# LE DEMANDEZ PROGRAMME

**Network, le processeur d'idées. Rythm'Oric. Par J.-P. Verpeaux notre gagnant du mois**

Ne cachez plus vos talents... Envoyez-nous un programme inédit que vous avez écrit et peut-être recevrez-vous une bourse de 1 000 F. Chaque mois, nous publions le programme de l'un de nos lecteurs dans notre cahier des programmes. Vous devez nous faire parvenir un listing complet du programme, une brève description de ses fonctionnalités, votre photographie et, bien sûr, une disquette ou une cassette. Envoyez-nous le tout à SVM, 5, rue de la Baume, 75008 Paris. Les programmes non primés vous seront retournés. A bientôt...

## PROCESSEUR D'IDÉES

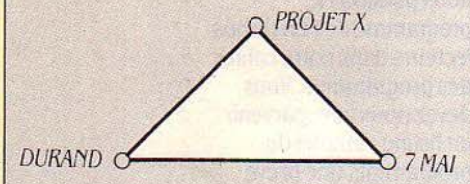
Ne manquez surtout pas le programme Network qui suit. Il est : a) utile à la plupart d'entre nous la plupart du temps ; b) introuvable dans le commerce ; c) adaptable à la quasi-totalité des micro-ordinateurs ; d) ridiculement simple, grâce à une astuce de programmation aussi géniale qu'inattendue. Côté application, c'est un processeur d'idées qui décuple l'efficacité de n'importe quel bloc-notes. Côté programmation, c'est un noyau de base de données relationnelles... en cent lignes de Basic. Qui dit mieux.

MAIS QUI DONC, BONSOIR !, M'A parlé du projet X ? « *Quel est ce truc important que m'a confié le patron hier ?* » « *Quel soir ai-je prévu pour conclure avec Lucienne ?* » Enervant, de tout oublier, non ? Autant que d'extirper la réponse du fouillis de votre agenda surchargé. Alors, vous vous prenez à rêver d'un bloc-notes intelligent, qui vous fournirait illico la pièce manquante du puzzle.

Ne rêvez plus : Network est là. Bien plus qu'un simple carnet de notes électronique, il jette des ponts entre tous les indices de vos énigmes quotidiennes, à la manière d'une base de données relationnelle. Pour Network, une personne, une date, un dossier, une chose à faire sont des « objets ». Le programme constitue une liste des objets que vous lui fournissez et des rapports qu'ils entretiennent entre eux : si le 7 mai, vous avez rendez-vous avec Durand à propos du projet X, vous donnez à Network la liste d'objets « 7 mai », « Durand », « Projet X », et celui-ci référencera ces trois objets dans sa liste, tout en créant

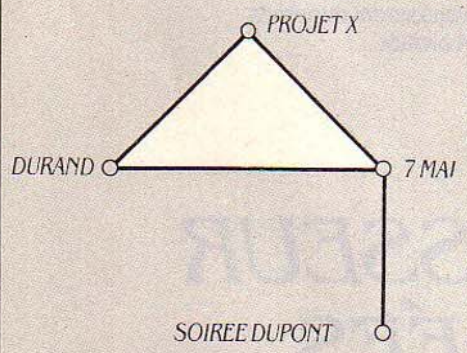
les liens entre eux. Si vous appelez alors « Durand », Network vous dira qu'il est en rapport avec « 7 mai » et « Projet X » ; si, au contraire, vous voulez savoir avec qui et quand vous avez rendez-vous pour le projet X, appelez « Projet X » et vous aurez « Durand » et « 7 mai ». Si, par ailleurs, le 7 mai vous êtes invité à une soirée chez les Dupont, vous pouvez rentrer la liste « 7 mai » et « Soirée Dupont ». Si vous appelez le « 7 mai », vous aurez toujours « Durand » et « Projet X » pour vous rappeler votre rendez-vous, mais aussi « Soirée Dupont » que vous venez de rajouter.

