CHAPITRE CHHU

# JPC

OCTOBRE 1986
NUMERO 38

Le numéro 35 FF.

## A PROPOS DU CLUB


| | | |
|---|---|---|
| P. David & J. Taillandier | Editorial | *1* |
| P. David | PPC Paris se réunit | *2* |
| J.J. Dhénin | Ah ! Vous écrivez | *2* |
| | Courrier des lecteurs | *3* |
| | Courrier du coeur | *4* |
| E. Gengoux | Choc en retour | *5* |


## HP41


| | | |
|---|---|---|
| L. Ancelet | Programmes astronomiques | *8* |


## HP71


| | | |
|---|---|---|
| G. Toublanc | Le nouveau DIVILEX | *18* |
| J.P. Bondu | Un Lex pour vos Basic | *24* |
| A. Goubault | Forthément complexe | *34* |
| P. David | Le coin des Lhex | *41* |


Edité par PPC-PC : rue Jean-Jacques Rousseau - 75001 PARIS

# EDITORIAL

Ave à tous

Nous rédigeons cet éditorial le 17 octobre 1986. Le retard se comble peu à peu. Le prochain numéro devrait être dans les temps. Enfin, nous espérons...

Ne vous laissez pas abuser par la "minceur" du sommaire : trois articles dont la longueur égale la qualité lui donnent cet aspect particulier.

Nous rappelons à votre attention l'Assemblée Générale qui aura lieu, comme prévu, le 7 décembre 1986. A noter dans vos tablettes...

Nous vous quittons maintenant, car il reste beaucoup de travail pour le prochain numéro. Bonne lecture.

A bientôt,

Pierre David
Janick Taillandier

## PPC PARIS SE REUNIT UNE FOIS PAR MOIS

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

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

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

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

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

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

- Métro Strasbourg Saint Denis :
Sortie porte St Martin / Bd St Denis, coté pairs

- Métro République :
Sortie Bd St Martin, coté pairs

- Métro Jacques Bonsergent :
Sortie Bd Magenta, coté impairs.

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

Les dates des prochaines réunions sont :

Samedi 4 octobre 1986
Samedi 15 novembre 1986
Samedi 6 décembre 1986
Samedi 17 janvier 1987
Samedi 7 février 1987
Samedi 7 mars 1987
Samedi 4 avril 1987
Samedi 16 mai 1987
Samedi 6 juin 1987

Pierre DAVID
(P37, SIG1, CHHU616)

---

## AH ! VOUS ECRIVEZ

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

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

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

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

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

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

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

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

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

Jean-Jacques Dhénin

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

---

Alain GOUBAULT DE BRUGIERE
27, Avenue de Brimont
78400 Chatou

Cher ami,

Je profite de l'occasion pour donner mon point de vue et échanger quelques idées. Je viens de lire le dernier numéro de JPC (35) et je tiens à féliciter tous ceux qui ont participé à sa réalisation. La présentation est remarquable. Pour ce qui est du contenu, tu l'avoues toi-même dans l'éditorial : ça plane haut, très haut même !

Je travaille actuellement sur un HP71 avec "simplement" un lecteur de cartes (minimum oblige) et quand même le module FORTH/ASSEMBLEUR ; j'ai acquis ce dernier pour le FORTH qui est un langage "sympa" et surtout bien adapté à ce genre de machine. Je ne suis pas encore familier avec le langage machine pour profiter pleinement ; jusqu'à présent, les Lex que j'utilise dans mes applications (scientifiques) ont été récupérés tout faits dans JPC et finalement, je me demande si je n'aurais pas dû lui préférer le module "Translator PAC" (dictionnaire FORTH étendu, simulation HP41, mode CALC amélioré). Bah ! je ne désespère pas de faire mes propres fonctions, même si je n'ai pas l'âme d'un SIGouilleur. Cela dit, je me demande s'il reste quelque chose d'intéressant à réaliser qui puisse être récupéré dans une application : à mon avis, la prochaine étape serait que le Club réalise un module (du genre ROM I/O April du HP75), qui rassemblerait les réalisations les plus importantes, les plus intéressantes. Je ne serais pas surpris d'apprendre que l'article "Assemblage d'enfer" dans JPC 35 est un prélude à ce qui se prépare ; et puisque nous y sommes, pourquoi pas un compilateur Basic ?

Mes applications sont donc plutôt scientifiques (je suis ingénieur, tout s'explique) et j'ai plutôt besoin de programmes utilitaires appliqués aux mathématiques, à la physique ou la mécanique (RDM). Il me tarde d'ailleurs de me procurer le module MATHS (question : puis-je me le procurer par l'intermédiaire du Club, et à quel prix ?).

Bien sûr, mon rêve serait d'avoir quelques périphériques HPIL (au moins imprimante et stockage de masse, mais bon sang ! que c'est cher !) et plus de mémoire (les modules HHP de 32 Ko sont abordables, mais j'ai entendu dire qu'il y a un problème si on les raccorde à la mémoire principale à cause de fichiers qui pourraient être à cheval sur ces deux mémoires. Qu'en est-il exactement ?).

Alain Goubault de Brugière

Réponse du docteur JPC :

Il est vrai que le niveau des articles est souvent très élevé dans JPC. Nous voudrions tous que cela change. Mais pas par un nivellement par le bas : nous ne sommes pas ridicules dans le domaine de l'assembleur HP71, et il serait dommage de perdre notre réputation. Cependant, ce souci ne doit pas nous faire perdre de vue que chacun doit pouvoir trouver de l'intérêt dans JPC. Alors, si ce sont toujours les mêmes qui écrivent, cela ne changera pas... Suivez donc l'exemple d'Alain Gillet !

Je pense que ton module Forth/Assembleur est un achat intelligent, que tu ne regretteras pas. Tu liras dans un prochain JPC ce que nous pensons du module "Translator Pac". Et nul n'a besoin d'être SIGouilleur pour utiliser les nouvelles fonctions qui paraissent. Quant à les faire soi-même, c'est un pas que peu franchissent. C'est d'ailleurs pour cela que Pierre David est en train de rédiger un petit article d'initiation, en espérant que cela permette à plus de membres de s'y mettre.

Quant à une Rom PPC-Paris, il n'en est pas encore question. Cela représente un travail monumental, et nos week-ends sont déjà pris par la fabrication du Journal. Un jour, peut-être...

Tu peux te procurer le matériel à 25% du prix officiel HP en t'adressant au Club. Les modules 32Ko se plaçant dans les ports avant du HP71 sont maintenant disponibles chez EduCalc aux Etats-Unis, au prix de 150 $. Il n'y a pas de problème de raccord à la mémoire principale.

Philippe Guez

---

Michel MARTINET
55 rue Véron
94140 ALFORTVILLE
Tél: 43 78 62 50

Vend :
Un HP75C + Visicalc + Text Formatter
pour 3500.00 Frs.

François LE GRAND
rue de la Poterie
Trévignon
29128 TREGUNC
tél : 98 50 00 10

Vend :
1 Lecteur de cartes HP-71 + 75 cartes : 1000 FF
1 Module FORTH-ASSEMBLER : 800 FF
1 Lecteur de Cassettes : 2000 FF

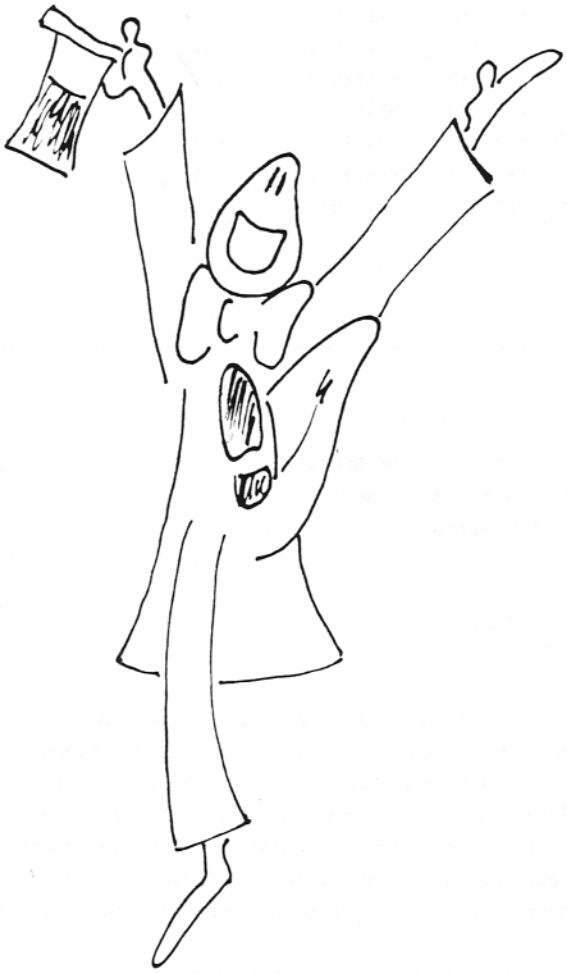


Figure IV.7 – Types d'arguments et modes de passage : définition, possibilités et incompatibilités.

## CHOC EN RETOUR

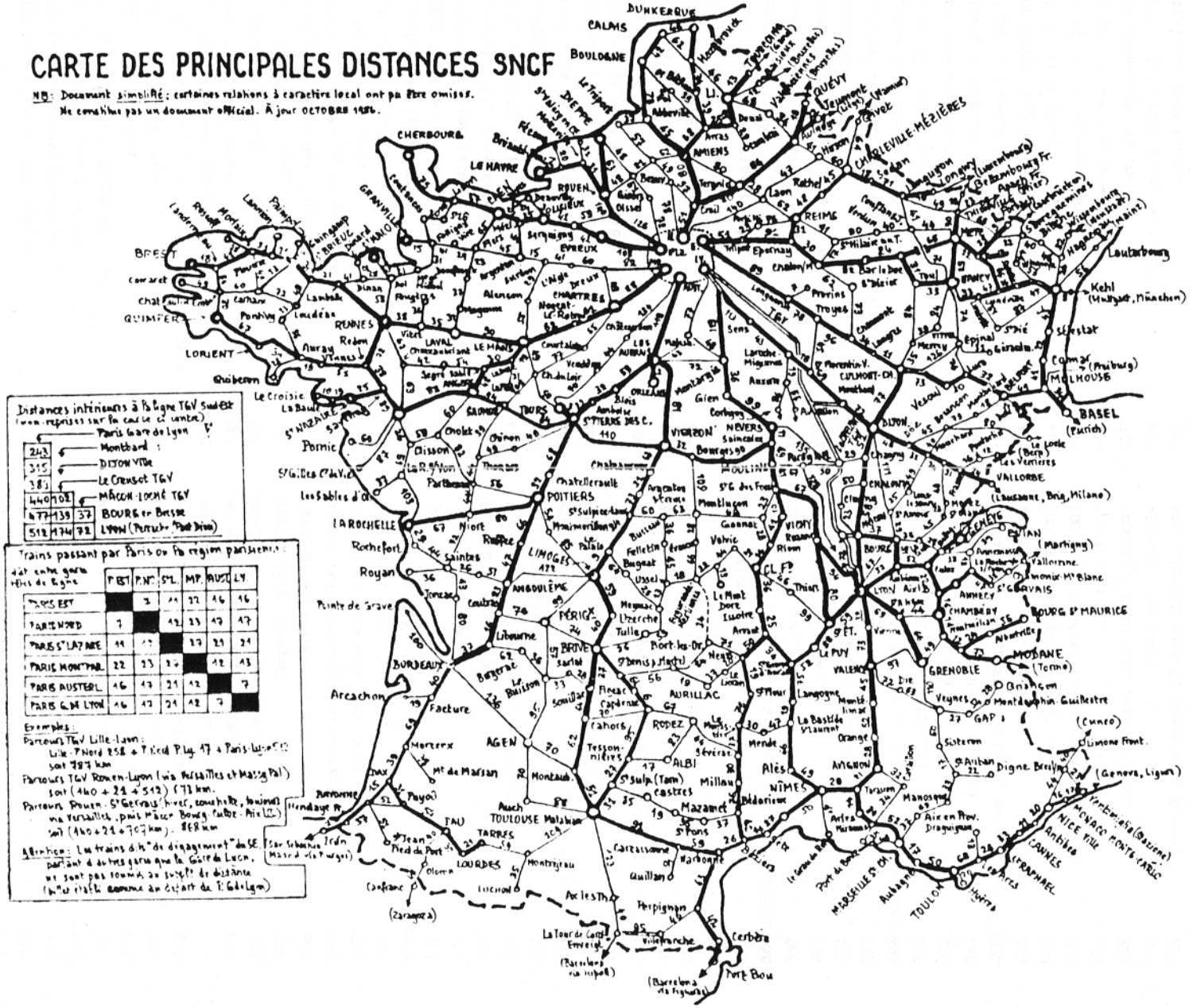
L'article "billet doux" (JPC36) de Eric Gengoux est complété par cette table, ainsi que la carte :

Quelques distances au départ de Paris :

| | |
|---|---|
| AGEN (via Vierzon, Périgueux) | 651 |
| AGEN (via Poitiers, Bordeaux) | 717 |
| AIX EN PROVENCE (via Marseille) | 900 |
| AIX LES BAINS (via Bourg ou Lyon) | 582 |
| ANGERS (via Le Mans) | 308 |
| ANGOULEME (via Poitiers) | 445 |
| ANNECY (via Bourg ou Lyon-St André le Gaz) | 621 |
| ANTIBES (via Lyon, Marseille) | 1068 |
| ARCACHON (via Poitiers, Bordeaux) | 640 |
| AVIGNON (via Lyon) | 742 |
| BASEL SBB (via Troyes, Belfort) | 526 |
| LA BAULE (via Le Mans, Nantes) | 479 |
| BAYONNE (via Poitiers, Bordeaux) | 780 |
| BELFORT (via Troyes) | 743 |
| BESANCON (via Dijon) | 406 |
| BEZIERS (via Lyon, Nîmes) | 912 |
| BIARRITZ (via Poitiers, Bordeaux) | 790 |
| BORDEAUX St Jean (via Poitiers) | 581 |
| BOULOGNE VILLE (via Amiens) | 254 |
| BOURG EN BRESSE (via Dijon ou TGV) | 477 |
| BOURGES (via Vierzon) | 232 |
| BREST (via Le Mans, Rennes) | 622 |
| BRIANCON (via Lyon, Valence) | 861 |
| BRIVE LA GAILLARDE (via Vierzon, Limoges) | 499 |
| CAEN (via Evreux) | 299 |
| CALAIS (via Amiens) | 296 |
| CALAIS MARITIME (via Amiens) | 299 |
| CANNES (via Lyon, Marseille) | 1057 |
| CARCASSONNE (via Vierzon, Toulouse) | 804 |
| CHALON SUR SAONE (via Dijon) | 382 |
| CHAMONIX-MONT-BLANC (via TGV ou Bourg ou Lyon) | 734 |
| CHAMBERY (via Dijon, Bourg, Culoz, ou TGV) | 596 |
| CHERBOURG (via Caen) | 371 |
| CLERMONT-FERRAND (via Vierzon ou Montargis) | 420 |
| COLMAR (via Troyes, Mulhouse) | 533 |
| LE CREUSOT (via TGV) | 389 |
| DAX (via Bordeaux) | 729 |
| DIEPPE (via Rouen ou Gisors) | 171 |
| DIJON (via Laroche-Migennes ou TGV) | 315 |
| DUNKERQUE (via Amiens, Arras, Hazebrouck) | 312 |
| EPINAL (via Nancy) | 427 |
| EVIAN (via Bourg ou Lyon) | 670 |
| FORBACH Frontière (via Metz) | 429 |
| GAP (via Lyon, Valence) | 779 |
| GENEVE (via TGV, Bourg ou Lyon) | 626 |
| GRANVILLE (via Flers) | 328 |
| GRENOBLE (via TGV, Lyon) | 641 |
| LE HAVRE (via Rouen) | 228 |
| HENDAYE Frontière (via Bordeaux, Dax) | 817 |
| JEUMONT Frontière (via St Quentin) | 241 |
| KEHL Frontière (via Nancy) | 512 |
| LILLE (via Douai) | 258 |
| LIMOGES (via Vierzon) | 400 |
| LISIEUX (via Evreux) | 191 |
| LORIENT (via Rennes ou Nantes) | 554 |
| LOURDES (via Bordeaux) | 853 |
| LYON PERRACHE/PART-DIEU (via TGV ou Dijon) | 512 |
| MACON (via TGV ou Dijon) | 440 |
| LE MANS (via Chartres) | 211 |
| MARSEILLE ST CHARLES (via Lyon) | 863 |
| MENTON (via Lyon, Marseille) | 1111 |
| METZ VILLE (via Bar-le-Duc) | 354 |
| MODANE Frontière (via TGV ou Lyon) | 706 |
| MONACO-MONTE-CARLO (via Lyon, Marseille) | 1104 |
| MONTPELLIER (via Lyon, Avignon) | 841 |
| MOULINS (via Montargis ou Vierzon) | 314 |
| MULHOUSE VILLE (via Troyes) | 492 |
| NANCY VILLE (via Bar-le-Duc) | 353 |
| NANTES (via Le Mans) | 396 |
| NARBONNE (via Vierzon, Toulouse) | 863 |
| NEVERS (via Montargis ou Vierzon) | 254 |
| NICE VILLE (via Lyon, Marseille) | 1088 |
| NIMES (via Lyon, Avignon) | 791 |
| NIORT (via Poitiers) | 412 |
| ORLEANS | 121 |
| PAU (via Bordeaux) | 814 |
| PERIGUEUX (via Limoges) | 499 |
| PERPIGNAN (via Vierzon, Toulouse) | 926 |
| POITIERS (via Blois) | 232 |
| PORT-BOU Frontière (via Vierzon, Toulouse) | 969 |
| QUEVY Frontière (via St Quentin) | 235 |
| QUIMPER (via Rennes ou Nantes) | 621 |
| REIMS (via La Ferté-Milon ou Epernay) | 156 |
| RENNES (via Le Mans) | 374 |
| LA ROCHELLE (via Poitiers) | 479 |
| ROUBAIX (via Lille) | 268 |
| ROUEN RIVE DROITE | 140 |
| ROYAN (via Poitiers, La Rochelle ou Angoulême) | 558 |
| SETE (via Lyon, Nîmes) | 868 |
| STRASBOURG (via Nancy) | 504 |
| ST BRIEUC (via Le Mans, Rennes) | 475 |
| ST ETIENNE-CHATEAUCREUX (via Lyon) | 512 |
| ST GERVAIS-LE FAYET (via TGV ou Bourg ou Lyon) | 696 |
| ST JEAN DE LUZ (via Bordeaux) | 803 |
| ST MALO (via Rennes) | 455 |
| ST NAZAIRE (via Le Mans, Nantes) | 460 |
| ST RAPHAEL-VALESCURE (via Lyon, Marseille) | 1024 |
| TARBES (via Bordeaux) | 874 |
| TOULON (via Lyon, Marseille) | 930 |
| TOULOUSE MATABIAU (via Vierzon, Cahors) | 713 |
| TOURS (via Blois) | 235 |
| TROUVILLE-DEAUVILLE (via Lisieux) | 221 |
| VALENCE (via Lyon) | 617 |
| VALENCIENNES (via St Quentin ou Douai) | 251 |
| VALLORBE Frontière (via Dijon, Dôle) | 460 |
| VANNES (via Rennes ou Nantes) | 501 |
| VINTIMILLE Frontière (via Lyon, Marseille) | 1115 |
| VICHY (via Montargis ou Vierzon, Moulins) | 365 |

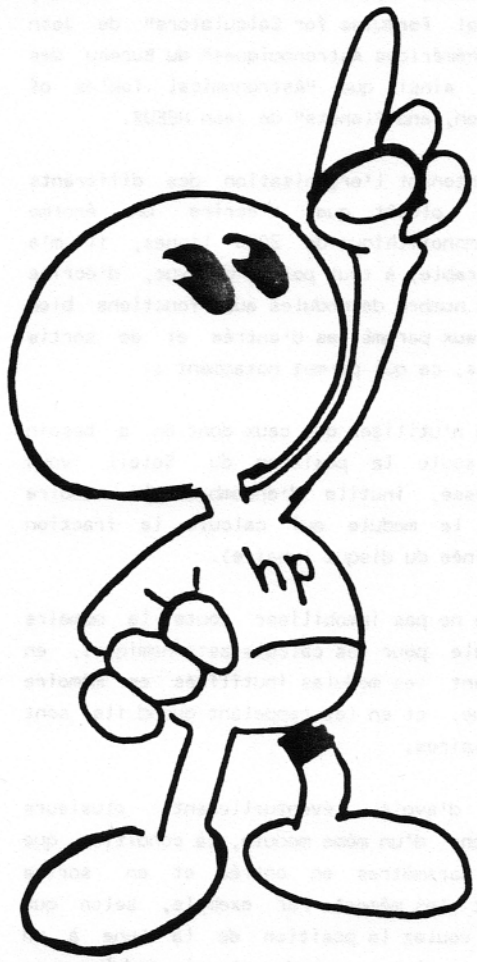
# CARTE DES PRINCIPALES DISTANCES SNCF

NB: Document simplifié ; certaines relations à caractère local ont pu être omises.
Ne constitue pas un document officiel. À jour OCTOBRE 1986.


