

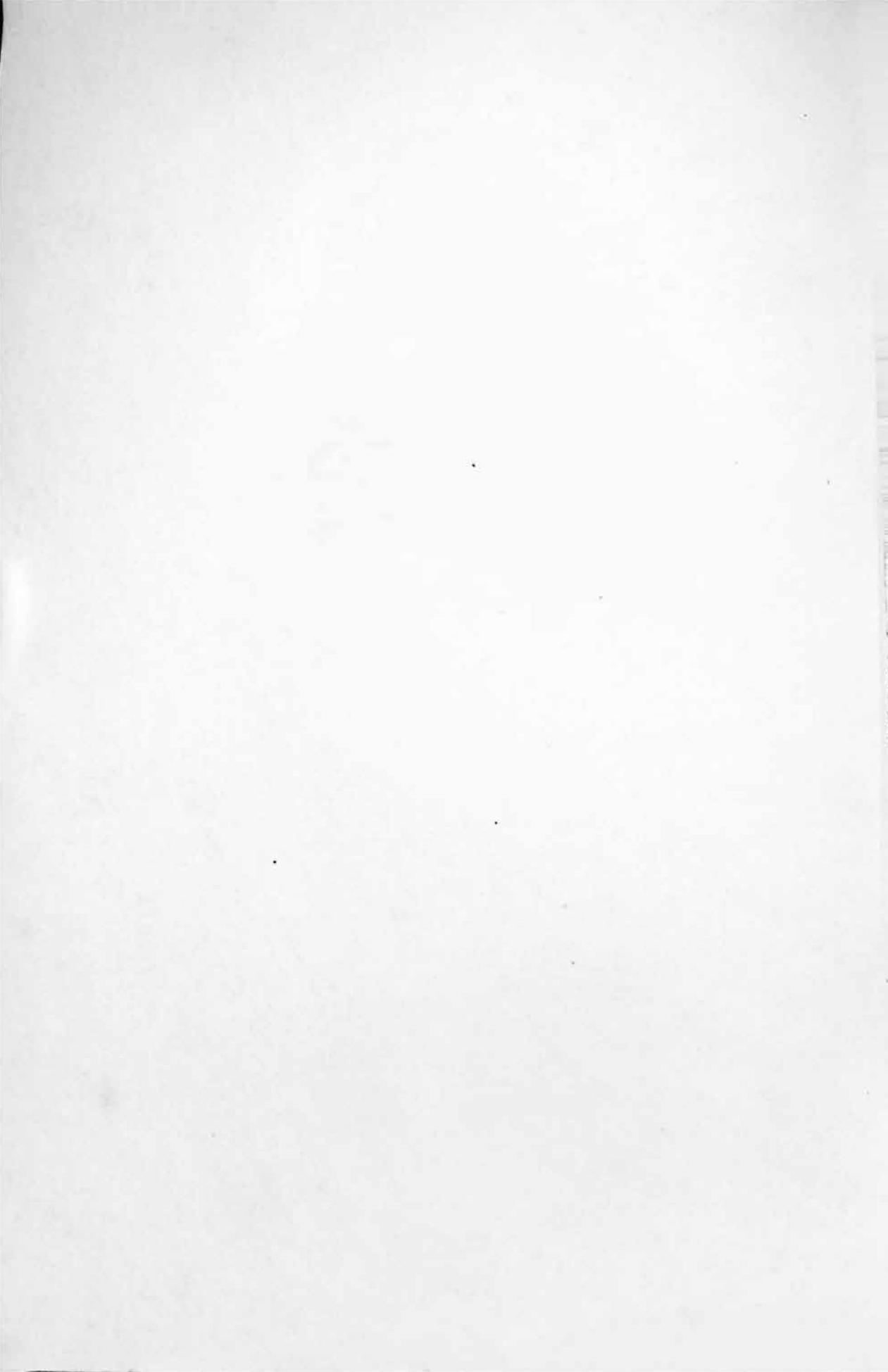
P.S.I.

MICRO POUR L'ÉCOLE

# EXL 100 A L'ÉCOLE



DANIEL NIELSEN



# **EXL 100**

## **A L'ÉCOLE**

---

## **Autre ouvrage relatif à l'EXL 100**

---

- 102 programmes pour EXL 100 - Jacques Deconchat
- 

## **Ouvrages relatifs à l'Enseignement Assisté par Ordinateur en Primaire**

---

- Le BASIC et l'école, tomes 1 et 2 - Jacques Gouet
- M05 et T07/70 à l'école - Daniel Nielsen
- M05 et T07/70 pour réussir en CM 1 - Daniel Nielsen
- Destination collège - Daniel Nielsen

## **A paraître :**

- M05 et T07/70 pour réussir en CE 2 - Daniel Nielsen
  - Logo à l'école - Daniel Nielsen et Gérard Bossuet
- 

## **Ouvrages généraux d'initiation à l'informatique**

---

- Visa pour l'informatique - Jean-Michel Jégo
- Visa pour le BASIC - Jean-Michel Jégo
- Le logotron informatique - Jean-Pierre Petit
- Les mots de la micro - Alan Freedman

Pour tout problème rencontré dans les ouvrages P.S.I.  
vous pouvez nous contacter au numéro ci-dessous :

**Numéro Vert/Appel Gratuit en France**

**16 (05) 21 22 01**

(Composer tous les chiffres, même en région parisienne)

La loi du 11 mars 1957 n'autorisant, aux termes des alinéas 2 et 3 de l'article 41, d'une part, que les «copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective», et, d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, «toute représentation ou reproduction intégrale, ou partielle, faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause, est illicite» (alinéa 1<sup>e</sup> de l'article 40).