## CONSTRUCTION D'UN RÉSEAU DE LIENS PAR « NETWORK »




*Etape 1*

*Le « 7 mai » vous avez rendez-vous avec « Durand », à propos du « Projet X ». Les trois objets sont en rapport les uns avec les autres. L'appel de l'un d'eux vous donne les deux autres.*




*Etape 2*

*Le « 7 mai », vous avez également la « Soirée Dupont ». Vous créez un nouveau lien pour le « 7 mai » qui est maintenant relié avec Projet X, Durand et Soirée Dupont.*
*« Soirée Dupont » n'est relié qu'à « 7 mai » et les liens de « Projet X » et « Durand » ne sont pas modifiés.*

Si vous voulez savoir quel jour tombe la soirée Dupont, appelez « Soirée Dupont » et vous aurez « 7 mai » (mais pas « Durand » ou « Projet X » qui n'ont rien à voir avec les Dupont). Ouf. Vous aurez compris (j'espère) que la principale originalité de Network est de s'intéresser plus aux liens entre les objets qu'aux objets eux-mêmes.

Une idée simple pour stocker la liste des liens entre les objets consiste à faire un grand tableau avec une ligne et une colonne par objet. Le lien se trouve donc à l'intersection de la colonne et de la ligne. Ceci marche en théorie mais se traduit par un énorme gâchis

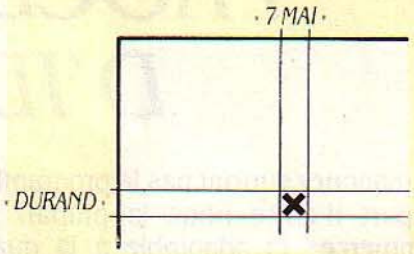
# NETWORK

Basic standard

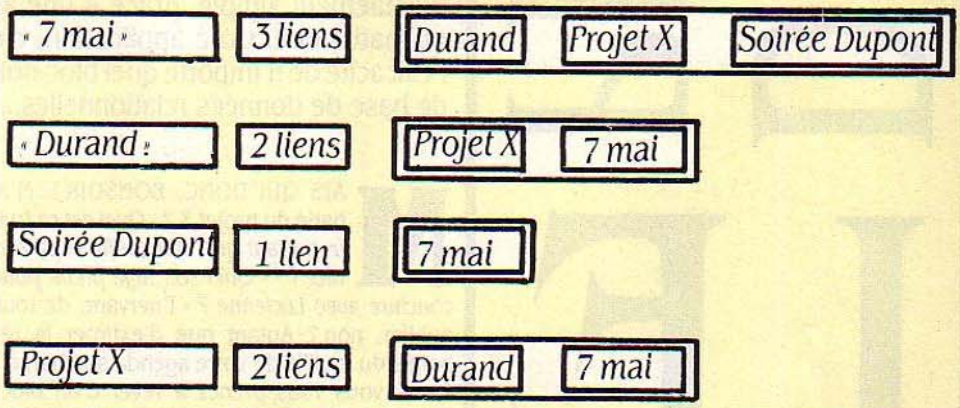
```
10    DIM A$(255),B$(255)
20    HOME : HTAB 15: VTAB 8
30    PRINT "NETWORK"
40    HTAB 15: PRINT "======="
50    HTAB 5: VTAB 12: PRINT "VOULEZ-VOUS :"
60    PRINT :
      PRINT " 1-CONSULTER LA LISTE DES OBJETS"
70    PRINT " 2-CONSULTER LES LIENS D'UN OBJET"
80    PRINT " 3-CREER DE NOUVEAUX LIENS ET/OU OBJETS"
85    PRINT " 4-DETRUIRE UN OBJET"
90    PRINT " 5-SAUVER LE BLOC-NOTE"
100   PRINT " 6-CHARGER UN BLOC-NOTE"
110   GET R$:R =  VAL (R$):
      ON R GOTO 1000,2000,3000,4000,5000,6000
120   GOTO 20

1000  HOME :NO = 0
1005  K = 0: FOR I = 0 TO 255: IF A$(I) = "" THEN 1100
1010  K = K + 1: IF K = 23 THEN  GET R$:K = 0
1020  PRINT A$(I):NO = NO + 1
1100  NEXT I
1110  PRINT : PRINT "Nombre d'objets : ";NO
1120  GET R$: GOTO 20

2000  HOME : HTAB 15: VTAB 4: PRINT "Liste des liens"
2010  HTAB 15: PRINT "==============="
2020  PRINT : INPUT "  - de quel objet --> ";N$
2025  IF N$ = "" THEN 20
2030  N$ =  LEFT$ (N$ + "          ",8)
```

