

## A PROPOS DU CLUB

Le Bureau  
J. Belin  
J. Belin  
J. Belin

Editorial 1  
Compte-rendu de l'Assemblée Générale 2  
Les nouveaux statuts 4  
Nouvelles Coordonnées du Support technique 6  
Courrier du Coeur 6

## HP28

G. Toublanc

Modes Symbolique/Numérique (acte II) 8

## HP48

C. Nober  
R. Amram  
J.F. Garnier  
L.V Huynh / G.T. Toublanc  
G. Toublanc  
G. Toublanc

Annuaire téléphonique 10  
Conversion de caractères accentués 13  
Numéro de semaine (acte 3) 14  
Trucs, Astuces et Utilitaires 15  
Conversion de fichiers Xmodem (acte II) 16  
PEEK et la Rom Cachée 12

## HP95 / HP100

J. Belin  
J. Belin  
J. Belin

Une nouvelle carte NewMédia 26  
Tracé de fougère fractale 26  
Récupération des fichiers de base de données 28

Le coin des codes 30

Index des articles parus en 1993 33

## EDITORIAL

Comme vous le savez, nous avons tenu notre assemblée générale le 22 Janvier dernier. Voir le compte rendu page suivante. Mais le fait le plus important de cette AG est sûrement le vote à une écrasante majorité de la modification des statuts visant à inciter tous les adhérents à écrire au moins un article pour le club. Maintenant, il ne vous reste plus qu'à appliquer cette mesure...

C'est avec grand plaisir que nous voyons que de nouveaux auteurs n'ont pas attendu ce vote pour nous envoyer des articles. Ce qui est intéressant, c'est que pour la plupart, il s'agit de nouveaux adhérents. Pouvons nous espérer que les arrivants aient plus de civisme que beaucoup d'anciens ? Si oui, nous pouvons voir l'avenir du club en rose...

Comme pour les gros listings assembleur PC et en langage C publiés dans des numéros précédents, ce JPC comporte un source HP48 présenté en format vertical. Bien que "cassant" un peu l'homogénéité de la mise en page, ce format permet d'ajouter plus de commentaires sur ces listings. Dites nous ce que vous pensez de cette présentation. Votre avis nous est utile, surtout si nous réalisons notre projet de changer la mise en page de JPC à partir du numéro 100...

Comme le mois dernier, la rubrique HP28 est assez courte. Cependant, cette fois-ci, c'est plutôt dû à des contraintes de mise en page. Mais l'avantage de la HP28 est qu'elle peut aussi utiliser les programmes RPL de la rubrique HP48, qui est très fournie ce mois-ci... En ce qui concerne la rubrique HP95, elle est aussi très courte ce mois-ci. Que les utilisateurs de cette machine ne s'inquiètent pas, cela n'est pas un indice du déclin de la rubrique, mais plutôt le contraire : de nombreux articles en cours d'écriture, mais qui ne seront prêts qu'après que certains derniers détails auront pu être trouvés. En attendant, il serait peut-être temps que les utilisateurs de HP95 suivent l'exemple des possesseurs de HP48 en participant un peu plus, le temps d'assurer le relais.

Pour finir, vous avez sûrement remarqué le changement de couleur de ce JPC. Ce vert "Kiwi" nous accompagnera tout au long des dix prochains numéros. Nous pensons sans aucun doute que vous le trouverez plus reposant que la couleur de l'année dernière !

## COMPTE-RENDU DE L'ASSEMBLEE GENERALE

Comme prévu, notre Assemblée Générale s'est déroulée au Centre Verdier, le 22 Janvier 1994.

### Assemblée Générale Extraordinaire.

Rappelons que cette assemblée générale avait pour but de discuter de modifications statutaires devant permettre de relancer l'écriture des articles pour JPC.

Une évidence est que tout le monde souhaiterait que le journal soit plus étoffé. Cela ne ferait en plus que nous permettre de creuser encore plus l'écart qui existe entre nous et les autres clubs.

Il semble aussi qu'une prise de conscience soit en train de se faire à propos du fait qu'il n'est pas très bon, pour la renommée du club, de laisser penser que seulement une poignée d'auteurs soient actifs. Cela ne fait que confirmer mon opinion qu'il vaut mieux trois articles de trois pages, écrits par trois auteurs différents, qu'un seul article de dix pages...

Pourtant, une des objections faites est que les adhérents actuels programmeraient moins que ceux du début du club. Si cela ne peut pas être vérifié, il est cependant certain que la majorité des programmeurs et utilisateurs actifs écrivaient régulièrement jusqu'en 1986 ou 1987, ce qui n'est plus le cas actuellement. A titre d'information, un rapide calcul (portant sur ces deux dernières années) montre que les deux principaux auteurs ont écrit plus de 50 articles et que le troisième n'en a fait que 10...

Une fois encore, il a été répété que les articles ne devaient pas être obligatoirement des programmes, mais pouvaient être aussi, par exemple, des "trucs" d'utilisateurs (ne demandant pas obligatoirement de programmer) ou des présentations de nouveaux livres, logiciels, ou matériels (le bureau recevant en fait peu de choses. Ceci peut aussi inclure des dessins servant à agrémenter un peu le journal.

Après discussion de ces points, la proposition est passée au vote. Parmi les votants (les adhérents présents ou représentés par un pouvoir), il n'y a eu qu'un vote "contre" et une abstention. La modification est donc approuvée à la quasi-unanimité.

Comme cela a été annoncé, les premiers adhérents touchés par cette mesure seront ceux dont l'adhésion se termine à la fin du mois d'avril 1994.

Hormis les statuts eux-mêmes, d'autres documents sont touchés par cette modification. En premier lieu, le bulletin d'adhésion portera maintenant une mention mentionnant brièvement l'article 7. Il y aura aussi quelques mots de rappel dans les lettres annonçant la fin de la période d'adhésion.

Enfin, nous allons profiter de ces modifications de statuts pour reprendre une habitude qui avait été perdue depuis plusieurs années : les envoyer systématiquement aux nouveaux adhérents. Pour les membres actuels, ils sont inclus dans ce JPC.

Tout ceci devrait nous permettre de revenir progressivement à un journal ayant une pagination moyenne de 50 pages (comme cela se faisait il y a quelques années), avec une segmentation idéale de 20% HP48, 40% HP48 et 40% HP95/100. Et accessoirement de recréer un stock d'articles, indispensable pour assurer une bonne diversité des articles et techniques de programmation présentes dans chaque numéro.

### Assemblée générale Ordinaire

#### *Rapport du président*

Comme je l'ai déjà dit, la situation de PPC Paris est plutôt bonne si on la compare à d'autres clubs étrangers. En effet, nous sommes un des seuls clubs à avoir pu mener une activité "normale", en sortant tous les journaux prévus, en continuant à organiser les réunions et en assurant toutes les autres tâches normales du bureau (courrier, trésorerie...).

Cependant, puisque les nouveaux utilisateurs ne semblaient pas avoir connaissance de nos activités par les moyens que nous avons auparavant (support technique HP, revendeurs...) nous avons du recourir à d'autres méthodes pour faire connaître PPC Paris. Cela a été fait principalement en effectuant un accord avec le nouveau journal *Poche Computer*, consistant à faire paraître une publicité du club en échange d'un article (qui était paru précédemment dans JPC). Ceci nous ayant apporté une soixantaine de demandes de renseignements (pour les personnes ayant explicitement cité *Poche Computer* dans leur lettre), dont environ le quart ont effectivement adhéré.

#### *Affaire Nalin*

Pour les nouveaux adhérents, je rapelle que cette affaire (nous occupant depuis le début de 1992), a débuté lorsque Jean-Paul Nalin, un ancien membre du bureau a disparu en détournant à son profit différents biens appartenant au club, dont notre LaserJet et la majeure partie du stock d'un très gros JPC devant être envoyé aux adhérents.

Après les événements survenus en 1992 (je conseille aux "vieux" adhérents de se reporter au compte-rendu de l'Assemblée Générale précédente pour plus de détails), nous avons attendu jusqu'en fin avril 1993 pour avoir les premières retombées juridiques de cette affaire : une *Citation à Victime* dans le cadre d'un jugement devant survenir le 19 Mai 1993.

Or, à la réception de cette convocation, je me suis aperçu que seule la LaserJet était citée. Pas les journaux, ni le reste du matériel. Était-ce un simple souci de simplifier la rédaction de cette convocation, un oubli "administratif" ou une décision de la justice ? Malheureusement, pour obtenir une réponse à ces questions, il est obligatoire de recourir à un avocat pouvant avoir accès au dossier et connaître les résultats de l'enquête. Le club ne pouvant pas engager de frais pour engager un avocat, j'ai déposé une demande d'aide juridique, puisque cela est théoriquement possible pour les associations à but non lucratif.

Les délais pour obtenir cette aide étant très longs, j'ai aussi dû demander un report d'audience. Il nous été accordé, au cours de l'audience originellement prévue, pour le 15 Septembre. Cependant, le juge m'a confirmé à l'occasion que seule la LaserJet serait prise en compte pour le jugement. Sans vouloir m'en expliquer la raison. Notez que Jean-Paul Nalin n'était pas présent et n'avait pas retiré sa convocation à la mairie d'arrondissement de son lieu déclaré de résidence.

Or, le lendemain, une personne a déposé chez Eric Gengoux (le président précédent), en son absence, un carton contenant notre LaserJet et le lecteur de disquette HPIL 9114 ! Les deux en excellent état de fonctionnement...

Cependant, puisque les journaux ne nous avaient pas été restitués (et représentent financièrement plus de 7000F), nous avons décidé de continuer la procédure et d'attendre d'avoir un avocat afin de discuter de la marche à suivre, avant d'annoncer officiellement cette restitution. En effet, il était dans notre intérêt de faire réintégrer d'abord les charges omises dans le jugement, car nous pensions que si le cas de la LaserJet était jugé seul, nous pourrions perdre toute possibilité d'action en ce qui concerne les journaux. D'autre part, il est à noter que dans le cas d'abus de confiance, le fait de restituer les éléments détournés ne stoppe pas l'action de la justice.

Le jour de la deuxième audience, en septembre, nous n'avions toujours aucune nouvelle de la demande d'aide juridique. Ce qui nous poussa à demander un deuxième report. En fait, j'ai de fortes chances de penser que la demande a été "enterrée" par une

secrétaire qui devait nous envoyer une demande d'informations complémentaires. Chose que j'ai déduite après avoir donné une multitude de coups de téléphone afin de connaître l'état d'avancement du dossier.

Ce n'est que le 28 Octobre (soit six mois jour pour jour après la demande !) que nous avons reçu un avis de rejet de la demande sous le prétexte que l'aide judiciaire n'était accordée *qu'à titre exceptionnel* aux associations.

Je me suis donc présenté à l'audience du 15 décembre, sans l'aide d'un avocat. Comme pour les deux audiences précédentes, Jean-Paul Nalin était absent. Il n'y a lieu d'en être étonné, si l'on sait que cette personne n'a pour adresse qu'une multitude de boîtes aux lettres disséminées dans Paris et que plusieurs personnes ont déposé des plaintes pour abus de confiance contre lui... Chose confirmée par la mimique du juge évoquant un casier judiciaire chargé... Comme prévu, je n'ai pas pu réintégrer le stock de journaux (ni savoir pourquoi ces éléments étaient ignorés). Par contre, j'ai pu démontrer que le fait de ne pas disposer de la LaserJet nous a causé un préjudice, en nous empêchant de porter le programme d'impression du journal sur IBM PC, puis revendre notre IPC (un petit compatible HP9000, tournant sous Unix) avant une période où il a fortement baissé de prix sur le marché de l'occasion. La sanction est donc tombée : deux mois de prison fermes plus 5000F de dommages et intérêts. Une condamnation relativement lourde si l'on sait que cela ne concernait que la LaserJet et que je venais d'annoncer qu'elle nous avait été restituée !

Cependant, il faut être réaliste, il y a peu de chances que cette condamnation soit appliquée, puisque Jean-Paul Nalin est actuellement introuvable. Toute information est bien sûr la bienvenue à ce sujet !

En ce qui concerne les journaux, il serait maintenant nécessaire de déposer une nouvelle plainte à ce sujet. Étant donné le temps qu'a duré la première procédure (un an et demi !), le fait que nous avons peu de chances de les récupérer (je pense qu'ils sont totalement dispersés), Poser une nouvelle plainte ne serait qu'une perte de temps, surtout si nous n'avons pas d'avocat pour nous assister.

#### *Les projets du club*

La conclusion de l'affaire Nalin va enfin nous permettre de remplacer l'ordinateur appartenant au club (L'IPC cité dans le paragraphe concernant l'affaire Nalin), par un portable, que nous pourrions faire circuler parmi les membres du bureau (trésorier, éditeur du journal...) mais surtout amener aux

réunions afin de faciliter les échanges de programmes. Financièrement, cela devrait se faire en revendant l'IPC (par exemple à un étudiant désirant utiliser Unix sur une machine conçue à cet effet) et en achetant un matériel d'occasion.

En ce qui concerne le journal, nous devrions arriver au numéro 100 en octobre. Il serait intéressant de profiter de l'occasion pour faire évoluer sa mise en pages. Tout en conservant le principe général du JPC actuel (sommaire en couverture, séparation des machines...) nous pourrions définir une maquette plus attractive, avec plus d'enrichissements graphiques et probablement une nouvelle façon de présenter les sources. Cependant, j'ai fixé une contrainte dans le choix du logiciel de PAO que nous pourrions utiliser : qu'il soit diffusé en "Freeware" ou en "Shareware". Ceci d'une part afin de ne pas trop grever le budget du club, et d'autre part afin de pouvoir redistribuer légalement des copies du programme aux auteurs réguliers. Notez que j'ai déjà fait des essais avec un logiciel nommé *Envision Publisher*, mais qu'il n'est pas satisfaisant. Cependant, il est très probable que des logiciels plus adaptés à nos besoins apparaissent au cours des prochains mois.