Cette représentation ou reproduction, par quelque procédé que ce soit, constituerait donc une contrefaçon sanctionnée par les articles 425 et suivants du Code Pénal.

**MICRO POUR L'ÉCOLE**

# **EXL 100 A L'ÉCOLE**

**DANIEL NIELSEN**

**ADAPTÉ PAR PATRICE SOFRIN**



**Éditions du P.S.I.  
1985**

2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
20100

# 2001-2009 AUDITION

2010-2019  
AUDITION

2020-2029  
AUDITION

2030-2039  
AUDITION

2040-2049  
AUDITION

# Sommaire

<b>Introduction</b>	<b>7</b>
Conseils techniques généraux	8
Manettes de jeu et clavier	9
Instructions particulières	10
Index de classement par thèmes	12
Index de classement par niveaux	13
Kim musical	15
Autodictée	23
Soutien	31
Safari	39
Lecture rapide	47
Lecture débutant	57
Pluriel des noms	65
Conjugaison	73
Addition	81
Soustraction	89
Multiplication	99
Division	109
Ravitaillement en vol	119
Le train capricieux	125
Villes de France	137
Construction de phrases	145
J'écris tout seul	153
Reconstitution de textes	163
Cocktail magique	171
Masse et volume	181
Kim nombres	193
<b>Conclusion</b>	<b>203</b>
<b>Conseils de lecture</b>	<b>205</b>

Daniel Nielsen, directeur d'école, travaille depuis deux ans à l'introduction de l'informatique dans l'enseignement primaire.

Il a pu constater que l'ordinateur est une aide précieuse, en particulier pour les enfants ayant des difficultés scolaires.

# Introduction

L'ordinateur à l'école n'est plus une innovation. En effet, de multiples recherches ont été effectuées et de nombreuses expériences actives sont en cours dans des écoles. Elles ont permis de constater que l'ordinateur ouvrait de nouvelles perspectives d'enseignement en apportant des motivations différentes, en créant des possibilités d'action originales (pour les enseignants et les enfants) dans le travail en classe.

Désormais, l'enfant n'est plus un sujet passif absorbant un lot de connaissances multiples et diverses, mais au contraire un élément actif de sa propre formation.

L'ordinateur à la maison fait aussitôt penser aux jeux, à la gestion familiale, mais il peut être aussi un auxiliaire éducatif pour les enfants, car l'ordinateur est un excellent répétiteur : l'exercice sur ordinateur détecte chaque erreur de l'enfant, l'amène à la corriger et lui permet de continuer l'exercice sur des bases exactes.

Enfin, pour les enfants, c'est cet instrument séduisant qui rappelle plus les jeux vidéo que l'école ou les devoirs à la maison et devant lequel s'assiéront avec plaisir pour participer à une activité également ceux que l'on dit inadaptés au monde scolaire (à moins que ce soit le monde scolaire qui ne leur soit pas adapté...).

Si vous partagez ces opinions, et si vous ne disposez pas du temps nécessaire pour réaliser vous-même vos propres programmes, ce livre a été écrit pour vous.

Les 21 programmes qui y sont présentés recouvrent toutes les matières étudiées à l'école et tous les niveaux d'âge (voir les index de classement par niveaux et par thèmes). Ces pro-

grammes ont été écrits par un enseignant, tout d'abord pour être utilisés par les enfants dans l'école dont il a la responsabilité.

C'est donc la finalité éducative et la présentation sur l'écran qui ont été les objectifs prioritaires lors de l'élaboration de ces programmes. Certains trouveront sans doute des procédures de travail plus courtes.

Avec l'adoption par l'éducation nationale de l'EXL 100 pour le plan 100 000 micros, c'est un bon matériel éducatif qui va pénétrer le monde de l'enseignement. Les programmes profitent en grande partie des possibilités originales de l'EXL 100, notamment au niveau de l'affichage.

**Finalité éducative** : à chaque programme, qu'il soit didactique ou de jeu, correspond un certain nombre de connaissances que l'enfant va acquérir ou ré-investir. De même, il devra, pour arriver au but, mettre en œuvre différents types de « savoir-faire » (attention, logique, etc.).

**Présentation sur l'écran** : c'est un point très important, et parfois malheureusement négligé, des programmes pour enfants : une quantité trop importante d'informations sur l'écran détourne l'attention. Il faut souvent « faire le ménage » et ne garder que ce qui est indispensable et bien le placer sur l'écran. Cela allonge les programmes, mais c'est indispensable.