●●●

## COMMENT GÉRER LES LIENS




*Une méthode vorace de mémoire : le tableau N x N pour N objets.*




*La méthode employée dans Network : pour chaque objet, on donne le nombre de liens puis la liste de ceux-ci. La longueur de cette liste est variable et précisément ajustée au nombre de liens, ainsi, aucune place mémoire n'est perdue.*

de place mémoire qui limite très vite le nombre d'objets : pour 200 objets, il faudrait un tableau d'entiers de 200 x 200 soit 40 000 cases. Près de 80 Ko de mémoire rien que pour les liens ! D'autant plus ridicule que dans la majorité des cas, presque toutes les cases du tableau sont vides : lorsqu'un objet n'a qu'un ou deux liens, seules une ou deux cases du tableau par colonne sont effectivement utilisées. Pour pallier ceci, il faut, pour chaque objet, ne stocker que la liste de ceux avec lesquels il est effectivement relié. Comme le nombre de liens d'un objet est a priori variable, cela revient à dire que chaque objet peut avoir une liste de longueur différente. Impossible de faire un tableau, puisque toutes les lignes n'auraient pas la même longueur. Il faut donc faire de la gestion dynamique de mémoire. Ça tombe très bien : ce bon vieil interpréteur Basic fait ça à la perfection.

## *Le Basic au boulot*

Si vous avez un tant soit peu programmé par vous-même, comme M. Jourdain, vous avez utilisé la gestion dynamique de mémoire sans le savoir. Sa particularité ? Une chaine de 90 caractères de long y occupe 90 cases mémoire, tandis qu'une chaine de 25 caractères n'en occupe que 25. La façon dont Basic gère cela n'est pas notre propos, mais nous allons utiliser cette particularité pour avoir des listes de liens de longueur variable, comme les chaines de caractères. Nous allons transcoder en caractères (avec la fonction CHR$ () du Basic) les index des liens et les stocker dans une chaine. Ainsi, si l'objet n° 1 est relié à l'objet n° 4, on placera le caractère CHR$ (4) (pour objet n° 4) dans la chaine de caractères correspondant à la liste des liens de l'objet n° 1. L'opération inverse de CHR$ (), ASC (), permet de récupérer le numéro de l'objet à partir de cette chaine. Grâce à cet artifice, il est possible de gérer sans problème un bloc-notes de 256 objets avec seulement 32 Ko de mémoire, alors que par une méthode matricielle, il faudrait plus de 128 Ko !

Network utilise deux tableaux de chaines de caractères A$ () et B$ () qui contiennent, l'un les intitulés des objets (« 7 mai », « Durand »...) et l'autre les listes des liens correspondant à chaque objet. Un objet est une chaine de 255 caractères au plus. Seuls les huit premiers caractères sont significatifs pour la recherche, il faut donc faire attention à ce que les noms de deux objets distincts ne

## *CONSEILS D'ADAPTATION*

Instructions spécifiques à l'Apple II :
HOME efface l'écran, peut être omis ou remplacé par l'instruction correspondante sur votre machine.
HTAB X : VTAB Y permettent de positionner le curseur à la position X.Y. Ces instructions affectent uniquement la présentation et peuvent être omises.
GET R$ permet de lire une seule touche tapée au clavier et peut être remplacé par INPUT R$.

