SCIENCE & VIE

MENSUEL
N° 831 DÉCEMBRE 1986

LA TÉLÉVISION A 50 ANS

NEUROPHYSIOLOGIE DE LA VIOLENCE

16 F
N° 831
SUISSE 5 FS
CANADA $2,95
BELGIQUE 110 FB
ESPAGNE 450 Ptas
MAROC 17 Dh
TUNISIE 2,09 DT

ISSN 0036 8369

Gérez vos stocks

LE MICRO DE L'INGÉNIEUR

La gestion des stocks dispose d'un vaste choix de méthodes, chacune devant être adaptée au cas particulier de l'entreprise concernée. Mais, si nous ne surprendrons personne en disant que ces méthodes peuvent être plus ou moins complexes, nous rassurerons les lecteurs en présentant ici une des plus simples qui soient. Nous allons, pour ce faire, recourir à la programmation dynamique stochastique, rien que ça ! En fait, stochastique signifie *grosso modo* "qui tient compte du hasard" ; et la programmation dynamique s'occupe de décomposer un problème en plusieurs phases (***figure 1***), l'état du système à résoudre évoluant au cours de chacune des phases d'un état E_n à un état E_{n+1} après prise d'une décision X_n ; la question étant de trouver à chaque phase la décision X_n la plus appropriée. L'exemple qui suivra la formulation permettra de mieux comprendre cette approche, mais nous pouvons en un premier temps donner un petit exemple de programmation dynamique tout court avec la recherche du plus court chemin entre les points A et I de la ***figure 2*** .

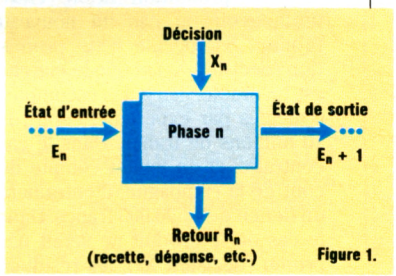


Figure 1.

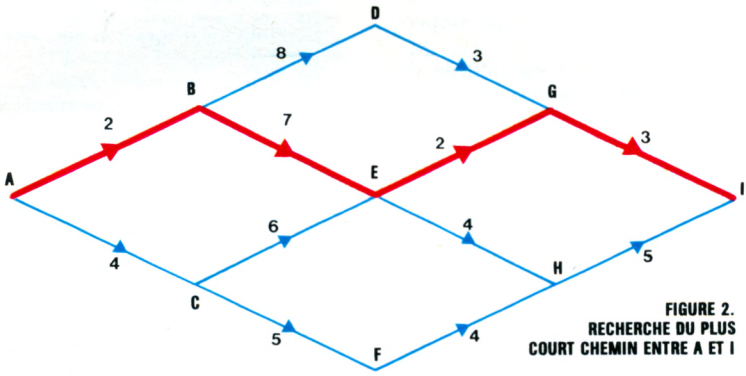


FIGURE 2.
RECHERCHE DU PLUS COURT CHEMIN ENTRE A ET I

Pour arriver en I, on peut venir de G ou de H, la première distance valant 3, la deuxième 5. Jusque-là, La Palisse n'aurait rien à redire. Dans la phase précédente, on peut partir de D, de E ou de F. Venant de D, le point de passage obligé est G, la distance totale valant alors 3 + 3 = 6. Pareillement, le chemin de F à I passe par H, et la longueur est de 9. Mais, à partir de E, deux solutions sont possibles : passer par G (distance = 2 + 3 = 5) ou par H (distance = 4 + 5 = 9). La minimisation de la distance totale étant notre but premier, le tronçon optimal partant de E est constitué par EGI, de longueur 5. Poursuivant avec la même méthode, nous aboutirons finalement au chemin optimal représenté en trait plus gras : ABEGI.

La recherche du plus court chemin entre E et I pouvait être considérée comme un problème partiel, ou intermédiaire. C'est que, précisément, la programmation dynamique consiste à rechercher les solutions optimales de tous les problèmes intermédiaires par lesquels dont absolument passer la solution finale. Ceci montre comment, même si la programmation dynamique reste un procédé énumératif, il ne passe pas par la recherche exhaustive de toutes les solutions possibles, d'où un gain de temps important.