Certains listings de programmes vous paraîtront parfois longs à taper : afin de rendre les listes claires et facilement copiables, les programmes sont ponctués d'instructions REM expliquant les fonctions des différents blocs d'instructions. De plus, les numéros de ligne se succèdent de 10 en 10 afin de faciliter la saisie.

Par précaution, et par habitude aussi des différents tracas subis lors des saisies de programmes, nous vous conseillons de sauvegarder les programmes périodiquement lorsque vous les tapez. De cette manière, même si un incident survient (panne de courant, ou ... saute d'humeur de l'EXL 100 !), vous ne perdrez pas trop de travail et n'aurez qu'un petit nombre de lignes à taper de nouveau.

## Conseils techniques généraux

Les fonctions RND et INTRND (tirage au sort de nombres aléatoires) sont présentes dans chaque programme. Ce sont elles qui permettent, à partir des mêmes données, d'obtenir des exercices ou des jeux différents chaque fois.

Sur l'EXL 100, ces fonctions sont en fait pseudo-aléatoires : elles utilisent une fonction mathématique relativement imprévisible qui utilise le nombre généré à l'appel précédent pour en générer un nouveau. En conséquence, si le premier nombre généré est fixé, la suite des nombres générés par chaque appel sera la même. C'est notamment le cas si on n'initialise pas le générateur au début du programme. A cet effet, l'Exelbasic propose une instruction RANDOMIZE. Si cette instruction est utilisée, le générateur est initialisé avec un nombre provenant de l'horloge interne de l'EXL 100 (nombre modifié plusieurs milliers de fois par seconde !) : on obtiendra alors des jeux à chaque fois différents.

Toutefois, il peut être souhaitable d'obtenir le même jeu pour tous les enfants. Dans ce cas, il est possible de mettre un nombre derrière RANDOMIZE. La suite générée dépendra du nombre en question, mais elle sera la même à chaque exécution du programme. Vous pouvez utiliser cette possibilité par exemple pour personnaliser les jeux afin que chaque élève ait son propre jeu. Il suffira (ce n'est qu'un exemple de l'application de ce principe) de placer une instruction RANDOMIZE LEN(P\$) au début du programme, la variable P\$ contenant le nom du joueur.

Ce second procédé, intéressant à l'école, l'est moins à la maison car l'intérêt va diminuer rapidement. A la maison, il faudra donc, après deux ou trois utilisations, modifier les programmes qui font appel à la deuxième technique, et se servir de la première.

## Manettes de jeux et clavier

Si vous possédez les manettes de jeux Exelvision, certains programmes seront beaucoup plus attractifs. En effet, la manette à touches orange est totalement compatible avec le clavier. Au lieu d'utiliser les touches « flèches » et « barre espace », l'enfant utilisera le manche et les boutons de tir.

Au sujet du clavier, vous devrez faire particulièrement attention à la touche LOCK (touche orange située en bas à droite sur le clavier). Un appui sur celle-ci bloque le clavier en mode « minuscules », et l'enfant risque d'être perdu face à un programme qui ne reconnaîtra que les majuscules. D'autre part, les chiffres ne sont plus accessibles facilement dans ce mode.

Les programmes font parfois référence à la touche « PRET ». Il s'agit en fait de la touche « ← », également appelée « READY », « RETURN » ou « ENTER » sur d'autres matériels. Il faudra la mettre en évidence auprès des enfants avant tout travail, puisque le clavier Exelvision ne donne pas de nom à cette touche.

Enfin, si vous possédez une cartouche de mémoire vive 16 Ko, il vous est conseillé de mettre dans cette cartouche les quatre programmes les plus utilisés (il sera difficile d'en faire tenir un cinquième) afin de permettre une mise en route rapide et aisée. N'oubliez pas dans ce cas de les protéger en écriture (grâce à l'option MFP du CROS) pour éviter un effacement accidentel.

## Instructions particulières

L'EXL 100 possède certaines instructions qui lui sont propres. Elles sont parfois moins de simples curiosités qu'elles n'en ont l'air.

Ainsi, l'utilisation de ACCEPT (forme sophistiquée de INPUT) permet de saisir des réponses de l'enfant sans risquer de message erreur du système du style « Err 3 in 1782 ». Il est de plus impossible à l'enfant de rentrer des lettres si des chiffres sont attendus, et réciproquement. Les possibilités de l'instruction ACCEPT rendent les entrées de données beaucoup plus faciles pour les néophytes que sur les autres machines.

De même pour toutes les entrées de phrases, l'instruction LINPUT a été préférée à INPUT car elle permet d'entrer tous les signes de ponctuation, notamment les virgules qui provoquent une erreur avec INPUT.

Les tirages aléatoires, généralement de la forme INT(N \* RND)+1 deviennent plus clairs grâce à l'équivalent EXL, INTRND(N), qui génère directement un nombre entier entre 1 et N.

Toutefois, rien n'est parfait : l'EXL 100 manque cruellement d'instruction sonore. C'est pourquoi un SUB PLAY a été développé dans les programmes de ce livre pour permettre à la machine de jouer un air directement à partir de notes de musique exprimées en clair.