Jacques Belin (123)

---

## LES NOUVEAUX STATUTS

### ARTICLE PREMIER :

Il est fondé entre les adhérents aux présents Statuts une Association, régie par la loi du 1er juillet 1901 et le décret du 16 août 1901, ayant pour dénomination : *PPC Paris*.

### ARTICLE 2 :

Cette Association a pour but de réunir des amateurs et utilisateurs de l'informatique ainsi que d'assurer la mise en commun des connaissances informatiques de chacun au profit de tous.

### ARTICLE 3 :

Le siège social est actuellement fixé à Paris. Il peut être déplacé par simple décision du Bureau.

### ARTICLE 4 :

La durée de l'Association est illimitée.

### ARTICLE 5 :

Les moyens d'action de l'Association sont :

- le bulletin de liaison *JPC* ;
- des réunions ;
- toute activité autorisée par la Loi dans le cadre de l'Article 2 et avec l'accord du Bureau.

### ARTICLE 6 :

L'Association se compose :

- des membres actifs, c'est à dire tout adhérent aux présents Statuts, à jour de sa cotisation.
- des membres honoraires nommés par le Bureau parmi les personnes qui ont rendu ou rendent des services à l'Association ; ils font partie de droit de l'Assemblée Générale sans pouvoir prendre part aux votes.

### ARTICLE 7 :

Pour devenir membre actif de l'Association, il faut avoir acquitté sa cotisation et être agréé par le Bureau.

Pour pouvoir réadhérer à l'association, il faudra en plus avoir transmis au club au moins un article (d'une page minimum) pour le journal *JPC*, au cours de l'année d'adhésion précédente. Les premiers adhérents concernés par cette mesure sont ceux dont l'adhésion se termine fin avril 1994.

### ARTICLE 8 :

Les ressources de l'Association se composent :

- des cotisations de ses membres ;
- des subventions ;
- des revenus de ses biens ;
- de toute ressource autorisée par les textes législatifs et réglementaires.

### ARTICLE 9 :

La qualité de membre de l'Association se perd :

- par démission ;
- par décès ;
- par radiation prononcée par le Bureau pour non paiement de la cotisation ou pour motif grave.

**ARTICLE 10 :**

L'Association est administrée par un Bureau composé de membres actifs élus pour un an par l'Assemblée Générale.

Le mode de scrutin est fixé par le Règlement Intérieur du Bureau. Les membres du Bureau sont rééligibles.

Le Bureau comporte au moins :

- un Président ;
- un Trésorier ;
- un Secrétaire.

Si, au bout d'un an, le Bureau sortant n'est pas renouvelé, ses pouvoirs sont prorogés jusqu'à la plus prochaine Assemblée Générale.

En cas de vacance ou d'incapacité, le Bureau pourvoit provisoirement au remplacement de ses membres en attendant la prochaine Assemblée Générale.

**ARTICLE 11 :**

Le Bureau se réunit au moins une fois tous les six mois sur convocation du Président ou sur demande d'un tiers de ses membres au moins.

Le quorum est fixé aux 2/3 des membres du Bureau. Les décisions sont prises à la majorité absolue des voix ; en cas de partage, la voix du Président est prépondérante.

Tout membre du Bureau qui, sans excuse valable, n'aura pas assisté à trois réunions consécutives sera considéré comme démissionnaire de son poste.

**ARTICLE 12 :**

Le Bureau est investi des pouvoirs les plus étendus pour autoriser tous actes qui ne sont pas réservés à l'Assemblée Générale.

Il assure la gestion matérielle et financière de l'Association.

Il fixe le montant de la cotisation.

Il se fait rendre compte des décisions prises en son nom.

Il autorise tout achat, aliénation ou location, emprunt et prêt nécessaires au fonctionnement de l'Association.

Il établit à l'unanimité un Règlement Intérieur portant application des présents Statuts.

Il peut faire toute délégation de son pouvoir pour un temps déterminé à tout membre actif de l'Association.

Le patrimoine de l'Association répondra seul des engagements pris en son nom et aucun des membres du Bureau ne pourra en être tenu responsable.

**ARTICLE 13 :**

Le Président convoque les Assemblées Générales et les réunions du Bureau.

Il représente l'Association dans tous les actes de la vie civile et est investi de tous pouvoirs à cet effet. En cas d'absence ou d'indisponibilité, il est remplacé par un membre du Bureau pris dans l'ordre de leur élection, ou, en cas d'empêchement, par le membre actif disponible ayant la plus grande ancienneté au sein de l'Association.

**ARTICLE 14 :**

Le Trésorier est chargé de la gestion des comptes et du patrimoine de l'Association.

Ses décisions financières sont contrôlées par le Président ou, à défaut, par tout autre membre du Bureau.

Il présente le rapport financier de l'année écoulée à l'Assemblée Générale et propose un projet de budget pour l'année à venir.

**ARTICLE 15 :**

Le Secrétaire est chargé de tout ce qui concerne la correspondance et les archives.

Il rédige les procès-verbaux des délibérations et en assure la transcription sur les registres.

**ARTICLE 16 :**

L'Assemblée Générale est convoquée au moins une fois par an par le Président.

L'ordre du jour est porté à la connaissance des membres sur leur convocation au moins une semaine avant la date fixée pour l'Assemblée Générale.

L'Assemblée Générale comprend tous les membres actifs, présents ou représentés, à jour de leur cotisation à la date de la convocation et est présidée par le Président assisté des membres du Bureau.

Le Président expose la situation morale de l'Association au moment de la convocation.

Le Trésorier rend compte de sa gestion et soumet son bilan à l'approbation de l'Assemblée.

L'Assemblée Générale peut conférer à tout membre du Bureau autorisation d'agir dans le cadre des buts de l'Association dans tout domaine où les présents Statuts se révéleraient insuffisants.

Après épuisement de l'ordre du jour, il est éventuellement procédé au renouvellement des membres du Bureau.

Toutes les décisions sont prises à la majorité absolue ; en cas de partage, la voix du Président est prépondérante.

## ARTICLE 17 :

Une Assemblée Générale Extraordinaire peut être convoquée dans les mêmes conditions qu'à l'Article 16 ou sur demande d'au moins un quart des membres actifs.

Son objet ne peut être alors que de décider :

- une modification des Statuts ;
- une radiation ;
- une fusion avec toute association de même objet ;
- la dissolution.

Le quorum d'une telle Assemblée est fixé à 20 % des membres actifs, présents ou représentés, et les décisions doivent y être prises à la majorité des 2/3. Si le quorum n'est pas atteint, une nouvelle Assemblée Générale Extraordinaire est convoquée qui délibérera sans contrainte.

## ARTICLE 18 :

La dissolution ne peut être décidée que par une Assemblée Générale Extraordinaire convoquée dans les conditions de l'Article 17.

L'Assemblée décide de l'attribution des actifs de l'Association conformément à l'Article 9 de la loi du 1er juillet 1901 et du décret du 16 août 1901.

Elle désigne un ou plusieurs commissaires, dont elle détermine les pouvoirs, chargés de la liquidation.

Fait à Paris le 22 janvier 1994

---

## LE NOUVEAU SUPPORT TECHNIQUE

Depuis peu de temps, le support technique calculatrices n'est plus pris en charge par Hewlett-Packard, mais par la société INFOPOINT. Cette société est un des deux principaux grossistes de matériel HP.

Si vous avez un problème quant à l'utilisation de votre calculatrice préférée, il vous faudra donc appeler le (1) 69 18 20 64, où Xavier Bouchet se fera un plaisir de vous répondre.

Jacques Belin (123)

## COURRIER DU COEUR

Jean-François GARNIER  
47 allée de la Dent de Crolles  
38420 DOMENE.

### Cherche :

Pour HP75 (oui, je parle bien du 75C!) :

- ROM Math,

- ROM I/O (APRIL),

dans le but de compléter ma collection. Si vous avez ces matériels au fond d'un placard (quelle honte!), ou si vous connaissez quelqu'un les possédant, faites moi une offre.

---

Philippe Guez  
56, rue Jean-Jacques Rousseau  
75001 Paris  
Tel : (1) 42 84 00 69

### Cherche :

Une Ram/Eprom, ou juste une Eprom 64 ko enfichable dans le port du lecteur de carte du HP-71; une interface HPIL-RS232.

Et un volontaire pour lui faire un Lex HP-71.

---

Florent Euanno  
3, rue Marie Fichet  
92140 CLAMART  
Tel : (1) 46 32 52 50

### Cherche :

Un HP95 avec 512 Ko, prix : environ 1800 FF

## **HP28**

G. Toublanc

Modes Symbolique/Numérique (acte II) 8

**Le coin des codes** 30

## TRUCS ASTUCES ET PETITS UTILITAIRES

### Un tout autre SYMB pour HP28S

Dans JPC 91 page 4, je proposais SYMB, un petit programme permettant de passer du mode symbolique au mode numérique et vice versa. Voici son cousin en assembleur, mais fonctionnant d'une manière qui paraîtra plus agréable à certains. En effet le nom du programme se modifie suivant le mode. En mode numérique le label du programme est SYM, lors du passage en mode symbolique un petit carré est ajouté à SYM.

#### Les limites du programme :

Ce petit programme simule ce qui se fait dans le menu MODE mais ne peut pas prétendre avoir la maîtrise complète du système de la machine comme ses cousines STD FIX etc...

La correspondance entre le label SYMx et le mode symbolique/numérique n'est valable que si l'état du flag 35 n'est pas modifié par un programme. Il est donc conseillé dans un programme de sauvegarder l'état des flags avec RCLF et de les restaurer en fin de programme. De toutes manières SYMx est fait avant tout (comme son prédécesseur SYMB) pour vous éviter de vous souvenir constamment de l'usage du flag 35.

Autre particularité : SYMx ne peut être activé que dans le répertoire où il se trouve, donc sa place toute indiquée pourra être dans le répertoire HOME.

#### Principe du programme :

Le nom du programme est SYM + 1 caractère, donc au total 4 caractères. Le programme change l'état du flag 35 et suivant le mode ainsi activé change le dernier caractère du nom :

- Passage en mode symbolique : caractère = CHR 0, ce qui se traduit à l'affichage par un petit carré à la fin du label.

- Passage en mode numérique: caractère = 1 espace, ce qui se traduit à l'affichage par le label SYM tout simplement.

#### Mode d'emploi:

Pour stocker le programme opérez de la manière suivante: ajouter une quelconque lettre au nom, par exemple 'SYMA'.

Le programme assemblé étant sur la pile faire "SYM " # 5B15h SYSEVAL STO pour transformer la chaîne "SYM " (se terminant par un espace) en nom global avant de stocher le programme.

Pour la première fois si vous activé SYMx il y aura passage dans l'autre mode, le label modifié perdra sa dernière lettre. Pour toutes les autres activations le label sera modifié en parallèle avec le changement de mode symbolique/numérique.

Pour contrôler le bon fonctionnement de SYMx, après sa deuxième activation, tapez  $\pi$  puis faire alternativement SYMx  $\pi$  et éventuellement ENTER si c'est le mode numérique pour constater l'effet.

Pour purger SYMx dans son répertoire faire: VARS puis repérez dans la liste la position N de SYMx que vous extrayez avec N GET. Un PURGE est maintenant réalisable.

Voici ce programme.

#### SYMx

cksum: # 9245h 40.5 octets

```
ASSEMBLE
con(5) #02c96 * DOCODE
rel(5) end
gosbvl #05081 * SAVPTR
gosub here
here c=rstk * adresse du prog. ici
d1=c * @ programme à "here"
d1=d1- 16 * pour pointer sur
d1=d1- 9 * dernier caractère du
* nom " " ou CHR 0
d0=(5) #c0117 * #C010Fh + 8 (32/4)
* @ sur flag 33
a=dat0 a * état flags depuis 33
c=0 a * pour CHR 0
?abit=0 2 * flag 35 à zéro ?
goyes num * oui alors en num
abit=0 2 * non symb. alors num
lcasc ' ' * et dernier caract.
gonc load * puis charge en mém.
num abit=1 2 * passage en symb.
load dat0=a a * change flag 35
dat1=c b * change dernier car.
govlng #125e1 * GETPTRLOOP
end
```

Voir la liste des codes dans le coin des codes.

Guy Toublanc (276)

## HP48

C. Nober	Annuaire téléphonique	10
R. Amram	Conversion de caractères accentués	13
J.F. Garnier	Numéro de semaine (acte 3)	14
L.V Huynh / G.T. Toublanc	Trucs, Astuces et Utilitaires	15
G. Toublanc	Conversion de fichiers Xmodem (acte II)	16
G. Toublanc	PEEK et la Rom Cachée	12
	<b>Le coin des codes</b>	<b>30</b>

# ANNUAIRE TELEPHONIQUE

NDLR : Un an après un programme de Laurent Grand gérant un annuaire téléphonique, en voici un autre traitant du même sujet.

Je vous envoie un programme de débutant pour d'autres débutants. C'est un petit répertoire d'adresses, téléphones et notes.

Le stockage des données se fera dans une liste contenant des sous-listes spécifiques à chaque NOM. Cette liste s'appellera 'LTEL'.

Par exemple :

```
'LTEL' ( { "NOM Prénom"
          "adresse"
          "tel : 12 34 56 78"
          "membre PPC" }
        { ... }
      )
```

Pour chaque nom, on créera une sous-liste contenant 4 chaînes alphanumériques.

On veillera à ne pas dépasser 6 lignes par chaîne et une seule ligne (ou plus à condition de limiter alors les autres chaînes) pour le nom-prénom.

