous utiliserez la flèche vers le haut pour avancer, et vers le bas pour reculer (sans vous retourner, attention). Ce n'est pas grave si vous vous cognez contre les murs, le programme est assez tolérant. Vous pourrez évidemment pivoter sur vous-même de 90 degrés à droite ou à gauche en utilisant respectivement les flèches à droite et à gauche.

Les autres touches ne servent à rien.

Voilà pour ce qui est du clavier.

Vous aurez droit à une pression de touche par cycle, un cycle représentant un déplacement d'une case de la part du fantôme. Vous appuyerez sur la touche de votre choix après l'affichage du temps, pendant une pause, pour le programme.

Passons maintenant au plus intéressant: le fantôme...

Vos outils de chasseurs de fantômes sont un plan du labyrinthe, un pistolaser à rayon cosmique et

à variateur bêta qui fait bip quand on appuie dessus si le cran de sûreté est débloqué et qui désintègre les molécules de fantôme qui peuvent malencontreusement se trouver dans votre champ de vision, et un radar détecteur de fantôme à deux tonalités: plus vous êtes près d'un fantôme, plus le 'blip' qu'il émet est aigu.

Si vous ne l'entendez pas, rassurez-vous: vous n'avez pas grand chose à craindre du fantôme là où vous êtes.

Celui-ci se représentera comme une barre verticale au milieu de votre champ de vision lorsqu'il sera devant vous.

Lorsqu'il sera derrière vous, je ne donnerai pas chère de votre peau...

Bien évidemment, il peut surgir par surprise au coin d'un croisement ou d'un virage, ou pendant que vous consultez votre plan, ou vous courir derrière (rassurez-vous: il est plus rapide que vous!)

Si vous réussissez à lui tirer dessus, vous le verrez se désintégrer, mais un autre fantôme viendra le venger...

Cependant, si le fantôme arrive à vous toucher, ce qui est beaucoup plus probable, il absorbera tout ce qu'il y a de vivant en vous et vous vous anihilerez: votre vue se brouillera, et il ne restera plus que votre score pour vous consoler.

Le labyrinthe se trouve codé en binaire dans les DATA de la ligne 90 à la ligne 310.

Vous pourrez bien entendu le modifier à condition de mettre ses dimensions à la ligne 20 (X9=23 @ Y9=17).