Les applications de SUB et SUBEND sont innombrables : elles permettent de créer des instructions supplémentaires avec des paramètres au choix (y compris des tableaux de variables si on le désire) et cette possibilité a été utilisée systématiquement

dans les programmes pour les affichages en double taille, l'initialisation des programmes (effacement de l'exaspérante ligne de contrôle du haut de l'écran, mise en place d'un générateur de musique), et de manière générale tout sous-programme dont l'action rappelle celle d'une instruction simple.

L'effet est avant tout de rendre le listing plus clair, mais également de procurer un gain non négligeable de mémoire (le programme est moins long), donc un temps de saisie plus réduit. De plus, les facilités de programmation procurées par SUB ne peuvent pas être ignorées. La mise au point des programmes a été beaucoup plus rapide que sur une autre machine grâce, notamment, aux variables locales (elles ne concernent que l'intérieur du SUB et n'interfèrent pas avec les variables situées au niveau au-dessus).

## Index de classement par thèmes

<i>Thèmes</i>	<i>Jeu</i>	<i>Didact</i>	<i>Etude de situation</i>	<i>Français</i>	<i>Maths</i>	<i>Eveil</i>	<i>Attention</i>	<i>Mémoire</i>	<i>Logique</i>	<i>Rapidité</i>
<i>Programmes</i>										
Kim musical	X					X	X	X		
Autodictée		X		X			X	X		
Soutien	X	X		X	X	X		X		
Safari	X				X		X			X
Lecture rapide			X	X			X	X		
Lecture débutant			X	X			X	X		
Pluriel des noms	X		X				X	X		
Conjugaison	X		X				X	X		
Addition	X			X			X			
Soustraction	X			X			X			
Multiplication	X			X			X			
Division	X			X			X			
Ravitaillement en vol	X				X		X		X	
Le train capricieux	X			X	X	X	X			X
Villes de France			X			X	X	X		
Construction de phrases			X	X			X		X	
J'écris tout seul			X	X			X		X	
Reconstitution de textes			X	X			X		X	
Cocktail magique			X		X		X	X	X	
Masse et volume			X		X		X	X	X	
Kim nombres	X				X		X	X		

## Index de classement par niveaux

Programmes	Niveaux (années)							
	5	6	7	8	9	10	11	12 et plus
Kim musical			X	X	X	X	X	X
Autodictée			X	X	X	X	X	X (1)
Soutien		X	X	X	X	X	X	X (1)
Safari			X	X	X	X		
Lecture rapide			X	X	X	X	X	X
Lecture débutant	X	X	X					
Pluriel des noms			X	X	X	X	X	X
Conjugaison				X	X	X	X	X
Addition		X	X	X	X	X	X	X
Soustraction		X	X	X	X	X	X	X
Multiplication				X	X	X	X	X
Division					X	X	X	X
Ravitaillement en vol			X	X	X	X	X	X
Le train capricieux	X	X	X	X	X	X	X	X (1)
Villes de France				X	X	X	X	X
Construction de phrases				X	X	X	X	X
J'écris tout seul		X	X					
Reconstitution de textes			X	X	X	X	X	X (1)
Cocktail magique						X	X	X
Masse et volume						X	X	X
Kim nombres								

selon niveau d'adaptation

(1) Le niveau dépend surtout des données entrées dans le programme.



# **KIM MUSICAL**

## **Thème**

Jeu musical sur le principe du jeu de Kim, recherche d'une note qui a disparu d'une séquence musicale de six notes, jouée par EXL 100 et affichée sur une portée.

## **Niveau**

A partir de 7 ans.

## **Intérêt pédagogique**

### **Connaissances**

- découverte de la gamme et des notes de musique ;
- place des notes de musique sur une portée.

### **Savoir-faire**

- mémoire ;
- perception auditive ;
- reconnaissance visuelle et auditive de la note manquante.

## Déroulement

Après une explication du jeu, la gamme est présentée au joueur : chaque note s'inscrit sur une portée, son nom s'affiche en dessous et elle est jouée par EXL 100.

Ensuite, une séquence aléatoire de six notes est affichée et jouée.

Cette séquence est jouée une deuxième fois, mais avec une note en moins. L'enfant doit taper le nom de la note qui a disparu. EXL 100 joue sa réponse.

Si la réponse est bonne, on passe à un jeu suivant. Sinon, le joueur a une seconde chance, après laquelle la bonne réponse lui est donnée, s'il s'est encore trompé.

Le jeu se répète cinq fois mais, à partir du quatrième jeu, le nom de la note n'est plus affiché. L'enfant doit trouver la bonne réponse à l'aide des notes dessinées sur la portée et des sons joués par EXL 100.