DUNKERQUE
CALAIS
BOULOGNE
CHERBOURG
LE HAVRE
ROUEN
AMIENS
LILLE
CHARLEVILLE-MÉZIÈRES
REIMS
BREST
QUIMPER
LORIENT
RENNES
LE MANS
CHARTRES
ORLÉANS
TOURS
POITIERS
LA ROCHELLE
NIORT
ANGOULÊME
LIMOGES
PÉRIGUEUX
BRIVE
BORDEAUX
Arcachon
AGEN
MONTAUBAN
TOULOUSE
TARBES
LOURDES
Irún
Hendaye
Biarritz
Bayonne
Dax
Mont de Marsan
Facture
Morcenx
Payan
Auch
Lucon
Les Sables d'Olonne
Royan
Rochefort
Pointe de Grave
Le Croisic
Pornic
La Baule
St Nazaire
Redon
Vitré
Segré
Laval
Châteauroux
Vierzon
Bourges
NEVERS
Saincaize
MOULINS
VICHY
Riom
Thiers
CLERMONT-FD
Montluçon
Guéret
Argenton s/Creuse
St Sulpice-Laurière
Châteauroux
Châteauneuf
Gien
Auxerre
Sens
Montargis
Les Aubrais
Orléans
Blois
Amboise
St Pierre des C.
Tours
Saumur
Cholet
Chinon
Thouars
Parthenay
Loudéac
Pontivy
Carhaix
Landerneau
Morlaix
Plouaret
Lannion
Paimpol
Guingamp
Paimboeuf
Dinan
Lamballe
Granville
Coutances
Lisieux
Serquigny
Evreux
Dieppe
St Valery en C.
Le Tréport
Abbeville
Hazebrouck
Dunkerque
Calais
Boulogne
Montreuil
Berck
Hesdin
Arras
Douai
Valenciennes
Maubeuge
Quiévy
Vervins
Hirson
Aulnoye
Rethel
Laon
Soissons
Creil
Compiègne
Montdidier
Beauvais
Gisors
Dreux
Nogent-le-Rotrou
Mamers
Alençon
Argentan
Lisieux
Pont-l'Evêque
Honfleur
Bernay
Pont-Audemer
Elbeuf
Barentin
Yvetot
Fécamp
Montivilliers
Le Havre
Harfleur
Bolbec
Caudebec
Dieppedalle
Pavilly
Barentin
Yvetot
Motteville
Clères
Buchy
Forges-les-Eaux
Gournay
Neufchâtel
Serqueux
Gisors
Pontoise
Persan-Beaumont
Creil
Chantilly
Ermont-Eaubonne
Argenteuil
Paris-St-Lazare
Paris-Nord
Paris-Est
Paris-Austerlitz
Paris-Bercy
Paris-Gare de Lyon
Paris-Montparnasse
Juvisy
Corbeil
Melun
Montereau
Fontainebleau
Nemours
Montargis
Gien
Briare
Cosne
Nevers
Decize
Moulins
Vichy
Riom
Clermont-Fd
Thiers
Issoire
Le Mont Dore
Brioude
Langeac
Le Puy
St Etienne
Firminy
Andrézieux
Roanne
Tarare
Lyon Perrache
Lyon Part-Dieu
Lyon Vaise
Lyon Gorge de Loup
Lyon St Paul
Lyon Jean Macé
Vénissieux
Givors
Rive-de-Gier
St Chamond
Firminy
Le Puy
Langogne
Mende
Marvejols
Séverac
Rodez
Capdenac
Figeac
Aurillac
Brive
Périgueux
Bergerac
Libourne
Coutras
Angoulême
Limoges
Poitiers
Châtellerault
Tours
Saumur
Angers
Nantes
Ancenis
Clisson
Cholet
Segré
Laval
Le Mans
Alençon
Mortagne
Argentan
Caen
Lisieux
Pont-l'Evêque
Honfleur
Trouville
Deauville
Cabourg
Dives
Caen
Bayeux
Lisieux
Evreux
Vernon
Mantes
Poissy
Versailles
Paris
Meaux
Coulommiers
Provins
Troyes
Romilly
Nogent
Chaumont
Langres
Culmont-Ch.
Is-sur-Tille
Dijon
Beaune
Chagny
Le Creusot
Montchanin
Montceau
Chalon/S
Mâcon
Tournus
Bourg-en-Bresse
Ambérieu
Culoz
Modane
Chambéry
Aix-les-Bains
Annecy
St Gervais
Bourg St Maurice
Grenoble
Gap
Veynes
Briançon
Montdauphin-Guillestre
Sisteron
St Auban
Digne
Nice
Cannes
Antibes
Menton
Vintimille
Genova
Cuneo
Limone Piem.
Ventimiglia
Marseille St Ch.
Aubagne
Toulon
Hyères
St Raphaël
Fréjus
Les Arcs
Draguignan
Aix-en-Prov.
Manosque
Miramas
Arles
Tarascon
Avignon
Orange
Alès
Nîmes
Bédarieux
Mazamet
Carcassonne
Narbonne
Béziers
Sète
Montpellier
Le Grau du Roi
Port de Bouc
Port-la-Nouvelle
Perpignan
Villefranche
Cerbère
Port Bou
La Tour de Carol
Enveitg
(Barcelona via Ripoll)
(Barcelona via Figueras)
Canfranc
(Zaragoza)
Oloron
Puigcerdá
Ax les Th.
Quillan
Montségur
Luchon
Tarascon
Foix
Pau
Lourdes
Tarbes
Auch
Toulouse Matabiau
Montauban
Montech
Castelsarrasin
Moissac
Agen
Marmande
Libourne
Bordeaux
Arcachon
Facture
Morcenx
Payan
Hendaye
Irún
(San Sebastián Madrid via Burgos)
Bayonne
Biarritz
Dax
Mont de Marsan
Pau
Oloron
Canfranc
(Zaragoza)
Lourdes
Tarbes
Montréjeau
Luchon
Ax les Th.
Perpignan
Villefranche
Cerbère
Port Bou
La Tour de Carol
Enveitg
(Barcelona via Ripoll)
(Barcelona via Figueras)
Quillan
Carcassonne
Narbonne
Béziers
Sète
Montpellier
Le Grau du Roi
Port de Bouc
Marseille St Ch.
Aubagne
Toulon
Hyères
St Raphaël
Fréjus
Les Arcs
Draguignan
Aix-en-Prov.
Manosque
Miramas
Arles
Tarascon
Avignon
Orange
Alès
Nîmes
Bédarieux
Mazamet
Albi
Millau
Séverac
Rodez
Capdenac
Figeac
Aurillac
Brive
Périgueux
Bergerac
Libourne
Coutras
Angoulême
Limoges
Poitiers
Châtellerault
Tours
Saumur
Angers
Nantes
Ancenis
Clisson
Cholet
Segré
Laval
Le Mans
Alençon
Mortagne
Argentan
Caen
Lisieux
Pont-l'Evêque
Honfleur
Trouville
Deauville
Cabourg
Dives
Caen
Bayeux
Lisieux
Evreux
Vernon
Mantes
Poissy
Versailles
Paris
Meaux
Coulommiers
Provins
Troyes
Romilly
Nogent
Chaumont
Langres
Culmont-Ch.
Is-sur-Tille
Dijon
Beaune
Chagny
Le Creusot
Montchanin
Montceau
Chalon/S
Mâcon
Tournus
Bourg-en-Bresse
Ambérieu
Culoz
Modane
Chambéry
Aix-les-Bains
Annecy
St Gervais
Bourg St Maurice
Grenoble
Gap
Veynes
Briançon
Montdauphin-Guillestre
Sisteron
St Auban
Digne
Nice
Cannes
Antibes
Menton
Vintimille
Genova
Cuneo
Limone Piem.
Ventimiglia

## HP41

L. Ancelet — Programmes astronomiques — *8*



PROGRAMMES ASTRONOMIQUES

L'ensemble des programmes décrits ci-dessous permet, à l'aide d'une HP-41 munie au minimum de ses 319 registres de mémoire centrale, de ses 124 registres de mémoire étendue, et d'un module TIME (idéalement, une HP-41CX), d'effectuer un certain nombre de calculs astronomiques tels que: position des principaux astres du système solaire, heures de lever et de coucher de ces astres, phases de la Lune, inclinaison des anneaux de Saturne, ainsi que quelques calculs astrologiques tels que position des maisons, noeuds de la Lune, Soleil Noir et Lune Noire, conversion de la longitude écliptique en une position dans un signe (il ne s'agit ici que de la partie calculatoire de l'astrologie. Je n'entrerai pas dans des querelles du style "faut-il croire à l'astrologie", "faut-il tenir compte de la précession", "les maisons ne doivent-elles pas être égales", etc...). Un rappel des notions élémentaires de mécanique céleste sortirait du cadre de cet article, aussi renverrai-je le lecteur à quelques ouvrages de référence comme "Astronomie Générale" de DANJON, "Astronomical Formulae for Calculators" de Jean MEEUS, "Ephémérides Astronomiques" du Bureau des Longitudes, ainsi que "Astronomical Tables of the Sun, Moon, and Planets" de Jean MEEUS.

Voyons maintenant l'organisation des différents programmes: plutôt que d'écrire un énorme programme monolithique de 2000 lignes, il m'a paru préférable, à tout point de vue, d'écrire un certain nombre de modules aux fonctions bien précises, aux paramètres d'entrée et de sortie bien définis, ce qui permet notamment :

1) de n'utiliser que ceux dont on a besoin (si seule la position du Soleil vous intéresse, inutile d'encombrer la mémoire avec le module qui calcule la fraction illuminée du disque lunaire).

2) de ne pas immobiliser toute la mémoire centrale pour les calculs astronomiques, en stockant les modules inutilisés en mémoire étendue, et en les rappelant quand ils sont nécessaires.

3) d'avoir éventuellement plusieurs versions d'un même module, (à condition que les paramètres en entrée et en sortie soient les mêmes). Par exemple, selon que vous voulez la position de la Lune à un degré ou à une seconde près, le module aura une taille de 200 ou de 2000 octets...

Voici la liste des modules présentés dans cet article, avec leur fonction. Les paramètres et les mémoires utilisées sont detaillées plus bas.

REF: cacule, à partir de la date et de l'heure en Temps Universel, le temps écoulé (en siècles juliens de 36525 jours) depuis la date de référence (1900,0 soit le 31 décembre 1899 à 12 heures Temps Universel).

SID: calcule le Temps Sidéral correspondant à un instant donné, au méridien origine.

KPL: résout l'équation de Kepler (ce module est utilisé dans le calcul de la position du Soleil et des planètes).

NUT: calcul de la nutation en obliquité et en longitude a un instant donné.

EQT: calcul des coordonnées équatoriales d'un astre, à partir de ses coordonnées écliptiques.

SUN: calcul de la position du Soleil (longitude et rayon vecteur, éventuellement ascension droite et déclinaison).

PLA: calcul de la position d'une planète (coordonnées hélio- et géo-centriques, éventuellement coordonnées équatoriales et élongation).

NOM: recopie en mémoire centrale, depuis la mémoire étendue, des coefficients des polynômes nécessaires au calcul des éléments orbitaux d'une planète.

ELT: calcul des éléments osculateurs d'une orbite planétaire.

LUN: calcul des coordonnées écliptiques de la Lune, et éventuellement coordonnées équatoriales.

HOR: calcul pour un lieu donné des coordonnées horaires (azimut et hauteur) d'un astre dont on connait les coordonnées équatoriales (ascension droite et déclinaison).

LCA: calcule, pour un lieu donné, l'heure du lever ou du coucher d'un astre quelconque (Soleil, Lune, Planète).

KLU: calcule la fraction illuminée du disque lunaire, et dit si la phase est croissante ou décroissante.

ANN: calcule l'angle sous lequel les anneaux de Saturne sont éclairés par le Soleil, et l'angle sous lequel on les voit depuis la la terre.

Viennent ensuite quelques programmes qui n'intéresseront que les astrologues en culotte courte :

CSP: calcule pour un lieu donné les positions des cuspides des maisons (méthode de Placidus).

SLN: calcul de la position du Soleil Noir.

LNN: calcul de la position de la Lune Noire.

N.B.: le Soleil Noir et la Lune Noire ne correspondent à aucun caillou céleste, sachez seulement qu'ils tournent à la même vitesse que la ligne des apsides des astres correspondants.

SGN: convertit une longitude écliptique (donc de 0 à 360 degrés) en une position à l'intérieur d'un signe (donc de 0 à 30 degrés).

Il serait peut-être temps de s'intéresser à l'utilisation des différents modules, c'est-à-dire à ce qu'on leur dit et ce qu'ils nous répondent.

```
REF: temps écoulé depuis la date de référence.
Entrée: Y = date (JJ.MMAA)
        X = heure T.U. (HH.MMSS)
Sortie: X et R00= Temps écoulé depuis 1900,0.

SID: temps sidéral.
Entrée: R00 = T (Temps écoulé depuis 1900,0).
Sortie: X = temps sidéral à l'instant T.

KPL: Résolution de l'équation de Kepler.
Entrée: Y = M (anomalie moyenne);
        X = e (excentricité).
Sortie: Y = v-M (v = anomalie vraie);
        X = r/a (r = Rayon vecteur,  a = demi
                 grand axe).

NUT: composantes de la nutation.
Entrée: R00 = T (Temps depuis 1900,0).
Sortie: T = obliquité moyenne
        Z =    "        "
        Y = nutation en obliquité (degrés)
        X = nutation en longitude (degrés)
        O = obliquité apparente

EQT : coordonnées équatoriales.
Entrée: Y = longitude écliptique
        X = latitude écliptique
Sortie: Y = déclinaison (degrés);
        X = ascension droite (heures)

SUN: position du soleil.
Entrée: R00 = T (Temps écoulé depuis 1900,0).
Sortie: * si CF00: Y = rayon vecteur (U.A.)
                   X = longitude écliptique
        * si SF00: Y = déclinaison (degrés)
                   X = ascension droite (h.)

PLA: position d'une planète.
Entrée: R00 = T (Temps écoulé depuis 1900,0)
        R16 à R26 = coefficients des polynômes
Sortie: R01 = longitude héliocentrique
        R02 = latitude héliocentrique
        R03 = distance héliocentrique
        R04 = longitude géocentrique
        R05 = latitude géocentrique
        R06 = distance géocentrique
        R09 = longitude du Soleil (Qui a dit : "
                                   héliocentrique
                                        ou
                                   géocentrique
                                    ???")
        R10 = rayon vecteur du Soleil sus-nommé.
        * Si CF00: X = longitude géocentrique(R04)
                   Y = latitude géocentrique (R05)
        * si SF00: X = ascension droite
                   Y = déclinaison
              et en prime : R07 = élongation,  >0
                                à l'ouest du soleil,
                                <0 à l'est du soleil.

NOM: rappel en mémoire des coefficients des
     polynômes qui, etc...
Entrée: ALPHA = nom de la planète sur 3 caractères
        (MER, VEN, MAR, JUP, SAT, URA, NEP, PLU).
Sortie: R16 à R26 = coefficients des polynômes qui
                    etc...
N.B.: l'exécution de ce programme suppose
l'existence en X-memory (soit mémoire étendue
pour les anglophobes) de 8 fichiers de type DATA
ayant pour nom: 'MER, VEN, etc...' et pour
contenu les coefficients des polynômes qui, etc.

ELT: calcul des éléments osculateurs à partir
     des fameux coefficients.
Entrée: R00 = T (temps perdu depuis 1900,0)
        R16 à R26 = coefficients des polynômes
                    qui, tout ça, or donc.
Sortie: R01 = anomalie moyenne de l'astre
              céleste considéré.
        R02 = exentricité de qui vous savez.
        R03 = demi-grand axe of the orbitre.
        R04 = longitude hélio sans trique moyenne.
        R05 = longitude héliocentrique
              (décidement...) ascendant (Oh!).
```

R06 = inclinaison du plan orbital sur l'écliptique.

LUN: calcul de la position de notre mère à tous, la LUNE !!!
Entrée: R00 = T (temp retrouvé par ce cher Marcel)
Sortie: * si CF01: X et R06 = long. géocentrique
Y et R07 = lat. géocentrique
* si SF01: X = ascencion droite
Y = déclinaison
R06 = longitude géocentrique
R07 = latitude géocentrique.