LTEL doit contenir au moins une sous-liste pour que le programme GO puisse "tourner".  
Exemple : { { "" "" "" "" } }

## Architecture du programme.

```
HOME - 'TELEF' « TEL GO »
      || (programme de démarrage depuis HOME)
      |
      v
TEL    - CST : { GO }
(sous-répertoire) - GO
                    - INTRO
                    - TEL2...TEL6
```

## Programme 'N°'

Ce programme prend dans la première ligne de la pile une chaîne alphanumérique et va, par dichotomie, trouver le numéro de classement qui lui correspond dans LTEL.

La variable locale X jouera le rôle de borne inférieure, sa valeur de départ sera 1.

La variable locale Y jouera le rôle de borne supérieure, sa valeur de départ sera : « LTEL SIZE 1 + », c'est à dire la taille de LTEL plus 1

(ce qui correspond à la dernière place de LTEL avec le nouvel élément).

La variable Z sera notre variable de comparaison, toujours choisie entre X et Y, donc « X Y + 2 / »

## Programme 'INTRO'

Ce programme demande "NOM Prénom", "adresse", "numéro de téléphone", "notes" et place les quatre chaînes dans la pile et renvoie au programme TEL3.

## Programme 'TEL2'

Efface de la liste LTEL la Nième liste. N est placé à la première ligne de la pile.

## Programme 'TEL3'

Ce programme reçoit les quatre chaînes alphanumériques générées par INTRO ou TEL4. En fait, une liste qu'il classe ensuite dans LTEL.

## Programme 'TEL4'

Ce programme permet de modifier les quatre chaînes alphanumériques qu'une sous-liste N (en fait, il efface la sous-liste dans LTEL tout en la conservant dans la pile pour la modifier puis la réenregistrer.

## Programme 'TEL5'

Affichage succinct de la Nième sous-liste.  
>NOBER Christophe  
tel : 12 34 56 78

## Programme 'TEL6'

Affichage Complet.

## Programme 'GO'

Programme de gestion du répertoire :  
- Affiche les menus.  
- Gère l'affichage.  
- Place au dessus de la pile le chiffre N qui correspond à la sous-liste pointée, sa première valeur est 1.

## MENU :

- "?!": Cherche un nom avec ses premières lettres.

- "↑" : Remonte la liste.
- "↓" : Descend la liste.
- "VOIR" : Permet de visualiser les éléments non visibles (adresse et notes). Le retour se fait par la touche ENTER. L'affichage du Nom est précédé de VOIR>NOM.
- "MODIF" : Permet d'effacer ou modifier le nom à l'affichage.
- "NOTER" : Enregistrer une nouvelle donnée.

### Comment tout cela fonctionne-il ?

Il suffit de taper directement depuis HOME sur TELEF ou entrer dans le répertoire TEL, puis taper GO.

- Pour noter un nom, taper NOTE, puis suivre les instructions et exemples, le retour au menu est automatique dès la saisie terminée.
- Pour chercher un nom avec ses premières lettres, taper "↑?↓".
- "↑" et "↓" font défiler les noms avec les numéros de téléphone.
- "VOIR" permet de faire afficher l'adresse puis, si l'on presse sur une touche du clavier, les notes. Le retour au menu se fait par ENTER.
- "MODIF" permet d'effacer une donnée (Kill) ou de modifier les 4 chaînes alphanumériques (Modif).
- Le retour au répertoire HOME se fait par ENTER.

### Listing du programme

NDLR : Dans les sources suivants, le caractère 'R' indique un retour chariot.

DIR

```
LTEL
  ( (
    "NOM prénom"
    "BP 604
    75028 PARIS CEDEX 01"
    " tel:12 34 56 78"
    "Notes diverses" )
  )
```

N°

```
«
  1 LTEL SIZE 1 +      @ Prépare X et Y
  DUP2 + 2 /          @ Calcule Z (milieu)
  → X Y Z             @ Crée les Vars Locales
  «
    DO
      DUP              @ Copie la chaîne
                      @ alphanum de la pile
      LTEL Z IP        @ Prend la partie entière
                      @ de Z pour être
                      @ compatible avec GET
      GET              @ Copie la 2eme liste
                      @ de LTEL
```

```
1 GET                @ Copie la 1ere chaîne
                    @ alphanum. (NOM Prénom)
                    @ de la 2eme liste de
                    @ LTEL
                    @ Compare les 2 chaînes

IF ≤
THEN
  Z IP 'Y' STO       @ La chaîne est classée
ELSE
  Z IP 'X' STO       @ avant la chaîne 'Z'.
                    @ Z est alors notre
                    @ nouvelle borne sup.
END                  @ qu'on stocke dans Y
Z                    @ Val. de Z dans pile
X Y + 2 /           @ Nouv. val. de Z dans
DUP 'Z' STO         @ pile + enregistrement
UNTIL == END        @ Quand Z n'a pas changé,
                    @ c'est que X et Y ne
                    @ changent plus.
                    @ (soit X=Y ou Y=X+1)

DROP                @ On efface la chaîne
Z                    @ On rappelle Z, qui
                    @ peut être soit un
                    @ entier, soit un
                    @ entier+½ (Y=X+1)
                    @ → Z=X ou Z=Y (2e cas)
DUP FP +            @ On rajoute à Z sa
                    @ partie décimale, ce
                    @ qui évite un test
```

»

INTRO

```
«
  "NOM prénom ?" { "" α } INPUT
  "adresse ?" { "" α } INPUT
  "n° téléphone ?" { "" α } INPUT
  " tel:" SWAP +
  "notes ?" { "" α } INPUT
  TEL3
  »
```

TEL2

```
«
  LTEL OBJ→         @ Démonte LTEL
  DUP 2 + ROLL       @ Rappelle N au dessus
                    @ de la pile
  NEG OVER + 2 +     @ Calcule la position
                    @ de la liste à effacer
  ROLL               @ Amène liste au niveau 1
  DROP              @ .. et l'efface
  1 -                @ Décrémente taille LTEL
  →LIST 'LTEL' STO  @ Reforme LTEL
  »
```

TEL3

```
«
  4 →LIST           @ Forme la liste
  DUP 1 GET         @ Affiche "NOM Prénom"
  N°                @ Trouve son classement
```

LTEL OBJ→	@ Démonte la liste LTEL	TEL6	
1 + → X	@ Rajoute 1 à la taille	«	
	@ de LTEL et stocke	{ } TMENU	@ Efface la ligne des
	@ dans X		@ menus
«		LTEL SWAP GET	@ Rappelle Nieme
X 1 + ROLL	@ Rappelle le classement		@ sous-liste
X 1 + ROLL	@ du dessus de la pile	OBJ→ DROP	
	@ ainsi la liste	SWAP DROP	@ Efface la ligne "tel:"
NEG X + 1 +	@ Calcule la position	ROT "VOIR" 134 CHR +	@ "VOIR>"
	@ que devra prendre	SWAP +	@ "VOIR>NOM Prenom"
	@ notre liste dans la	"X" + 1 DISP	@ Ajoute un retour
	@ pile [X+1 - pos. N°]		@ chariot et affiche
ROLLD	@ Place		@ ligne 1
X →LIST	@ Reforme LTEL	IF DUP2 + "" ≠	@ Si "adresse" ou
'LTEL' STO		THEN	@ "notes" sont non vides
»		DO	
»		SWAP DUP	
		IF "" SAME	@ Si la chaîne à afficher
		THEN	@ est vide, on passe
		SWAP	@ à la suivante
		END	
		DUP "XXXXXX" +	@ 5 retours chariots
		2 DISP 3 FREEZE	
		-1 WAIT	
		UNTIL 51.1 ==	@ Correspond à la touche
			@ ENTER
		END	
		ELSE	@ 2eme clause du test,
		"X	
		rien de plus	
		X	
		"	
		3 DISP 2 WAIT	@ affichage 2 secondes
		END	
		DROP2	@ Efface la pile
		»	
		GO	
		«	
		1	@ 1ere place du pointeur
		DO	
		{ "↑↓" "↑" "↓" "VOIR"	
		"MODIF" "NOTER" } TMENU	
		DUP TEL5	@ Affichage suivant
		-1 WAIT → X	
		«	
		CASE X 11.1 ==	@ Efface le pointeur et
		THEN	@ le remplace dans
		DROP	@ la pile
		"1ères lettres ?"	
		{ "" α } INPUT N°	
		END	
		X 12.1 ==	@ Décrémente le pointeur
		THEN	
		1 -	
		END	
		X 13.1 ==	@ Incrémente le pointeur
		THEN	

```

1 +
END
X 14.1 == @ Affichage restant
THEN
DUP TEL6
END
X 15.1 ==
THEN
" R
R
effacer (↑↓)R
R
modifier (MODIF)
R
"
3 DISP 3 FREEZE
-1 WAIT → Y
«
CASE Y 11.1 == @ Efface
THEN
DUP TEL2
END
Y 15.1 ==
THEN
DUP TEL4 @ modifie
END
END
»
END
X 16.1 ==
THEN
INTRO @ enregistrer
END
END X
»
SWAP
1 - LTEL SIZE MOD 1 + @ Vérifie que le
SWAP @ le pointeur soit
@ compris entre 1
@ et LTEL SIZE
UNTIL 51.1 == @ Touche ENTER
END
1 MENU @ Menu utilisateur CST
UPDIR @ Retour au répertoire
@ supérieur
DROP @ Efface le pointeur
»

CST { GO }
END

```

Christophe Nober (594)

## LA HP48SX : OUTIL DU TRADUCTEUR PROFESSIONNEL

Lorsque j'ai exprimé l'avis à notre ami Jacques Belin que notre journal risquait de paraître un peu "élitiste" à la vaste majorité des utilisateurs de nos machines, il m'a répondu que ce qui lui manquait cruellement c'était justement des articles écrits par des débutants ou des semi-débutants.

Me trouvant en vacances dernièrement pour quelques jours, j'en ai donc profité pour soumettre ma propre contribution de semi-débutant.

Il s'agit d'un petit programme pour le HP48SX permettant de convertir les caractères accentués d'un texte généré sur une machine autre qu'un Macintosh (au moyen d'un traitement de texte quelconque), en caractères dont les valeurs ASCII représentent ces mêmes caractères accentués dans le tableau ASCII utilisé par le Macintosh. Ceci afin de permettre une exploitation directe du texte en question par le Mac.

Ici un mot pour expliquer les raisons qui ont conduit à ce programme. Je suis traducteur technique de mon métier, et lorsque je suis en déplacement j'utilise souvent un Z88 (de Cambridge Computer Ltd) à clavier français pour traduire des textes de l'anglais vers le français, que je transforme ensuite en fichier ASCII. Ce fichier est alors envoyé sur le HP48 (via Kermit), qui effectue automatiquement la conversion des caractères souhaités afin de permettre ensuite le transfert du fichier - contenant le texte ainsi modifié - vers le Mac.

Très succinctement, il y'a lieu tout d'abord de constituer deux listes. La première devra contenir, chacun entre double guillemets, la suite des caractères accentués (et/ou non accentués éventuellement) à convertir. L'autre liste contient, dans le même ordre bien entendu, une suite de caractères dont les valeurs ASCII sur le HP48 représentent, pour le Macintosh, les caractères d'origine qui étaient à convertir. Personnellement j'ai préféré stocker ces deux listes dans le port 0, sous les variables Z et MAC respectivement.

Quant au programme lui-même, celui-ci EVALUE, au niveau 1 de la pile, le contenu de la variable utilisée pour stocker le texte lors de son transfert depuis le Z88. Le programme fait ensuite appel à une boucle qui sélectionne, séquentiellement, chacun des caractères de la première liste pour le remplacer par le caractère correspondant de la deuxième liste. Une fois la boucle "bouclée", la variable qui contenait le texte Z88 non compatible Macintosh à l'origine, contiendra maintenant un texte qui le sera devenu.

Dans mon cas personnel, étant donné que je me trouve souvent en Grande Bretagne, j'utilise le Kermit du HP48 et un modem pour transférer ensuite le texte directement vers le Mac de mon client en France.

J'attire l'attention des lecteurs sur plusieurs points :

- Il va de soi que le type d'ordinateur sur lequel est rédigé le texte d'origine est sans importance, puisqu'il suffit de modifier la première liste en conséquence, celle-ci devant contenir les caractères HP48 correspondant aux codes que génère l'ordinateur pour représenter les caractères à convertir. Il se trouve que l'exemple du Z88 est commode parce que cette machine utilise le même code ISO que le HP48; donc les caractères que je souhaite convertir (ë â ê é à è ï î ô ù û ç) sont rendus identiquement par le HP48 lorsque celui-ci les reçoit du Z88; c'est-à-dire que la fameuse première liste contiendra, dans ce cas particulier, la même suite de caractères. Par contre, si le texte d'origine avait été rédigé sur un PC, ladite liste contiendrait des caractères tout à fait différents. (Par exemple, le "é", qui est généré par le code ASCII 130 de l'IBM, serait représenté, dans la première liste, par le caractère de code 130 de la HP48 (un petit triangle inversé). Dans mon exemple, les caractères à convertir ont respectivement pour valeurs ASCII Macintosh :

145 137 144 142 136 143 149 148 153 157 158 141

..et seront donc représentés, dans la deuxième liste, par les caractères HP48 obtenus en tapant :

145 ENTER CHR ENTER, 137 ENTER CHR ENTER etc...

- J'ai écrit ce programme il y a bientôt deux ans, et j'ai utilisé une commande tirée du *Programmer's ToolKit* de James Donnelly : REPLACE. Cette commande sert à substituer une sous-chaîne quelconque par une autre sous-chaîne partout où elle est rencontrée à l'intérieur d'une chaîne donnée. Par exemple :  
"ABCB" "B" "-" REPLACE donne "A-C-D"

- Il est très important de respecter l'ordre des caractères contenus dans la première liste; autrement on risque de voir un caractère de remplacement (inséré lors d'une précédente passe) remplacé lui-même par inadvertance lors d'une passe ultérieure !