## Commentaires

- |                  |  |
|------------------|--|
| <b>130 à 210</b> | Initialisation et lecture des notes de la gamme. Trois tableaux sont utilisés pour mémoriser la gamme : N contient la fréquence de la note destinée au générateur sonore, CO\$ sa couleur et NOM\$ son nom. Les notes sont ainsi définies par un indice de 1 à 7.  |
| <b>220 à 330</b> | Présentation du jeu.   |
| <b>340 à 350</b> | Affichage et jeu de la gamme.  |
| <b>360 à 460</b> | Affichage et jeu de la séquence musicale. Chaque note de la séquence est tirée au hasard à la ligne 400, affichée en 410 (avec son nom pour les trois premiers jeux, ligne 420) et enfin jouée en ligne 430 avant d'être mémorisée en ligne 440.   |
| <b>470 à 560</b> | Affichage et jeu de la séquence tronquée. La note à enlever est tirée au sort à la ligne 480. Ensuite, on affiche et on joue la séquence de la même façon que précédemment mais en ignorant la note choisie grâce au test de la ligne 520. La position des notes dépendant de leur indice dans la séquence, il y a un espace libre à l'endroit de la note enlevée. |

<b>570 à 650</b>	Entrée de la réponse, test de la réponse. Message d'erreur en cas de réponse fausse. La ligne 650 renvoie à une seconde lecture de la séquence tronquée.
<b>660 à 700</b>	Affichage de la bonne réponse en cas de seconde réponse erronée.
<b>710 à 740</b>	Cas de réponse correcte.
<b>750 à 840</b>	Fin du programme. Si le troisième jeu n'a pas encore été effectué, on recommence, sinon on affiche un message pour avertir le joueur de la disparition du nom des notes.
<b>860 à 910</b>	SUB INIT. Sous-programme effaçant la ligne de contrôle et mettant en place le générateur sonore.
<b>920 à 940</b>	SUB PLAY. Sous-programme jouant une note en connaissant sa fréquence. Voir SUB CONVER pour plus de détails.
<b>950 à 1060</b>	SUB GAMME. Dessine et joue la gamme.
<b>1070 à 1120</b>	SUB PORTEE. Dessine une portée à l'écran. Ce SUB est utilisé par GAMME et par les boucles d'affichage des séquences musicales.
<b>1130 à 1170</b>	SUB CONVER. Convertit le nom d'une note en fréquence destinée au générateur sonore.

## Particularités techniques

L'absence d'instructions musicales dans l'Exelbasic est palliée par la mise en place d'un générateur sonore logiciel, basé sur un principe d'interruptions et utilisant le port cassette pour envoyer les sons sur le téléviseur. Son utilisation est moins souple que celle d'une instruction spécialisée, mais le générateur remplit néanmoins parfaitement son rôle. Ce dernier est mis en place par la ligne 870. On génère un son par CALL POKE(258,L,H) où L est un nombre de 15 à 255 et H un nombre au-dessus de 128. On arrête le son par CALL POKE(259,0).

Les instructions LOCATE présentes lors des boucles d'affichage des notes ou de la gamme ont une particularité : elles sont suivies en premier du numéro de la ligne d'affichage visée (de 1 à 22) et en second du numéro de la colonne (1 à 40).

Les instructions et fonctions RANDOMIZE (ligne 330) et INTRND nécessitent quelques commentaires : INTRND(X) génère un nombre entre 1 et X. L'initialisation du générateur aléatoire est obtenue par RANDOMIZE. Si vous désirez que le jeu soit le même à chaque exécution du programme, il suffit d'ajouter un nombre derrière cette instruction à la ligne 330. Cela vous permettra de mieux comparer les résultats des enfants, les séquences musicales étant les mêmes pour tous.

La ligne 440 attribue à chaque note aléatoire de la première séquence un numéro d'ordre D(l) qui permet de déterminer ensuite dans la deuxième séquence la note à supprimer, et de jouer et afficher de nouveau les autres.

## Modifications

Il est possible d'augmenter le nombre de notes de la séquence musicale (lignes 270, 390 et 510). Mais si vous voulez placer plus de neuf notes sur l'écran, il faut aussi diminuer l'écartement entre l'affichage des notes (lignes 410/420 et 530/540).

Si ce programme est utilisé par des enfants n'ayant aucune connaissance des notes de musique (il en existe encore quelques-uns !), il est souhaitable de faire rejouer la gamme plus souvent :

- avant chaque jeu (lignes 760 et 800 : 340 au lieu de 370) ;
- en cas de première réponse erronée, avant de rejouer la séquence