Reste la programmation dynamique stochastique. Dans beau-

coup de problèmes, on rencontre des phases où chaque décision est suivie de phénomènes hasardeux. Un exemple simple est indiqué dans la ***figure 3***, dans lequel un chef de promotion hésite entre promouvoir ou pas un nouveau produit. S'il décide cette promotion (décision X_1) et que le marché est favorable, des gains escomptés (calculés par expérience) sont de 10 millions de francs. Dans le cas contraire, et en tenant compte du coût de la campagne, il entraînera un déficit de 1 500 000 francs. Dans le cas où le produit ne serait pas promu (décision X_2), les résultats respectifs seront vraisemblablement égaux à 3 et 1 million respectivement.

A ce stade, il est difficile de choisir entre les décisions X_1 et X_2 ; mais si, de plus, nous supposons que le marché a 30 % de chances d'être favorable (et donc 70 % de ne pas l'être), nous nous servirons de la notion d'espérance mathématique pour établir un critère. En effet l'espérance de gain associée à la décision X_1 vaut : $10 \times 0{,}3 - 1{,}5 \times 0{,}7 = 1{,}95$ million. A la décision X_2 est associée l'espérance : $3 \times 0{,}3 + 1 \times 0{,}7 = 1{,}6$ million.

Cette dernière est la plus faible, donc la première décision est optimale. Il est vrai que d'autres critères que l'espérance mathématique existent, mais c'est celui que nous allons garder pour traiter notre problème : la gestion de stocks, dont nous énonçons ci-après le libellé.

Formulation

Il s'agit de gérer au mieux le stock d'un produit, sur un nombre n de périodes, ou phases. Au début de chaque phase, on peut acheter "a" unités de ce produit, a étant compris entre ai et ax, au coût c (n), fonction de n. Ensuite, pendant cette même phase, l'ensemble des clients peuvent demander v unités, v compris entre vi et vx, au prix pr(n). A chaque v est associée une probabilité p (v), sachant que p (vi) + p (vi + 1) +... + p (vx) = 1.

Partant d'un stock nul, il s'agit de gérer au mieux le stock, donc déjà de décider l'achat en première phase, en maximisant le revenu sur ces n périodes. Nous pouvons considérer que les unités stockées non vendues à l'issue de la phase n seront soldées au prix unitaire SOL. Le principe consiste à partir des divers états finals possibles en se plaçant d'emblée en début de phase n, et de remonter ainsi jusqu'au point de départ en cumulant les revenus perçus en chemin.

1° Entrée des données : vi, vx, ai, ax, n, SOL, et les divers pr(i), c(i), p (i), pr (i) dépendant de la phase où