- Ce petit programme n'a aucune prétention, si ce n'est d'empêcher que je ne sois mal vu par le bureau pour manque de solidarité !! Si Jacques Belin décide de le publier, peut-être servira-t-il au moins à inciter d'autres membres - dont les compétences en matière de programmation ont été acquises, comme moi,

strictement grâce aux manuels HP - à sortir de leur timidité et à contribuer quelque chose à leur tour.

*Last but by no means least*, étant moi-même de langue anglaise, je demande aux lecteurs d'être indulgents s'ils éprouvent, ici et là, des difficultés à me comprendre. Croyez-moi, j'ai vraiment transpiré pour rédiger cet article en français.

En tout cas, je suis à la disposition de tout membre du Club qui aurait des problèmes de compréhension de textes, notices techniques, manuels etc. en langue anglaise. Qu'ils n'hésitent pas à me contacter.

Enfin, je voudrais profiter de cette occasion pour dire combien j'admire la tenacité et le courage de Jacques Belin dans le rôle, souvent ingrat, qui est le sien. Je pense franchement que sans lui et Guy Toublanc le Club sombrerait bien vite.

Voici le programme, que j'ai nommé ZMC : (Je rappelle qu'il utilise deux listes, "Z" et "MAC", contenant respectivement les codes d'origine et les codes de remplacement (12 dans mon exemple) et stockées toutes deux dans le port 0).

```
«
DUP EVAL :0: MAC EVAL :0: Z EVAL
ROT m pc f
«
f 1 12
FOR i
pc i GET m i GET
REPLACE
NEXT
»
SWAP STO
»
```

Robert Amram (390)

---

## NUMERO DE SEMAINE ACTE III

Faisant suite aux articles de Robert Pulluard et de Guy Toublanc, je vous propose ma contribution à l'art de l'optimisation.

Tout d'abord, il me faut relever une erreur dans les programmes de Guy qui a semble-t-il regroupé un peu rapidement les lignes 3 et 7 (je reprend ses notations, voir JPC92, p10). Pour vous en convaincre, essayez avec un jour quelconque de 1992. Voici une version corrigée possible de NSEMUSER, renommée NSEM2 :

#### NSEM2

checksum: #8145h 106.5 octets

```
«
DUP 0 DATE IFTE + @ 1
-2 ALOG OVER OVER MOD + 1 + @ 2
DUP ROT DDAYS 3 + @ 3
"THFRSASUMOTUWE" @ 4
ROT 0 TSTR @ 5
1 2 SUB POS @ 6
2 / + 7 / @ 7
FLOOR @ 8
»
```

Je me suis limité pour ma part à l'optimisation de NSEM en langage RPL utilisateur. Je me suis interdit toute astuce telle que suppression des délimiteurs de programme, insertion d'Externals, etc... Voici donc ma version, dénommée NSEM3 :

#### NSEM3

checksum: #70ECh 78 octets

```
«
DUP DATE IFTE
-2 ALOG @ jj.mmaaaa 0.01
DUP2 MOD + 1 + @ jj.mmaaaa 1.01aaaa
DUP ROT @ 1.01aaaa 1.01aaaa jj.mmaaaa
DDAYS 3 + @ 1.01aaaa (n+2)
2.021989 ROT @ (n+2) 2.021989 1.01aaaa
DDAYS 7 MOD @ (n+2) r
+ 7 / FLOOR @ résultat !
»
```

Le gain en encombrement est dû principalement au calcul du jour de la semaine à l'aide de DDAYS. En effet, la fonction DOW (Day Of Week) peut être réalisée par la séquence :

```
1.011989 SWAP DDAYS 7 MOD
```

J'ai intégré une variante de ce calcul à NSEM3 de façon à optimiser le code. Notez que j'avais déjà publié une version de DOW sur le même principe dans JPC 78 à l'occasion d'un programme de calendrier perpétuel (je constate que ma prose laisse un souvenir impérissable ...). Toute la subtilité reside dans le fait que cette séquence fonctionne quelque soit le format de la date (flag -42). Vous trouverez un autre exemple d'utilisation de DDAYS dans la dernière version de mon

programme d'astronomie (à paraître le moins prochain), où je l'utilise pour calculer un temps exprimé en jours julian de façon très efficace.

J'ai gagné encore quelques quartets en utilisant de façon adéquate les fonctions IFTE et DUP2. (Une question m'obsède: à quoi pensait donc notre bon docteur JPC en écrivant cette horrible séquence OVER OVER ?). L'astuce de Robert pour placer 0.01 sur la pile consistant à faire le calcul -2 ALOG est effectivement très efficace en terme de taille de code (5 octets), mais si le temps d'exécution est votre souci principal, vous aurez intérêt à remplacer ces 2 instructions par la constante 0.01, même au prix de ses 10.5 octets.

La conclusion de tout ceci pourrait être que le langage RPL utilisateur, est d'une très grande richesse, et peut, bien maîtrisé, donner des programmes extrêmement performants et compacts, du moins dans les domaines qui sont le sien, c'est à dire les utilitaires à base mathématique. Le system RPL est d'une richesse encore plus grande et permet justement d'élargir le champ d'action de la 48, mais je ne l'ai jusqu'ici qu'effleuré superficiellement.

Qui réalisera NSEM4 à moins de 75 octets ? (Je pense que c'est impossible, mais ...)

Jean-François GARNIER (242)

---

## TRUCS ASTUCES ET PETITS UTILITAIRES

Auteur: Lu Vy HUYNH (579)

Sujet : retourner n°LIB et n°XLIB d'un Rom-Word ou Romptr (sur HP48xx).

#### XLIB+

bytes: 68, chksum: #DA81h

```
::
CK1NoBlame * objet quelconque
::
CODE
GOSBVL =SAVPTR
GOSBVL =Rom-Word? * ROM-WORD ?
GOC inrom * oui
```

## A PROPOS DE XMODEM et XMOD->KER

```

D0=A          * non
A=DATO A
D0=D0+ 11
LC(5) =DOROMP
?A=C A       * ROMPTR ?
GOYES ok     * oui

GOSBVL =GETPTR * ce n'est pas ROMPTR
GOVLNG #0501B * alors "Undefined XLIB
           * Name" DOERR

inrom D0=A

ok  D0=D0- 3   * n°XLIB
    C=0 A
    C=DATO X
    R1=C       * dans R1
    D0=D0- 3   * n°LIB
    C=DATO X
    R0=C       * dans R0
    GOSBVL =PUSH2# * push 2:<n°LIB>
           *      1:<n°XLIB>

    GOVLNG =Loop

ENDCODE

ROTDROP
UNCOERCE2    * UNCOERCE SWAP UNCOERCE
              * SWAP
;
;

```

### Commentaires de G.T :

Très heureux de recevoir dans cette rubrique, qui est celle de tout le monde, cette contribution de notre ami Lu Vy Huynh, j'ajouterais simplement: l'adresse #050B1 bien que n'étant pas un point d'entrée supporté par HP, correspond bien au même programme dans les versions A à P.

### Exemple d'utilisation :

```

{ SWAP } OBJ-> DROP
XLIB->  → 2
        271

```

Donc SWAP appartient à la librairie en Rom de numéro 2 et a pour numéro de commande 271 (#10Fh).

Même procédure pour les librairies non en Rom.

Echappent à ce programmes les instructions telles IF THEN etc... ou celles de la librairie de commande DEV.

David Fabiani, un adhérent italien de HPCC et de 48Sxtant m'a fait part de problèmes rencontrés lors du transfert d'objets HP48 de son HP95 vers la HP48 série G et sous protocole Xmodem. S'il est satisfait du Kermit HP48 il émet des réserves concernant la fiabilité du Xmodem HP48, déplorant le manque de contrôle par CRC.

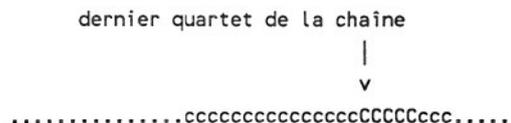
En ce qui me concerne je n'ai pas rencontré de problèmes en utilisant Xmodem de Windows 3.0. Par contre avec LotusWorks 1.0 je reçois toujours dans la HP48 des chaînes ce qui m'oblige à récupérer l'objet binaire avec XMOD-KER (JPC-91 page 14).

L'alerte de David Fabiani m'a fait prendre mieux conscience de dégradations éventuelles des données d'un objet HP48 lors d'une réception Xmodem. Aussi il n'est pas inutile de rappeler le principe de XMOD-KER et de ses limites.

Ce programme recherche, dans les données de la chaîne reçue, le code du premier prologue de l'objet binaire HP48. C'est ce principe qui a été utilisé avant moi dans le programme ASCLOAD de Voyage 48G (voir l'intitulé *très très* précis de cet ouvrage dans JPC-91 page 4). Dans la partie *assembleur* de ce programme il est vérifié :

- la présence d'un objet
- le type chaîne

Si la chaîne est de longueur nulle ou si aucun prologue n'a été trouvé, la chaîne est recrée sur la pile donc ici pas d'avertissement de problème rencontré. Autre problème : le programme cherche un prologue en partant du premier quartet des données de la chaîne et allant éventuellement jusqu'au dernier quartet, le prélèvement des quartets se se faisant toujours par 5. Ceci peut nous amener au cas de figure suivant dans le cas d'un objet dégradé :



la suite ccccc de code, du dernier quartet de la chaîne et des 4 quartets suivants en mémoire (4 premiers quartets de longueur de zone de l'objet lui-même avec son nom dans le répertoire), peut correspondre éventuellement au code d'un prologue et alors qu'obtiendrons nous comme objet à partir de cette adresse ?

## ROMs ET REGARDS INDISCRETS

Pour prévoir cette éventualité j'ai modifié mon programme XMOD-KER. Celui-ci, comme ASCLOAD, est un utilitaire qui ne peut travailler qu'avec ce qu'on lui donne et donc ne peut pas faire des miracles si le code a été dégradé.

Voici donc XMOD-KER nouvelle mouture. Les commentaires peuvent être lus dans la 1ère version les lignes de remarques correspondent à :

- Remarque 1 : Calcul du nombre maxi de prélèvements de 5 quartets puis refus d'une chaîne de longueur nulle ou de longueur inférieure à 5 nibs.
- Remarque 2 : refus si sortie de la boucle sans avoir trouvé de code de prologue.

Si cette nouvelle version plus sécurisante fait 5 octets de plus elle reste encore moins octivore que ASCLOAD qui est 36% plus long, donc à vous de choisir.

### XMOD-KER

67.5 octets cksum # D03Ch

```
::
CK1NoBlame CK&DISPATCH0 str
::
CODE
    gosbvl  =SAVPTR
    a=dat1  a
    d1=a
    d1=d1+  5
    c=dat1  a
    d1=d1+  5
    c=c-con a,10      * remarque n° 1
    d=c      a
    lc(5)   =PRLG
    goc     bad      * remarque n° 1
loop      a=dat1  a
          d0=a
          a=dat0  a
          ?a=c    a
          goyes   ok
cont      d1=d1+  1
          d=d-1  a
          gonc    loop      * remarque n° 2
bad       lchex  0203
          govlng  =GPErrjmpC
ok        ad1ex
          govlng  =GPOverWrAlp
ENDCODE
    TOTEMPOB
;
;
```

Voir la liste des codes dans le coin des codes.

Guy Toublanc (276)

### Compatibilité S et X

Dans un précédent JPC (n°89 pages 15-16) était abordé le problème de la compatibilité logicielle entre HP48S(X) et HP48G(X). Un rapide inventaire, des programmes parus dans JPC et en langage System Rpl ou assembleur, révélait quelques programmes pour HP48S(X) inexploitable sur HP48G(X). Parmi eux le programme XPEEK permettant de faire un PEEK en Rom cachée.

### Peek en Rom cachée

Le programme pour S(X) n'opérait que dans la plage # 70000h-# 7FFFFh, cette dernière adresse étant la limite supérieure de la Rom, celle-ci pouvant être explorée en dessous de # 70000h avec un classique PEEK. En ce qui concerne la Rom G(X) le problème est un peu plus compliqué car celle-ci va jusque # FFFFFh et comporte plusieurs plages cachées.

La technique de programmation doit donc être revue pour pouvoir explorer la Rom quelle que soit la plage mémoire et quelle que soit la position du programme en Ram, IRam, Rom.

Le problème de départ : il faut à la fois tenir compte de la position du programme en mémoire et de l'adresse où *peeker*. Vu le nombre de cas de figures et pour simplifier la programmation on pourra adopter la stratégie suivante :

Le programme sera recopié dans une zone mémoire et exécuté depuis sa nouvelle position. Deux zones sont à notre disposition: le buffer de transfert de mémoire (532 quartets) et le buffer de données de l'interface série RS232c (512 quartets) qui ne seront pas utilisés autrement pendant l'exécution du programme. Pour ma part j'ai retenu le 2ième buffer, la partie de programme copiée (entre *start* et *end* dans le listing du fichier source) faisant 364 quartets pour le XPEEK polyvalent et 353 pour la version spécifique HP48G(X).

### Choix et fonctionnalités

Deux versions vous sont offertes:

- Un XPEEK compatible HP48-G(X) et S(X) pour ceux qui ont les deux machines. Dans les deux cas on

peut faire un PEEK sur toute l'étendue de la Rom de chaque machine, donc sans les contraintes de la version HP48S(X).

- Un XPEEKG spécifique HP48G(X) donc moins octivore (235 octets contre 287,5).

Ces programme sont exécutables en Ram ou en port 0 pour toutes les machines, en port 1 pour les X ou en port 2 pour la SX.