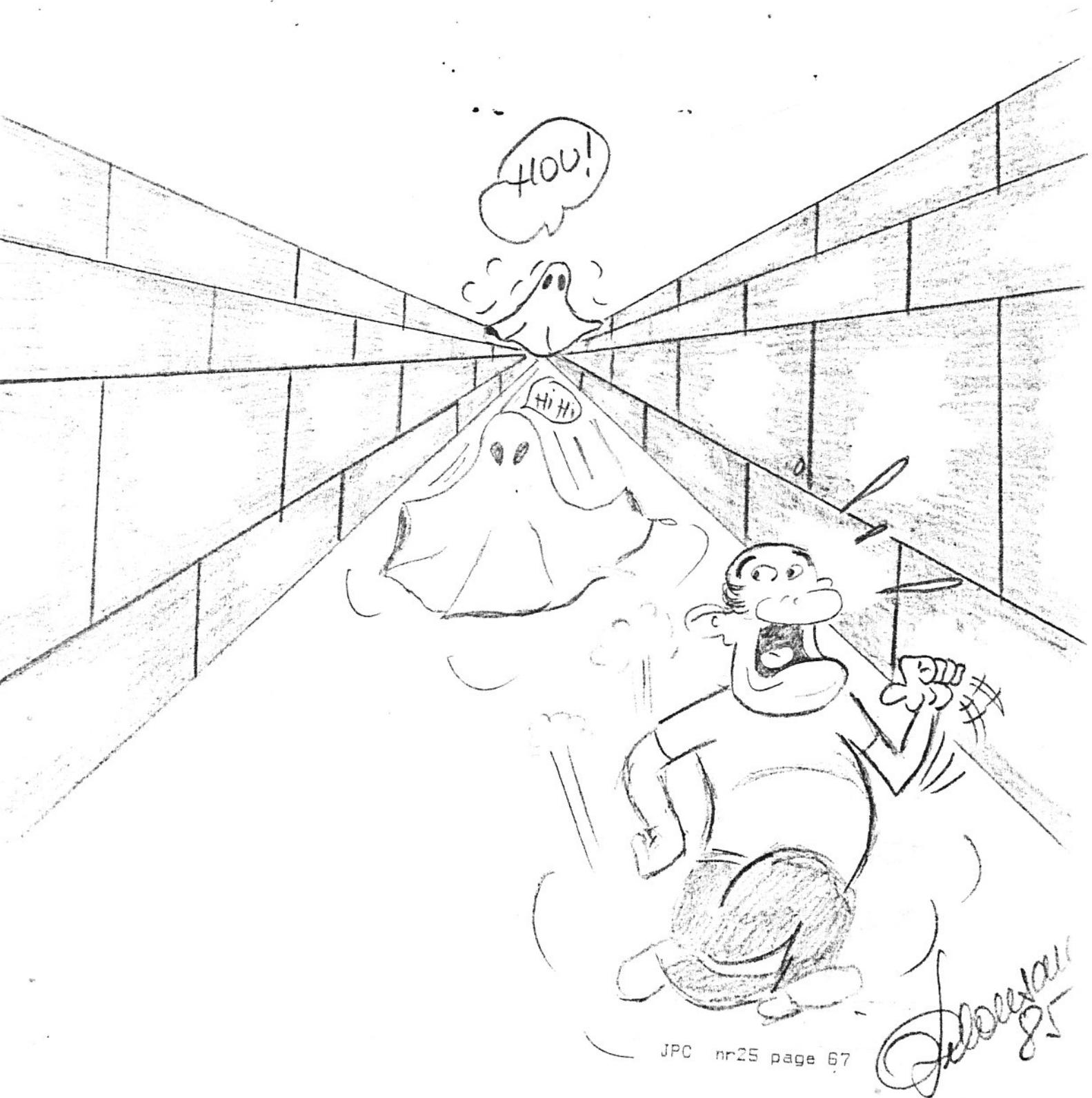
Le programme utilise le LEX POINTSET (140 octets) par sa fonction PSET. Je ne sais pas qui a écrit ce LEX, je sais simplement qu'il s'est retrouvé dans mon 71 à la fin d'une réunion du Club. Si l'auteur ne s'est pas décidé à le faire paraître dans le journal, et si vous n'avez pas ce LEX, vous pourrez remplacer les DISP PSET(x,y) par des CALL PSET(x,y) car j'ai écrit un SUB équivalent en BASIC. C'est plus lent, mais ça marche.

Voilà, je ne vous ennuierais pas plus longtemps avec mes élucubrations mensuelles, je vous demanderai simplement de ne pas trop maltraiter mes fantômes et je vous dirai au mois prochain.

Serge VAUDENAY.

Note de Pierre David:

- 1- Je n'ai pas l'habitude d'égorger les gens.
- 2- POINTSET est de Jean-Jacques MOREAU. Il est remplacé par PAINTLEX, paru le mois dernier.
- 3- Ce programme est génial.



"MASTERP" (Programme de Master-Mind amélioré)

- Initialisation

Entrée de la longueur de la combinaison
A contient cette longueur

10 INPUT "Array (3-9) ?","5";A @ IF A<3 OR A>9 THEN BEEP 1500,.1 @ GOTO 10

- Choix du type de combinaison. C est ce type

20 DISP "N(1), "&CHR\$(4)&"N(2), "&CHR\$(4)&"N(3)"; @ INPUT "", "1";C @ C=MOD(C-1,3)+1

- Initialisation des constantes suivant C

30 ON C GOTO 40,50,60

- U>V: bornes entre lesquelles CHR\$(Z) ne représente pas un chiffre ou une lettre (code ASCII)

X,Y: bornes ASCII telles que CHR\$(Z) représente chiffre ou lettre.