LCA: lever et coucher des astres en un lieu
Entrée: R13 = latitude du lieu (degrés, positive au Nord)
R14 = longitude du lieu (heures, positive à l'Ouest, rien de nouveau).
choix de l'astre: SF00 = Soleil
SF01 = Lune
SF02 = Planète
Sortie: CF03 ===> R00 = instant du lever
SF03 ===> R00 = instant du coucher
N.B.: l'instant du lever (ou du coucher) est exprimé en siècles juliens écoulés depuis 1900,0... mais le programme suivant les exprime sous une forme plus sympathique:

DTE: convertit T en une date et une heure T.U.
Entrée: R00 = T (siècles juliens depuis 1900,0)
Sortie: X = date (JJ.MMAAAA)
Y = T.U. (HH.MMSS)

KLU: fraction illuminée du disque lunaire.
Entrée: R00 = Temps écoulé depuis 1900,0
Sortie: X = fraction illuminée, >0 si la phase croit, <0 si elle décroit.
N.B.: la fraction illuminée est un nombre entre 0 et 1, 0 correspond à la Nouvelle lune, 1 à la Pleine Lune, +0.5 au Premier Quartier, -0.5 au Dernier Quartier.

ANN: latitude saturnicentrique de la Terre et du Soleil.
Entrée: R00 = T (siècles juliens depuis 1900,0)
R01 = longitude héliocentrique de Saturne à cet instant
R02 = latitude héliocentrique de Saturne à cet instant
R04 = longitude géocentrique de Saturne à cet instant
R05 = latitude géocentrique de Saturne à cet instant
Sortie: X = latitude saturnicentrique de la Terre
Y = latitude saturnicentrique du Soleil

N.B.: Si ces deux latitudes sont de même signe, on voit les anneaux du côté éclairé, sinon on en voit la face obscure ; quand la latitude saturnicentrique de la Terre s'annule, on voit les anneaux par la tranche.

Voilà, nous avons fait le tour des programmes d'astronomie, jetons à présent un coup d'oeil sur ceux d'astrologie.

CSP: Calcul des cuspides des maisons par la méthode de Placidus.
Entrée: R00 = T
X = longitude (heures, >0 à l'est de Greenwich)
Y = latitude (degrés, >0 au Nord de l'équateur)
Sortie: R01 = longitude écliptique de l'ASC
R02 = " " de II
R03 = " " de III
R04 = " " du MC
R05 = " " de XI
R06 = " " de XII
N.B.: quand la latitude du lieu est négative (hémisphère SUD), les registres R01 à R06 contiennent les longitudes des cuspides VII, VIII, IX, IV, V et VI (respectivement).

SGN: conversion de longitude écliptique en position dans un signe.
Entrée: X = longitude écliptique de l'astre ou cuspide.
ALPHA = nom de l'astre (ou cuspide) sur 4 caractères maximum.
Sortie: derrière le nom de l'astre est rajouté le signe et la position dans le signe.
N.B.: les 12 signes sont représentés par les mnémoniques (latins) suivants: 'ARI, TAU, GEM, CAN, LEO, VIR, LIB, SCO, SGT, CAP, AQR, PIS'.

NOD: noeud ascendant de la Lune.
Entrée: R00 = T (siècles juliens depuis 1900,0)
Sortie: X = longitude du noeud ascendant de la Lune.

SLN: Soleil Noir.
Entrée: R00 = T (siècles juliens depuis 1900,0)
Sortie: X = longitude du Soleil Noir.

LNN: Lune noire.
Entrée: R00 = T (Siècles juliens depuis 1900,0)
Sortie: X Longitude de la Lune Noire.

Ainsi s'achève la description des différents modules. Avant de passer à leur contenu, si nous examinions quelques exemples ?

Exemple 1: calcul de la position du Soleil le 12/02/81 à 12 heures T.U.

```
 12.02198100 ENTER^
 12.00000000
XEQ "REF"
 0.811170431 ***     (temps ecoulé depuis 1900,0)
SF 00                (on veut les coordonnées
XEQ "SUN"             équatoriales)
 21.73439793 ***     (ascension droite en degrés
HMS                   décimaux)
 21.44038326 ***     (les éphémérides donnent
X<>Y                  21 44 04)
-13.62291147 ***     (déclinaison en degrés
HMS                   décimaux)
-13.37224813 ***     (les éphémérides donnent
                      -13 37)
```

Exemple 2: calcul de la position de la Lune le 10/05/81 à 0 heure T.U.

```
 10.05198100 ENTER^
 0.000000000
XEQ "REF"
 0.813538672 ***
CF 01                (on veut les coordonnées
XEQ "LUN"             écliptiques)
 128.0151769 ***     (longitude degrés décimaux)
X<>Y
 0.282566336 ***     (latitude)
X<>Y                 (on les remet dans le bon
XEQ "EQT"             ordre)
HMS
 8.420114070 ***     (les éphémérides donnent
X<>Y                  08 41 54)
HMS
 18.32175456 ***     (les éphémérides donnent
XEQ "KLU"             18 32)
 0.403816228 ***     (phase croissante - les
                      éphémérides donnent 0.40)
```

Exemple 3: position de Saturne le 27/11/81 à 0 heure T.U.

```
SAT
XEQ "NOM"            (rappel des coefficients en
 27.11198100 ENTER^   mémoire)
 0.000000000
XEQ "REF"
 0.819041752 ***
SF 02                (on veut les coordonnées
XEQ "PLA"             équatoriales)
HMS
 13.14072504 ***     (les éphémérides donnent
X<>Y                  13 12 46)
HMS
-5.213305423 ***     (les éphémérides donnent
RCL 07                -05 14
 45.57338082 ***     (élongation Ouest)
RCL 01
 194.9418468 ***     (longitude héliocentrique)
RCL 02
 2.461589758 ***     (latitude héliocentrique)
RCL 03
 9.638908288 ***     (distance au Soleil en U.A.)
RCL 04
 199.1348610 ***     (longitude géocentrique)
RCL 05
 2.302654838 ***     (latitude géocentrique)
RCL 06
 10.30381336 ***     (distance à la terre -
XEQ "ANN"             éphémérides: 10.289)
 11.45783111 ***     (la Terre voit la face Nord -
X<>Y                  éphémérides: 11.32)
 9.540904181 ***     (le Soleil éclaire la face
                      Nord - éphémérides: 9.38)
```

Exemple 4: lever et coucher de Jupiter à Paris le 10/07/81.

```
 10.07198100 ENTER^
12.00000000          (heure approchée initiale: 12
XEQ "REF"             heures T.U.)
CF 00                (on ne veut pas le soleil)
CF 01                (on ne veut pas la lune)
SF 02                (on veut une planète)
CF 03                (on cherche l'heure du lever)
JUP
XEQ "NOM"            (rappelle les élts orbitaux)
 48.50000000         (latitude de Paris = 48 50 N)
HR
STO 13
-.092100000          (longitude de Paris = 00 09
HR                    21 Est)
STO 14
XEQ "LCA"
RCL 00
 0.815221134 ***     (instant du lever)
XEQ "DTE"
 10.07198100 ***     (c'est bien le 10/07/81)
X<>Y
 10.50441600 ***     (10 50 44 T.U. - éph.: 10 48)
SF 03                (cette fois le coucher)
XEQ "LCA"
RCL 00
 0.815234782 ***     (temps écoulé depuis 1900,0)
XEQ "DTE"
 10.07198100 ***     (c'est bien le 10/07/81)
X<>Y
 22.48345600 ***     (à 22 48 35 - éphémérides:
                      22 53)
```

Exemples 5: quelques calculs astrologiques.

```
 28.09193400 ENTER^  (naissance le 28/09/34)
 12.15000000         (à 12h15 T.U.)
XEQ "REF"
 0.347406172 ***
 48.50000000 HR      (à Paris - ou pas loin)
 .092100000 HR
XEQ "CSP"
RCL 01
 256.2481550         (longitude de l'Ascendant)
ASC
XEQ "SGN"
ASC SGT 16.2
RCL 02
 294.5 ***           (longitude de la maison II)
II
XEQ "SGN"
II CAP 24.5
RCL 03
III
XEQ "SGN"
III PIS  9.6         (etc...)
RCL 04
MC
XEQ "SGN"
MC LIB 15.5
XI
RCL 05
XEQ "SGN"
XI SCO 10.7
RCL 06
XII
XEQ "SGN"
XII SCO 29.7
XEQ "NOD"
 307.3 ***           (longitude du noeud ascendant
NOD                   de la lune)
XEQ "SGN"
NOD AQR  7.3
XEQ "SLN"
 101.8 ***           (longitude du soleil noir)
SLN
XEQ "SGN"
SLN CAN 11.8
XEQ "LNN"
128.1 ***            (longitude de la lune noire)
LNN
XEQ "SGN"
LNN LEO  8.1

Voilà, c'est tout!
          !


 01*LBL "REF"
HR 24 / ,5 + 1,0119
RCL Z DDAYS + 36525
/ STO 00 END
```

```
 01*LBL "SID"
RCL 00 36525 * ,5 -
FRC LASTX INT ,5 +
365,2421988 /
,276919398 + FRC X<>Y
1,0027 * + 24 *
LASTX MOD .END.

 01*LBL "KPL"
STO N X<>Y STO M
STO O
 06*LBL 00
RCL M RCL N R-D RCL O
SIN * + RCL O -  E
RCL N LASTX COS * -
STO Z / ST+ O ABS
 E-4 X<=Y? GTO 00
RCL Z X<> O 2 / TAN
 E RCL N +  E LASTX
- / SQRT * ATAN
ST+ X RCL M - RCL O
END


 01*LBL "NUT"
23,452294 RCL 00
,0130125 * - RCL 00
X^2 164 E-8 * -
STO O 259,18 RCL 00
1934,142 * - 360 MOD
 E P-R 256 E-5 *
ST+ O X<>Y -479 E-5 *
.END.


 01*LBL "EQT"
STO M X<>Y STO N
XEQ "NUT" ST+ M RCL O
COS RCL N SIN *
RCL O SIN RCL N COS
* RCL M SIN * +
ASIN RCL O COS RCL N
COS * RCL M SIN *
RCL O SIN RCL N SIN
* - RCL N COS RCL M
COS * R-P CLX 15 /
24 MOD .END.


 01*LBL "SUN"
358,47583 RCL 00
35999,04975 * +
RCL 00 X^2 15 E-5 *
- 360 MOD ,01675104
RCL 00 418 E-7 * -
RCL 00 X^2 126 E-9 *
- XEQ "KPL"  E 2 E-7
+ * STO M X<>Y
```

```
279,69668 + RCL 00
36000,76892 * +
RCL 00 X^2 3025 E-7 *
+ 360 MOD STO N
FC? 00 RTN XEQ "NUT"
,00569 - ST+ N RCL O
SIN RCL N SIN * ASIN
RCL O COS RCL N SIN
* RCL N COS R-P CLX
15 / 24 MOD END
```

```
 01*LBL "PLA"
XEQ "ELT" RCL 01
RCL 02 XEQ "KPL"
ST* 03 X<>Y RCL 04 +
RCL 05 - STO 01 SIN
RCL 06 COS * RCL 01
COS R-P X<>Y RCL 05
+ 360 MOD X<> 01 SIN
RCL 06 SIN * ASIN
STO 02 RCL 03 P-R
RCL 01 X<>Y P-R
STO 04 RDN STO 05 RDN
STO 06 CF 00 XEQ "SUN"
STO 09 X<>Y STO 10
P-R RCL 04 + X<>Y
RCL 05 + X<>Y R-P
RCL 06 X<>Y R-P
STO 06 RDN STO 05
X<>Y 360 MOD STO 04
FC? 02 RTN RCL 09
RCL 04 - STO Y COS
RCL 05 COS * ACOS
X<>Y SIN SIGN *
STO 07 RCL 05 RCL 04
XEQ "EQT" END
```

```
 01*LBL "NOM"
, SEEKPTA 16,026
GETRX END
```

```
 01*LBL "ELT"
RCL 16 RCL 00 RCL 17
* + 360 MOD STO 01
RCL 18 RCL 00 RCL 19
* + STO 02 RCL 20
STO 03 RCL 21 RCL 00
RCL 22 * + 360 MOD
STO 04 RCL 23 RCL 00
RCL 24 * + STO 05
RCL 25 RCL 00 RCL 26
* + STO 06 END
```

```
 01*LBL "LUN"
270,434164 RCL 00
481267,8831 * + 360
MOD STO 01 296,104608
RCL 00 477198,8491 *
+ 360 MOD STO 02
358,475833 RCL 00
35999,0498 * + 360
MOD STO 03 350,737486
RCL 00 445267,1142 *
+ 360 MOD STO 04
11,250889 RCL 00
483202,0251 * + 360
MOD STO 05 RCL 01
RCL 02 SIN 6,28875 *
+ RCL 04 ST+ X RCL 02
- SIN 1,274018 * +
RCL 04 ST+ X SIN
,658309 * + RCL 02
ST+ X SIN ,213616 *
+ RCL 03 SIN ,185596
* - RCL 05 ST+ X SIN
,114336 * - RCL 04
ST+ X RCL 02 ST+ X -
SIN ,058793 * +
RCL 04 ST+ X RCL 03 -
RCL 02 - SIN ,057212
* + RCL 04 ST+ X
RCL 02 + SIN ,05332
* + 360 MOD STO 06
RCL 05 SIN 5,128189 *
RCL 02 RCL 05 + SIN
,280606 * + RCL 02
RCL 05 - SIN ,277693
* + RCL 04 ST+ X
RCL 05 - SIN ,173238
* + RCL 04 ST+ X
RCL 05 + RCL 02 -
SIN ,055413 * +
RCL 04 ST+ X RCL 05 -
RCL 02 - SIN ,046272
* + RCL 04 ST+ X
RCL 05 + SIN ,032573
* + RCL 02 ST+ X
RCL 05 + SIN ,017198
* + STO 07 RCL 06
FC? 01 RTN XEQ "EQT"
END
```

```
 01*LBL "HOR"
X<>Y STO M X<>Y
XEQ "SID" R^ - RCL 14
- 15 * STO N RCL 13
 E P-R X<>Y RCL M  E
P-R R^ * RCL N COS
* X<> Z * + ASIN
STO O RCL 13  E P-R
```

```
RCL M  E P-R R^ *
X<> Z * RCL N  E P-R
R^ * R^ - X<>Y
RCL M COS * X<>Y R-P
X<>Y 360 MOD RCL O
.END.
```

```
 01*LBL "LCA"

 02*LBL 00
FS? 00 XEQ "SUN"
FS? 01 XEQ "LUN"
FS? 02 XEQ "PLA" X<>Y
TAN RCL 13 TAN * CHS
ACOS 15 / FC? 03 CHS
+ RCL 14 + 24 MOD
XEQ "SID" - ,997 *
876600 / ST+ 00 LASTX
* ABS  E-2 X<=Y?
GTO 00 .END.
```

```
 01*LBL "KLU"
358,476 RCL 00
35999,0498 * + 360
MOD STO M 296,105
RCL 00 477198,8491 *
+ 360 MOD STO N
350,737 RCL 00
445267,1142 * + 360
MOD STO O RCL N SIN
6,289 * + RCL M SIN
2,1 * - RCL O ST+ X
RCL N - SIN 1,274 *
+ RCL O ST+ X SIN
,658 * + RCL N ST+ X
SIN ,214 * + RCL O
SIN ,112 * +  E P-R
CHS LASTX + 2 /
X<>Y SIGN * .END.
```

```
 01*LBL "ANN"
168,12 RCL 00 1,3938
* + STO M 28,07
RCL 00 ,013 * -
STO N  E P-R RCL 05
LASTX P-R RDN *
X<> Z * RCL 04 RCL M
- SIN * X<>Y - ASIN
STO O RCL N  E P-R
RCL 02 LASTX P-R RDN
* X<> Z * RCL 01
RCL M - SIN * X<>Y
- ASIN RCL O .END.
```

```
 01*LBL "CSP"
HR X<>Y HR TAN STO M
X<>Y XEQ "SID" + 15
* STO 04 1 30 XEQ 01
STO 05 2 60 XEQ 01
STO 06 -3 90 XEQ 01
STO 07 -2 120 XEQ 01
STO 02 -1 150 XEQ 01
STO 03 RCL 04 XEQ 02
STO 04 CF 00 RCL 01
RTN

 39*LBL 01
RCL 04 + STO N X<>Y
STO O CF 00 X<0?
SF 00

 48*LBL 03
RCL N SIN 23,44 TAN
* RCL M * FC? 00 CHS
ACOS RCL O * 3 /
FC? 00 GTO 00 180 +

 67*LBL 00
RCL 04 + X<> N RCL N
- ABS ,1 X<=Y?'
GTO 03 RCL N
 78*LBL 02
 E P-R 23,44 COS *
R-P X<>Y 360 MOD
.END.
```

```
 01*LBL "NOD"
259,1833 RCL 00
1934,142 * - 360 MOD
END
```

```
 01*LBL "SLN"
101,220833 RCL 00
1,719175 * + 360 MOD
END
```

```
 01*LBL "LNN"
154,47 RCL 00
4069,034033 * + 360
MOD END
```

```
 01*LBL "SGN"
360 MOD 30 / INT
LASTX FRC 30 * X<>Y
XEQ IND X X<>Y FIX 1
RND  E1 X>Y? "├ "
ARCL Y AVIEW RTN
```

```
 22*LBL 00
"├ ARI " RTN

 25*LBL 01
"├ TAU " RTN

 28*LBL 02
"├ GEM " RTN

 31*LBL 03
"├ CAN " RTN

 34*LBL 04
"├ LEO " RTN

 37*LBL 05
"├ VIR " RTN

 40*LBL 06
"├ LIB " RTN

 43*LBL 07
"├ SCO " RTN

 46*LBL 08
"├ SGT " RTN

 49*LBL 09
"├ CAP " RTN

 52*LBL 10
"├ AQR " RTN

 55*LBL 11
"├ PIS " .END.


 01*LBL "DTE"
RCL 00 36525 * ,5 -
INT LASTX FRC 24 *
HMS X<>Y 1,0119 X<>Y
DATE+ .END.


 01*LBL "LOE"
"MER" XEQ 01 "VEN"
XEQ 01 "MAR" XEQ 01
"JUP" XEQ 01 "SAT"
XEQ 01 "URA" XEQ 01
"NEP" XEQ 01 "PLU"
XEQ 01 BEEP RTN


 20*LBL 01
AVIEW , SEEKR 16,026
READRX VIEW 20 11
CRFLD 16,026 SAVERX
.END.
```

Les valeurs des coefficients des polynômes, pour chaque planète, sont données ci-après:

Pour Mercure:

```
R16=  102.2793800
R17=  149472.5153
R18=  0.205614210
R19=  0.000020460
R20=  0.387098600
R21=  178.1790780
R22=  149474.0708
R23=  47.14594400
R24=  1.185208300
R25=  7.002881000
R26=  0.001860800
```

Pour Venus:

```
R16=  212.6032200
R17=  58517.80387
R18=  0.006820690
R19= -0.000047740
R20=  0.723331600
R21=  342.7670530
R22=  58519.21191
R23=  75.77964700
R24=  0.899850000
R25=  3.393631000
R26=  0.001005800
```

Pour Mars:

```
R16=  319.5191300
R17=  19139.85475
R18=  0.093312900
R19=  0.000092064
R20=  1.523688300
R21=  293.7373340
R22=  19141.69551
R23=  48.78644200
R24=  0.770991700
R25=  1.850333000
R26= -0.000675000
```