Côté syntaxe j'ai opté pour la double possibilité de donner le nombre de quartets à *peeker* à l'aide d'un entier binaire ou d'un réel.

Malgré tous ces "+" mon programme compatible S et G fait 10 octets de moins que son homologue d'un certain livre sur la G(X) et qui est spécifique G(X). Ce dernier programme est 30 % plus long que le mien pour G(X) (sans la double option) et dans tous les cas de figures mes programmes sont au moins aussi rapides.

Maintenant concernant le choix du programme soit spécifique G(X) ou compatible S(X) et G(X) cela dépend de la ou des machines possédées :

- Vous n'avez qu'une HP48G(X) alors pas de problème c'est le programme spécifique et le moins long qui convient.

- Vous possédez les 2 types de machines et ne tenez pas à avoir 2 versions à sauvegarder, ni à faire de confusions, alors le programme compatible S et G est la bonne solution, d'autant que celui-ci a la possibilité de *peeker* dans toute la Rom cachée ou non, ce que ne fait pas l'ancienne version pour S(X).

### Usages possibles

A quoi peuvent servir ces programmes ? Si vous êtes un curieux vous pouvez :

- Transférer le code de la Rom HP48 dans un PC pour ensuite le désassembler.

- Comparer les Roms HP48S(X) et HP48G(X) et ainsi voir ce qui a changé.

- Ou tout simplement faire un éditeur de mémoire. A ce propos je signale que les HP48G(X) ne possèdent pas d'éditeur de mémoire intégré comme chez leurs aînées les HP48S(X), ce qui était très pratique pour transférer le code de la Rom dans un PC.

### Inspiration et voies diverses

L'allusion au programme d'un certain livre se rapporte évidemment à celui intitulé ROMPEEK et le livre en question est aussi évidemment Voyage 48G (réf. dans JPC-91 p4). Si pour le principe de transférer le programme dans une zone de buffer je

me suis inspiré de la dernière version de ROMPEEK, pour y arriver j'emploie une méthode plus économe en mémoire. En effet ROMPEEK utilise un programme d'installation d'un programme non évalué puis lance celui-ci, technique intéressante, alors que mon programme est d'un seul tenant avec une seule séquence de code *assembleur*. Le reste des octets économisés est le résultat d'un plus grand soin dans ma programmation (ce qui n'exclut nullement des possibilités de faire mieux).

### XPEEKs spécialisés

Vous trouverez à la fin de cet article le source du programme XPEEKU compatible HP48 S et G. J'ai essayé de le commenter largement. Pour ceux qui n'ont qu'une HP48Gx voici tout de suite le source de la version courte. Il n'est pas commenté car la programmation est de même type que celle de XPEEKU auquel vous êtes renvoyé. D'autre part pour ne pas monopoliser les colonnes de JPC j'ai mis à côté le source d'un programme de la même famille mais plus particulier.

La syntaxe de XPEEKG est rappelé dans le source de XPEEKU.

Voici donc les deux programmes en question.

XPEEKG	XPEEK32K
cksum # 404Ah 235 octets	cksum # 7AB0h 210.5 oct.
::	::
OLASTOWDOB! CK2NOLASTWD	CK1NoBlame
DUPTYPEREAL? IT %>#	
CK&DISPATCH1 <BBh>	CK&DISPATCHO ELEVEN
::	::
CODE	CODE
=<BBh> EQU #64dce	
gosub trans	gosub trans
gosbvl =PopASavptr	
d0=a	
d0=d0+ 10	
c=dat0 a	
rstk=c	
c=c+c a	lchex 10000
gosbvl =MAKE\$N	gosbvl =MAKE\$N
c=rstk	lchex fff
b=c a	b=c a
c=dat1 a	c=dat1 a
d1=c	d1=c
d1=d1+ 10	d1=d1+ 10
c=dat1 a	c=dat1 a
d1=c	d1=c
goto decr	
loop gosbvl =DisableIntr loop gosbvl =Disab...	
ad1ex	ad1ex
d1=a	d1=a

```

cd0ex          cd0ex
r1=c           r1=c
lchex 00140   lchex 00140
?a>=c a       ?a>=c a
goyes noram0  goyes noram0
c=0 b         c=0 b
?a<c a        ?a<c a
goyes noram0  goyes noram0
a=0 a         a=0 w
gonc load     gonc load
noram0 lchex 7e000  noram0 lchex 7e000
?a>=c a       ?a>=c a
goyes testram1 goyes testram1
a=dat1 1      a=dat1 16
gonc load     gonc load
testram1      testram1
lchex 80000   lchex 80000
?a>=c a       ?a>=c a
goyes testram3 goyes testram3
lchex 7f000   lchex 7f000
?a>=c a       ?a>=c a
goyes noram1  goyes noram1
lchex e000    lchex e000
gosub sub0    gosub sub0
gonc load     gonc load
noram1 gosub sub00  noram1 gosub sub00
cd0ex         cd0ex
config        config
cd0ex         cd0ex
config        config
gonc load     gonc load
testram3      testram3
lchex c0000   lchex c0000
?a>=c a       ?a>=c a
goyes ram3    goyes ram3
gosub gram    gosub gram
c=r2          c=r2
d1=c          d1=c
a=dat1 1      a=dat1 16
r4=a         r4=a
rtn          rtn
ram3 gosbvl #70b66  ram3 gosbvl #70b66
a=dat1 1      a=dat1 16
gosbvl #70bb7 gosbvl #70bb7
load gosbvl =AllowIntr load gosbvl =Allow...
c=r1          c=r1
d0=c          d0=c
lcase '9'
acex p
?c<=a p
goyes inf10   goyes inf10
c=c+con b,7
inf10 dat0=c b      dat0=a 16
d0=d0+ 2       d0=d0+ 16
d1=d1+ 1       d1=d1+ 16
decr b=b-1 a    decr b=b-1 x
goc out        goc out
goto loop      goto loop

```

```

out a=r0
govlng =GPOverWrAL
gram cd1ex
r2=c
c=b a
r3=c
c=rstk
p= 12
cpex 4
p= 0
gosbvl #700fe
c=r3
b=c a
c=r2
d1=c
a=r4
gonc load
sub00 uncnfg
sub0 uncnfg
a=dat1 1
cd0ex
lchex ff000
config
cd0ex
config
rtnc
trans ad0ex
d0=a
gosbvl =SAVPTR
c=rstk
d0=c
a=a-c a
lchex 80319
rstk=c
d1=c
c=a a
gosbvl =MOVEDOWN
govlng =GETPTR
ENDCODE
;
;

out a=r0
govlng =GPOve...
gram cd1ex
r2=c
c=b a
r3=c
c=rstk
p= 12
cpex 4
p= 0
gosbvl #700fe
c=r3
b=c a
c=r2
d1=c
a=r4
gonc load
sub00 uncnfg
sub0 uncnfg
a=dat1 16
cd0ex
lchex ff000
config
cd0ex
config
rtnc
trans ad0ex
d0=a
gosbvl =SAVPTR
c=rstk
d0=c
a=a-c a
lchex 80319
rstk=c
d1=c
c=a a
gosbvl =MOVEDOWN
govlng =GETPTR
ENDCODE
;
;

```

Vous trouverez les listes de codes de ces deux programmes dans le coin des codes.

### Indiscretions confortables

Pour ceux qui ont la chance de posséder un PC, il est plus confortable de désassembler les Roms HP48 ou d'éditer le code sur un PC. Après avoir utilisé le programme VOYAGER des Goodies Disk #2, j'ai adopté le décompilateur pour PC et sous DOS : MON48 de notre ami Jean-François Garnier.

Par rapport à l'autre décompilateur utilisant un fichier Ascii du code de la Rom HP48, MON48 utilise un fichier binaire donc deux fois moins gourmand en mémoire. Ceci est particulièrement appréciable

lorsque les fichiers de plusieurs versions de Roms cohabitent sur le disque dur.

Le transfert de la Rom des HP48Sx pouvait se faire aisément grâce à son éditeur de mémoire et Kermit. Cela n'est plus réalisable avec les HP48Gx.

Il serait possible de transférer la Rom HP48 via ROMPEEK ou XPEEK mais cela présenterait deux gros inconvénients :

- Les 512K de Rom (fragmentée) seraient transformés en 1024K de chaînes et les programmes lisant un quartet à la fois, puis faisant la conversion ascii cela n'est pas favorable à la rapidité.
- Puis pour le transfert il y aurait à la fois le temps et l'usure des piles.

Le programme ci-dessus XPEEK32K permet d'opérer de la manière suivante :

Les données dans une chaîne représentant un nombre pair de quartets, il suffira de poker dans une chaîne un nombre pair de quartets de la Rom sans faire de conversion ascii, d'où des chaînes 2 fois moins grosses qu'avec ROMPEEK ou XPEEK. D'autre part les zones de Rom cachée sont toujours à des adresses et sur des longueurs de code multiples de 16 ce qui permet au programme de prélever 16 quartets à la fois donc pratiquement 16 fois plus rapidement que XPEEK.

Suivant ce principe, les 512K de Rom sont prélevés en 16 blocs de 32K. Ces 16 fichiers seront envoyés en mode binaire, via Kermit, dans le PC.

Le transfert HP->PC devant être très fiable pour récupérer l'intégralité de la Rom (sans décalage) je n'envisage que le transfert sous Kermit.

Sur le PC un programme réassemblera les 16 blocs de code de Rom en un seul bloc de 512K. Pour faire cela j'ai réalisé un mini programme en Turbo Basic que vous trouverez plus bas et avec quelques explications complémentaires.

Pour automatiser le transfert des 512K de la Rom voici un petit programme qui utilise XPEEK32K. Il suffira donc de connecter la HP48 au PC, de lancer Kermit en 9600 bauds et mode serveur, faire de la place en Ram sur HP48, XPEEK32K étant présent le programme ROM48G+ se chargera du reste.

Utile précaution : assurez vous d'avoir des piles en état de fraîcheur.

En cas de problème lors des transferts, il vous suffirait de modifier ce programme de la manière suivante :

Si le transfert s'est bien effectué pour ROM1 à ROMn, modifiez le départ de la boucle avec n+1 au lieu de 1 et remplacez # 0h en 3ième ligne par la valeur correspondant à # 10000h \* n. Purgez la ou les variables ROMx éventuellement. Relancez Kermit puis ROM48G+.

Voici donc ce programme en User Rpl.

#### ROM48G+

cksum: # 52B3h 314 octets

```
« RCLF 'FLAGS' STO
  STD HEX -33 CF -35 SF
  # 0h DUP 'DEBUP' STO 'DEBUN' STO
  0 'P' STO
  1 16
  FOR I
    "ROM" P + # 5B15h SYSEVAL PURGE
    DEBUN DUP 'DEBUP' STO
    XPEEK32K "ROM" I DUP 'P' STO
    + OBJ→ SWAP OVER STO
    DEBUP # 10000h + 'DEBUN' STO
    SEND CLOSEIO
  NEXT FLAGS STOF
»
```

Après l'assemblage sur le PC des fichiers binaires ROM1 à ROM16 vous obtiendrez le code des 512K de la Rom en un unique fichier ROM48.BIN utilisable par MON48.

Les possesseurs de HP48G avec 32K de Ram doivent adapter ces programmes en fonction de la place mémoire disponible. Si certains expriment le désir que je fasse ces modifications moi-même, alors qu'ils n'hésitent pas à se manifester.

#### Et un peu de Basic...

Comme promis voici le programme en Turbo Basic permettant de reconstituer le code de la Rom des HP48Gx dans un fichier binaire DOS.

Le principe:

Chaque fichier ROM1 à ROM16 a l'en-tête Kermit HPH48-x où x est la version de la HP48, suit le code "c2A20" pour la chaîne transférée, puis les 5 quartets de la longueur de la chaîne ce qui fait au total 8 + 5 = 13 octets à sauter pour pointer sur le début du code de la partie de Rom HP48.

Pour chacun des fichiers ROM1 à ROM16 on pointe sur l'octet n° 14 (13 pour SEEK) puis on lit le maximum pour Turbo Basic : 32767 octets qui seront chargés dans le fichier ROM48.BIN. La lecture du dernier octet,

pour faire 32K, puis le chargement termineront le transfert d'un module de 32K.

Il ne restera plus qu'à détruire les fichiers ROM1 à ROM16 et, si vous possédez plusieurs Roms, à renommer ROM48.BIN en ROM48x.BIN suivant la version de la Rom x = ...M, P,...

Vous voilà prêts pour jouer les curieux avec le décompilateur MON48 de Jean-François Garnier.

#### Remarque:

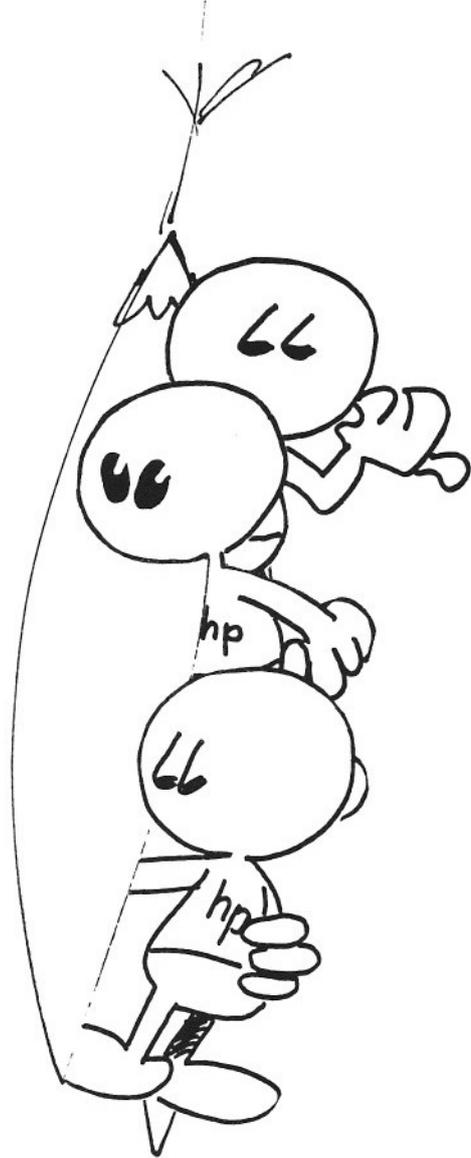
Etant donnée la très grande simplicité du principe il est très facile de transposer le programme ci-dessous dans un autre langage.