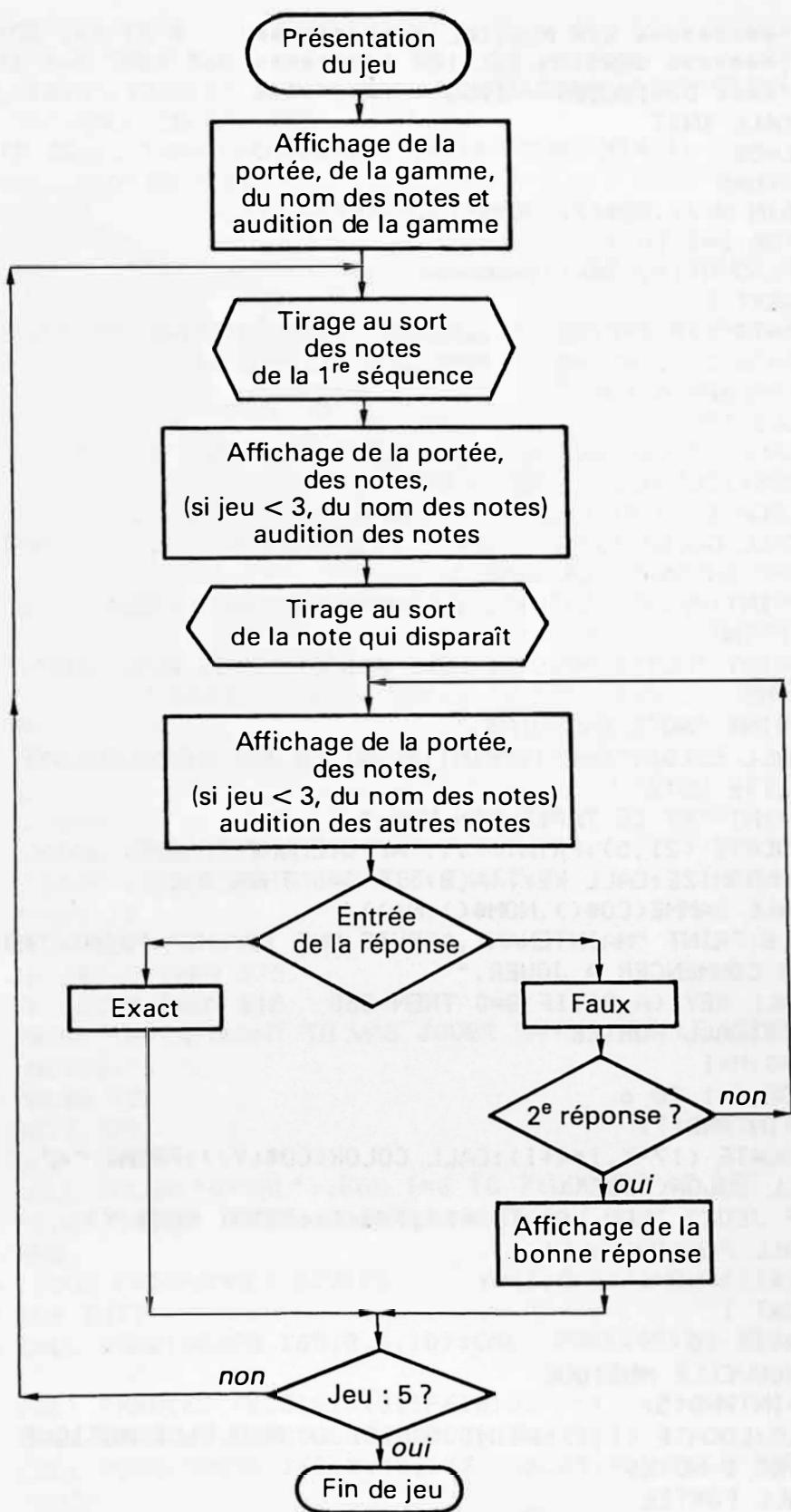
650 C=I :CALL GAMME (CO\$(),NOM\$(),N())

655 GOTO 490

Par contre, avec des enfants ayant de bonnes connaissances musicales, on peut supprimer les lignes 420, 540 et 760, les notes apparaîtront ainsi sans leur nom dès le premier jeu.

## Adaptations

Si vous préférez que EXL 100 vous joue de vraies mélodies, au lieu d'une suite de notes déterminées au hasard et dont l'harmonie peut paraître parfois douteuse, vous pouvez remplacer la ligne 400 par un READ Y, en ajoutant une ligne 405 DATA contenant les indices des notes de votre mélodie.



```

100 !***** KIM MUSICAL *****
110 !***** VERSION EXL 100 *****
120 !*** D.NIELSEN - 1983 *****
130 CALL INIT
140 L=50
150 JEU=0
160 DIM N(7),C0$(7),NOM$(7),NT$(7),D(7)
170 FOR I=1 TO 7
180 READ N(I),C0$(I),NOM$(I)
190 NEXT I
200 DATA 199,1RC,D0,176,1BC,RE,158,1BC,MI,149,1RC,FA
210 DATA 133,1RC,SOL,118,1BC,LA,105,1WC,SI
220 !PRESENTATION
230 CLS "BCC"
240 CALL COLOR("0BYLH"):LOCATE (4,8):PRINT "KKI IMM MM
UUSSIICCAALL"
250 LOCATE (5,8):PRINT "KKI IMM MMUSSIICCAALL"
260 CALL COLOR("0BC"):LOCATE (10,8):PRINT "TU VAS D'AB
ORD ENTENDRE LA GAMME"
270 PRINT:PRINT "ENSUITE TU ENTENDRAS UN MORCEAU DE 6"
:PRINT
280 PRINT "NOTES,PUIS LE MEME MORCEAU MAIS AVEC UNE":P
RINT
290 PRINT "NOTE EN MOINS."
300 CALL COLOR("0BC"):PRINT:PRINT "A TOI DE RETROUVER
 CETTE NOTE"
310 PRINT "ET DE TAPER SON NOM."
320 LOCATE (21,5):PRINT "... APPUIE UNE TOUCHES ..."
330 RANDOMIZE:CALL KEY1(A,B):IF B=0 THEN 330
340 CALL GAMME(C0$(),NOM$(),N())
350 CLS:PRINT "MAINTENANT,APPUIE UNE TOUCHES":PRINT "PO
UR COMMENCER A JOUER."
360 CALL KEY1(A,B):IF B=0 THEN 360
370 CLS:CALL PORTEE
380 C=0:M=I
390 FOR I=1 TO 6
400 Y=INTRND(7)
410 LOCATE (17-Y,I*4+1):CALL COLOR(C0$(Y)):PRINT "A":C
ALL COLOR("0BC")
420 IF JEU<3 THEN LOCATE (18,I*4+1):PRINT NOM$(Y)
430 CALL PLAY(N(Y),L)
440 NT$(I)=NOM$(Y):D(I)=Y
450 NEXT I
460 PAUSE 10
470 !NOUVELLE MUSIQUE
480 F=INTRND(5)
490 CLS:LOCATE (1,1):PRINT "VOICI LA NOUVELLE MUSIQUE
 AVEC 5 NOTES"
500 CALL PORTEE