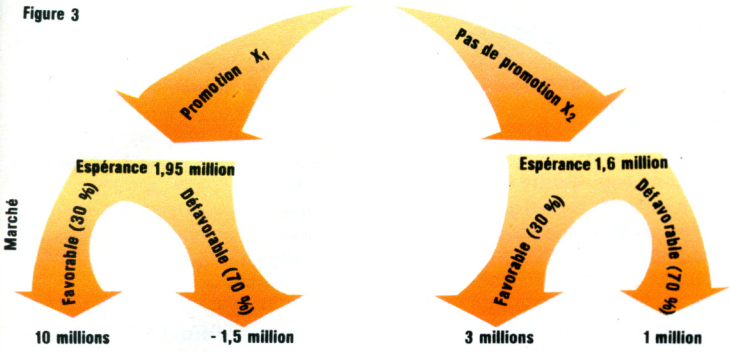
```
10  REM  PROGRAMME DE GESTION ELEMENTAIRE DE STOCKS
20  REM
30  HOME : VTAB 5: PRINT "*** GESTION SIMPLIFIEE DU STOCK D'UN PRODUIT S
    UR UN HORIZON PREDETERMINE AVEC DEMANDE ALEATOIRE DU PRODUIT **"
40  PRINT : INPUT "SUR COMBIEN DE PERIODES S'ETEND VOTRE RECHERCHE? ";N:
     REM  POINT N° 1 DE LA FORMULATION
50  PRINT : PRINT "INTRODUISEZ LES LIMITES POUR VOS ACHATS:": PRINT
60  INPUT "NOMBRE D'UNITES MINIMAL= ? ";AI
70  INPUT "NOMBRE D'UNITES MAXIMAL= ? ";AX
80  PRINT : PRINT "INDIQUEZ LES VARIATIONS POSSIBLES DE LA CLIENTELE:": PRINT

90  INPUT "NOMBRE D'UNITES MINIMAL= ? ";VI
100  INPUT "NOMBRE D'UNITES MAXIMAL= ? ";VX: PRINT
110 NF = (N - 1) * (AX - VI)
120  DIM C(N),P(N),PR(N,VX),ETA(N,NF),DECIS(N,NF)
130  FOR M = 1 TO N
140  PRINT "INTRODUISEZ LES PARAMETRES DE LA PERIODE n° ";M: PRINT
150  INPUT "PRIX D'ACHAT= ? ";C(M)
160  PRINT : INPUT "PRIX DE VENTE= ? ";P(M)
170  PRINT :S = 0: PRINT "PROBABILITE D'ACHAT :": PRINT
180  FOR K = VI TO VX
190  PRINT "DE ";K;: INPUT " UNITES= ? ";PR(M,K):S = S + PR(M,K): PRINT

200  NEXT K
210  FOR K = VI TO VX
220 PR(M,K) = PR(M,K) / S: REM  POUR AVOIR UNE SOMME DE PROBABILITES EGA
    LE A 1
230  NEXT K
240  NEXT M
250  INPUT "PRIX SOLDé A LA FIN=  ? ";SOL: PRINT : PRINT "     RECHERCHE
     EN COURS"
260 M = N: REM  POINT n° 2
270  FOR K = 0 TO NF
280 ETA(M,K) = 0: REM  POINT n° 3
290  FOR J = AI TO AX: REM  POINT n° 4
300 MM = K + J:RECETTE =  - J * C(M): REM  POINT n° 5
310  FOR L = VI TO VX: REM  POINT n° 6
320 MX = L: IF MM < MX THEN MX = MM: REM  POINT n° 7
330 RECETTE = RECETTE + PR(M,L) * (P(M) * MX + (MM - L) * SOL): REM  POI
    NT n° 8
340  NEXT L: REM  POINT n° 9
350  IF RECETTE > ETA(M,K) THEN ETA(M,K) = RECETTE:DECIS(M,K) = J: REM
    POINT n° 10
360  NEXT J: REM  POINT n° 11
370  NEXT K: REM  POINT n° 12
380 M = M - 1: IF M = 0 THEN  GOTO 520: REM  POINT n° 13
390 NF = (M - 1) * (AX - VI)
400  FOR K = 0 TO NF
410 ETA(M,K) = 0
420  FOR J = AI TO AX
430 MM = K + J:RECETTE =  - J * C(M): REM  POINT n° 15
440  FOR L = VI TO VX
450 MX = L: IF MM < L THEN MX = MM: REM  POINT n° 16
460 RECETTE = RECETTE + PR(M,L) * (P(M) * MX + ETA(M + 1,MM - MX))
470  NEXT L: REM  POINT n° 17
480  IF RECETTE > ETA(M,K) THEN ETA(M,K) = RECETTE:DECIS(M,K) = J: REM
    POINT n° 18
490  NEXT J: REM  POINT n° 19
500  NEXT K: REM  POINT n° 20
510  GOTO 380: REM  POINT n° 21
520  HOME : PRINT "LA RECHERCHE EST TERMINEE": PRINT
530  PRINT "JE VOUS RECOMMANDE L'ACHAT DE ";DECIS(1,0);" UNITES  DE PROD
    UIT AU DEBUT, DE FAçON A POUVOIR ESPERER UN GAIN OPTIMAL DE ";ETA(1,
    0);" FRANCS"
540  PRINT : INPUT "VOULEZ-VOUS CONNAITRE LES DECISIONS A PRENDRE EN COU
    RS DE ROUTE ? (O/N) ";R$
550  IF R$ = "N" THEN  GOTO 620
560  PRINT : INPUT "QUEL EST LE NUMERO DE PHASE QUI VOUS INTERESSE ? ";M

570 NF = (M - 1) * (AX - VI)
580  FOR K = 0 TO NF
590  PRINT : PRINT "SI VOTRE STOCK COMPORTE ";K;" UNITES, RACHETEZ-EN ";
    DECIS(M,K);" DE FAçON A VISER UN REVENNU GLOBAL DE ";ETA(M,K);" FRAN
    CS"
600  NEXT K
610  GOTO 540
620  END
```

Figure 3



l'on se trouve.

2° m = n.
Nf = (m − 1) × (ax − vi) = nombre maximal d'unités en stock au début de la phase m, en supposant que l'on a acheté le maximum et vendu le minimum. Poser k = 0.