#### ROM48ADD.BAS

```
' auteur Guy Toublanc  
,  
' programme à compiler en Turbo Basic et  
' pour assembler les fichiers HP-48 ROM1 à ROM16  
' générés par les programmes HP48 ROM48GX et  
' XPEEK32K et reconstituer dans le fichier DOS  
' ROM48.BIN le code des 512K de la Rom HP48Gx  
' ce fichier binaire est utilisable avec MON48,  
' le décompilateur de Jean-François Garnier
```

```
close  
f0$ = "rom48.bin"  
open f0$ for binary as #1  
for i%=1 to 16  
f1$ = "rom" + mid$(str$(i%),2)  
open f1$ for binary as #2  
seek 2,13  
get$ #2,32767,c$  
put$ #1,c$  
get$ #2,1,c$  
put$ #1,c$  
close #2  
next  
close  
print "fin"  
end
```

Guy Toublanc (276)







```

sub00 uncnfg
sub0  a=dat1 1
      rstk=c
      lchex ff000
      c=stkg
      config
      rtncc

end

*****
* trans : sauvegarde le nombre de codes à peeker et l'adresse de début du
* code
* détecte le type de Rom HP48S(X) ou HP48G(X)
* transfère la partie de programme entre start et end
* dans le buffer RS232c correspondant au type de machine
entrées : l'entier binaire n au niveau 2 pour l'adresse du peek
          (quartets ou nibbles) à peeker
          l'adresse de start sur la pile des retours
          D0 pointe sur end
* sorties : R1(A) := nombre n de quartets à peeker
            R2(A) := adresse début du code à peeker
            début de ce programme transféré en
            # 701FCh pour HP48S(X)
            # 80319h pour HP48G(X)
            SI(0) = 0 pour HP48G(X) et = 1 pour HP48S(X)
            message d'erreur "Bad Argument Value" et arrêt si pour
            HP48S(X) l'adresse du peek + nombre de codes > # 80000h
*****

```

```

trans ad0ex
      d0=a
      r0=a
      gosbvl =PopASavptr
      d0=a
      d0=d0+ 10
      a=dat0 a
      r1=a
      c=dat1 a
      d0=c
      d0=d0+ 10
      c=dat0 a
      r2=c
      a=a+c
      d0=(5) #0011b
      c=dat0 a

      st=0 0
      c=c+c a
      goc hp48g
      st=1 0
      lchex 80000
      ?a<c a
      goyes ok48s
      lc(5) =argvalerr
      govling =gPErrJmpC

      a=r0
      c=rstk
      d0=c
      a=a-c
      lc(5) a
      ?st=0 0
      goyes HP48G

*
      * adresse fin programme (end)
      * sauvegarde cette adresse
      * A(A) := adresse nombre nibs, DROP, SAVPTR
      * @ entier binaire 1er niveau
      * nombre de quartets à peeker
      * sauvegarde
      * adresse entier binaire 2ième niveau
      * adresse où peeker
      * sauvegarde
      * fin du peek pour hp48s
      * C(4) := 7 pour Ram interne HP48G(X)
              8 pour Ram interne HP48S(X)
      * pour HP48G(X)
      * pour tenter de mettre Carry à 1
      * si HP48G(X) 8 + 8 -> dépassement -> Carry
      * pour HP48S(X)
      * vérification adresse du peek + n <= 80000

      * hors Rom HP48S(X) : Bad Argument Value
      * GETPTR et sortie avec erreur

      * adresse fin programme (end)
      * adresse début programme (start)
      * pour début mémoire à déplacer
      * fin - début := longueur à déplacer
      * adresse buffer RS232c HP48G(X)
      * Rom HP48G(X) ?

```

```

HP48G
*
lc(5) =uart_buffers
rstk=c
d1=c
c=a
gosbvl =MOVEDOWN
govling =GETPTR

*
* sinon adresse buffer RS232c HP48S(X)
* mise sur la pile pour exécution du
  programme à partir de start
* pour début mémoire déplacée
* nombre de quartets à déplacer
* déplacement
* après GETPTR saut à la nouvelle position
  de start dans le buffer RS232c
*
* ENDCODE
*
*

```



## HP95 / HP100

J. Belin  
J. Belin  
J. Belin

Une nouvelle carte NewMédia 26  
Tracé de fougère fractale 26  
Récupération des fichiers de base de données 28

## JEU DE CARTES

New Média vient de commercialiser une nouvelle carte pour HP95 et HP100. Il s'agit de la *Flash Bak* qui, comme son nom l'indique est une carte Flash permettant d'effectuer une sauvegarde du disque C. Un logiciel intégré à la carte permet d'effectuer un *Back Up* d'une sélection de fichiers ou de l'image totale du disque. L'intérêt de cette carte consiste surtout en son faible prix : 49\$ pour la version 512 ko et 99\$ pour la version 1 Mo.

D'autre part, ayant discuté dernièrement avec un représentant de cette société, j'ai eu la confirmation qu'ils n'avaient pas en projet de version plus rapide de leur *PalmModem* pour HP95 (en PCMCIA version 1.0). Par contre, en ce qui concerne le HP100, je rappelle qu'ils disposent d'une version allant jusqu'à 14400 bits/s en fax et modem.

Il est à noter que les cartes New Media sont maintenant distribuées en France par la société Distrilogie.

Jacques Belin (123)

---

## A L'OMBRE DES FOUGERES FRACTALES...

De nombreux adhérents m'ont souvent reproché de publier des programmes dans des langages qu'ils ne maîtrisaient pas, comme le C ou l'assembleur... Or, je voudrais préciser que si je n'utilise plus le Basic, je n'ai absolument rien contre le fait que d'autres programment avec, et encore moins qu'ils publient des programmes dans ce langage !

La preuve ? Pour une fois, je vais changer mes habitudes, et vous présenter moi-même un programme en Basic ! Et j'espère que cela devrait suffire pour vous inciter à utiliser ce langage pour écrire des petits programmes, puis nous en faire profiter...

Le sujet qui va nous servir de d'exemple m'a été inspiré à la vision d'un programme HP48 paru dans un récent *48SXTANT*, la revue de notre ami Robert Pulluard.

Cependant, je dois avouer qu'une grande partie du texte est honteusement (si si !) recopié sur l'article original ! J'espère que Robert m'en excusera. Il faut dire que si je suis intéressé par ce genre de programmes, je n'y comprends strictement rien à ce type de mathématiques... Pour résumer, je n'ai fait que le portage du programme HP48 vers le HP95. Soit le plus facile, étant donné la simplicité du code de ce programme !

Le but de ce petit programme est d'afficher une des images fractales les plus spectaculaires, car semblant sortir tout droit du monde végétal : la fougère de Barnsley.

La théorie de cette fougère est un peu plus complexe que pour certaines autres images fractales (Mandelbrot, triangle de Sierpinski). En effet, si elle est "autosimilaire", ses règles de transformation ne sont pas aussi directes pour passer d'une "niveau" au suivant. Il s'agit ici de quatre transformations linéaires affines dont les coefficients de réduction sont différents. D'autre part, nous n'avons pas nous n'avons pas affaire à une ligne, comme pour la courbe de Koch, mais à une surface qui est reproduite en quatre exemplaires réduits, étirés, retournés puis juxtaposés (avec des recouvrements).

S'il existe des algorithmes pour effectuer ces transformations de manière littérale, ils sont cependant trop lourds et trop lents pour nos modestes moyens. Heureusement, la théorie montre qu'une figure telle que la fougère de Barnsley est un "attracteur" pour la transformation considérée, et ceci a comme conséquence que tout point appartenant à la figure résulte en un autre point de la figure lorsqu'on lui fait subir l'une de ces transformations. La solution est alors simple : opérons directement sur les points (pixels, en partant d'un point connu de la figure, tel que l'extrémité de la tige (point de coordonnées  $[0.5*w \ 0]$ , si "w" est la largeur de la figure) en lui appliquant l'une des transformations, puis, partant du point obtenu, en évaluant le point suivant de la même manière, et ainsi de suite jusqu'à ce qu'on ait suffisamment de points pour que la figure ressemble à quelque chose.

Chacune des quatre transformations (affines linéaires) est de la forme :

$$\begin{aligned}x_{n+1} &= a \times x_n + b \times y_n + e \\y_{n+1} &= c \times x_n + d \times y_n + f\end{aligned}$$

Connaissant les divers coefficients (a, b, c, d, e, f), il suffira donc d'appliquer ces transformations un peu au hasard pour voir apparaître sur l'écran de votre HP48 (non, de votre HP95, je recopie trop vite !) la fougère en question. C'est la méthode utilisée par le

programme ci-après, dans lequel le "hasard" est établi par la génération d'un nombre aléatoire au moyen de la fonction RND. Une astuce, cependant, pour obtenir une répartition "équitable" des différents points successifs consiste à "piper le dés", c'est à dire que selon la valeur du nombre aléatoire, la possibilité de choisir l'une ou l'autre des quatre transformations ne sera pas la même pour toutes. Un ensemble de tests permet de faire le tri et la sélection. La largeur des intervalles est respectivement de :

- 0.02 pour la tige,
- 0.17 pour la feuille du haut,
- 0.3 pour la feuille du bas,
- 0.781 pour la partie droite de la fougère.

Ce programme devant utiliser des fonctions graphiques, nous devons utiliser le *Swift!Basic*, qui est le seul Basic adapté au HP95. Heureusement, je sais que de nombreux adhérents le possèdent déjà (pour des programmes ne faisant pas appel aux spécificités du HP95, je vous rappelle que vous pouvez utiliser les Basics classiques du PC). Voici donc le listing du programme, tel que vous devrez le taper. Les signes dièse (#) indiquent des variables en simple précision. Notez que l'éditeur intégré au programme ajoutera automatiquement les caractères '#X' après chaque constante décimale. Je ne les ait pas laissés pour rendre le programme plus lisible.

```

INPUT "Nombre d'iterations : ";N
X#=0
Y#=64
GRAF
FOR I=1 TO N
  A#=RND(0)
  IF A#<0.03 THEN
    X1#=57.6
    Y1#=0.27*Y#
  ELSE
    IF A#<0.17 THEN
      X1#=-0.139*X#+0.263*Y#+65.644
      Y1#=0.246*X#+0.224*Y#-8.424
    ELSE
      IF A#<0.3 THEN
        X1#=0.17*X#+0.215*Y#+47.016
        Y1#=0.222*X#+0.176*Y#+20.896
      ELSE
        X1#=0.781*X#+0.034*Y#+12.384
        Y1#=-0.032*X#+0.739*Y#+63.18
      ENDIF
    ENDIF
  ENDIF
  X#=X1#
  Y#=Y1#
  DOT(Y#,128-X#)
NEXT I
SOUND(1000,250)
END

```

Par rapport au programme de Robert, j'ai aussi changé quelques constantes (la dernière de chaque ligne de calcul de X1 et Y1) afin d'adapter l'affichage à la taille de l'écran.

Le fonctionnement du programme est simple, puisqu'il suffit d'entrer le nombre d'itérations, et d'attendre le Bip final, tout en admirant l'image se former petit à petit. Très reposant après une journée de travail.

Si vous entrez la valeur 500, vous obtiendrez :



Pour une valeur de 1000...



Pour une valeur de 2500...



Et enfin pour 4000...



Il est bien sûr possible de continuer avec des valeurs supérieures, afin d'avoir une image plus précise (la tige complètement tracée, par exemple), mais cela sera au détriment du temps de calcul. La raison est que le nombre de pixels à tracer est limité et que plus le temps passera, plus le programme passera de temps à re-noircir des pixels déjà affichés...

Avant de terminer, je voudrais préciser (ce qui devrait faire enrager encore plus Robert Pulluard !) que malgré le fait que les vitesses des microprocesseurs des deux machines soient relativement proches, mon programme est près de quatre fois plus rapide, puisqu'il effectue 4000 itérations en 1'35" contre 6'15" pour la HP48 !

Il n'y a plus de respect pour les anciens...

Jacques Belin (123)

---

## UN DEBUG CONTRE UN-DELETE

Sur HP100, si nous récupérons un fichier effacé à l'aide de la commande `Undelete` du Filer, il y a de fortes chances que les derniers octets de ce fichier soient perdus. Si cela n'est pas trop gênant pour un fichier Texte (que nous pouvons corriger avec un simple éditeur), cela peut rendre un fichier binaire (programme ou données) inutilisable.

Cependant, les derniers octets des fichiers de base de données du HP100 (y compris les fichiers du carnet d'adresse ou de l'agenda, qui ont la même structure) ne sont pas toujours significatifs.

Il est donc possible de tenter de récupérer le fichier, mais dans certains cas seulement.

La "recette" qui va suivre a été (si mes souvenirs sont bons) communiquée par Everett Kaser. Je vous expliquerai ensuite ce que vous venez de faire.

### *La manipulation*

Tout d'abord, il est conseillé de sauvegarder le fichier (dans une autre unité, de préférence), car les manipulations sont un peu complexes, et il y a toujours un risque de faire une erreur.