Pour Jupiter:

```
R16=  225.3283300
R17=  3034.692020
R18=  0.048334750
R19=  0.000164180
R20=  5.202561000
R21=  238.0492570
R22=  3036.301986
R23=  99.44341400
R24=  1.010530000
R25=  1.308736000
R26= -0.005696100
```

Pour Saturne:

R16= 175.4662200
R17= 1221.551470
R18= 0.055892320
R19= -0.000345500
R20= 9.554747000
R21= 266.5643770
R22= 1223.509884
R23= 112.7904140
R24= 0.873195100
R25= 2.492519000
R26= -0.003918900

Pour Uranus:

R16= 72.648774800
R17= 428.3791132
R18= 0.046344400
R19= -0.000026580
R20= 19.21814000
R21= 244.1974700
R22= 429.8635460
R23= 73.47711100
R24= 0.498667800
R25= 0.772464000
R26= 0.000625300

Pour Neptune:

R16= 37.73063000
R17= 218.4613396
R18= 0.008997040
R19= 0.000006330
R20= 30.10957000
R21= 84.45799400
R22= 219.8859140
R23= 130.6813890
R24= 1.098935000
R25= 1.779242000
R26= -0.009543600

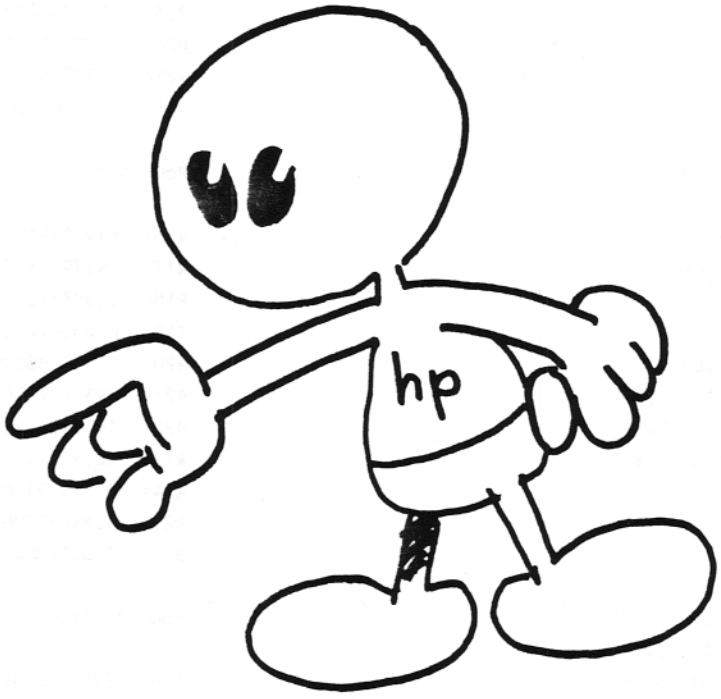
Pour Pluton:

R16= 229.2180000
R17= 145.3486800
R18= 0.250236000
R19= 0.000000000
R20= 39.43871200
R21= 93.22800000
R22= 145.3486800
R23= 109.7400000
R24= 0.000000000
R25= 17.17047000
R26= 0.000000000

Pour créer le fichier de données relatif à une planète, remplir les registres 16 à 26 avec les valeurs correctes (ou d'autres si ça vous amuse), mettre le nom de la planète en ALPHA (3 caractères), puis exécuter l'instruction SAVERX.

On remarquera que certains "polynômes" relatifs à Pluton sont en fait des constantes: ceci est dû au fait que la théorie du mouvement de cette planète n'a pas été calculée. De ce fait, les expressions donnant directement les éléments osculateurs ne sont pas connues. On a donc utilisé des valeurs moyennes.

Lionel Ancelet.



## ASSEMBLEUR


| | | |
|---|---|---|
| G. Toublanc | Le nouveau DIVILEX | *18* |
| J.P. Bondu | Un Lex pour vos Basic | *24* |


## BASIC / FORTH


| | | |
|---|---|---|
| A. Goubault | Forthément complexe | *34* |
| G. Toublanc | Programme "LISTPRIM" | *37* |
| A. Goubault | Programme "COMPLEX" | *38* |


## LE COIN DES LHEX

*41*

## LE NOUVEAU DIVILEX

Dans l'article DIVORCE (JPC No 37) vous étaient annoncés les nouveaux DIVILEX et LISTPRIM avec les raisons qui ont motivé ces refontes. Voici les résultats :

- rapidité légèrement accrue et gain d'octets.
- PGCD et PPCM acceptent maintenant jusqu'à 10 nombres au lieu de 2.
- FPRM recherche un nombre en amont, mais aussi en aval d'un nombre donné. Le 2ème paramètre n'est plus qu'optionnel et non obligatoire.
- PRIM peut tester un nombre avec maximum de 15 chiffres au lieu de 12. En revanche, il renvoie 0 si le nombre est premier ce qui facilite la programmation en Basic avec gain d'octets, et rend possible le même type de sortie pour les nombres à 12 ou 15 chiffres.
- Et malgré ces "plus" ajoutés, gain de 25 octets.

Le fichier Basic utilisant DIVILEX a été revu pour tenir compte des nouvelles syntaxes. Voici les changements :
- création de sous-programmes acceptant des nombres jusqu'à 15 chiffres sous forme de chaînes alphanumériques :
- PRIM équivalent du mot PRIM
- FPRM équivalent du mot FPRM
- sous-programme DECOMP adapté aux possibilités ci-dessus.
- sous-programme REDUC acceptant maintenant une ou plusieurs fractions.

Remarques : la 2ème syntaxe de PRIM est sûrement critiquable. Mon choix a été motivé par les raisons suivantes :

- il faut rester réaliste : PRIM utilisé pour les nombres > 10^12 sera le fait de la curiosité et de l'exceptionnel.

Les temps de réponse pour les grands nombres premiers sont longs (7 heures 3/4 pour 10^15-11 ! aussi confiera-t-on au basic le soin de récupérer les résultats que Titan aura trouvés par l'intermédiaire de PRIM pendant une nuit par exemple.

C'est pour cela que FPRM et NPRM ne vont pas au delà de 12 chiffres. Par contre, les sous-programmes FPRM et DECOMP acceptent 15 chiffres.

Ce choix m'a permis de ne pas trop grossir le Lex mais d'exploiter presque gratuitement la possibilité offerte par les registres de calculer avec 15 chiffres.

LISTPRIM est composé de sous-progammes totalement indépendants, donc possibilité d'en supprimer.

## MODE D'EMPLOI
## et exemples avec DIVILEX

Seuls les changements par rapport à l'ancien DIVILEX sont signalés. Pour le reste, voir JPC No 35.

```
PRIM renvoie le premier diviseur
     ou 0 si le nombre est premier .

1ère syntaxe : PRIM (n)
               avec 0 < | n | < 10^12

2ème syntaxe : PRIM (H,B)
               avec 0 <= | H | < 10^12
                    0 <= B < 1000
               ou   0 < B < 1000 si H = 0
               donc n = H*1000 + B
               et   0 < | n | < 10^15

PRIM(311) ou PRIM(-311)   ( ou PRIM(0,311) )
 -> 0          n est premier
PRIM(143) ou PRIM(-143)   ( ou PRIM(0,143) )
 -> 11        | 143 | = 11*13
PRIM(999997874844,049) ou
PRIM(-999997874844,049)
 -> 31622743  (en 3"01)
| 999997874844049 | = 31622743 ^ 2
PRIM(100071000730,021)
 -> 10007     (en 11"15)
100071000730021 = 10007 * 10000100003
PRIM(999999999999,827)
 -> 0         (en 7 heures 3/4 !)
n est premier
PRIM(1000000000000,021)
 -> invalid arg
H > 10^12 donc n > 10^15
PRIM(1,8981)
 -> invalid arg
B > 999

ATTENTION aux "0" en excès dans B car il n'y a
pas de "garde-fou" :
exemple : n = 10007 nombre premier
PRIM(1,0007)
 -> 19    Titan pose n = 1000*1 + 7 = 1007
```

```
OPTION interruption : voir JPC No 35


FPRM renvoie le nombre premier n
     le plus proche d'un nombre
     donné n1 dans un intervalle
     défini ou non
     on aura toujours 5 < n1 < 10^12
                   ou 5 < n2 < 10^12

1ère syntaxe : FPRM(n1)
               avec n >= n1
               limite supérieure
               implicite n2 = 10^12
2ème syntaxe : FPRM(-n1)
               avec n <= n1
               limite inférieure
               implicite n2 = 5
3ème syntaxe : FPRM(n1,n2)
               avec n1 <= n <= n2
               si   n1 <= n2
               avec n2 <= n <= n1
               si   n1 >= n2

FPRM(300) ou FPRM(300,350)
 -> 307
1er nombre premier >= 300 ou >= 300 et <= 350
FPRM(-300) ou FPRM(300,250)
 -> 293
1er nombre premier <= 300 ou <= 300 et >= 250
FPRM(300,294)
 -> 0
pas de nombre premier entre 300 et 294

Remarque : "-" devant n1 ou n2 ou n1 et n2
implique aussi la recherche en aval comme pour
la 2ème syntaxe, mais restons simples.
( FPRM(-300,250) -> 293 )

Pour l'extension à 15 chiffres voir LISTPRIM


NPRM : pas de changement : voir JPC No 35
PHI  : dito
     : opère aussi avec les entiers négatifs


PGCD : plus grand commun diviseur
       de plusieurs nombres

syntaxe : PGCD(n1,n2,...,ni)
          avec 2 <= i <= 10
          et | ni | < 10^12

PGCD(385,210,715) ou PGCD(-385,210,-715)
 -> 5
PGCD de 385,210 et 715 ou de -385,210 et -715
pour i > 10

PGCD(n1,n2,...n9,PGCD(n10,n11,...))
   etc...   voir JPC No 35


PPCM : plus petit commun multiple
       de plusieurs nombres

syntaxe : PPCM(n1,n2,...,ni)
          avec 2 <= i <= 10
          et | ni | < 10^12

PPCM(385,210,715)
 -> 30030
PPCM de 385,210 et 715
pour i > 10 voir PGCD


                Le fichier basic
                    LISTPRIM

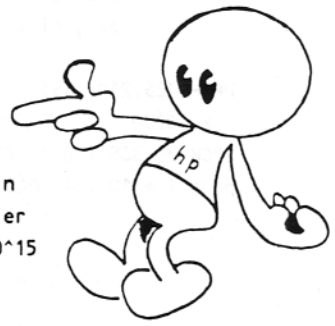

sous-progammes LPRIM , EQUZ , JUM , QUAD
               inchangés : voir JPC No 35


CALL DECOMP("n")
 -> décomposition d'un nombre en
    ses facteurs premiers
    n entré sous forme de chaine
    avec 0 < | n | < 10^15
CALL DECOMP("99999999927")
 -> 3^3 59 62774639
CALL DECOMP("100011000110021")
 -> 7 11 13 5227 19114423


CALL REDUC(n)
 -> réduction de n fractions
    si n = 1  la fraction est réduite
    à sa plus simple expression
    si n > 1  les fractions sont réduites
    suivant leur plus petit commun dénominateur
    valeur absolue des
    numérateurs et dénominateurs < 10^12

CALL REDUC(1)
 -> N1,D1 ?    99999999927,99999999999
 -> 11111111103 / 11111111111
CALL REDUC(3)
 -> N1,D1 ?    48,56
 -> N2,D2 ?    65,135
 -> N3,D3 ?    1200,360
 -> F1 = 162 / 189        fCONT
 -> F2 = 91 / 189         fCONT
 -> F3 = 630 / 189
    ( 48/56=6/7 165/135=12/27 1200/360=10/3
    PPCM(7,27,3)=189 48/56=162/189
    65/135=91/189 1200/360=630/189 )
```

```
CALL FPRM("n")
 -> nombre premier le plus proche de n
    et supérieur ou égal à n
    avec 5 < n < 10^15
CALL FPRM("-n")
 -> nombre premier le plus proche de n
    et inférieur ou égal à n
CALL FPRM("1000000000000")
 -> 1000000000039   (en 12'34")
CALL FPRM("-9999999928")
 -> 9999999881      (en 77")
```



```
CALL PRIM("n")
 -> renvoie le 1er diviseur de n
           ou 0 si n est premier
           avec 0 < | n | < 10^15
CALL PRIM("999999999999839")
 -> 502543  1er diviseur de n
CALL PRIM("999999999999877")
 -> 0       n est premier
```

L'OPTION interruption du mot PRIM est évidemment utilisable avec ce sous-programme (voir JPC No 35)

Vous trouverez ci-dessous le source du nouveau DIVILEX. Les algorithmes étant les mêmes, les commentaires sont réduits à ce qui est nouveau.

Bonne chasse aux nombres entiers remarquables...

Guy TOUBLANC (P276)