40 SFLAG 1 @ X=10 @ Y=48 @ U=INF @ V=-INF @ GOTO 'GAME'

50 SFLAG 2 @ X=26 @ Y=65 @ U=INF @ V=-INF @ GOTO 'GAME'

60 SFLAG 1,2 @ X=43 @ Y=48 @ U=57 @ V=65

- Combinaisons:

F: compte les chiffres bien placés

G: compte les chiffres mal placés

I\$: Introduction de la combinaison du joueur

R\$: combinaison source/référence

T\$(=I\$) et W\$(=R\$) sont des variables modifiables pour les tests

S: nombre d'essais

=====

70 'GAME': DISP "A second..." @ DESTROY R\$,I\$,S,F,G @ RANDOMIZE @ FOR J=1 TO A

- J: indice de boucle

Z: variable aléatoire

80 Z=IP(RND*X)+Y @ IF Z>U AND Z<V THEN 80

- Vérification de la bonne "fourchette"

(Choix 3: éviter les codes ASCII 57 à 64)

90 R\$=R\$&CHR\$(Z) @ I\$=I\$&"." @ NEXT J

100 DISP "Ok ! Try and guess:" @ WAIT 2

- Jeu: test et intro

110 S=S+1 @ WINDOW 1 @ DISP TAB(17); @ DISP USING 'D,"%",D,B,2D';F,G,7,S @ WINDOW 1,16

- Affichage des résultats (premier coup: I\$ est remplacé par A points).

Introduction de l'Essai

120 INPUT "",I\$;I\$ @ IF LEN(I\$)>A THEN BEEP 1500,.1 @ I\$=I\$[1,A]

130 W\$=R\$ @ T\$=UPRC\$(I\$) @ DESTROY F,G

140 FOR J=1 TO A @ IF T\$[J,J]=W\$[J,J] THEN F=F+1 @ W\$[J,J]="+" @ T\$[J,J]="*"

- Y-a-t-il un chiffre à la bonne place ?

150 NEXT J @ IF F=A THEN 'WIN'

160 FOR J=1 TO A @ P=POS(W\$,T\$[J,J])

- Y-a-t-il un chiffre existant, mais mal placé ?

170 IF P THEN G=G+1 @ T\$[J,J]="*" @ W\$[P,P]="+"

180 NEXT J @ GOTO 110

- En cas de victoire...

=====

190 'WIN': BEEP @ WINDOW 1 @ DISP "Correct in";S;"attempts."

200 WAIT 1 @ DISP "Game Over (Y/N)?" @ ON TIMER #1,5 GOTO 220

210 K\$=KEY\$ @ IF K\$="" THEN 210

- Le jeu est-il fini ? Sinon retour.

220 OFF TIMER #1 @ IF UPRC\$(K\$)="N" THEN 'GAME' ELSE CFLAG ALL @ END


```

140 DATA 1,0,0,1,0,1,0,1,1,0,0,1,1,0,1,0,1
150 DATA 1,0,1,1,0,1,0,0,1,1,0,0,1,0,1,0,1
160 DATA 1,0,0,0,0,0,0,0,0,1,1,0,0,0,1,0,1
170 DATA 1,0,1,1,1,1,1,1,0,0,1,1,1,1,1,0,1
180 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
190 DATA 1,0,1,1,1,1,1,1,0,1,1,1,1,1,1,0,1
200 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
210 DATA 1,0,1,0,1,1,0,1,0,1,1,0,1,1,1,0,1
220 DATA 1,0,1,0,0,0,0,1,0,1,1,0,1,0,0,0,1
230 DATA 1,0,1,1,1,1,1,1,0,0,0,0,1,0,1,0,1
240 DATA 1,0,1,0,0,0,0,1,0,1,1,0,1,0,1,0,1
250 DATA 1,0,1,0,1,1,0,1,0,1,1,0,1,0,1,0,1
260 DATA 1,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,1
270 DATA 1,0,1,1,1,1,1,1,0,1,1,1,1,1,1,1,1
280 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
290 DATA 1,0,1,1,1,1,1,1,1,1,1,1,1,1,1,0,1
300 DATA 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1
310 DATA 1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1