3° Constituer la matrice ETA et poser ETA (m, k) = 0.

4° j = ai (achat du nombre minimum).

5° MM = k + j = nombre d'unités disponibles à ce moment. Recette = − j × c(m), égale pour l'instant à la dépense liée à l'achat.

6° i = vi : on commence par le cas le plus défavorable

7° mx = max (i,MM) = nombre d'unités maximum que peut demander l'ensemble des clients.

8° Recette = Recette + p (i) ×(pr (m) × mx + (MM − i) × SOL).

ETA (m,k) = Recette et noter la décision optimale : DECIS (m,k) = j.

11° Faire j = j + 1 (on incrémente le stock disponible) et aller en 5° tant que j ⩽ ax.

12° Faire k = k + 1 (stock résiduel en début de phase n augmenté d'une unité) et aller en 3° tant que k ⩽ Nf.

13° faire m = m − 1 ; si m = − 1, finir en 22°. Autrement, poser k = 0 et Nf = (m − 1) × (ax − vi).

14° ETA (m,k) = 0 ; j = ai.

15° MM = k + j ; Recette = −j × c (m) ; i = vi.

16° mx = max (i, MM) ; Recette = Recette + p (i) × (pr (m) × mx + ETA (m + 1, MM − mx)) : on cumule le gain de la vente en phase m avec la recette espérée à la phase m + 1.

TABLEAU 1 : Exemple de travail

Phase	Paramètre	1	2	3
Coût à l'achat	©	10	10	30
Prix de vente	pr	30	30	50
Probabilité de vente d'une unité	$p(^1)$	0,6	0,6	0,5
Probabilité de vente de deux unités	$p(^2)$	0,4	0,4	0,5
σ = 0	σx = 2	vi = 1		vx = 2

9° Faire i = i + 1 (on considère les clients qui achètent une unité de plus) et aller en 7°. Recommencer tant que i ⩽ vx.

10° Si Recette > ETA (m,k), faire

17° Faire i = i + 1 et aller en 16°. Recommencer tant que i ⩽ vx.

18° Si Recette > ETA (m, k), faire ETA (m,k) = Recette et DECIS (m,k) = j.

19° Faire j = j + 1 et aller en 15° ; recommencer tant que j ⩽ ax.

20° k = k + 1 ; aller en 14° tant que k ⩽ Nf.
21° Aller en 13°.
22° La recherche est terminée. ETA (0,0) indique le profit final maximal que l'on peut espérer. DECIS (0,0) indique le nombre d'unités à acheter en début de phase 1. A la fin de n'importe quelle phase m (0 ⩽ m ⩽ n − 1), si l'on se trouve avec k unités en stock, il est recommandé d'acheter le nombre DECIS (m + 1, k) d'unités pour la phase m + 1.

Exemple

Le ***tableau 1*** ci-contre montre les divers paramètres de travail pour cet exemple qui ne comporte que trois phases, et relativement peu de cas, puisque 0 ⩽ a ⩽ 2 et 1 ⩽ v ⩽ 2.

Sans développer en détail les calculs, indiquons que la stratégie optimale recommande l'achat de deux unités au tout début et de

TABLEAU 2 : Achat recommandé en fonction du stock disponible au début de la phase considérée

Phase \ Stock	0	1	2	3
1	2	—	—	—
2	2	2	—	—
3	2	1	0	—

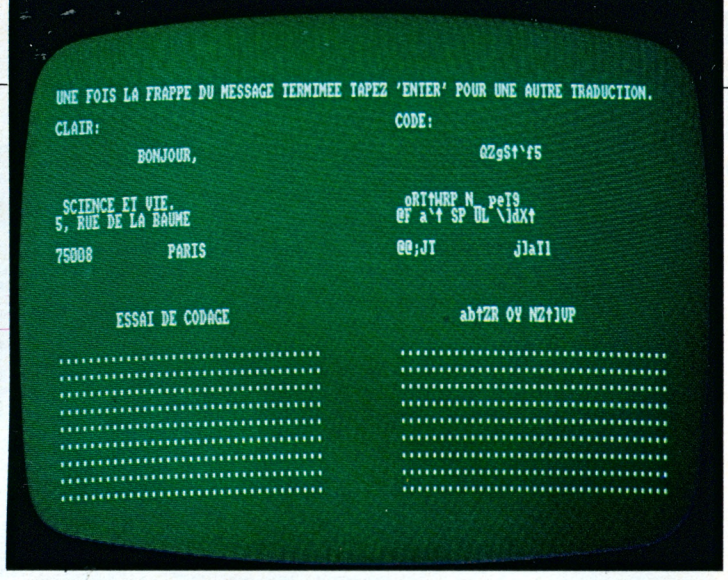
gérer le stock de façon résumée au ***tableau 2***.