La nouvelle version de DIVILEX

```
       LEX    'DIVILEX'
       ID     #E1
       MSG    0
       POLL   0

t      EQU    82
ARGERR EQU    #0BF19
DCHXW  EQU    #0ECDC   Conversion décimal -> hexa
FLOAT  EQU    #1B322
FNRTN4 EQU    #0F238
IDIV   EQU    #0EC7B   Division entière en hexa ou
         *             décimal A(W)/C(W)
MPY    EQU    #0ECBB
POP1N  EQU    #0BD1C
RJUST  EQU    #12AE2
SQR17  EQU    #0C553   racine carrée d'1 arg. fini

       ENTRY  FPR
       CHAR   #F
       ENTRY  NPR
       CHAR   #F
       ENTRY  PGD
       CHAR   #F
       ENTRY  PH
       CHAR   #F
       ENTRY  PPM
       CHAR   #F
       ENTRY  PRM
       CHAR   #F

       KEY    'FPRM'
       TOKEN  t
       KEY    'NPRM'
       TOKEN  1+t
       KEY    'PGCD'
       TOKEN  2+t
       KEY    'PHI'
       TOKEN  3+t
       KEY    'PPCM'
       TOKEN  4+t
       KEY    'PRIM'
       TOKEN  5+t

       ENDTXT

       NIBHEX 811
PH     GOSUB  pop1n    Calcul de PHI : même algo-
       R0=A             rithme (JPC no 35) mais rou
       R2=A             tines en décimal + optimisa
       C=0    W         tion
       C+P+1
       ?A=C   W
       GOYES  RESA
HIO    GOSUB  GTES
ROA    R0=A
       ?A#C   W
       GOYES  LOO
       R1=C
       GOSUB  PHI
       GONC   RESA
LOO    GOSUB  div
       C=R1
       ?B=0   W
       GOYES  ROA
       GOSUB  PHI
       GONC   HIO
PHI    A=R2
       C=R1
       GOSUB  div
       C=R1
       C=C-1  W
       GOSUB  R2A
       R2=A
       RTN
```

```
RESC    A=C    W
RESA    GOSBVL FLOAT
        C=A    W
        ?ST=0  0
        GOYES  OUT
        C=C-1  S
OUT     GOVLNG FNRTN4

        NIBHEX 88888888882A param. : mini 2 maxi 10
PPM     ST=1   2          repère pour PPCM
        GONC   pgd

        NIBHEX 88888888882A dito ci-dessus
PGD     ST=0   2          repére PGCD
pgd     GOSUB  DCS        pope et teste le dernier
          *               param.
        R1=A              sauve en R1
        GOSUB  D1         pope et teste le param.
          *               précédent
        GONC   pgc
LOOPG   R3=C              sauve le PGCD ou le PPCM
        GOSUB  D1         pope et teste le param.
          *               précédent
pgc     GOSUB  PGC        calcul du PGCD
        ?D#0   S          encore 1 param. à poper ?
        GOYES  LOOPG      oui retour à la boucle
        GONC   RESC       non alors résultat
PGC     C=R1              calcul PGCD des param.
        ?C#0   W
        GOYES  DIF0
        ACEX   W
DIF0    R3=A
        R2=C
EUCL    GOSUB  R1C
        ?B=0   W
        GOYES  PGCD
        A=R1
        GONC   EUCL
PGCD    C=R1              C(W)=PGCD
        ?ST=0  2          seulement calcul du PGCD ?
        RTNYES            oui alors retour
        A=R3              non alors calcul du PPCM
        GOSUB  div
        R0=A
        A=R2
        GOSUB  E12
        ACEX   W
        GOSUB  div
        C=R0
        ?A<C   W
        GOYES  pop
        GOSUB  dh
        R1=A
        RTN
pop     C=RSTK
ERR     GOVLNG ARGERR
```

```
DCS     D=C    S          nombre de param. -> D(S)
        D1=D1- 16         acrobatie pour le 1er param
D1      D1=D1+ 16         actualise D1 : param. suiv.
pop1n   GOSBVL POP1N      pope le paramètre
        ?A=0   S          pas "-" avec le 1er param.?
        GOYES  C11        oui alors test C11
        ST=1   4          non alors repère de "-"
C11     LCHEX  011
        ?A>C   X          N >= 10^12 ?
        GOYES  pop        oui alors erreur
rjust   GOSBVL RJUST      conversion déci. flot.->déc
        D=D-1  S          actualise nombre param.
        RTN

E12     P=     1          C(W)=10^12 en décimal
        C=0    W
        C=P    12
        P=     0
        RTN

CONST   C=0    W
        LCHEX  2
        D=C    W          D(W)=2 en décimal
        LCHEX  10
        R4=C              R4=10 en décimal
        RTN

        NIBHEX 8812       param. : 1 mini 2 maxi
FPR     ST=1   1          repère pour FPRM
        GONC   prim       goto routine commune

        NIBHEX 8822       param. : 2 mini 2 maxi
NPR     ST=0   1          repère pour NPRM
prim    ST=0   4          pas de param. négatif
        GOSUB  DCS        pope et teste le paramètre
        C=0    W
        ?ST=1  4          repère de param. négatif ?
        GOYES  LIM2       oui alors test de ce param.
        GOSUB  E12        non alors C(W)=10^12
LIM2    R2=C              R2=0 ou 10^12 (limite)
        ?D=0   S          1 seul paramètre ?
        GOYES  LIM1       oui alors 1 limite
        R2=A              sauve le 2e paramètre
        GOSUB  D1         pope et teste le 1er param.
LIM1    R0=A              sauve le param. ou le 1er
        C=R2              C(W)=limite sup. ou inf.
        ?A<=C  W          n <= limite ?
        GOYES  const      oui alors S4=0
        ST=1   4          non alors S4
const   GOSUB  CONST      D(W)=2  R4=10=C(W)
        LCHEX  06         C(W)=6
        GOSUB  div        A(W)=N/6 B(W)=C(W)=MOD(N,6)
        ?A#0   W          N > 5 ?
        GOYES  PL5        oui alors on continue
        GOTO   ERR        non alors invalid arg
PL5     A=0    W
        R3=A              initialise compteur
        C=R0
```

```
        C=C-B  W
        C=C+1  W        C(W)=N+1-MOD(N,6)
        ?ST=0  4        pas de param. négatif ?
        GOYES  CONTNP   alors recherche en amont
        ?B=0   B        N multiple de 6 ?
        GOYES  START2   oui alors départ de boucle
        ACEX   B        C(B)=0
        P=     0
        LCHEX  5        C(B)=5
        ACEX   B        A(B)=5  C(W)=N+1-MOD(N,6)
        B=B-A  B        B(B)=MOD(N,6)-5
        GOC    start    MOD(N,6) < 5
        C=R0            MOD(N,6) = 5   C(W) = N
LOOPF   GOSUB  det      boucle testant les nombres
        C=C-D  W        non multiples de 2 ou 3 en
        C=C-D  W        aval de N
start   GOSUB  det
START2  C=C-D  W
        GONC   LOOPF
CONTNP  B=B-1  B        N multiple de 6 alors
        GOC    LOOP1    départ de boucle en LOOP1
        B=B-1  B
        GONC   START    MOD(N,6)=1 alors START
LOOP1   GOSUB  DET      boucle testant les nombres
START   C=C+D  W        non multiples de 2 ou 3 en
        C=C+D  W        amont de N
        GOSUB  DET
        C=C+D  W
        GONC   LOOP1
det     R0=C            sauve n nouveau
        A=R0
         C=R2
        GOTO   DETF
DET     R0=C
        A=R2
DETF    ?C>A   W        routine commune à FPRM/NPRM
        GOYES  RESN     pour tester la primalité
        GOSUB  NTES     des nombres obtenus par
        ?A#C   W        incrémentation ou décrémen-
        GOYES  CONT     tation
        ?ST=1  1        est-ce FPRM ?
        GOYES  RESU     oui alors résultat
        C=R3            non alors incrémentation
        C=C+1  W        du compteur
        R3=C
CONT    C=R0            C(W)=dernière valeur
        RTNCC
RESN    A=R3
RESU    C=RSTK
        GOTO   RESA     résultat

R1C     R1=C
div     GOSBVL IDIV     division entière A(W)/C(W)
        RTNCC

        NIBHEX 8812     param. : 1 mini 2 maxi
PRM     GOSUB  DCS      pope et teste le paramètre
        ?D=0   S        1 seul paramètre ?
        GOYES  TVAL     oui alors N < 10^12
        C=0    W
        LCHEX  999
        ?A>C   W        B > 999 ?
        GOYES  err      oui alors erreur
        C=C+1  A        C(W)=1000
        R1=C            sauve 1000
        R0=A            sauve B
        GOSUB  D1       pope et teste H
        C=R1            C(W) = 1000
        GOSUB  R2A      C(W) = H*1000
        A=R0            A(W) = B
        A=A+C  W        A(W) = H*1000 + B
TVAL    R1=A            sauve N
        ?A#0   W        N # 0 ?
        GOYES  VAL      oui alors on continue
err     GOTO   ERR      non alors erreur
VAL     B=A    W        ( équivalent de RJUST mais
        A=0    A        ( sur 15 chiffres au lieu
        LCHEX  15       ( de 12
        A=C    B        (
        P=     14       (
loop    A=A-1  B        (
        ?B#0   P        (
        GOYES  SQR      (
        BSL    W        (
        GONC   loop
SQR     GOSBVL SQR17    ( teste si N est carré
        A=B    M        ( parfait
        GOSUB  rjust    (
        R0=A            (
        C=R0            (
        GOSUB  R2A      (
        A=R1            (
        ST=1   0        sortie anticipée possible
        ST=1   3        repère N carré parfait
        ?A=C   W        N carré parfait ?
        GOYES  LTES     oui alors tests
        ST=0   3        non alors repère
        R0=A            sauve N et non sa racine
LTES    GOSUB  GTES     test primalité
        ?ST=1  3        N carré parfait ?
        GOYES  resc     oui alors résultat
        ?A#C   W        N non premier ?
        GOYES  resc     oui alors résultat
        C=0    W        non alors résultat = 0
resc    GOTO   RESC

GTES    GOSUB  CONST    D(W) = 2   R4 = 10
        C=D    B
        GOSUB  TES0     divisib./2
        C=C+1  B
        GOSUB  TES0     divisib./3
NTES    C=D    W
        C=C+1  B
        GOSUB  TES2     dito JPC no 35
        GOSUB  TES2
```

```
        GOSUB  TES4
LOOP    GOSUB  TES2
        GOSUB  TES4
        GOSUB  TES2
        GOSUB  TES4
        GOSUB  TES6
        GOSUB  TES2
        GOSUB  TES6
        GOSUB  TES4
        GOSUB  TES2
        GOSUB  TES4
        GOSUB  TES6
        GOSUB  TES6
        GOSUB  TES2
        GOSUB  TES6
        GOSUB  TES4
        GOSUB  TES2
        GOSUB  TES6
        GOSUB  TES4
        GOSUB  TES6
        GOSUB  TES8
        GOSUB  TES4
        GOSUB  TES2
        GOSUB  TES4
        GOSUB  TES2
        GOSUB  TES4
        GOSUB  TES8
        GOSUB  TES6
        GOSUB  TES4
        GOSUB  TES6
        GOSUB  TES2
        GOSUB  TES4
        GOSUB  TES6
        GOSUB  TES2
        GOSUB  TES6
        GOSUB  TES6
        GOSUB  TES4
        GOSUB  TES2
        GOSUB  TES4
        GOSUB  TES6
        GOSUB  TES2
        GOSUB  TES6
        GOSUB  TES4
        GOSUB  TES2
        GOSUB  TES4
        GOSUB  TES2
        GOSUB  TES10
        GOSUB  TES2
        A=IN
        ?A#0   X
        RTNYES
        A=R4            A(W) = 10
        C=C+A  WP       dernier diviseur + 10
        A=R0
        GOSUB  R1C      N/diviseur
        C=R1            le dernier diviseur
        P=     7        pour tests de comparaison
        *               sur 8 nibbles seulement
        *               ( gain de rapidité )
        ?B=0   WP       par exemple ici
        GOYES  rnp
        ?C>A   W
        GOYES  RESP
        GOTO   LOOP

RESP    C=R0
rnp     GOC    RNP
TES10   C=C+D  WP       dernier diviseur + 10
TES8    C=C+D  WP
TES6    C=C+D  WP
TES4    C=C+D  WP
TES2    C=C+D  WP       dernier diviseur + 2
TES0    R1=C            sauve nouveau diviseur
        A=R0            dito JPC no 35
        P=     0
HI      ?C>=A  W
        GOYES  LO
        CSL    W
        P=P+1
        GONC   HI
LO      A=A-C  W
        GONC   LO
        A=A+C  W
        P=P-1
        GOC    LO1
        CSR    W
        GONC   LO
LO1     C=R1
        P=     7
        ?A#0   WP
        RTNYES
        C=RSTK
        C=R1
RNP     A=R0
        ST=0   0
        RTN

R2A     R2=A            ( A(W)*C(W) en décimal
dh      GOSBVL DCHXW    (
        SETDEC          (
        A=R2            (
mpy     GOSBVL MPY      (
        RTN

        END
```

## UN LEX POUR VOS BASIC

### FONCTIONS BLIST & PBLIST

Pour lister un programe de la manière dont il paraît dans JPC (mise en évidence des labels, fonctions utilisateurs, data...), il fallait jusqu'à présent utiliser le programme JPCLISTE. Celui-ci transforme le Basic en texte, opération longue et délicate lorsque des XFN ou XWORD sont utilisés. En effet, ceux-ci ne sont pas reconnus lors de la transformation contraire (texte en basic). Je rappelle aux distraits que XFN et XWORD sont l'interprétation, lors de la décompilation, des mots appartenant à des Lex non présents en mémoire, ou désactivés.

Désormais, il suffira de taper BLIST (Basic LIST) pour obtenir un résultat similaire à l'affichage. Cet ordre ajouté au BASIC travaille sur la version "tokenisée" du programme et n'en demande donc pas le changement de type. Les mots END DEF, END SUB... sont localisés n'importe où sur la ligne et les fonctions utilisateur monolignes différenciées, sans erreur possible, des multilignes. L'ordre PBLIST est similaire mais effectue la sortie sur imprimante en décalant le texte de 20 colonnes à droite. Il faudra donc déclarer une largeur de ligne supérieure à 96+20=116 caractères. Utilisez donc PRINTLEX à cet effet !

### FONCTION RENUMREM

Pour faire paraitre un programme Basic, vous le documentez en insérant entre chaque ligne des remarques. Résultat : au prochain RENUMBER vous aurez une numérotation "tordue" des seules lignes BASIC. De plus, lors de la parution dans notre JPC favori, les numéros précédant une ligne de REM sont supprimés : le lecteur a l'impression que vous avez numéroté votre programme en dépit du bon sens. Mais pire, l'incrément entre chaque ligne BASIC n'étant pas constant, il lui sera impossible d'utiliser AUTO pour taper le programme.

J'ai donc écrit l'ordre RENUMREM qui admet la même syntaxe et les mêmes paramètres par défaut que RENUMBER, mais qui renumérote différement les lignes de REM. Il "colle" au maximum ces lignes à la ligne BASIC immédiatement précédente. Un exemple :

```
05 ! DEBUT
10 PRINT TAB(18);"TOTO"
30 ! affiche le titre
40 ! à droite de l'affichage
42 ! pour les besoins de
45 ! notre exemple.
50 IF KEY$="" THEN 50
```

Deviendra après RENUMREM 100,5 :

```
100 ! DEBUT
105 PRINT TAB(18);"TOTO"
106 ! affiche le titre
107 ! à droite de l'affichage
108 ! pour les besoins de
109 ! notre exemple.
110 IF KEY$="" THEN 110
```

Ou après RENUMREM 100,3 :

```
100 ! DEBUT
103 PRINT TAB(18);"TOTO"
104 ! affichage du titre
105 ! à droite de l'affichage
106 ! pour les besoins de
107 ! notre exemple.
109 IF KEY$="" THEN 109
```

Comme demandé, la première ligne porte le No 100. La suivante est BASIC, elle porte donc le No 100+3=103. Viennent les lignes de REM dont l'incrément est toujours de 1. Nous en sommes donc à la ligne 107. La ligne IF KEY$... aurait dû porter le No 106, qui est déja pris, on lui alloue donc le No : 106+Incrément = 106+3 = 109. La prochaine ligne aura le No 109+1=110 si REM, 109+3=112 si c'est une ligne BASIC.

J'espère que vous jugerez ce Lex digne d'occuper 1096 octets de votre EPROM. A bientôt !