Ensuite, nous allons utiliser `DEBUG` pour éditer le fichier. Pour cela, il est conseillé de passer sur la ligne de commande du Dos, en l'appelant à l'aide de l'icône Dos du Filer. Cela permettra donc de disposer de suffisamment de mémoire vive pour travailler.

Ensuite, si vous désirez récupérer le fichier `TOTO.PDB`, par exemple, exécuter la commande :

```
debug fichier.pdb
```

Après chargement du fichier, le programme affichera le caractère `'-'`, qui indique qu'il est en attente d'une commande de l'utilisateur.

Tapez alors :

```
d 112,115
```

Ce qui fera apparaître un "dump" du contenu du fichier. Supposons que nous obtenions ceci :

```
2116:0110      12 34 56 78      .4vx
```

Les données qui nous intéressent sont celle affichées au milieu. Si elles indiquent `"00 00 00 00"`, cela veut malheureusement dire qu'il n'est pas possible de récupérer le fichier.

Sinon, tapez la commande :

```
r cx
```

Le programme fera alors apparaître un nombre hexadécimal (qui n'a aucun intérêt pour nous) et vous demandera d'entre une valeur :

```
-r cx  
CX CDAD  
:
```

Il vous faudra alors taper le code hexa décimal des deux premiers octets du dump, mais inversés et sans espaces. Pour notre exemple, il nous faudra donc répondre :

```
-r cx  
CX CDAD  
:3412
```

Ensuite, tapez :

```
r bx
```

Encore une fois, le programme demandera une deuxième fois une entrée de données. Suivant le même méthode que précédemment (inversion; sans

espaces) entrez le code hexadécimal des deux octets suivants :

```
-r bx  
BX 0000  
:7856
```

Ensuite, tapez :

```
e 112
```

puis :

```
00<Espace>00<Espace>00<Espace>00<Entrée>
```

Le fichier est maintenant corrigé. Il ne reste plus qu'à le sauver sur le disque, puis quitter DEBUG :

```
w<Entrée>  
q<Entrée>
```

*Que venons nous de faire ?*

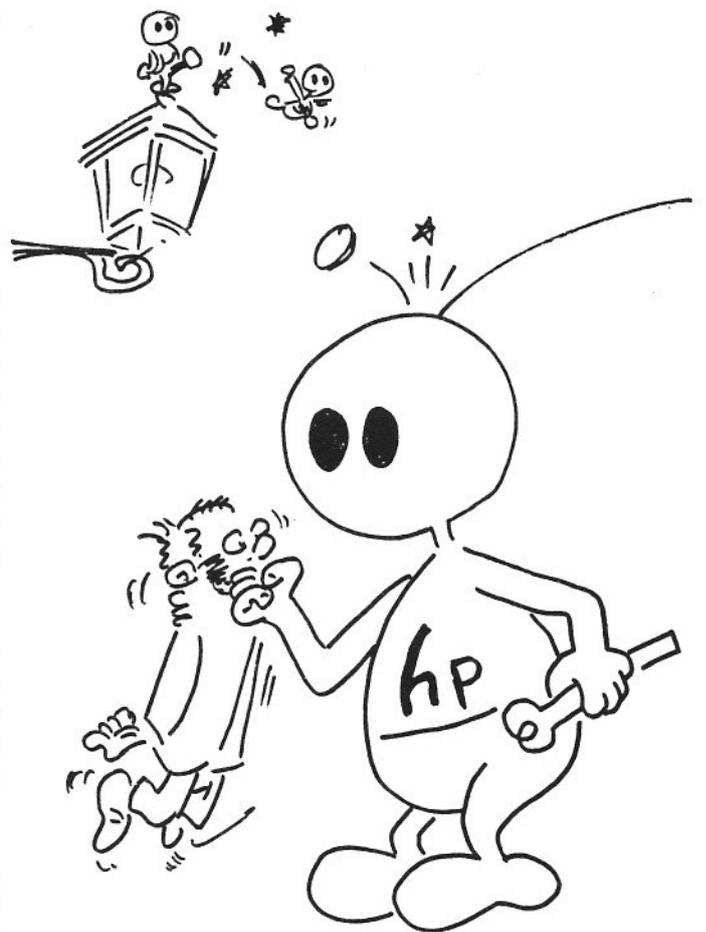
En fait, à l'adresse 12h du fichier de base de données (si nous avons entré le chiffre 112, c'est à cause d'une convention de DEBUG) se trouve une taille de fichier, codée sur 4 octets. Apparemment (je n'ai pas vérifié), cette taille correspond à l'encombrement des données effectivement utilisées, alors que la taille réelle du fichier est supérieure. Le programme ajustera la taille au cours d'une prochaine utilisation. Si cette valeur est égale à 0, cela voudrait dire qu'un ajustement de la taille a été fait.

La manipulation consiste donc à lire la taille du fichier, et à l'écrire dans les CX et BX, qui sont les registres du microprocesseur utilisés par la commande w de DEBUG (voir le chapitre concernant DEBUG dans votre manuel du Dos pour plus de détails de cette commande).

Entre l'initialisation des registres CX et BX et la réécriture du fichier, nous avons réinitialisé la zone de données à 0, afin d'interdire toute modification ultérieure.

Comme vous le voyez, si il est possible de récupérer certains fichiers, cela ne se fait pas sans mal (bien que je ne devrait pas tarder à vous proposer un programme faisant cela automatiquement), mais la récupération n'est possible que dans certains cas. Je ne ferais donc que vous conseiller de réfléchir à deux fois avant d'effacer un fichier. Car après tout, c'est encore la meilleure méthode pour ne pas avoir à utiliser Undelete...

Jacques Belin (123)





Donc à partir de maintenant pour tout assemblage de chaîne de codes procédez de manière suivante :

- 1- lancez le programme ASSCOD48 ou ASSCOD28
- 2- donnez le nombre d'octets (1/2 octet compris) puis validez avec ENTER.
- 3- tapez chaque ligne de codes, correspondant au numéro de ligne à 3 chiffres, sans les espaces et validez.
- 4- tapez la somme de contrôle et validez. S'il y a erreur la ligne de codes sera demandée à nouveau après émission d'un BEEP. L'appui sur EDIT fera apparaître la ligne des codes qui pourra être corrigée. Alors relancez avec CONT.
- 5- après l'affichage de "fin" stockez le programme assemblé dans la variable donnée en tête.
- 6- si tout s'est bien déroulé vous pouvez purger tmpcod qui contient la chaîne de codes.

### Programmes MS-DOS

Afin d'être utilisé par tous, ce programme est destiné à être écrit en GWBASIC. Il devrait cependant être facile de le convertir pour un autre programme (QBASIC, Turbo BASIC...).

#### Programme MAKEDOS.BAS

```
10 INPUT "Nom du fichier : ",NOM$ : INPUT "Nombre d'octets : ",N : N=N*2
20 OPEN NOM$ FOR OUTPUT AS #1 : CLOSE #1 : KILL NOM$
30 OPEN "bin.tmp" FOR OUTPUT AS #1 : S=0 : P$="-----"
40 NLINES=N\16 : LENLAST=(N MOD 16)+(N MOD 16)\5
50 IF (N MOD 16)=0 THEN NLINES=NLINES-1 : LENLAST=LEN(P$)
60 FOR X=0 TO NLINES
70 IF X=NLINES THEN P$=LEFT$(P$, LENLAST)
80 C$=P$
90 X2$="00"+HEX$(X) : PRINT RIGHT$(X2$,3);": ";
100 Y=CSRLIN : LOCATE Y,6 : PRINT C$; : LOCATE Y,6 : INPUT "",C$ : IF Y=24 THEN Y=Y-1
110 LOCATE Y,27 : PRINT " sm = ---" : LOCATE Y,33 : INPUT "",D$
120 M=S
130 FOR Z=1 TO LEN(C$)
140 IF MID$(C$,Z,1)<>" " THEN M=(M+((Z-(Z\5))*ASC(MID$(C$,Z,1)))) MOD 4096
150 NEXT Z
160 D2$="00"+HEX$(M) : D2$=RIGHT$(D2$,3)
170 IF D2$<>D$ THEN PRINT "Erreur de somme" : BEEP : GOTO 90
180 FOR Z=1 TO LEN(C$) STEP 2
190 IF MID$(C$,Z,1)=" " THEN Z=Z-1 : GOTO 230
200 CH=ASC(MID$(C$,Z,1))-48 : IF CH>9 THEN CH=CH-7
210 CL=ASC(MID$(C$,Z+1,1))-48 : IF CL>9 THEN CL=CL-7
220 PRINT#1,CHR$((16*CH)+CL);
230 NEXT Z
240 S=M
250 NEXT X
260 CLOSE #1 : NAME "bin.tmp" AS NOM$ : END
```

#### Mode d'emploi :

- 1- Lancer le programme : GWBASIC MAKEDOS.BAS
- 2- Entrer le nom du fichier destination.
- 3- Entrer la taille du fichier.
- 4- Entrer les listes de codes puis le checksum (en prenant soin d'entrer les codes hexadécimaux en majuscules). En cas d'erreur corriger la ligne, en prenant soin de placer le curseur après le dernier caractère avant de taper sur la touche d'entrée.
- 5- Une fois que toutes les lignes sont entrées, sortir du GWBASIC en exécutant la commande SYSTEM. Le nouveau programme est immédiatement disponible.

Note : La taille du fichier résultant peut être supérieure d'un octet à ce qui est affiché dans le listing. Cela n'est pas un problème.

SYMX (HP28)  
 # 5BA1h 41 octets

0123 4567 89AB CDEF sm

000: 69C2 0C40 008F 1805 C94  
 001: 0700 0071 351C F1C8 A60  
 002: 1B71 10C1 42D2 8086 71C  
 003: 2E08 0842 3102 5708 2A4  
 004: 0852 1401 4D8D 1E52 042  
 005: 10 0D3

XLIB-> (HP48)  
 # DA81h 68 octets

0123 4567 89AB CDEF sm

000: D9D2 0D29 51D9 D20C E87  
 001: CD20 0600 08FB 9760 BB7  
 002: 8FFC F164 5213 0142 791  
 003: 16A3 429E 208A 2318 46D  
 004: F2D7 608D B105 0130 078  
 005: 182D 2156 3109 1821 BFE  
 006: 5631 088F 9256 08D4 974  
 007: 65D2 12F0 6B05 91B2 6CA  
 008: 130B 2130 DFE

XMOD->KER (HP48)  
 # D03Ch 67.5 octets

0123 4567 89AB CDEF sm

000: D9D2 0D29 51D9 F81D F14  
 001: 0040 D9D2 OCCD 20A5 DBE  
 002: 0008 FB97 6014 3131 992  
 003: 1741 4717 4818 FA9D 860  
 004: 734C F820 4811 4313 419  
 005: 0142 8A27 1170 CF5C 289  
 006: E333 0208 D04F 0113 EBA  
 007: 38D2 7630 7566 0B21 B12  
 008: 30B2 130 0AA

XPEEK32K (HP48)  
 # 7AB0h 210.5 octets

0123 4567 89AB CDEF sm

000: D9D2 0D29 51D9 F81D F14  
 001: 5040 D9D2 OCCD 20D7 E10  
 002: 1007 4413 4000 018F A02

003: D7B5 032F FFD5 1471 836  
 004: 3517 9147 1358 F511 4AA  
 005: 1013 3131 1361 0934 FCC  
 006: 0410 08BE 01AE 28B2 E26  
 007: 80AF 05E7 3400 0E78 B74  
 008: BE90 15BF 5B63 4000 817  
 009: 088B EE23 4000 F78B 650  
 00A: EF03 3000 E72A 0564 2F1  
 00B: 7890 1368 0513 6805 E93  
 00C: 5333 4000 0C8B E517 C0B  
 00D: C401 1A13 515B F104 8E6  
 00E: 018F 66B0 715B F8F7 838  
 00F: BB07 8F5E 0101 1913 461  
 010: 4158 F16F 17FA 3D46 34C  
 011: 0693 F110 8D27 6301 F7C  
 012: 3710 AD91 0B07 2C80 CC8  
 013: F420 8FEF 0071 1BD5 B39  
 014: 11A1 3511 45AA 8048 830  
 015: 0415 BF13 6340 00FF 5F6  
 016: 8051 3680 5031 3213 0D8  
 017: 08FB 9760 0713 4EA3 E71  
 018: 4913 0806 135D 68FC CFC  
 019: 0760 8D2D 760B 2130 992  
 01A: B213 0 C87

XPEEKG (HP48)  
 # 404Ah 235 octets

0123 4567 89AB CDEF sm

000: D9D2 0D48 81D8 A819 E38  
 001: 6126 CB91 69F3 452B C4E  
 002: F81E CD46 D9D2 OCCD CD7  
 003: 20A9 1007 1618 FC15 A24  
 004: 2313 0169 1460 6C68 6D8  
 005: FD7B 5007 D514 7135 34B  
 006: 1791 4713 563D 08F5 0DA  
 007: 1110 1331 3113 6109 BD4  
 008: 3404 1008 BEF0 AE28 ABB  
 009: B270 D05E 7340 00E7 7C9  
 00A: 8BE9 015B 05B6 3400 430  
 00B: 0088 BEE2 3400 0F78 1C3  
 00C: BEF0 3300 0E72 B056 EAE  
 00D: 478A 0136 8051 3680 A65  
 00E: 5533 3400 00C8 BE51 7CC  
 00F: 7C50 11A1 3515 B010 39F  
 010: 4018 F66B 0715 B08F 1E1  
 011: 7BB0 78F5 E010 1191 E51  
 012: 3431 93A8 E98E 8081 C3A  
 013: 8626 14C1 6117 OCD4 9AB  
 014: 606A 2F11 08D2 7630 644  
 015: 1371 0AD9 10B0 72C8 401