```

```

=====
320 ASSIGN #1 TO DISPLAY
330 T1=0 @ X=2 @ Y=2 @ A=0 @ T=TIME @ WINDOW 11
340 X5=IP(RND*(X9-2)+2) @ Y5=IP(RND*(Y9-2)+2) @ IF (X5-X)^2+(Y5-Y)^2<16 OR L(X5,Y5) THEN 340

```

```

=====
350 'MASTER': CALL DSP(L(,),X,Y,A,A$,#1,G)
360 IF X=X5 AND Y=Y5 THEN 600
370 B=X5-SGN(X5-X) @ C=Y5-SGN(Y5-Y)
380 IF RND*2>1 AND L(B,C)=0 THEN X5=B @ Y5=C @ GOTO 400
390 IF L(X5,C)=0 AND C#Y5 THEN Y5=C ELSE IF L(B,Y5)=0 THEN X5=B
400 IF X=X5 AND Y=Y5 THEN 600
410 D=(X5-X)^2+(Y5-Y)^2
420 GDISP A$
430 F=SGN(X5-X)=COS(A) AND SGN(Y5-Y)=SIN(A) AND D<17 AND 6^2=D @ IF F THEN GOSUB 'PHAN'
440 DISP STR$(IP((TIME-T)/60))&" "&T1$ @ BEEP 50*((D<9)+(D<25)),.05
450 K=0
460 K$=KEY$ @ K=K+1 @ IF K<40 AND K$=' ' THEN 460
470 ON POS(' #47#48#50#51#38#43M',UPRC$(K$))/3+1 GOTO 'MASTER',540,550,560,570,580,650,480
480 C=1 @ IF Y>7 THEN C=Y-4
490 B=1 @ IF X>120 THEN B=X-66
500 CALL DISLAB(L(,),B,C,X9)
510 A$=GDISP$ @ DISP PSET(X-B,Y-C) @ A$[133]=GDISP$
520 FOR I=1 TO 200 @ GDISP A$[133*MOD(I,2)] @ NEXT I
530 GOTO 'MASTER'
540 A=A-90 @ GOTO 'MASTER'
550 A=A+90 @ GOTO 'MASTER'
560 B=X+IP(COS(A)) @ C=Y+IP(SIN(A)) @ IF L(B,C) THEN 'MASTER' ELSE X=B @ Y=C @ GOTO 'MASTER'
570 B=X-IP(COS(A)) @ C=Y-IP(SIN(A)) @ IF L(B,C) THEN 'MASTER' ELSE X=B @ Y=C @ GOTO 'MASTER'
580 BEEP 5000,.1 @ BEEP 100,.05 @ IF NOT F THEN 'MASTER'
590 CALL OK @ T1=T1+1 @ T1$=F1$[0,IP(T1/5)]&F2$[0,MOD(T1,5)] @ GOTO 340
600 T=TIME-T @ BEEP 25,1
610 CALL LOST
620 WINDOW 1 @ DISP STR$(T1)&' fant'&CHR$(130)&'me'; @ IF T1>1 THEN DISP 's';
630 DISP ' en '&STR$(IP(T/60))&" "&STR$(100+IP(MOD(T,60)))[2]&" " @ DESTROY ALL
640 POKE '2F441','0' @ END
650 T=TIME-T @ GOTO 620