Jean-Pierre BONDU (PC 33, SIG 3)

```
        LEX     'BASICLEX'

        ID      #E1
        MSG     0
        POLL    0

t       EQU     90
```

```
        ENTRY   BLSTs
        CHAR    #D
        ENTRY   PBLSTs
        CHAR    #D
        ENTRY   RENUMe
        CHAR    #D
        KEY     'BLIST'
        TOKEN   t
        KEY     'PBLIST'
        TOKEN   1+t
        KEY     'RENUMREM'
        TOKEN   2+t

colprt  EQU     19          indentation de PBLIST
len"=   EQU     62

ARGERR  EQU     #0BF19
BASCHA  EQU     #07741
BASCHK  EQU     #0773E
BSERR   EQU     #0939A
CK"ON"  EQU     #076AD
CKINFO  EQU     #18542
COMCK   EQU     #036CD
CPL#10  EQU     #07887
CSLW5   EQU     #0ED3D
CSRW5   EQU     #0ED2C
CURREN  EQU     #2F56C
CURRL   EQU     #2F7E8
CURRST  EQU     #2F55D
D0=PCA  EQU     #09B37
DISPt   EQU     #00000
EOLCK   EQU     #02A7E
EOLXCK  EQU     #05405
FINDF+  EQU     #09F63
FINDL   EQU     #0FFE4
FINDL0  EQU     #0FFFD
FSPECe  EQU     #02F02
FSPECp  EQU     #03CC5
FSPECx  EQU     #09F2D
FUNCR0  EQU     #2F89B
GETPR1  EQU     #06BFB
GETPR0  EQU     #06BEE
GETSTC  EQU     #07726
IVPARe  EQU     #02E3F
LDCM10  EQU     #04F6F
LDCSPC  EQU     #2F6C1
LISTDC  EQU     #05839
MFERR   EQU     #09393
MLFFLG  EQU     #2F870
NTOKNL  EQU     #048E6
NXTSTM  EQU     #08A48
OUT3TK  EQU     #02D15
OUTBS   EQU     #2F58F
PCADDR  EQU     #2F679
POLL    EQU     #12337
PRINTt  EQU     #00001
PRPSND  EQU     #06B17
PRSC00  EQU     #07B93
RENSUB  EQU     #1A753
RESPTR  EQU     #03172
S-R0-2  EQU     #2F87B
S-R1-1  EQU     #2F886
SEND20  EQU     #17DFA
SENDEL  EQU     #17DC1
SENDWD  EQU     #17E15
STMTD1  EQU     #2F896
STMTR0  EQU     #2F871
STSAVE  EQU     #2F6BE

eFACCS  EQU     #0003C  ILLEGAL ACCESS
eFSPEC  EQU     #0003A  INVALID FILESPC
eSTMNF  EQU     #0001E  STATMENT NOT FOUND

fBASIC  EQU     #0E214
lEOL    EQU     #00002
oBSsod  EQU     #00011
oFLENh  EQU     #00020
oFLSTr  EQU     #00031
oFTYPh  EQU     #00010
pLIST   EQU     #0000C
t!      EQU     #000FC
tCOMMA  EQU     #000F1
tDATA   EQU     #000C6
tDEF    EQU     #000B9
tEND    EQU     #000DA
tENDDF  EQU     #000BA
tENDSB  EQU     #000C2
tLBLST  EQU     #000F6
tLINE#  EQU     #0000F
tREM    EQU     #000E6
tSUB    EQU     #000C1
        ENDTXT
```



```
        STITLE  Listing Structuré
*************************************************
*************************************************
* Nom : BLIST / PBLIST   ;     Type :  Statment
*
* But : Lister un programme BASIC au format  de
*       JPCLISTE, c.à.d. en sautant une ligne /
*       tracant  des  "="  avant  chaque  label,
*       SUB, DEF FN, DATA  ...  ou après chaque
*       END DEF, END SUB, END ALL. Ces derniers
*       seront pris en compte quelle  que  soit
*       leur position sur la ligne.
*
* Syntaxe: identique à LIST
*
*       BLIST [<filename>] [,début [,fin]]
*
* Entrée :
*       P= 0
*       D0 past tBLIST
*
*
```

```
* Utilisé :
*       R0-R3,S0-S11,STMTR0,STMTR1,STSAVE,FUNCR0,
*       FUNCR1 + tous les registres.
*
* Stack levels : 8
*
* Auteur : Jean-Pierre BONDU 05/86
************************************************
************************************************

       REL(5) BLSTDC
       REL(5) BLSTP

PBLSTs LC(2)  (PRINTt)*16+#F
       A=0    X        lève 'notre' S0
       A=A+1  X        *
       GONC   LIST05   B.E.T.

       REL(5) BLSTDC
       REL(5) BLSTP

BLSTs  LC(2)  (DISPt)*16+#F
       A=0    X        baisse 'notre' S0

LIST05 D1=(5) MLFFLG
       DAT1=C B
       D1=D1+ 11       D1 @ S-R0-2
       DAT1=A X        sauve nos drapeaux
       D1=(4) CURRST
       C=DAT1 A
       D1=C            D1 @ Début fichier
       A=DAT0 B
       GOSBVL EOLXCK   paramètres suivent ?
       GOC    LIST50   non
       LC(2)  tCOMMA
       ?A=C   B
       GOYES  LIST50
LIST10 GOSBVL FSPECx   spécificateur valide ?
       GOC    bserr    non

LIST20 GOSBVL FINDF+   cherche le fichier
       GOC    LSTEXT   pas trouvé ?

LIST50 D1=D1+ (oFTYPh)-1
       A=DAT1 A        FILE TYPE
       ASR    A
       D1=D1- (oFTYPh)-1
       GOSBVL BASCHA   BASIC FILE ?
       GOC    bserr    no

       GOSUB  PARMXQ
LIST60 GOSBVL GETPR1   TEST FILE PROTECTION
       GOC    bserr    "File Protect"

LIST65 D1=D1+ (oFLENh)-(oFTYPh)
       A=DAT1 A
       GOSUB  GETST1
       C=D    A
       D1=(5) S-R1-1
       DAT1=C A
       GOSUB  LNARGS
       D0=(5) S-R0-2
       C=DAT0 X
       D0=(4) STSAVE
       DAT0=C X
       GOC    LIST76   Cy vient de LNARGS
       GOTO   LIST80

bserr  GOVLNG BSERR

LSTEXT ?ST=1  6
       GOYES  bserr
       GOSBVL POLL
       CON(2) pLIST
pPOLL2 GOC    bserr    Handled and errored ?
       ?XM=0           Handled Okay ?
       GOYES  Nxtstm
* Not handled - Give correct error
       D=D+1  S
       D=D+D  S        external device
       GOC    fspcer
       ?A=0   W        No file name specified ?
       GOYES  fspcer
       LC(2)  eFACCS
Mferr  GOTO   mferr
fspcer LC(2)  eFSPEC
       GOC    Mferr    B.E.T.
Nxtstm GOTO   nxtstm

GETST1 CD1EX
       D0=C
       C=C+A  A
       D=C    A
       D0=D0+ (oBSsod)-(lEOL)
       GOVLNG BASCHK

       *
       * D1 @ Line To Print
       *
LIST76 GOSUB  ST<>RM   rétablit 'nos' drapeaux
       ST=0   3        baisse Multi line FN flag
JPL05  GOSUB  READtk   A[B]= token

*
* Traduit Ligne 2010
*
       LC(2)  tDATA
       ?A=C   B        DATA ?
       GOYES  JPL10    oui
       ST=0   7
       GONC   JPL20    B.E.T.

*
* Traite une ligne de DATA
*
```

```
JPL10  ST=1   9
       ?ST=0  7
       GOYES  dat1
dat1   ST=1   7
       GONC   JPL20    7 était levé
       ?ST#0  6
       GOYES  JPL20    6 était levé
       ST=1   8

*
* Traduit Ligne 2020
*
JPL20  ?ST=0  9
       GOYES  JPL22    9 est baissé
       ?ST#0  7
       GOYES  JPL22    7 est levé
       ST=1   8

*      Ajuste S9 sur S7
JPL22  ST=0   9
       ?ST=0  7
       GOYES  JPL30    on ne change rien
       ST=1   9

*
* Traduit Ligne 2030
*
JPL30  LC(2)  tLBLST
       ?A#C   B
       GOYES  JPL32
       ST=1   6
       ST=1   8
       GONC   JPL40    B.E.T.
JPL32  ST=0   6
       LC(2)  tDEF
       ?A#C   B
       GOYES  JPL40
*
* Une User FN est tokenisée comme suit :
*     tDEF <5 nibbles field> tFN  tVAR ...
*     tFN = 00 => FN multiligne
* Il faut que DEF FN, SUB, LABEL, DATA soient en
* début de ligne par contre END DEF, END SUB, END
* ALL sont repérés quelle que soit leur position.
* Cette recherche est assurée par SRCtk.
*
       ST=1   8
       ST=0   3        Multi line FN flag

* S3 = 1 : FN monoligne (DEF FNx(..)= ...)
* S3 = 0 : FN multiligne (DEF FNx(..) @ ...)

       D1=D1+ 13       D1 @ tFN
       C=DAT1 B
       D1=D1- 13
       ?C=0   B        FN multiligne ?
       GOYES  JPL40
       ST=1   3        FN monoligne
* Il reste à vérifier que la FN n'est pas une
* 'fausse' monoligne (i.e. qu'un END DEF n'est
* pas sur la même ligne.) Cela sera éffectué  par
* JPL60.

*
* Traduit Ligne 2040
*
JPL40  D0=(5) STMTD1
       CD1EX           D1 en STMTD1
       DAT0=C A
       CD1EX
       LC(2)  tSUB
       ?A=C   B
       GOYES  PRT"=
       ?ST=0  8
       GOYES  JPL50

*
* IMPRIME  62  '='
*
PRT"=  ST=0   8
       GOSUB  ST<>RM   Sauve 'nos' drapeaux
       GOSBVL CKINFO
       GOSBVL SENDEL   saute une ligne
       C=0    A
       GOSUB  SENDSP   effectue indentation
       GOSUB  D10@BS
       C=0    A
       LC(2)  len"=
       B=C    B        compteur
       A=C    A        A[A]= longueur buffer
       B=B-1  B        B[B]= compteur
       LCASC  '='
loop1  DAT0=C B
       D0=D0+ 2
       B=B-1  B
       GONC   loop1
       ST=1   4
       GOSBVL SENDWD   Affiche le buffer de
*                      longueur A[A], D1 @ début
       GOSBVL SENDEL   SEND End of Line
       *
       * Restore D1 en début de ligne
       *
       GOSUB  RCALD1
       GOSUB  ST<>RM   rétablit 'nos' drapeaux

*
* Traduit Ligne 2050
*
JPL50  ST=0   4
       ST=0   2
       LC(2)  tREM
       ?A=C   B
       GOYES  JPL52
       LC(2)  t!
```

```
       ?A#C   B
       GOYES  JPL55
       ST=1   2

JPL52  ST=1   4        signale ligne de REM
JPL55  GOSUB  ST<>RM   sauve 'nos' drapeaux
       GOSBVL LDCM10   décompile
       GOSUB  SETPOS   effectue tabulation
       GOSUB  ST<>RM   rétablit 'nos' drapeaux
       GOSUB  D10@BS   D1 @ ligne décompilée

* Maintenant il faut modifier le No de ligne
* si c'est une ligne de REM.

       ?ST=0  4        ligne REM ?
       GOYES  JPL60    non

* ================================================
* FORMATE UNE LIGNE DE REM :
*    - Remplace le No de ligne par des espaces
*    - Supprime le symbole "!"
*    - Rajoute '-' si 1er ligne de REM
* Entrée:
*    D1 & D0 @ début No ligne (OUTBS)
*    B[A] = longueur ligne en octets
*           ( NE PAS Y TOUCHER ! )
*    S5=0 => 1er ligne de REM
*    S4=1 => ligne de REMarque
*       0 => <> ligne de REMarque
*    S2=1 => ... de type "! "
*       0 => ... de type "REM "
* ================================================

       LCASC  ' '
loop3  A=DAT0 B        A[B]=chifrre du No ligne
       ?A=C   B        fin No ligne ?
       GOYES  JPL57    oui
       DAT0=C B        efface chiffre
       D0=D0+ 2
       GONC   loop3    B.E.T.

*
* Traduit ligne 2140
*
JPL57  D0=D0+ 2        D0 @ "! " ou "REM "
       A=0    W
       P=     7
       DAT0=A 4        efface "! " (ou "RE")
       ?ST=1  2        commence par "! " ?
       GOYES  suit
       DAT0=A WP       efface "REM "
suit   P=     0
       D0=D0- 4        D0 @ fin No ligne
       ?ST=0  5        1er ligne REM ?
       GOYES  JPL58    oui: Carry Set

JPL58  ST=1   5
       GONC   JPL59    <> 1er ligne REM
       D0=D0+ 8
       ADOEX
       D0=A
       D0=D0- 8
       CD1EX
       A=A-C  A        longueur du No de ligne
       ASRB                 en octets.
       ?B<=A  A        ligne de REM vide ?
       GOYES  JPL59
       LCASC  '-'
       DAT0=C B
JPL59  GOTO   Next

*
* Imprime une ligne <> REM
* Traduit  ligne  2060
*
JPL60  ST=0   5
       GOSUB  RCALD1
       ?ST=1  3        1 line User FN ?
       GOYES  JPL62
       LC(2)  tEND
       ?A=C   B
       GOYES  JPL62
       LC(2)  tENDDF
       GOSUB  SRCtk
       GOC    JPL62
       LC(2)  tENDSB
       GOSUB  SRCtk
       GONC   Next
JPL62  ST=1   8


Next   GOSUB  ST<>RM   sauve 'nos' drapeaux
       GOSBVL PRPSND   affiche Output Buffer


       D1=C            D1 @ NEXT LINE
LIST80 A=0    A
       A=DAT1 4        READ IN LINE#
       C=R3            LAST LINE TO DISPLAY
       ?A>C   A        DONE LISTING ?
       GOYES  nxtstm

*
* CHECK FOR ATTN KEY DOWN & ANY SERVICE REQUESTS
*

       CD1EX
       GOSBVL CK"ON"
       D1=C
       GONC   nxtstm   ATTN KEY DOWN
       GOTO   LIST76
nxtstm GOVLNG NXTSTM
```

```
************************************************
*               SOUS ROUTINES                  *
************************************************

LNARGS D0=D0+ (lEOL)   -- LiNe ARGumentS --
       CD0EX           HP71B : #06B62
       D1=C
       A=R1
       B=A    A
       A=0    A
       GOSBVL FINDL0
       RTNC
       ?ST=1  0
       GOYES  nxtstm
       ?ST=1  10
       GOYES  nxtstm
       RTNCC

RCALD1 D1=(5) STMTD1   -- ReCALl D1 --
       C=DAT1 A
       D1=C            D1 @ ligne compilée
READtk D1=D1+ 6        skip over LINE# & tLEN
       A=DAT1 B        A[B]= token du premier mot
       D1=D1- 6
       RTN             RETURN Carry Clear

*
* Echange STSAVE avec STatus
*
ST<>RM D0=(5) STSAVE   -- STatus <> RaM --
       C=DAT0 X
       CSTEX
       DAT0=C X
       RTN

D10@BS D0=(5) OUTBS
       C=DAT0 A
       D0=C
       D1=C
       RTN
* Sortie: C[A] = D0 = D1 = Output Buffer Start

* ================================================
*         ->  SRCtk  <-
* Recherche sur une ligne compilée un token
*
* Entrée :
*      P=0
*      C[B]= token recherché
* SRCtk  : D1 @ No ligne
* SRCtk+ : D0 @ Stmt lenght
*
* Sortie :
*      D1 = D1 en entrée
*      B[A]=B[A]    "
*      CARRY SET => trouvé, D0 @ token
*      CARRY CLR => pas trouvé
* ================================================

SRCtk  CD1EX           -- SeaRCh token --
       D0=C
       D0=D0+ 4
       CD1EX
SRCtk+ BCEX   A        B[B]= token recherché
       RSTK=C          sauve B[A] dans RSTK
cont   C=0    A
       C=DAT0 B
       D=C    A        D[A]= longeur du Statment
       D0=D0+ 2
       A=DAT0 B        lit token
       ?A=B   B        token recherché ?
       GOYES  rtn
       CD0EX
       C=C+D  A
       CD0EX
       D0=D0- 2        D0 @ tEOL ou t@
       C=DAT0 B
       D0=D0+ 2
       ?C#0   P        Not EOL ?
       GOYES  cont
rtn    C=RSTK
       B=C    A
       RTN

* ================================================
*         ->  SETPOS  <-
* Positionne le curseur où commencera
* l'impression de la ligne en fonction de
* la longueur du No de ligne.
*
* Entrée :  = sortie de LDCM10
*       (OUTBS) @ DC Line
*       B[A]= LEN(OUTBS)
*       STSAVE contient 'nos' drapeaux
*
* Sortie :  = sortie de SENDSP
*       B[A]= LEN(OUTBS)
* POSition du curseur a été écrite dans STMTR0.
* ================================================

SETPOS GOSUB  D10@BS   C @ BS
       D1=(5) LDCSPC
       A=DAT1 A        A @ espace
       A=A-C  A
       ASRB   B        A[0]= LEN(#line) octets
       LC(2)  4        longueur MAX d'1 ligne#
       C=C-A  B        offset pour TAB

* ================================================
*         ->  SENDSP  <-
* Envoie (colprt+offset) espaces au périphérique
* de sortie (DISPLAY ou PRINTER).
*
* Entrée :
*      C[B]= offset
*      B[A]= quelconque
*      STSAVE contient 'nos' drapeaux
```

```
* Sortie :
*      B[A]= B[A] en entrée
* ================================================
SENDSP A=C    B
       D1=(5) STSAVE   charge 'nos' drapeaux
       A=DAT1 XS       0 à 3 en A[XS]
* S0=0 => offset = 0      (BLIST)
* S0=1 => offset = colprt (PBLIST)
       C=R3
       GOSBVL CSLW5
       C=B    A
       GOSBVL CSLW5
       C=0    A
       C=C+1  XS
       A=A&C  XS
       ?A=0   XS       S0 baissé ?
       GOYES  TAB10    oui
       LC(2)  colprt
TAB10  C=0    XS
       C=C+A  B        #espaces total à envoyer
       R3=C
       GOSBVL CKINFO
       D1=(5) (STMTR0)+6
       C=DAT1 A
       D1=C
       C=0    A
       C=DAT1 B
       A=R3
       B=C    A
       D1=D1+ 2
       C=DAT1 B        lit WIDTH
       ?C#0   B
       GOYES  TAB20
       C=C+1  XS
TAB20  ?A<C   A        TAB>WIDTH ?
       GOYES  TAB30
       A=A-C  A
       GONC   TAB20    B.E.T.
TAB30  A=A-B  A
       R3=A
       GONC   TAB40
       A=A+B  A
       R3=A
       GOSBVL SENDEL
TAB40  LCASC  '        ' C=8 espaces
       D1=(5) FUNCR0
       DAT1=C W
TAB50  LC(5)  FUNCR0
       D=C    A        adresse
       C=0    A
       LC(1)  8        C=8
       A=R3
       A=A-C  A
       GOC    TAB60
       R3=A
       A=C    A
       GOSBVL SEND20
       GOTO   TAB50

TAB60  A=A+C  A
       GOSBVL SEND20
       C=R3            RESTORE B[A]
       GOSBVL CSRW5
       B=C    A
       GOSBVL CSRW5
       R3=C
       RTN


PARMXQ ST=0   9        HP71B : #06887
       C=0    A
       C=C+1  A
       R1=C            Parm 1 = 00001 par défaut
       LCHEX  9999
       R3=C            Parm 2 = 09999 par défaut
       ST=0   10

PRMXQ0 A=DAT0 B
       LC(2)  tCOMMA
       ?A#C   B        pas de paramètres ?
       RTNYES
       D0=D0+ 1
       A=DAT0 A
       ASR    A
       D0=D0+ 5
       R1=A
       ?ST=1  9
       GOYES  PRMXQ1
       ST=1   10
       A=0    A
       R3=A

PRMXQ1 A=DAT0 B
       ?A#C   B
       RTNYES
       ST=0   10
       D0=D0+ 2
       A=DAT0 4
       R3=A
       ?ST=1  9
       RTNYES

       C=R1
       ?C>A   A        parm1 > parm 2 ?
       GOYES  invarg   oui
       RTN

invarg GOVLNG ARGERR

       STITLE  Renumérotation des Remarques
```

```
************************************************
************************************************
* Nom :  RENUMREM         ;     Type :  Statment
*
* But : Renuméroter un programme BASIC sans
*       que  les  lignes  de  REM viennent
*       perturber cette numérotation (dans
*       la mesure du possible).
*
* Syntaxe: identique à RENUMBER
*
* Note: RENUMREM 100 , 10 , 200 , 500
*                -v-   -v-  -v-   -v-
* registres ->    R0    R1  @R2    R3
************************************************
************************************************

       REL(5) RNUMDC
       REL(5) RNUMP

RENUMe CD0EX
       R0=C
       GOSUB  CHKPSF
       GOSUB  GETSTe
       GOSBVL PRSC00

REN005 ST=1   1
       ST=1   2
       GOSBVL RENSUB
       GOC    REN010
       D1=(5) PCADDR
       C=R2
       DAT1=C A
       GOSUB  UPDCRL
       LC(2)  eSTMNF   "Statement Not Found"
       GOTO   mferr

REN010 A=R0
       D0=A
       GOSUB  LINE#1
       R2=A
       C=0    A
       C=C+1  A
       CSL    A
       R0=C
       R1=C
       CSL    A
       CSL    A
       CSL    A
       R3=C
       GOSUB  GETLN#
       GOC    REN100
       R0=A
       GOSUB  GETLN#
       GOC    REN100
       R1=A
       GOSUB  GETLN#
       GOC    REN100
       C=A    A
       AD0EX
       R2=A
       GOSBVL FINDL
       ?ST=0  0
       GOYES  REN020
       GOTO   nxtstm

REN020 CD1EX
       CR2EX
       D0=C
       GOSUB  GETLN#
       GOC    REN100
       SETDEC
       P=     3
       A=A+1  WP
       P=     0
       SETHEX
       GOC    REN100
       C=A    A
       GOSBVL FINDL
       A=0    A
       A=DAT1 4
       ?ST=1  0
       GOYES  REN100
       R3=A

REN100 GOSUB  LINE#1
       CD1EX
       A=R2
       ?A=C   A
       GOYES  REN105
       D0=A
       D0=D0- 4
       GOSBVL CPL#10
       C=0    A
       C=DAT1 4
       A=R0
       ?A>C   A
       GOYES  REN105
       GOTO   invarg

REN105 D1=(5) CURREN
       C=DAT1 A
       D=C    A
       C=R2
       D1=C
       C=0    A
       C=DAT1 4
       D0=C
       C=R0
       RSTK=C

REN110 C=R2
       D1=C
       A=R1
       B=A    A
```

```
REN120 CD1EX
       ?C>=D  A
       GOYES  ren170
       D1=C
       A=0    A
       A=DAT1 4
       C=R3
       ?A>=C  A
       GOYES  ren170
       A=R0
       ?A<C   A
       GOYES  REN140
REN130 SETHEX
       CD0EX
       R0=C
       C=0    A
       C=C+1  A
       R1=C
       GONC   REN110
ren170 GOTO   REN170

************************************************
*   ORGANIGRAMME DE REN140 à REN155:
*   --------------------------------
*
* Entrée: RSTK = Anc [Next REM Line#]
*         R0   = Nv  [Next BASIC Line#]
*         B[B] = Inc [Incrément]
*
*     -----------
*     |ECRIRE Nv|
*     -----------REN140
*         |
*     -----------
*    <LIGNE REM ?>-----non-------
*     -----------                |
*         |oui                   |
*     ------------               |
*     |ECRIRE Anc|               |
*     ------------REN145         |
*         |                      |
*     -----------          ----------
*     |Anc=Anc+1|          |Anc=Nv+1|
*     -----------          ----------
*         |                      |
*     ---------                  |
*    < Anc>NV ?>--non--          |
*     ---------       |          |
*         |<----------------------
*    -----------      |
*    |Nv=Nv+Inc|REN   |
*    -----------150   |
*         |<-----------
*         |
*     #########
*     #REN 155#
*     ######### FIN
************************************************

* --------------------
* ----- ECRIRE Nv -----
* --------------------
REN140 DAT1=A 4        LINE# = Nv
       GOSUB  READtk   A[B]= token
       LC(2)  t!
       ?A=C   B        Ligne REM ?
       GOYES  REN145   oui
       LC(2)  tREM
       ?A=C   B        Ligne REM ?
       GOYES  REN145   oui
       A=R0            non: Anc = Nv+1
       C=RSTK
       C=A    A
       SETDEC
       C=C+1  A
       RSTK=C
       GOTO   REN150
* --------------------
* ----- ECRIRE Anc ----
* --------------------
REN145 C=RSTK          C=Anc
       DAT1=C 4        Line# = Anc
       SETDEC
       C=C+1  A        Anc = Anc+1
       RSTK=C          Sauvegarde Anc
*
* On est toujours en DEC
*
       A=R0            A=Nv
       P=     3
       ?C<=A  WP       Anc <= Nv ?
       GOYES  REN155   oui alors FIN
*
* Entree: A=Nv ; B=Inc ; DEC
*
REN150 A=A+B  A        Nv = Nv+Inc
       R0=A            Sauve Nv
*
* FIN: on remet tout en ordre pour REN160
*
REN155 SETHEX
       P=     0
       D1=D1+ 4        D1 @ tLEN
       C=0    A

REN160 C=DAT1 B
       AD1EX
       A=A+C  A
       D1=A
       A=DAT1 B
       D1=D1+ 2
       ?A#0   P
       GOYES  REN160
       GOTO   REN120
```

```
REN170 C=RSTK
       ST=1   1
       ST=0   2
       GOSBVL RENSUB
       GOTO   nxtstm

GETLN# A=DAT0 B
       D0=D0+ 2
       LC(2)  tCOMMA
       ?A#C   B
       RTNYES
       A=0    A
       A=DAT0 4
       D0=D0+ 4
       RTNCC

LINE#1 D1=(5) CURREN
       C=DAT1 A
       D=C    A
       D1=D1- 15
       A=DAT1 A
       C=0    A
       LC(2)  oFLSTr
       A=A+C  A
       D1=A
       RTN

CHKPSF GOSUB  GETSTe
       GOSBVL GETPRO
       GOC    mferr
       ?SB=0
       RTNYES
       GONC   mferr

GETSTe GOSBVL GETSTC
       RTNNC
mferr  GOVLNG MFERR

UPDCRL D1=(5) CURRST
       C=DAT1 A
       D1=C
       D1=D1+ (oFTYPh)-1
       A=DAT1 A
       ASR    A
       LC(5)  fBASIC
       ?A#C   A
       GOYES  UPDCR1
       GOSBVL D0=PCA
       GOSBVL CPL#10
       C=DAT1 A
       GONC   UPDCR3
UPDCR1 C=0    A
UPDCR3 D0=(5) CURRL
       DAT0=C 4
       RTNCC
```

```
*************************************************
*************** ROUTINES D'ANALYSE ***************
*************************************************

* [<filename>] [,line#1 [,line#2]]
* Très inspiré de LISTP (#03B92 / HP71B)
BLSTP  GOSUB  eolck+
       GOC    NAMEP7   pas de paramètre : fin
       GOSBVL FSPECp   spécificateur OK ?
       GONC   LSTP12   oui
* ILLEGAL  FILE  SPEC
       ?ST=0  7
       GOYES  LSTP20

RNMP65 GOVLNG FSPECe

LSTP12 GOSBVL COMCK
       GOC    LSTP20
NAMEP7 GOVLNG RESPTR

LSTP20 GOSUB  LSTP40
       GOSBVL COMCK
       GONC   NAMEP7

       ST=1   9
LSTP40 GOSBVL NTOKNL
       LC(2)  tLINE#
       ?A#C   B
       GOYES  LSTPE
       LC(2)  tCOMMA
       A=C    B
       GOVLNG OUT3TK

LSTPE  GOVLNG IVPARe   INVALID (MISSING PARM)

eolck+ GOSBVL EOLCK    fin de commande ?
       RTNC           oui : Cy=1
       GONC   NAMEP7   Cy=0 & RESPTR (B.E.T.)

RNUMP  GOSUB  eolck+
       GOC    NAMEP7
       ST=0   9
       GOSUB  LSTP20
       ?ST=1  9
       GOYES  LSTP12
LSTPDN RTNCC

*************************************************
************ ROUTINES DE DECOMPILATION ***********
*************************************************

RNUMDC
BLSTDC GOVLNG LISTDC

       END
```

## FORTHEMENT COMPLEXE

A la suite de l'article de Alain Herreman "HP71: Tellement complexe !" (JPC No 28), j'ai essayé son programme et je l'ai trouvé très lent. Il n'en fallait pas plus pour que je me mette à l'ouvrage.