Commentaires sur le programme pour Apple IIc

Comme d'habitude, le programme suit fidèlement les points de la formulation, et les variables reçoivent les mêmes appellations, excepté la lettre i, remplacée par l pour éviter des confusions avec le chiffre 1, et les probabilités p (i), qui se notent en réalité p (i, m), car elles dépendent aussi de la phase. Rajoutons que toute modification des données est facilitée avant de recommencer un calcul. Les participations aux bénéfices de nos lecteurs commerçants sont les bienvenues. Merci à l'avance !

Daniel Ferro

Dessin A. Meyer



Codez vos messages

INFORMATIQUE AMUSANTE

Si vous vous sentez une âme d'espion ou d'agent secret, ce programme de codage et décodage de messages fera certainement votre bonheur. Certes, de nombreux procédés de codage ou de chiffrage des textes existent. Pour notre part, nous en avons choisi un relativement simple à adapter sur ordinateur et néanmoins très efficace. Son principe est le suivant. L'utilisateur devra choisir une clé. Cette clé pourra être un mot, une phrase ou un texte, à condition de comporter moins de 160 caractères hormis les espaces et la ponctuation.

Une fois cette clé tapée, l'ordinateur n'en conservera que les lettres en les mettant bout à bout pour ne former, au total, qu'une suite de caractères. Dès lors, le codage du texte pourra commencer. Pour cela le code ASCII du premier caractère du texte sera additionné au code ASCII du premier caractère de la clé, et le nouveau symbole issu de cette opération sera le caractère, codé à transmettre. La même opération sera appliquée au second caractère, puis au troisième, et ainsi de suite.

Si le texte est plus long que la clé, ce qui est le plus souvent le cas, lorsque le dernier caractère de cette dernière aura été utilisé, le programme reprendra le premier, puis le second, etc. Pour le décodage du texte obtenu, la même clé devra être utilisée, et l'opération inverse sera effectuée. Le destinataire du message devra donc être mis au courant de la clé employée afin qu'il puisse prendre connaissance du message ainsi envoyé.

Notons que si un déchiffrage du message reste possible avec une clé ne comportant que deux ou trois caractères, il devient extrêmement délicat si la clé comporte plus de 20 caractères. Une phrase est donc l'exemple-type de la clé idéale, surtout si celle-ci n'est pas un proverbe ou un dicton connus de tous. Enfin, pour compliquer encore le codage, les espaces séparant les mots pourront également être codés. Cependant, cette option rend délicate la frappe du texte à décoder.

Ces quelques règles mises en place, passons donc à l'écriture du programme de ce codeur-décodeur. Nous commencerons par effacer l'écran, puis l'ordinateur sera positionné en mode 2 de manière à disposer d'une place suffisante pour l'affichage du message (ligne 10). Comme nous devrons mémoriser la clé de codage, nous la placerons dans un tableau. Celui-ci sera créé par l'instruction DIM de la ligne 20. Ensuite, la page de présentation sera affichée : lignes 30 à 90.