●●●

```
2035  I1 =  - 1
2040  FOR I = 0 TO 255
2050  IF A$(I) = "" THEN 2200
2060  IF N$ < > LEFT$ (A$(I) + "          ",8) THEN 2200
2070  I1 = I:I = 255
2200  NEXT I
2210  IF I1 =  - 1 THEN
        PRINT :
        PRINT " Objet inconnu ":
        GET R$: GOTO 2000
2220  IF B$(I1) = "" THEN
        PRINT :
        PRINT A$(I1): PRINT : PRINT " AUCUN LIEN ":
        GET R$: GOTO 2000
2225  PRINT : PRINT "Liens avec :": PRINT
2230  FOR I = 1 TO  LEN (B$(I1))
2240  PRINT " - ";A$( ASC ( MID$ (B$(I1),I,1)))
2250  NEXT I
2260  GET R$: GOTO 2000

3000  HOME : HTAB 15: VTAB 4: PRINT "Création de liens"
3010  HTAB 15: PRINT "================="
3020  PRINT :
      PRINT "Tapez le nom de tous les objets reliés":
      PRINT "(Return = fin)"
3030  I2 = 0: PRINT
3040  INPUT "- ";L$
3050  IF L$ = "" THEN 3300
3055  I1 =  - 1:L1$ = L$:L$ =  LEFT$ (L$ + "          ",8)
3060  FOR I = 0 TO 255: IF A$(I) = "" THEN 3080
3065  IF L$ = LEFT$ (A$(I) + "          ",8) THEN I1 = I:I = 255
3080  NEXT I
3090  IF I1 <  > - 1 THEN 3200
3092  PRINT "OBJET INCONNU, VOULEZ-VOUS LE CREER O/N":
      GET R$: IF R$ < > "O" THEN 3040
3100  FOR I = 0 TO 255
3110  IF A$(I) = "" THEN I1=I:I = 255
3120  NEXT I
3130  IF I1 =  - 1 THEN  PRINT : PRINT " PLUS DE PLACE":
      GOTO 3040
3140  A$(I1) = L1$
3200  M(I2) = I1:I2 = I2 + 1: GOTO 3040
3300  IF I2 = 0 THEN GOTO 20
3310  FOR I = 0 TO I2 - 1
3320  FOR J = 0 TO I2 - 1
3330  IF M(I) = M(J) THEN 3380
3335  IF B$(M(I)) = "" THEN 3370
3340  FOR K = 1 TO  LEN (B$(M(I)))
3350  IF  ASC ( MID$ (B$(M(I)),K,1)) = M(J) THEN 3380
3360  NEXT K
3370  B$(M(I)) = B$(M(I)) +  CHR$ (M(J))
3380  NEXT J: NEXT I: GOTO 3000

4000  HOME : HTAB 15: VTAB 4: PRINT "DESTRUCTION D'OBJET"
4010  HTAB 15: PRINT "================="
4020  PRINT : PRINT "TAPEZ LE NOM DE L'OBJET A DETRUIRE"
4025  PRINT
4030  INPUT "- ";L$
```

*Suite page 86*

commencent pas par les mêmes huit lettres. Un menu général (lignes 10 à 120) permet d'accéder aux diverses fonctions du programme. La fonction clé du programme est la création de liens (fonction 3 du menu, lignes 3000 à 3380) qui permet de définir les liens entre les objets, groupe par groupe, et éventuellement de créer de nouveaux objets au passage. Les fonctions 5 et 6 (lignes 5000-6110) sont spécifiques à l'Apple II, elles permettent de sauvegarder et de recharger les tableaux A$ () et B$ () pour archiver le bloc-notes. Elles peuvent être omises sans affecter le fonctionnement du reste du programme.

Frédéric NEUVILLE

# A VOUS DE PLANCHER

SVM vous met au pied du mur. Vous trouverez ici, chaque mois, un micro-problème de programmation à résoudre en Basic. Ce n'est pas un concours : même si vous nous envoyez votre solution, vous ne recevrez ni abonnement, ni Macintosh. Mais une saine stimulation du cortex. La solution au prochain numéro.