```

```

510 FOR I=1 TO 6
520 IF I=F THEN 560
530 LOCATE (17-D(I),I*4+1):CALL COLOR(C0$(D(I))):PRINT
  "A":CALL COLOR("0BC")
540 IF JEU<3 THEN LOCATE (18,I*4+1):PRINT NT$(I)
550 CALL PLAY(N(D(I)),L)
560 NEXT I
570 !REPOSE
580 LOCATE (20,1):INPUT "QUELLE EST LA NOTE EN MOINS ?
  ";REP$
590 CALL CONVER(REP$,NO):IF NO<>0 THEN CALL PLAY(NO,L)

600 IF REP$=NT$(F)THEN 710
610 IF C=1 THEN 660
620 CALL COLOR("0BC"):LOCATE (20,1)
630 PRINT "NON,CE N'EST PAS CELA. ECOUTE UNE":PRINT "DE
  UXIEME FOIS."
640 PAUSE 10
650 C=1:GOTO 490
660 !AFFICHAGE BONNE REPOSE
670 LOCATE (19,1):PRINT RPT$(" ",40); "NON,C'EST ENCORE
  "
680 PRINT "LA BONNE REPOSE ETAIT ";:CALL COLOR("0BC")
  :PRINT NT$(F)
690 CALL PLAY(N(D(F)),L)
700 GOTO 740
710 !GAGNE
720 LOCATE (19,1):PRINT RPT$(" ",80);
730 LOCATE (20,1):PRINT "C'EST EXACT."
740 PAUSE 10
750 JEU=JEU+1
760 IF JEU<3 THEN 370
770 IF JEU=5 THEN 810
780 PRINT "MAINTENANT TU VAS JOUER SANS LIRE LE NOMDES
  NOTES."
790 PAUSE 10
800 GOTO 370
810 CLS "CBB"
820 CALL COLOR("0RBHL"):FOR I=6 TO 7:LOCATE (I,12)
830 PRINT "AAUU  RREEVVVOOIIRR":NEXT
840 END
850 !SOUS PROGRAMMES DIVERS
860 SUB INIT
870 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

880 CALL CHAR(65,"01010101011F6181827C")
890 CALL CHAR(66,"0000000000000FF0000")
900 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
  50850)

```

```
910 SUBEND
920 SUB PLAY(NOTE,DUREE)
930 CALL POKE(258,NOTE,129):PAUSE DUREE/50:CALL POKE(2
59,45)
940 SUBEND
950 SUB GAMME(CO$(),NOM$(),N())
960 CLS
970 CALL PORTEE
980 CALL COLOR("0BC")
990 LOCATE (20,1):PRINT "VOICI LA GAMME"
1000 FOR I=1 TO 7
1010 LOCATE (17-I,4*I+1):CALL COLOR(CO$(I)):PRINT "A":C
ALL COLOR("0BC")
1020 LOCATE (18,I*4+1):PRINT NOM$(I)
1030 CALL PLAY(N(I),40)
1040 NEXT I
1050 PAUSE 8
1060 SUBEND
1070 SUB PORTEE
1080 CALL COLOR("1BC")
1090 FOR I=6 TO 14 STEP 2
1100 LOCATE (I,1):PRINT RPT$("B",40);
1110 NEXT I
1120 SUBEND
1130 SUB CONVER(NOTE$,PERI)
1140 PERI=0:K=POS("DO.RE.MI.FA.SOLLA.SI.",SEG$(NOTE$&".
",1,3),1)
1150 IF K=0 THEN PERI=0:SUBEXIT
1160 PERI=VAL(SEG$("199176158149133118105",K,3))
1170 SUBEND
```

# **AUTODICTEE**

## **Thème**

Programme d'autodictée de cinq phrases courtes.

## **Niveau**

A partir de 6 ans (Cours préparatoire 2<sup>e</sup> trimestre).

## **Intérêt pédagogique**

### **Connaissances**

- comme pour un travail de dictée : savoir utiliser une orthographe correcte.