Principal problème, donc : comment faire plus rapide ?

Tout d'abord, éviter de manipuler des chaînes de caractères ; ensuite et surtout penser très FORTH.

En effet, le FORTH du HP71 est doté d'une pile dans le style du HP41 avec des mots associés. Alors, quoi de plus naturel que d'utiliser ce vocabulaire ?

Commençons donc par créer un double de la pile dans un buffer. Un mot spécial permute ces deux piles. L'une contient les parties réelles, l'autre les parties imaginaires. Le principe des mots que l'on définit ensuite est qu'ils agissent successivement sur ces deux piles dans des opérations distinctes, mais qui utilisent les mots standard du FORTH.

Exemple: l'addition. Il suffit d'additionner les deux parties réelles, permuter les deux piles, additionner les deux parties imaginaires et remettre les piles en ordre. Simple, non ?

Bien sûr, ce schéma ne permet pas d'inclure dans le vocabulaire des fonctions plus... complexes comme la multiplication, à moins de détruire toute ou partie de la pile. D'où l'intérêt, entre autres, d'utiliser un programme BASIC.

Les principaux mots FORTH utiles sont les suivants :

COMPLEX_MODE passe en mode complexe et dédouble la pile si ce n'était déjà fait.

REAL_MODE repasse en mode normal. La pile n'est pas détruite.

R<>I permute les deux piles réelle et imaginaire.

FUNCTION_COMPLEX récupère le CFA d'une fonction sur la pile de données et l'exécute sur les deux piles flottantes seulement si on est en mode complexe (voir fonction Z+ par exemple).

Le programme BASIC ci-joint facilite grandement l'exploitation de ces fonctions :

L'introduction des données et l'affichage des résultats sont beaucoup plus souples.

L'affectation des commandes aux touches du clavier rend le programme très pratique d'emploi. La liste de ces assignations doit normalement figurer avec le programme. Elles sont de la forme :

DEF KEY '+','ADD':

Le nom du fichier des assignations doit être CMPLXKEY ; le programme sauvegarde vos propres assignations dans le fichier TEMPKEYS et remet tout en place à la fin.

La gestion de la pile est dévolue au FORTH qui s'en sort très bien tout seul.

Les fonctions plus complexes peuvent être réalisées sous BASIC sans perturber la pile. Dans ce cas, il peut être tout de même nécessaire d'en assurer la gestion (voir la fonction MUL par exemple).

L'introduction d'un nombre est particulièrement simple : appuyez sur I ('INPUT') ; le programme vous demande d'introduire la partie réelle puis la partie imaginaire. N'importe quelle expression arithmétique valide peut être introduite, la lecture étant assurée par un LINPUT suivi d'un VAL.

Quant aux fonctions de stockage sur la pile, au message STO ou RCL, appuyez sur X, Y, Z, T ou L suivant votre choix. Le programme fait écho de la touche pressée avant d'exécuter la fonction.

Toute erreur renvoie à la demande d'une commande.

Il n'y a pas d'inhibition de la montée de la pile.

Les possesseurs du module MATHS peuvent très bien incorporer les fontions complexes qu'il apporte pour compléter ce programme ou l'améliorer.

Les fonctions déjà programmées sont: ADD (+), SOUS (-), MUL (*), DIV (/), INV (1/Z), CLZ, RUP RDN, LASTZ, ENTER, DROP, STO et RCL (X Y Z T L) ECH (X<>Y), MOD (module et argument), INPUT, END.

Pour terminer, je remercie Alain Herreman, car je me suis (largement) inspiré de son programme pour le graphisme de l'affichage.

Alain Goubault de Brugière

Fichier CMPLXKEY :

```
DEF KEY 'Q','END':
DEF KEY 'E','ENTER':
DEF KEY 'R','RCL':
DEF KEY 'I','INPUT':
DEF KEY '/','DIV':
DEF KEY 'S','STO':
DEF KEY 'D','DISP':
DEF KEY 'K','DROP':
DEF KEY 'L','LASTZ':
DEF KEY '*','MUL':
DEF KEY 'X','ECH':
DEF KEY 'C','CLZ':
DEF KEY 'M','MOD':
DEF KEY '-','SOUS':
DEF KEY '#43','';
DEF KEY '#47','';
DEF KEY '#48','';
DEF KEY '#50','RUP':
DEF KEY '#51','RDN':
DEF KEY '+','ADD':
DEF KEY '#99','';
DEF KEY 'f ','BAR':
DEF KEY 'f0','';
DEF KEY ':','INV':
DEF KEY '#155','';
DEF KEY '#159','';
DEF KEY '#160','';
DEF KEY '#162','';
DEF KEY '#163','';
```

Fichier CMPLXFTH :

```
( Vocabulaire définissant des mots pour la       )
( gestion d'une pile complexe en FORTH           )
( A. Goubault 08/86                              )

VOCABULARY COMPLEX_WORDS

: COMPLEX COMPLEX_WORDS DEFINITIONS ;

COMPLEX

( COMPLEX_BUFFER contient l'adresse de la pile   )
( imaginaire                                     )

VARIABLE COMPLEX_BUFFER 0 COMPLEX_BUFFER !
VARIABLE COMPLEX_STATUS 0 COMPLEX_STATUS !
HEX

( Construction de la double pile dans un "buffer")
( de 40 octets                                   )
: COMPLEX_STACK   ( -- adr )
  28 MAKEBF
  IF
    COMPLEX_BUFFER !
    DUP 28 0 FILL
  ELSE
    ABORT " No Room"
  THEN ;

( On construit la double pile si ce n'est pas    )
( déjà fait                                      )
: COMPLEX_STACK?  ( -- adr )
  COMPLEX_BUFFER @
  FINDBF DUP 0=
  IF COMPLEX_STACK
  THEN ;

( Permutation des deux piles                     )
: R<>I            ( -- )
  COMPLEX_STACK? L
  DUP PAD 28 CMOVE
  OVER SWAP 28 CMOVE
  PAD SWAP 28 CMOVE ;

( Activation du mode complexe                    )
: COMPLEX_MODE    ( -- adr )
  COMPLEX_STACK?
  DROP -1 COMPLEX_STATUS ! ;

( Retour au mode normal                          )
: REAL_MODE       ( -- adr )
  0 COMPLEX_STATUS ! ;

( Fonction agissant de manière identique sur les )
( deux piles si le mode complexe est activé      )
: FUNCTION_COMPLEX ( CFA -- )
  DUP EXECUTE COMPLEX_STATUS @
  IF
    R<>I EXECUTE R<>I
  ELSE
    DROP
  THEN ;

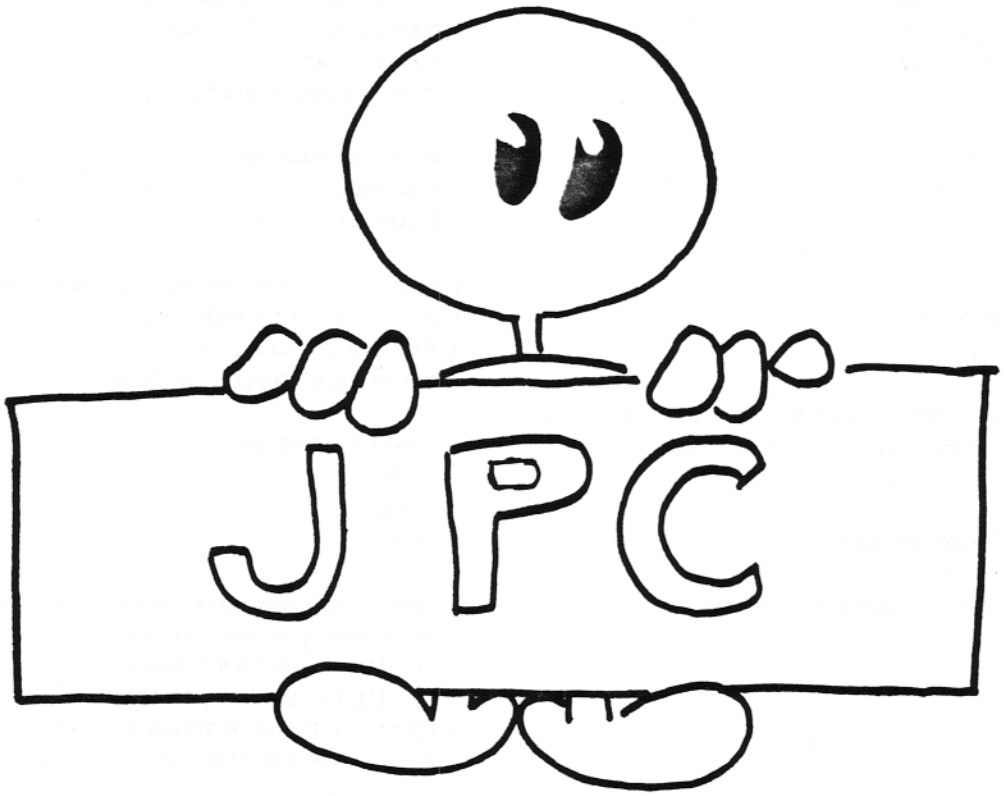
( Fonctions élémentaires agissant sur les deux   )
( si le mode complexe est activé                 )
: Z+  ['] F+ FUNCTION_COMPLEX ;
: Z-  ['] F- FUNCTION_COMPLEX ;
: COPYZ ['] FENTER FUNCTION_COMPLEX ;
: ZDN ['] RDN FUNCTION_COMPLEX ;
```

```
: ZUP ['] RUP FUNCTION_COMPLEX ;
: ECHZ ['] X<>Y FUNCTION_COMPLEX ;
: LASTZ ['] LASTX FUNCTION_COMPLEX ;
: ZDROP ['] FDROP FUNCTION_COMPLEX ;

( Fonction conjugué                              )
: ZBAR
  COMPLEX_STATUS @
  IF
    L STO R<>I L STO CHS R<>I
   ELSE
    1/X
  THEN ;

( Fonctions de stockage sur la pile              )
: ZSTO            ( adr -- )
  DUP STO
  COMPLEX_STATUS @
  IF
    R<>I STO R<>I
   ELSE
    DROP
  THEN ;
: ZRCL            ( adr -- )
  DUP RCL
  COMPLEX_STATUS @
  IF
    R<>I RCL R<>I
   ELSE
    DROP
  THEN ;

DECIMAL
```



Programme "LISTPRIM" (sous-programmes pour l'utilisation de DIVILEX)

```
================================================================
  10 SUB LPRIM(N,M)
  20 N=FPRM(N,M) @ IF N THEN DISP N; @ N=N+2 @ GOTO 20
  30 END

================================================================
  40 SUB DECOMP(A$) @ DIM A,B,C,H,R @ L=LEN(A$) @ IF L>15 THEN DISP "N < 10^15" @ END
  50 IF L<4 THEN H=0 @ B=VAL(A$) @ GOTO 70
  60 H=VAL(A$[1,L-3]) @ B=VAL(A$[L-2,L])
  70 A=PRIM(H,B)
  80 D=1 @ IF NOT A THEN DISP A$ @ END
  90 R=MOD(H,A) @ H=H DIV A @ B=R*1000+B @ B=B DIV A @ C=PRIM(H,B) @ IF C=A THEN D=D+1 @ GOTO 90
 100 IF H<1000000000 AND H*1000+B=A THEN D=D+1
 110 IF D=1 THEN DISP A; @ GOTO 130
 120 DISP STR$(A)&CHR$(94)&STR$(D)&" ";
 130 IF H<1000000000 AND H*1000+B=A THEN END
 140 IF NOT H THEN A$=STR$(B) @ A=C @ GOTO 80
 150 A$=STR$(H) @ IF B<100 THEN A$=A$&"0"
 160 IF B<10 THEN A$=A$&"0"
 170 A$=A$&STR$(B) @ A=C @ GOTO 80

================================================================
 180 SUB EQUZ(A,B,C) @ D=PGCD(A,B)
 190 A=A/D @ B=B/D @ C=C/D @ IF FP(C) THEN 300
 200 X1=0 @ Y2=0
 210 Y1=(C-X1*A)/B @ IF NOT FP(Y1) THEN 260
 220 X1=-X1+1*(X1<=0)
 230 X2=(C-Y2*B)/A @ IF NOT FP(X2) THEN 250
 240 Y2=-Y2+1*(Y2<=0) @ GOTO 210
 250 X1=X2 @ Y1=Y2
 260 DISP "X =";X1;"+";ABS(B);"* K  ";
 270 DISP "Y =";Y1;
 280 IF SGN(A*B)+1 THEN DISP "-"; ELSE DISP "+";
 290 DISP ABS(A);"* K" @ END
 300 DISP "Pas de solution" @ END

================================================================
 310 SUB JUM(N) @ A=FPRM(N)
 320 B=FPRM(A+2) @ IF B=A+2 THEN DISP A;B
 330 A=B @ GOTO 320

================================================================
 340 SUB QUAD(N) @ A=FPRM(N)
 350 B=FPRM(A+2) @ IF B#A+2 THEN A=B @ GOTO 350
 360 C=FPRM(B+4) @ IF C#B+4 THEN A=C @ GOTO 350
 370 D=FPRM(C+2) @ IF D#C+2 THEN A=D @ GOTO 350
 380 DISP A;B;C;D @ A=C @ GOTO 350

================================================================
 390 SUB FPRM(A$) @ IF LEN(A$)<13 THEN DISP FPRM(VAL(A$)) @ END
 400 CFLAG 0 @ IF A$[1,1]="-" THEN SFLAG 0 @ A$=A$[2]
 410 L=LEN(A$) @ IF L>15 THEN 550
 420 H=VAL(A$[1,L-3]) @ B=VAL(A$[L-2,L])
 430 IF H=1.E+12-1 AND B>989 THEN 550
```

```
440 IF FLAG(0) THEN 490
450 IF NOT MOD(B,2) THEN B=B+1
460 A=PRIM(H,B) @ IF NOT A THEN 520
470 B=B+2 @ IF B=1001 THEN H=H+1 @ B=1
480 GOTO 460
490 IF NOT MOD(B,2) THEN B=B-1
500 IF H AND B<0 THEN B=999 @ H=H-1
510 A=PRIM(H,B) @ IF NOT A THEN 520 ELSE B=B-2 @ GOTO 500
520 IF H THEN DISP STR$(H);
530 IF B<10 THEN DISP "00"; ELSE IF B<100 THEN DISP "0";
540 DISP STR$(B) @ END
550 DISP "N <= 999999999999989" @ END

============================================================
560 SUB REDUC(K) @ IF K=1 THEN J=2 ELSE J=K
570 OPTION BASE 1 @ DIM N(J),D(J) @ STD @ IF K=1 THEN N(2)=1 @ D(2)=1
580 FOR I=1 TO K @ DISP "N"&STR$(I)&",D"&STR$(I)&" "; @ INPUT N(I),D(I)
590 G=PGCD(N(I),D(I)) @ N(I)=N(I)/G @ D(I)=D(I)/G @ NEXT I
600 IF K=1 THEN P=D(1) @ GOTO 620
610 P=PPCM(D(1),D(2)) @ FOR I=3 TO J @ P=PPCM(P,D(I)) @ NEXT I
620 DELAY 8 @ FOR I=1 TO K @ N(I)=P/D(I)*N(I) @ IF N(I)>1.E+12-1 THEN DISP "OVFL" @ END
630 DISP "F"&STR$(I)&" = "&STR$(N(I))&" / "&STR$(P) @ NEXT I @ DELAY .5 @ END

============================================================
640 SUB PRIM(A$) @ L=LEN(A$) @ IF L>15 THEN DISP "N < 10^15" @ END
650 IF L<13 THEN DISP PRIM(VAL(A$)) @ END
660 H=VAL(A$[1,L-3]) @ B=VAL(A$[L-2,L])
670 DISP PRIM(H,B) @ END
  - Guy Toublanc  (P276)      28/9/86
```