L'ordinateur demandera ensuite si le résultat du codage, ou du décodage, doit être imprimé. Le

```
10 CLEAR:MODE 2:CLS
20 DIM C$(160):LET M=0:LET C=0:LET L=0
30 LOCATE 35,1:PRINT "BONJOUR,"
40 LOCATE 1,6:PRINT "JE SUIS UN PROGRAMME DE CODAGE ET D
E DECODAGE DE MESSAGES."
50 LOCATE 1,8:PRINT "POUR M'UTILISER VOUS DEVREZ, EN PRE
MIER LIEU, CHOISIR VOTRE CLE: LA COMPLEXITE"
60 LOCATE 1,10:PRINT "DU CODAGE SERA DIRECTEMENT FONCTIO
N DE SA LONGUEUR."
70 LOCATE 1,12:PRINT "VOUS POURREZ ALORS TAPER VOTRE MES
SAGE COMME SUR UNE SIMPLE MACHINE A ECRIRE."
80 LOCATE 1,14:PRINT "LE MESSAGE APPARAITRA SIMULTANEMEN
T EN CLAIR ET EN CODE SUR L'ECRAN ET,"
90 LOCATE 1,16:PRINT "SI VOUS LE DESIREZ, SERA IMPRIMME.
"
100 LOCATE 20,18:PRINT "DESIREZ VOUS IMPRIMER LE MESSAGE
 ?"
110 LOCATE 20,20:PRINT "POUR OUI TAPER O"
120 LOCATE 20,22:PRINT "POUR NON TAPER N"
130 LET K$=INKEY$
140 IF K$="" THEN GOTO 130
150 IF K$="O" THEN LET L=1
160 IF L=0 THEN GOTO 200
170 LOCATE 1,24:PRINT "N'OUBLIEZ PAS DE METTRE VOTRE IMPR
IMANTE 'ON LINE' ET DE LA MUNIR DE PAPIER."
180 PRINT #8,CHR$(27);"Q";CHR$(50)
190 PRINT #8,CHR$(27);"l";CHR$(15)
200 CLS
210 LOCATE 1,5:PRINT "N'OUBLIEZ PAS DE PASSER EN MAJUSCU
LES (TOUCHE CAPS LOCK) PUIS TAPEZ"
220 LOCATE 1,7:PRINT "VOTRE CLE (160 CARACTERES AU MAXIM
UM); ESPACES ET PONCTUATION SERONT ELIMINES."
230 LOCATE 1,9:PRINT "SA FRAPPE TERMINEE VEUILLEZ LA VAL
IDER EN APUYANT SUR LA TOUCHE 'ENTER'. MERCI."
240 LOCATE 25,24:PRINT "POUR CONTINUER TAPEZ UNE TOUCHE.
"
250 IF INKEY$="" THEN GOTO 250
260 CLS: LOCATE 20,5
270 PRINT "VOTRE CLE S.V.P. PUIS 'ENTER'":LOCATE 1,10
280 FOR P=1 TO 160: PRINT ".";:NEXT P: LOCATE 1,10
290 LET A=1
300 LET K$=INKEY$
310 IF K$="" THEN GOTO 300
320 LET C$(A)=K$
330 IF ASC(K$)=13 OR A=160 THEN GOTO 370
340 PRINT K$;
350 IF ASC(K$)>64 AND ASC (K$)<91 THEN LET A=A+1
360 GOTO 300
370 CLS: LOCATE 20,5:PRINT "VOTRE CLE, UNE FOIS EPUREE,
EST DONC:"
380 LOCATE 1,15
390 FOR I=1 TO A
400 PRINT C$(I);
410 NEXT I
420 LOCATE 25,24:PRINT "POUR CONTINUER TAPEZ UNE TOUCHE.
"
430 IF INKEY$="" THEN GOTO 430
440 CLS
450 LOCATE 10,5:PRINT "VOULEZ VOUS CODER OU DECODER UN M
ESSAGE ?"
460 LOCATE 15,10:PRINT "- POUR CODER TAPER C"
470 LOCATE 15,12:PRINT "- POUR DECODER TAPER D"
480 LET K$= INKEY$
490 IF K$="" THEN GOTO 480
500 IF K$="C" THEN LET C=1
510 IF C=0 THEN GOTO 590
520 CLS
530 LOCATE 10,5:PRINT "DANS LE TEXTE, UNE FOIS CODE, DES
IREZ-VOUS CONSERVER LES ESPACES ?"
540 LOCATE 15,10:PRINT "- POUR OUI TAPER O"
550 LOCATE 15,12:PRINT "- POUR NON TAPER N"
560 LET K$= INKEY$
570 IF K$="" THEN GOTO 560
580 IF K$="O" THEN LET M=1
590 CLS:WINDOW #1,1,35,4,25:WINDOW #2,46,80,4,25
600 PRINT "UNE FOIS LA FRAPPE DU MESSAGE TERMIMEE TAPEZ
'ENTER' POUR UNE AUTRE TRADUCTION."
610 LOCATE 1,3:PRINT "CLAIR:":LOCATE 46,3:PRINT "CODE:"
620 FOR I=1 TO 770
630 PRINT #1,".";:PRINT #2,".";
640 NEXT I
650 IF C=0 THEN GOTO 770
660 LET B=1:LOCATE #1,1,1:LOCATE #2,1,1
670 LET K$=INKEY$
680 IF K$="" THEN GOTO 670
690 PRINT #1,K$;
700 LET X=ASC(K$)+ASC(C$(B))-58
710 IF K$=" " AND M=1 THEN LET X=32
720 PRINT #2,CHR$(X);:IF L=1 THEN PRINT #8,CHR$(X);
730 LET B=B+1
740 IF B=A THEN LET B=1
750 IF ASC(K$)=13 THEN GOTO 10
760 GOTO 670
770 LET B=1:LOCATE #1,1,1:LOCATE #2,1,1
780 LET K$=INKEY$
790 IF K$="" THEN GOTO 780
800 PRINT #1,K$;
810 LET X=ASC(K$)+58-ASC(C$(B))
820 IF K$=" " THEN LET X=32
830 PRINT #2,CHR$(X);:IF L=1 THEN PRINT #8,CHR$(X);
840 LET B=B+1
850 IF B=A THEN LET B=1
860 IF ASC(K$)=13 THEN GOTO 10
870 GOTO 780
```