```

```

=====
660 'PHAN': A$=DTH$(2^(10-2*SQR(D))-1) @ A$=A$[5]&A$[4,4] @ POKE '2E124',A$&A$ @ RETURN

```

```

=====
670 SUB DSP(L(,),X,Y,A,E$,#1,6)
680 D0=IP(COS(A)) @ D1=IP(SIN(A))
690 X1=X-IP(SIN(A)) @ Y1=Y+IP(COS(A))
700 X2=X+IP(SIN(A)) @ Y2=Y-IP(COS(A))
710 D$=STR$(L(X,Y)=1)&'11111111' @ IF L(X,Y) THEN 780
720 FOR I=0 TO 2
730 D#[5+I,5+I]=STR$(L(X1+I*D0,Y1+I*D1)=1)
740 D#[8+I,8+I]=STR$(L(X2+I*D0,Y2+I*D1)=1)
750 F=L(X+(I+1)*D0,Y+(I+1)*D1)=1
760 D#[I+2,I+2]=STR$(F) @ IF F THEN I=3
770 NEXT I
780 ON 1+VAL(D#[3,3])+VAL(D#[2,2])+VAL(D#[1,1]) GOTO 790,810,820,830
790 B=VAL(D#[10])+2*VAL(D#[9,9])+4*VAL(D#[8,8])+8*VAL(D#[7,7])+16*VAL(D#[6,6])+32*VAL(D#[5,5])
800 B=B+64*VAL(D#[4,4]) @ G=4 @ GOTO 840
810 B=VAL(D#[9,9])+2*VAL(D#[8,8])+4*VAL(D#[6,6])+8*VAL(D#[5,5])+128 @ G=2 @ GOTO 840
820 B=VAL(D#[8,8])+2*VAL(D#[5,5])+144 @ G=1 @ GOTO 840
830 B=148 @ G=0
840 RESTORE #1,B @ READ #1;E$

```

```

=====
850 SUB DISLAB(L(,),A,B,X9)
860 GDISP '' @ FOR X=A TO MIN(A+131,X9) @ FOR Y=B TO B+7
870 IF L(X,Y) THEN DISP PSET(X-A,Y-B)
880 NEXT Y @ NEXT X
890 END SUB

```

```

=====
900 SUB LOST
910 DIM I(8) @ READ I()
920 R=RND @ DATA 1,5,0,7,6,3,2,4
930 FOR A=1 TO 8 @ RANDOMIZE R @ FOR X=0 TO 33 @ DISP PSET(X,MOD(IP(RND*8)+I(A),8))
940 NEXT X @ NEXT A

```

```

=====
950 SUB OK
960 FOR I=5000 TO 1000 STEP -1000 @ READ A @ POKE '2E124',DTH$(A)[2] @ BEEP I,.05 @ NEXT I

```

```

=====
970 DATA 65535,30702,21930,4488,0

```

```

=====
980 BEEP 100,.5

```

```

=====
990 SUB PSET(X,Y)
1000 A=NUM(GDISP#[X+1]) @ A=2^Y+MOD(A,2^Y)+2^(Y+1)*IP(A/2^(Y+1))
1010 GDISP GDISP#[0,X]&CHR$(A)&GDISP#[X+2]

```

```

*****
"CRDIS" (A utiliser avec "FANTOME"; voir article)

```

```

10 CALL CRF

```

```

=====
20 SUB TOD(D$,E$) @ DELAY 0,0

```

```
=====
30 DEF FNF(D$,A)=VAL(D$[A,A])
=====
```

```
40 DEF FNG$(A)=CHR$(A)&CHR$(A)
=====
```

```
50 B=FNF(D$,1) @ C=FNF(D$,2) @ D=FNF(D$,3) @ E=FNF(D$,4)
60 E$=FNG$(255)&FNG$(127*FNF(D$,8)+128*B)&FNG$(63+192*B)&FNG$(31*FNF(D$,9)+32*C+192*B)
70 E$=E$&FNG$(15+48*C+192*B)&FNG$(7*FNF(D$,10)+8*D+48*C+192*B)&FNG$(3+12*D+48*C+192*B)
80 E$=E$&FNG$(3*E+12*D+48*C+192*B)
90 E$=E$&E$[15]&E$[15]&E$[13,14]&FNG$(7*FNF(D$,7)+8*D+48*C+192*B)&E$[9,10]
100 E$=E$&FNG$(31*FNF(D$,6)+32*C+192*B)&E$[5,6]&FNG$(127*FNF(D$,5)+128*B)&FNG$(255)
=====
```

```
110 SUB DECBIN(D,B$)
120 B$='' @ C=D @ FOR A=1 TO B
130 C=C/2 @ IF FP(C) THEN B$='1'&B$ ELSE B$='0'&B$
140 C=IP(C) @ NEXT A
=====
```

```
150 SUB CRF @ CREATE TEXT DISPLAY
160 ASSIGN #1 TO DISPLAY @ DIM E$[34]
170 FOR A=0 TO 127
180 CALL DECBIN(A,D$) @ D$='00'&D$ @ GOSUB 300
190 DISP A @ NEXT A
200 FOR A=0 TO 15
210 CALL DECBIN(A,D$)
220 D$='0011'&D$[5,6]&'1'&D$[7]&'1' @ GOSUB 300
230 NEXT A
240 D$='0111011011' @ GOSUB 300
250 D$='0111011111' @ GOSUB 300
260 D$='0111111011' @ GOSUB 300
270 D$='0111111111' @ GOSUB 300
280 D$='1111111111' @ GOSUB 300
290 END
=====
```

```
300 CALL TOD(D$,E$)
310 PRINT #1;E$
320 RETURN
=====
```