**********

Programme "COMPLEX" (calculs sur nombres complexes)

```
  - HP71 Complex Stack
    A.Goubault JULY 86
    Uses FORTH and KEYWAIT$ (LEX)
 10 ON ERROR GOTO 20 @ COPY KEYS TO TEMPKEYS @ OFF ERROR
 20 PURGE KEYS @ MERGE CMPLXKEY
 30 FORTHX 'FORTH COMPLEX COMPLEX_MODE'
 40 GDISP CHR$(62)&CHR$(65)&CHR$(127)&CHR$(65)&CHR$(34) @ WINDOW 3
 50 DIM Z$[22],C$[5],K$[1],N$[3]
 60 USER ON @ DELAY 0,0 @ ENG 2 @ DEGREES

============================================================
 70 'D': GOSUB 'GET'

============================================================
 80 'DISP': S=SGN(Y)+2 @ Z$=STR$(X)&'-++'[S,S]&STR$(ABS(Y))&'i'
 90 ON ERROR GOTO 100
100 WINDOW 3 @ DISP Z$ @ WINDOW 22 @ INPUT C$ @ GOTO C$

============================================================
110 'INPUT': USER @ WINDOW 3
120 LINPUT 'Re? ','0';X$ @ LINPUT 'Im? ','0';Y$
```

```
130 X=VAL(X$) @ Y=VAL(Y$) @ USER
140 GOSUB 'PUT' @ GOTO 'DISP'

================================================================
150 'END': FORTHX 'REAL_MODE FORTH DEFINITIONS' @ WINDOW 1 @ STD @ USER OFF
160 ON ERROR GOTO 170 @ PURGE KEYS @ RENAME TEMPKEYS TO KEYS
170 END

================================================================
180 'ADD': FORTHX 'Z+' @ GOTO 'D'

================================================================
190 'SOUS': FORTHX 'Z-' @ GOTO 'D'

================================================================
200 'CLZ': FORTHX 'ZDN' @ X=0 @ Y=0 @ GOSUB 'PUT' @ GOTO 'DISP'

================================================================
210 'ECH': FORTHX 'ECHZ' @ GOTO 'D'

================================================================
220 'ENTER': FORTHX 'COPYZ' @ GOTO 'DISP'

================================================================
230 'INV': SFLAG 7 @ 'BAR': FORTHX 'ZBAR' @ Y=-Y
240 IF FLAG(7,0) THEN A=X^2+Y^2 @ X=X/A @ Y=Y/A @ GOSUB 'PUT'
250 GOTO 'DISP'

================================================================
260 'RUP': FORTHX 'ZUP' @ GOTO 'D'

================================================================
270 'RDN': FORTHX 'ZDN' @ GOTO 'D'

================================================================
280 'LASTZ': FORTHX 'LASTZ' @ GOTO 'D'

================================================================
290 'DIV': SFLAG 5 @ 'MUL': U=X @ V=Y @ FORTHX 'L ZSTO ZDROP' @ GOSUB 'GET'
300 N=1 @ IF FLAG(5,0) THEN V=-V @ N=U^2+V^2
310 A=U*X-V*Y @ B=U*Y+V*X @ X=A/N @ Y=B/N
320 FORTHX 'ZDROP' @ GOSUB 'PUT' @ GOTO 'DISP'

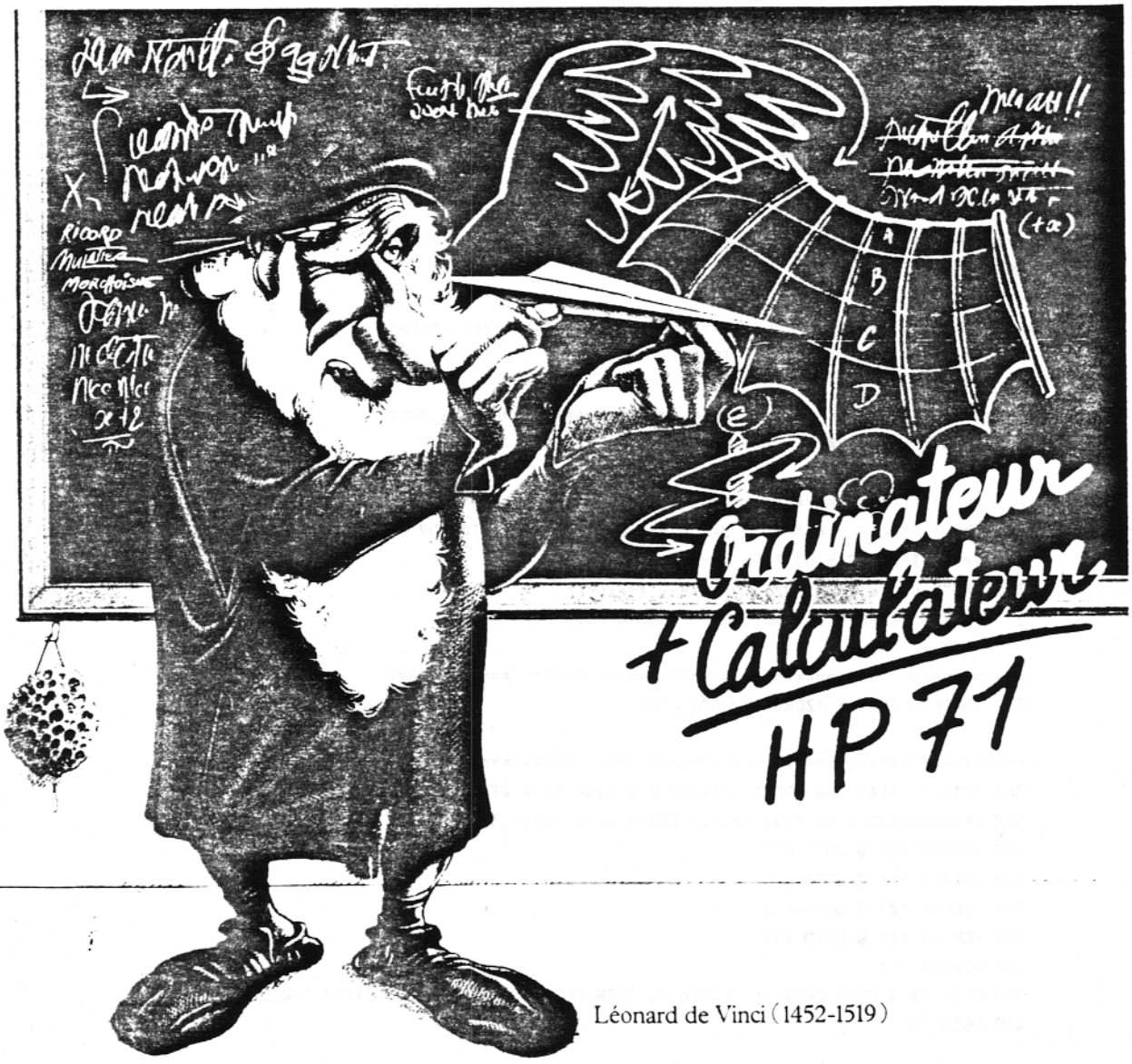
================================================================
330 'DROP': FORTHX 'ZDROP' @ GOTO 'D'

================================================================
340 'STO': SFLAG 6 @ 'RCL': WINDOW 3 @ DISP C$ @ WINDOW 7 @ USER OFF
350 K$=KEYWAIT$ @ ON POS('XYZTL',K$)+1 GOTO 'RCL',360,370,380,390,400
360 FORTHX 'X' @ GOTO 410
370 FORTHX 'Y' @ GOTO 410
380 FORTHX 'Z' @ GOTO 410
390 FORTHX 'T' @ GOTO 410
400 FORTHX 'L'
410 DISP K$ @ USER ON @ IF FLAG(6,0) THEN FORTHX 'ZSTO' ELSE FORTHX 'ZRCL'
420 GOTO 'D'
```

```
================================================================
 430 'MOD': A=SQR(X^2+Y^2) @ B=ANGLE(X,Y)
 440 WINDOW 3 @ DISP '|z|=';A @ WAIT 3 @ DISP '=';B;'Deg' @ WAIT 3
 450 GOTO 'DISP'

================================================================
 460 'PUT': FORTHX '" X" BASICF R<>I " Y" BASICF R<>I' @ RETURN

================================================================
 470 'GET': X=FORTHF @ FORTHX 'R<>I' @ Y=FORTHF @ FORTHX 'R<>I' @ RETURN
```




Léonard de Vinci (1452-1519)

## LE COIN DES LHEX

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

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

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

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

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

Vous trouverez, outre les Lex de la rubrique "assembleur", le Lex CHARLEX nécessaire à la rédaction de votre article. Voir l'article "Ah ! Vous écrivez !".

P. David (PC37 SIG1 CHHU616)

```
DIVILEX   FPRM      225082
          NPRM      225083
          PGCD      225084
          PHI       225085
          PPCM      225086
          PRIM      225087

BASICLEX  BLIST     225090
          PBLIST    225091
          RENUMREM  225092

CHARLEX
```

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

```
DIVILEX  ID#E1   641 octets

     0123456789ABCDEF sm

000: 44946594C4548502 36C
001: 802E005401910168 6BE
002: 705001E257500000 A03
003: F440000000000000 D31
004: 09D100FB00AD100F 0C3
005: 6108F000F120E500 42B
006: 0FA204D000F530CA 7BE
007: 200F7640525D4257 B29
008: E40525D435705743 E98
009: 4444550584945570 1F1
00A: 50534D4657052594 555
00B: D4751FF811741110 8CE
00C: 0102AF2809972747 C3F
00D: FD2100976C010976 FBC
00E: 1051374221199794 30D
00F: E730059D1121197B 686
010: 02119A7E73041020 9E1
011: 1AFA8F223B1AF686 D9D
012: 050A4E8D832F0888 133
013: 88888882A8525118 4AC
014: 8888888882A84277 833
015: 7010176705901087 B88
016: C60780094F2F54A1 F16
017: 1997E50AFE10310A 2AC
018: 76819798011153F1 615
019: 19862001137F6110 969
01A: 01127F40AFE7E511 CFD
01B: 189F2B0775310101 062
01C: 078D91FB0AC71CF1 41D
01D: 7F8FC1DB09485085 7C9
01E: 4321109B67D8F2EA B66
01F: 21A4F0121AF280CC EFF
020: 2001AF2302AF7310 272
021: 110C018812851590 5C7
022: 88228418447B9FAF 967
023: 2874607DBF10A94B D06
024: 90102768F10011A9 06A
025: FA5085479AF31607 3FF
026: AA097C60695FAF01 7AC
027: 03118B79B76864F2 B34
028: 96942AEE20305AEE EE2
029: B684F01187030B7B 26B
02A: B7B7620B7B5EEA6D 63A
02B: 480A6D5607D10A7B 9D2
02C: A7B7310A7B5EE108 D79
02D: 11011A6900108112 0BA
02E: 9F2E17BD09760187 454
02F: 13111BB7610B1180 7B9
030: 3113076D0E1098FB B41
031: 7CE00388127BAE94 EE2
032: B82AF2329999F632 281
033: E6109100739E1197 5F1
034: CC1110A7A10197C6 97E
035: 0607EAF8D03151AE D23
036: A2EA6C90D80BF154 0DC
037: F8F355C0AD47C6E1 49F
038: 0011876811118508 7E7
039: 5397280843100711 B32
03A: 0873A097650AF262 EB1
03B: 6D7E4EAEB7211B66 271
03C: 7B01AFBB667EF07A 63C
03D: F073F072F07BE07A 9E7
03E: E073E07CD07ED074 D94
03F: D073D072D07BC074 12A
040: C070C072C078B077 4B2
041: B076B07CA07BA074 853
042: A07D907F907E9077 BF1
043: 9076907F80758074 F64
044: 8073807C707E7077 2E0
045: 7070707270786074 635
046: 607360726078 5074 993
047: 5076507C407B407A D0D
048: 407340724072307A 060
049: 3080293C00114A12 3B6
04A: 110736E11927919E 729
04B: 09F2606F1F118494 AAE
04C: A1BA1BA1BA1BA1B1 E73
04D: 09110209FAA0BF20 1FB
04E: C55FB7A5CFA7A0D4 5D7
04F: 80BF65EE1192791C 97D
050: 0007119110840011 CAF
051: 028FCDCE0051128F 055
052: BBCE001          1F9

BASICLEX ID#E1   1093 octets

     0123456789ABCDEF sm

000: 2414359434C45485 364
001: 802E007401910168 6B8
002: F88001EA5C500000 A32
003: F920000000000000 D63
004: 0B6000DD00B4000D 0DD
005: C1075500D924C494 459
006: 3545A5B0524C4943 7D4
007: 545B5F2554E455D4 B6A
008: 2554D4C51FFD1800 F04
009: 2970031F1AB0B345 282
00A: 3160800B770031F0 5DE
00B: AB01F078F214D17A 981
00C: 15131ED55F147135 CFA
00D: 14A8F504504F1311 06C
00E: F962618FD2F90456 40C
00F: 8F36F9042617E143 793
010: F41CE8F147704647 B2C
011: 5248FBFB6048317F ED2
012: 1437A60DB1F688F2 273
013: 1457C521BB78F215 600
014: 631AEB6F15434856 992
015: A128DA93908766F8 D31
016: F73321C04AE831F1 0C7
017: B47A474F0978A031 457
018: C3645631A347F6EF 7FC
019: 1137134C2D716E8D B88
01A: E377078228437612 EF0
01B: 316C962808475818 25F
01C: 59867208575A0876 5D3
01D: 50858869A0877508 948
01E: 5884986750859316 CB4
01F: F966B08568585528 03D
020: 46319B9669185884 3B6
021: 317C14F1CC96A508 753
022: 531B698F21371441 AC2
023: 37311C9627086885 E2F
024: 8487A818F245818F 1C4
025: 1CD71D273F17581D 564
026: 231E3AE5DAA6D31D 925
027: 314C161A6D56F854 CB5
028: 8F51E718F1CD7177 062
029: 217B31844842316E 3CF
02A: 962E031CF9668085 75C
02B: 285479118FF6F407 AEC
02C: 1717A01791186406 E48
02D: 310214A962B014C1 1B0
02E: 6151F161AF027158 529
02F: 3872601501201838 870
030: 6520855522167132 BBC
031: 130187137EA81C8B F47
032: 89031D214C613084 2B0
033: 574808731231AD96 625
034: 28131AB7BA04D031 9AD
035: 2C70A05508587170 D18
036: 8F71B60135D015B3 09E
037: 11B8B6611378FDA6 43D
038: 701355606C1E8D84 7BF
039: A801611361351 11D B1A
03A: 8D08FDFFF0400870 ECD
03B: CD87A7D031F698F2 288
03C: 14713517514B1C50 5EA
03D: 11BEB6F215630B15 97A
03E: 43011BF85F214613 CEF
03F: 4135011371341631 02B
040: 37DD06D214ED7161 3C6
041: 14A96081136CB136 73D
042: 18114E16190EAD07 AC3
043: D5017CAF1F1C6F21 E78
044: 43EA81C3140B62AE 219
045: A1FEB6F215321188 5BD
046: FD3DE0D98FD3DE0D 982
047: 2B260E2692860313 D1B
048: 1AA2A6210B8F2458 0AD
049: 11F778F2147135D2 42E
04A: 14F113D517114F96 7AA
04B: E50B268B270EA58F B5A
04C: E01035E0C01038F1 ED3
04D: CD713F0202020202 235
04E: 0202021FB98F2155 5A8
04F: 734B98F2D7D23081 93E
050: 13EA421103DA8FAF CED
051: D716DDFCA8FAFD71 0E2
```

```
052: 11B8FC2DE0D58FC2 4AE
053: DE010B01849D2E61 842
054: 0933999910B84A14 BBA
055: A311F96600160142 F19
056: F4164101879A085A 297
057: D010314A9660084A 607
058: 16115A3103879001 956
059: 198B240018D91FB0 CDF
05A: 80300EE200136108 039
05B: 730275128F39B708 3AF
05C: 518528F357A14B11 72B
05D: F976F211A1457EF1 ACF
05E: 31E162F111013076 E2B
05F: A1102D2E6F210810 1A3
060: 9F2F2F210B707143 52D
061: 610076614951017C 886
062: 514F4D61321028F4 BFF
063: EFF086060625D137 F90
064: 12A1347431472052 2DF
065: 3B142004491D68F4 659
066: EFF0D015B3870501 9EE
067: 0372211371128A21 D38
068: 21301838F78870D2 0AC
069: 15F31108B6606CFE 447
06A: 1FC65F2147D711A1 7DF
06B: 35D215F313411806 B44
06C: 11A135111D81378B EB4
06D: FD2135D015B311B8 244
06E: BEC11108B2810413 5BE
06F: 6108D2E610954C6C 94D
070: 6015937CBC31CF96 CF0
071: 2C1316E962311100 04C
072: 7D605E6066?10071 3BB
073: 5D305E6061102399 726
074: E70C01000420173D A8B
075: 214F133CA13114B1 E00
076: 7190CCE636F07851 196
077: 8428F357A1611C14 513
078: A161311F96600D01 881
079: 5A3163031FC65F21 C05
07A: 47D71CE143D23113 F8A
07B: CA1310172108FEEB 328
07C: 60441832005C08F6 690
07D: 27705008D393901F A00
07E: D55F214713517E14 D7D
07F: 3F434412E08A6618 0FB
080: F73B908F78870147 48E
081: 540D21B8E7F215C3 828
082: 0370604228F5CC30 B9C
083: 5E0867A18D20F208 F2B
084: FDC6304908D27130 2B7
085: 7D008FDC6305DE85 66B
086: 98F6E84031F09660 9F8
087: 1311FAEA8D51D208 D9B
088: DF3E208FE7A20400 140
089: 58B7FEF41B8497FA 511
08A: F879B9038D93850  871

CHARLEX  ID#E1   624 octets

     0123456789ABCDEF sm

000: 34841425C4548502 35E
001: 802E009401910168 6B4
002: 5E4001E000000000 9F8
003: FE0000000800001F D52
004: F31BF9614000032BF 0E5
005: 38F14A11DB10AD23 47F
006: 07D532BFB8FD7911 832
007: 11AD754D7A101743 BB5
008: 11014D1CB15D0000 F20
009: 71450375FF864834 29D
00A: 5655581008355654 5F4
00B: 5810002455565870 941
00C: 0026555658700836 C93
00D: 555658100836 4545 FE9
00E: 4A30000A49724000 33C
00F: 0808094A2C180814 6A5
010: A464242008355455 9FF
011: 581000054C714000 D45
012: 0C3142404C700832 0A1
013: 41414A70002078A0 3F9
014: 2F30000000000000 724
015: 0000000000000000 A34
016: 0000000000000000 D44
017: 0000000000000000 054
018: 0000000000000000 364
019: 0000000000000000 674
01A: 0000000000000000 984
01B: 0000000000000000 C94
01C: 0000000000000000 FA4
01D: 0000000000000000 2B4
01E: 0000000000000000 5C4
01F: 0000000000000000 8D4
020: 0000000000000000 BE4
021: 000000000000080C F0F
022: 1A28080008080A2C 279
023: 180008040E340800 5C2
```


```
023: 180008040E340800 5C2
```


```
024: 08001E3018000000 8FC
025: 0000000000000000 C0C
026: 0000000000000000 F1C
027: 0000000000000000 22C
028: 0201000000010200 542
029: 0000000201020000 857
02A: 0001000100000002 B6B
02B: 0102010000000000 E7F
02C: 0000000000000000 18F
02D: 045E755142400101 4DB
02E: 0101010000000000 7EE
02F: 0000000000000000 AFE
030: 0000070507000000 E21
031: 000000000083444C4 15F
032: 44400D7901112D70 4BF
033: 050D750509700000 809
034: 0D70000000384540 B4C
035: 4020014E322E3140 EA0
036: 084E794142400000 1F0
037: 00000000002E4559 52E
038: 3200000000000000 843
039: 0000000000000026 B5B
03A: 5556587008365556 EBA
03B: 5810083645464830 20B
03C: 0832414248700024 54C
03D: 5655587008345655 8A9
03E: 5810083446454830 BF8
03F: 0C3042414C700024 F4D
040: 5556587008355654 2AA
041: 5810083546444830 5F9
042: 0C3142404C700025 94F
043: 5455587008355455 CA9
044: 5810083544454830 FF7
045: 0C3140414C700875 359
046: 14141870000A4972 6AA
047: 40000E3159454E30 A0A
048: 0C7A0F7949400024 D82
049: 5554587000084A71 0DE
04A: 40000C523A262D10 43F
04B: 0424587458400875 796
04C: 1415187000094A70 AE6
04D: 4000083544454830 E2A
04E: 0C3140414C300C74 192
04F: 5655545000054C71 4E9
050: 40000            5E2
```

# PPC PARIS CHAPTER

Octobre 1986

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

## BULLETIN D'ADHESION

NOM |_|_|_|

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

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

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