016: 0F42 08FE F007 11BD 2B6  
 017: 511A 1351 145A 9804 F1F  
 018: 8041 5B01 3634 000F B4C  
 019: F805 1368 0503 1321 642  
 01A: 308F B976 0071 34EA 429  
 01B: 3491 3080 6135 D68F 1E7  
 01C: C076 08D2 D760 B213 EF4  
 01D: 0B21 30 321

XPEEKU (HP48)  
 # 9092h 287.5 octets

0123 4567 89AB CDEF sm

000: D9D2 0D48 81D8 A819 E38  
 001: 6126 CB91 69F3 452B C4E  
 002: F81E CD46 D9D2 OCCD CD7  
 003: 2030 2007 C611 19C6 9AA  
 004: 8FD7 B501 19D5 11A1 6DB  
 005: 3562 E08F 5111 0136 2BC  
 006: 1091 3313 1860 E034 EB5  
 007: 0000 78BE 0734 0410 A8C  
 008: 08BE FOAE 28B2 70D0 8FA  
 009: 5C73 4000 E78B E901 6AC  
 00A: 5B05 9634 0000 88BE 425  
 00B: C234 000F 78BE F033 251  
 00C: 000E 75C0 5447 BB00 FA6  
 00D: 6805 0780 5533 3400 AB5  
 00E: 00C8 BE51 7C50 11A1 7EF  
 00F: 3515 B010 4018 F66B 56F  
 010: 0715 B08F 7BB0 78F5 474  
 011: E010 1191 3431 93A8 0EC  
 012: E98E 8081 8626 14C1 DBB  
 013: 6117 OCD4 606B 1F11 B30  
 014: 08D2 7630 1371 0AD9 88A  
 015: 10B0 7860 212F 80F4 5EA  
 016: 8FFB 6205 112C 80F4 379  
 017: 208F EF00 711B D511 0ED  
 018: A135 1145 5880 4804 CA5  
 019: 15B0 0634 000F F805 9A5  
 01A: 0780 5031 3213 0100 3ED  
 01B: 8FC1 5231 3016 9142 F6D  
 01C: 1011 4713 4169 1461 ADD  
 01D: 0ACA 1BB1 1001 4684 748  
 01E: 0C64 F185 0340 0008 2AE  
 01F: 8BA0 1343 0200 8D04 EAC  
 020: F011 1007 134E A349 BA8  
 021: 1308 8609 034C F107 8AD  
 022: 0613 5D68 FC07 608D 721  
 023: 2D76 0B21 30B2 130 FAA

# INDEX DES ARTICLES PARUS EN 1993

## (JPC 83 à 92)

### Informations générales

Titre	Auteur	JPC	Page
Compte-rendu de la réunion Anniversaire	J. Belin	83	2
Compte-Rendu de l'Assemblée Générale	J. Belin	83	3
Echos sur les futures HP (48G/X & 100)	J. Belin	83	6
SOS	P. Tenand	83	6
Nouvelles Fraiches	J. Belin	85	2
Libres Propos	C. Vaillant	86	2
Un club HP en Espagne	J. Belin	86	4
Nouvelles coordonnées du Support Technique	J. Belin	86	4
L'OmniBook 300	J. Belin	87	2
Libres Propos (Acte II)	R. Pulluard	87	3
SOS	G. Toublanc	90	2
Droit de réponse	P. Courbis	91	2
Des nouveaux statuts pour PPC PARIS ?	J. Belin	92	2

### HP28

Titre	Auteur	JPC	Page
Quel Avenir pour la HP28 ?	G. Toublanc	91	6
Rappel aux auteurs d'articles	G. Toublanc	92	4

### Programmes et applications

Titre	Auteur	JPC	Page
Test du Chi2 d'indépendance	R. Oughlissi	83	8
Traitement de données statistiques	P. Jebeily	84	4
Communications et nouveaux outils	G. Toublanc	84	7
Fast-Orielles, version HP28	G. Toublanc	84	11
Gestionnaire de fichiers	P. Jebeily	85	4
Outils HP48 sur HP28	G. Toublanc	85	7
Réponses et factorielles	G. Toublanc	86	6
Fractions de PI	E. Dougados	87	6
Evaluation du polynôme de Newton	E. Dougados	87	6
Programmation System RPL sur HP28	J.F. Garnier	87	7
Points d'entrées compatibles HP48 (Acte I)	J.F. Garnier	87	8
Le binôme de Newton (Acte II)	G. Toublanc	88	4
TIMED pour HP28S	G. Toublanc	88	7
Points d'entrées compatibles HP48 (Acte 2)	J.F. Garnier	88	7
Programmation System Rpl, correctif	J.F. Garnier	88	9
Outils pour bidouilleurs	G. Toublanc	89	4

Titre	Auteur	JPC	Page
Désactivation de l'écran	G. Toublanc	90	4
Graphismes et animations	P. Jebeily	90	4
Recherche de Chaîne	G. Toublanc	91	6
Graphismes et animations, correctif	P. Jebeily	91	8
Passage en modes Symbolique/Numérique	G. Toublanc	92	4

## HP48

Titre	Auteur	JPC	Page
Sur l'attribution de paternité des programmes	G. Toublanc	83	17
Connections entre HP48	G. Toublanc	85	17
Présentation de la RPL48 Toolkit	C. Vaillant	86	12
Détournements mineurs	C. Vaillant	86	14
Détournements majeurs	G. Toublanc	86	18
Transferts de fichiers sur HP48GX	G. Toublanc	89	8
Des bugs sur les HP48G(X)	J. Belin	89	8
Programme bizarre et dérives littéraires	G. Toublanc	90	14
Courrier des lecteurs	G. Toublanc	92	8

## Nouveautés

Titre	Auteur	JPC	Page
Nouvelles cartes STORC	J. Belin	83	12
Présentation des Goodies Disks	J.F. Garnier	83	12
Notes de lecture : Maîtrise de la HP48	G. Toublanc	83	13
Nouvelle version du kit de développement	J. Belin	84	16
Présentations des Goodies disks 8	J.F. Garnier	85	14
La HP48GX	J. Belin/P. de Sacy	87	15
Les Secrets de la HP48 tome II	P. Silvestre de Sacy	87	17
Les cartes Ram 1Mo Hewlett-Packard	G. Toublanc	90	14
Le kit de connection HP48GX	G. Toublanc	91	10
Présentation de CAT-48	G. Toublanc	91	10
HP48 en Prépa : 2ème édition.	G. Toublanc	92	6

## Programmes et applications

Titre	Auteur	JPC	Page
Changement de Répertoire	P. Silvestre de Sacy	83	14
Trucs et Astuces	G. Toublanc	83	16
Optimisation d'enregistrements musicaux	J.M. Mermet	84	14
Récupération de port	P. Silvestre de Sacy	84	15
Extension des menus	P. Silvestre de Sacy	85	15
Tracé de polyèdres	A. Buchmann	85	16

Titre	Auteur	JPC	Page
Désactivation de l'écran	G. Toublanc	90	4
Recherche de chaînes dans la mémoire	L. Grand	85	18
Récupération de port, correctif	P. Silvestre de Sacy	85	19
Résolution de polynômes, correctif	L. Grand	85	20
La HP48 mène grand train	R. Pulluard	86	8
Récurtivité et grains de sable	P. Leckler	86	11
Nouveaux points d'entrées supportés	J.F. Garnier	86	19
Mise en majuscules	L. Grand	86	22
Extension des menus (Acte II)	J.F. Garnier	87	18
Programmation System RPL	J.F. Garnier	87	18
Trucs et Astuces	G. Toublanc	87	19
Le binôme de Newton (Acte II bis)	G. Toublanc	88	12
Aides au désassemblage	G. Toublanc	88	12
Récupération des mnémoniques HP	G. Toublanc	88	15
REV : Nouvelle version	G. Toublanc	88	15
Les triangles de Sierpinski	G. Toublanc	89	9
Compatibilité et HP48	G. Toublanc	89	15
Calculs astronomiques	J.F. Garnier	90	18
Calcul du numéro de semaine	R. Pulluard	91	11
Calculs hyperboliques	G. Toublanc	91	12
Trucs et astuces	G. Toublanc	91	13
Points d'entrées correspondant SX & GX	P. Silvestre de Sacy	91	15
Cartes TDS sur le port 2 (48GX)	P. Silvestre de Sacy	91	16
Changement de répertoire (Acte II)	P. Silvestre de Sacy	91	18
Numéro de semaine (acte 2)	G. Toublanc	92	10
Voyage bizarre ... et code nul.	G. Toublanc	92	12
Trucs sur l'éditions de chaînes	G. Toublanc	92	12
Traduction des messages d'erreurs	A. Ripoll	92	14
Attachement de librairies	P. Silvestre de Sacy	92	15
Choc en retour	G. Toublanc	92	20

## HP71

Titre	Auteur	JPC	Page
Triangles de Sierpinski sur HP-71	G. Toublanc	89	18

## HP95 / HP100 / MS-DOS

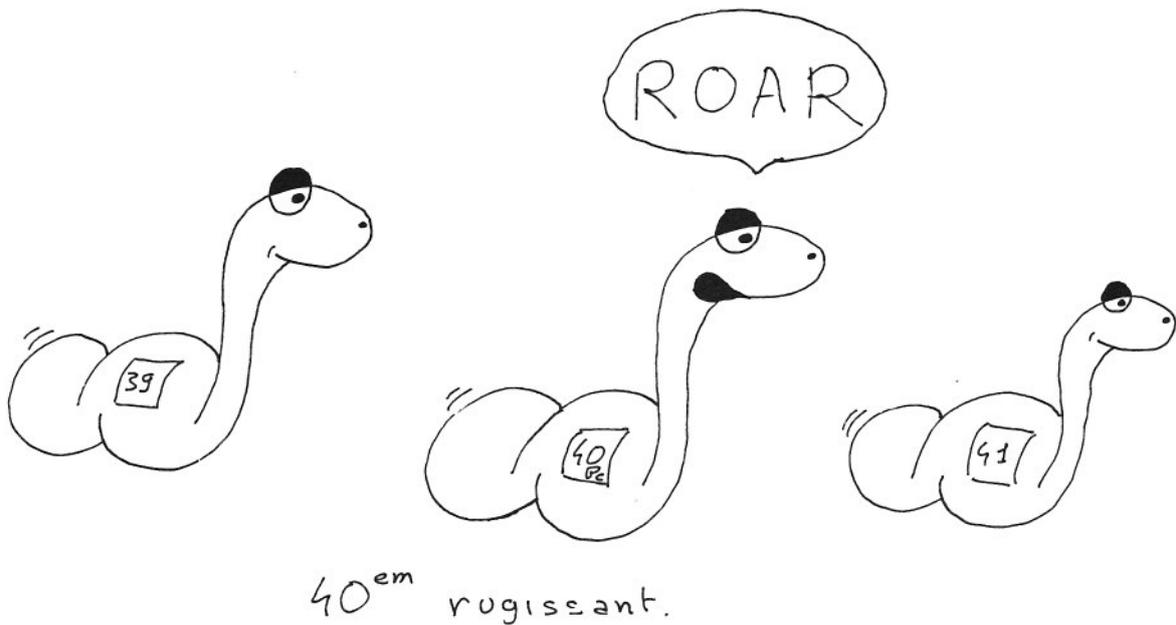
Titre	Auteur	JPC	Page
Jeu de cartes (PCMCIA)	E. Gengoux	86	27
Configuration de QEDIT sur HP95	J. Belin	89	22
La configuration du Drive95	G. Merlier	91	22
Conversion d'images pour HP95	J. Belin	91	25
Conversion d'images (Acte II) (HP100)	J. Belin	92	26

## Nouveautés

Titre	Auteur	JPC	Page
Banc d'essai des cartes SunDisk	J. Belin	83	20
Nouveaux Produits	J. Belin	83	23
Le nouveau HP100LX	J. Belin	86	24
Les Goodies Disks HP95	J. Belin	88	18
PC de Poche : HP100 & 95LX (livre)	J. Belin	92	22

## Programmes et applications

Titre	Auteur	JPC	Page
Contrast (Acte II)	J. Belin	83	24
La programmation du graphisme sur HP95	J. Belin	84	18
Programmation graphique, correctif	J. Belin	85	22
Editeur de motifs de remplissages	J. Belin	85	22
Visualisation d'encombrement du disque	J. Belin	87	22
Correction de Q.C.M. avec Lotus	P. Guerin	88	19
Recherche de fichiers sur disque	J. Belin	88	22
Panneau de paramétrage du HP95	J. Belin	90	24
Programmation du graphisme, correctif	J. Belin	91	28
Configuration commune des cartes Ram	J. Belin	92	24
Commande de sélection de mode écran (HP100)	J. Belin	92	27



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

PPC Paris est le représentant Français de HEX (Handhelds European Clubs EXchange), la fédération des principaux clubs Européens d'utilisateurs de calculateurs HP.

La maquette de ce numéro a été préparée et réalisée par Jacques Belin et Asdin Aoufi.

Les dessins sont de Jean-Jacques Dhénin et Paul Courbis.

Les informations et programmes parus dans ce journal sont publiées "Tels quels" et ne peuvent en aucun cas engager la responsabilité de Hewlett-Packard ou de PPC Paris. Hewlett-Packard se réserve le droit de ne pas répondre aux questions concernant le sujet de certains articles.

Les programmes publiés peuvent être utilisés librement. Cependant, ils ne peuvent être vendus ou fournis dans un ensemble commercialisé, sous quelque forme que ce soit, sans l'accord écrit de l'auteur ou de PPC Paris.

Directeur de la publication : Jacques Belin  
Numéro ISSN : 0762 - 381X

Veuillez adresser toute correspondance à :  
PPC Paris, BP 604, 75028 Paris Cedex 01.

Imprimé par Paris Copie, 4 rue Linné, 75005 Paris