HEWLETT-PACKARD FRANCE : S. A. au capital de 250.000.000 de Francs régie par les articles 118 A 150 de la loi sur les sociétés commerciales - R.C.S. Corbeil-Essonnes : B 709805030.

SIÈGE SOCIAL : Parc d'activité du Bols Briard - 2, Avenue du Lac - 91040 EVRY CEDEX - Tél. (6) 077.83.83 - Telex 692315 F

Dans le cadre de sa campagne générale BACK TO SCHOOL, HEWLETT PACKARD, pour aider ses partenaires, lance une grande promotion.

Pour chaque achat de HP 41 CV, le client recevra gratuitement un module mathématique ;

Pour chaque achat de HP 41 CX, le client recevra gratuitement un module mathématique et un module statistique.

Date : du 1er Juillet au 30 Septembre 1985
chez tous les distributeurs agréés Hewlett Packard.

DEPECHEZ-VOUS, LES QUANTITES SONT LIMITEES !...

DEMANDEZ VOTRE "HP 41 PROMOTION"

HEWLETT PACKARD ACTUALITE

Nouveau Produits :

Pour HP 71 : Module AMPI STATISTIQUE Réf. : 82489 A
Translator Pac Réf. : 82490 A
Datacom Réf. : 82488 A

Nouveaux Manuels :

- Enfin la HP 41 CV à son "manuel d'utilisation et fonctionnement en détail" en français :

Edition Janvier 1985
Réf. : 00D41 - 90533

- Pour la HP 71
5954 - 1278 : Manuel de Contrôle et d'acquisition de données par la HP 71 en anglais
Comprenant : 5954 - 1241 (voir cadre ci-dessous) ;

P/N	Title
5954-1241	HP-71 Instrument Control Systems brochure
5954-1268	HP-71 to HP 3000 and HP 1000 File Transfers
5954-1274	HP-71/HP Touchscreen and HP-71/HP 2392A Terminal Development Station
5954-1264D	HP-71 Control Price List

- Pour la HP 75
5954 - 1248 : Manuel traitant de la saisie de donnée électronique sur la productivité et fiabilité des données.
comprenant : 5954 - 1239 (voir cadre ci-dessous)

P/N	Title
5954-1239	HP-75D Data Collection Systems brochure
5954-1142	HP-75 to HP 3000 and HP 1000 File Transfers
5953-5657 Rev. A	HP-75D Bar Code Reading System
5954-1247	Field Service Application Description
5954-1234D	HP-75D Data Collection Price List

LE TOUT EST DISPONIBLE CHEZ VOTRE DISTRIBUTEUR AGREE HP.

Le journal J.P.C. est un bulletin de liaison entre les membres de l'association loi de 1901 PPC.P.C.. Le club est éditeur du J.P.C. et son siège est au 56 rue Jean-Jacques Rousseau - 75001 PARIS.

Directeur de la publication: Philippe GUEZ
Numéro d'ISSN: 0762 - 381X