contenu de la clé sera fourni et chacun de ses caractères mémorisé dans notre tableau (ligne 320). La frappe de la touche ENTER provoquera la fin cette séquence. La clé, débarrassée des espaces et de la ponctuation, sera présentée sur l'écran (lignes 370 à 410).

Les opérations de codage ou de décodage pourront commencer et le programme demandera quelle fonction doit être utilisée. La réponse sera fournie en tapant C pour codage et D pour décodage. En fonction de la réponse, le programme se rendra soit en ligne 520, soit en ligne 590. Les lignes 520 à 580 concernent les renseignements complémentaires relatifs à la conservation des espaces lors du codage. L'écran sera alors effacé puis séparé en deux zones : la première sera réservée à l'affichage du texte en clair, la seconde à celui du texte codé. Elles seront visualisées à l'aide de points. Cette opération terminée, la fonction souhaitée pourra être utilisée. Pour cela le programme utilisera deux routines différentes suivant qu'il s'agit d'un codage ou d'un décodage. L'aiguillage se fera toujours en fonction de la valeur prise par la variable C ; variable de mémorisation du choix de la fonction. Les lignes 650 à 760 concernant le codage et les lignes 770 à 870, le décodage.

Pour le codage, nous commencerons par positionner les curseurs des deux zones d'écran précédemment déterminées en début de page ; c'est-à-dire respectivement en haut et à gauche de chaque zone (ligne 660). De plus, nous utiliserons une variable B pour compter les caractères au fur et à mesure de leur frappe. Celle-ci sera utilisée pour "faire tourner" les lettres de la clé.

Chaque caractère frappé sera affiché dans la première zone (ligne 690) puis, après traitement dans la seconde zone, une fois codé (ligne 720). La variable B sera incrémentée, puis le programme se rebouclera jusqu'à ce que la touche ENTER soit tapée. Si tel est le cas, la réinitialisation de l'ensemble du programme sera effectuée en vue de l'utilisation d'une autre de ses fonctions ou d'une autre clé.

Le fonctionnement de la routine de décodage est tout à fait similaire à celle du codage. Seul le traitement effectué sur le caractère en cours de traduction sera différent. Cette opération est effectuée par la ligne 810.

La frappe de ce programme ne doit pas présenter de difficulté. Ici encore, de manière à en simplifier la compréhension, nous avons évité l'utilisation du Basic abrégé. Son utilisation est également très simple. Après avoir demandé RUN, la page de présentation s'affichera sur l'écran. Il ne faudra pas oublier, du moins pour entrer la clé et pratiquer un codage, de passer en mode CAPS LOCK. Après avoir tapé une touche, le programme demandera si le message doit être imprimé, puis vérifiera que l'imprimante est bien sous tension et équipée de papier. Si tel n'est pas le cas, cet oubli sera mentionné en bas de l'écran et le programme attendra sa réparation avant de continuer. Notons que ce test n'est effectué que si OUI a été répondu pour l'impression du message. La clé sera alors entrée puis validée en tapant ENTER. Cette validation sera confirmée par son affichage épuré sur l'écran.

La machine demandera alors si on désire coder ou décoder un message, puis la visualisation des deux zones écran sera effectuée. La frappe du message pourra commencer comme sur une machine à écrire : le texte s'affichera simultanément en clair sur la première zone et codé sur la seconde.

Henri-Pierre Penel ▲