AUJOURD'HUI, LA MAGIE DES grands nombres. Comment calculer le produit, la somme ou le quotient exact de deux nombres de 50 chiffres que votre micro-ordinateur refuse de digérer ? En effet, sur la plupart des appareils, la précision et la taille des nombres manipulables sont restreintes : les entiers sont le plus souvent limités à 65535 et les réels à 1037 ou 1099. Si votre Basic accepte la double précision, le nombre de chiffres significatifs variera de la demi-douzaine à deux douzaines au plus. Les amateurs de Pascal sont un peu mieux lotis, puisqu'il existe des variables du type « entier long » qui permettent de « caser » de grands entiers. Mais eux aussi sont souvent limités à quelques dizaines de chiffres significatifs. Paradoxal, de la part d'une machine réputée plus douée que l'homme pour les calculs...

Comment dès lors, réaliser un programme de codage inviolable, calculer pi avec 300 décimales, trouver le nombre exact des différentes anagrammes d'une phrase de quatre lignes ou déterminer si 237-1 est un nombre premier ? Ces quatre applications partagent en effet la même difficulté : manipuler de très grands nombres ou – ce qui revient au même – des nombres parfois pourvus de plusieurs centaines de chiffres significatifs. Alors ? le mois prochain, nos méthodes pour résoudre ces quatre problèmes ainsi que la réponse à la première question.

*Suite de la page 85*

```
4035  IF L$ = "" THEN 20
4040  L$ =  LEFT$ (L$ + "          ",8):I1 =  - 1
4050  FOR I = 0 TO 255:
       IF L$ =  LEFT$ (A$(I) + "          ",8) THEN
          I1 = I:I = 255
4055  NEXT I
4060  IF I1 =  - 1 THEN
       PRINT : PRINT " OBJET INCONNU":
       GET R$: GOTO 4000
4070  PRINT : PRINT " DETRUIRE ";A$(I1);" ? (O/N)":
      GET R$: IF R$ <  > "O" THEN 4000
4080  A$(I1) = "":B$(I1) = ""
4090  FOR I = 0 TO 255: IF B$(I) = "" THEN 4200
4100  FOR J = 1 TO  LEN (B$(I))
4110  IF I1 <  > ASC ( MID$ (B$(I),J,1)) THEN 4150
4120  B$(I) =  MID$ (B$(I),1,J - 1) +  MID$ (B$(I),J + 1)
4130  GOTO 4200
4150  NEXT J
4200  NEXT I
4210  GOTO 4000

5000  HOME : HTAB 15: VTAB 4:
      PRINT "SAUVEGARDE DU BLOC-NOTE"
5010  HTAB 15: PRINT "======================"
5020  PRINT : PRINT "TAPEZ LE NOM DU FICHIER A SAUVER"
5025  PRINT
5030  INPUT "- ";L$
5040  PRINT : PRINT CHR$(4);"OPEN";L$
5050  PRINT CHR$(4);"WRITE";L$
5060  FOR I = 0 TO 255
5070  PRINT A$(I):
      PRINT  LEN (B$(I)):
      IF B$(I) <  > "" THEN
       FOR J = 1 TO  LEN (B$(I)):
        PRINT  ASC ( MID$ (B$(I),J,1)):
       NEXT J
5072  NEXT I
5080  PRINT CHR$(4);"CLOSE ";L$
5090  GOTO 20

6000  HOME :PRINT : PRINT CHR$(4);"CATALOG":
      PRINT : PRINT "QUEL FICHIER VOULEZ-VOUS CHARGER ?"
6010  PRINT : INPUT " - ";L$
6020  PRINT CHR$(4);"OPEN";L$
6030  PRINT CHR$(4);"READ";L$
6040  FOR I = 0 TO 255
6050  INPUT A$(I): INPUT N:B$(I) = "":
      IF N <  > 0 THEN
       FOR J = 1 TO N:
        INPUT I1:B$(I) = B$(I) +  CHR$ (I1):
       NEXT J
6060  NEXT I
6100  PRINT CHR$(4);"CLOSE";L$
6110  PRINT : GOTO 20
```