### **Savoir-faire**

- mémorisation des phrases ;
- utilisation du clavier ;
- esprit de synthèse en cas d'erreur précédente.

## **Déroulement**

Après le chargement du programme, l'adulte (enseignant ou parent) entre lui-même les cinq phrases sur lesquelles les enfants vont travailler.

Ensuite, le programme s'adresse aux enfants. La phrase est affichée et reste à l'écran aussi longtemps que l'enfant le souhaite pour la mémoriser complètement (c'est lui qui relance la suite du programme).

La phrase s'efface et l'enfant est invité à la taper. Son texte apparaît en bas de l'écran. Lorsqu'il appuie sur la touche PRET, pour confirmer la fin de son texte, EXL 100 compare celui-ci avec la phrase donnée. Si le nombre de lettres ou d'espaces est différent, un message d'erreur s'affiche. Si le nombre de lettres est le même, le texte de l'enfant est inscrit au milieu de l'écran et les fautes éventuelles sont visualisées par un carré rouge.

Si le texte est exact, on passe à la phrase suivante, sinon l'enfant doit recommencer. Au bout de trois essais infructueux, la phrase correcte est affichée.

A la fin du jeu, le score de l'enfant est commenté par EXL 100.

## Commentaires

- |                  |   |
|------------------|---|
| <b>100</b>       | Permet de sortir du programme en supprimant la protection du haut de l'écran pour éviter que celle-ci ne reste présente dans les programmes suivants. |
| <b>140 à 160</b> | Présentation du programme.  |
| <b>170 à 370</b> | Entrée des cinq phrases. L'instruction LINPUT est utilisée pour permettre la saisie des virgules.   |
| <b>380 à 510</b> | Travail sur les cinq phrases (le travail est effectué à l'intérieur du sous-programme GOSUB 630).   |
| <b>520 à 620</b> | Score –commentaires– fin du jeu. Le jeu est relancé pour un autre enfant.   |
| <b>630 à 710</b> | Affichage de la phrase dans un rectangle de couleur et attente d'une touche.  |
| <b>720 à 790</b> | Entrée du texte de l'enfant.  |
| <b>800 à 860</b> | Test de la réponse. Les lignes 820 à 860 comparent lettre par lettre les deux phrases.  |
| <b>870 à 890</b> | Affichage (aucune faute).   |
| <b>900 à 920</b> | Affichage (nombre de fautes).   |

<b>930 à 1000</b>	Affichage de la bonne réponse si l'enfant en était à son troisième essai.
<b>1020 à 1130</b>	Affichage lorsque le nombre de lettres n'est pas le même dans les deux phrases.
<b>1140</b>	Fin du programme.
<b>1150 à 1160</b>	Affichage d'un carré rouge à l'emplacement de la faute.
<b>1170 à 1190</b>	SUB INIT. Efface la ligne de contrôle et met en place le générateur sonore.
<b>1200 à 1210</b>	SUB LINO. Efface la ligne de contrôle.
<b>1220 à 1270</b>	SUB PLAY. Joue des notes de musique de longueur donnée.

## Particularités techniques

Il serait dommage de faire des fautes d'orthographe dans un programme de dictée ! C'est pourquoi les lignes 900-910 ainsi que 560-570, 970-980 et 1060-1090 tiennent compte de la valeur des variables T, K et X afin de ne pas afficher « 1 FAUTES » !

Dans le test réponse, la ligne 810 compare la longueur des phrases puis, si les deux phrases ont le même nombre de lettres, celles-ci sont analysées puis affichées une à une à l'intérieur de la boucle des lignes 820 à 860.

Il peut être utile pour de jeunes enfants d'y ajouter une temporisation (exemple : 855 PAUSE 2) afin que cet affichage se fasse plus lentement, ce qui permet de mieux visualiser les lettres fausses.

Ce programme, comme « Soutien » ou « Le train capricieux » fonctionne sur des données qui sont introduites au début du programme. Cela permet d'utiliser des phrases correspondant au niveau des enfants (voir le paragraphe « adaptations »). Mais il ne faut pas interrompre le programme (touche CTL puis C) car on doit alors entrer de nouveau les phrases.

## Modifications

Comme dans « Soutien », on peut supprimer le commentaire du score (lignes 580 à 600).

Les instructions PAUSE situées aux lignes 620, 930, 1000 et 1100 servent de temporisation aux différents messages. Il est possible de modifier leur longueur pour les adapter au niveau de chacun.

Il est intéressant, lorsque les phrases sont relativement longues et utilisées par des enfants ayant des difficultés en orthographe, de « ressortir » la phrase modèle sur l'écran avant un nouvel essai car il arrive parfois qu'il y ait plus de fautes encore au 3<sup>e</sup> essai qu'au 1<sup>er</sup>, si la mémoire du joueur est fragile. Il faudrait ajouter :

```
745 PRINT « RELIS BIEN LA PHRASE »
746 LOCATE(10,1) : PRINT V$
747 PAUSE 5
```

Le nombre d'essais possibles (variable M) peut être réduit à deux, ou au contraire augmenté en modifiant les tests portant sur la valeur de M dans les lignes 740 et 900-980.

## Adaptations

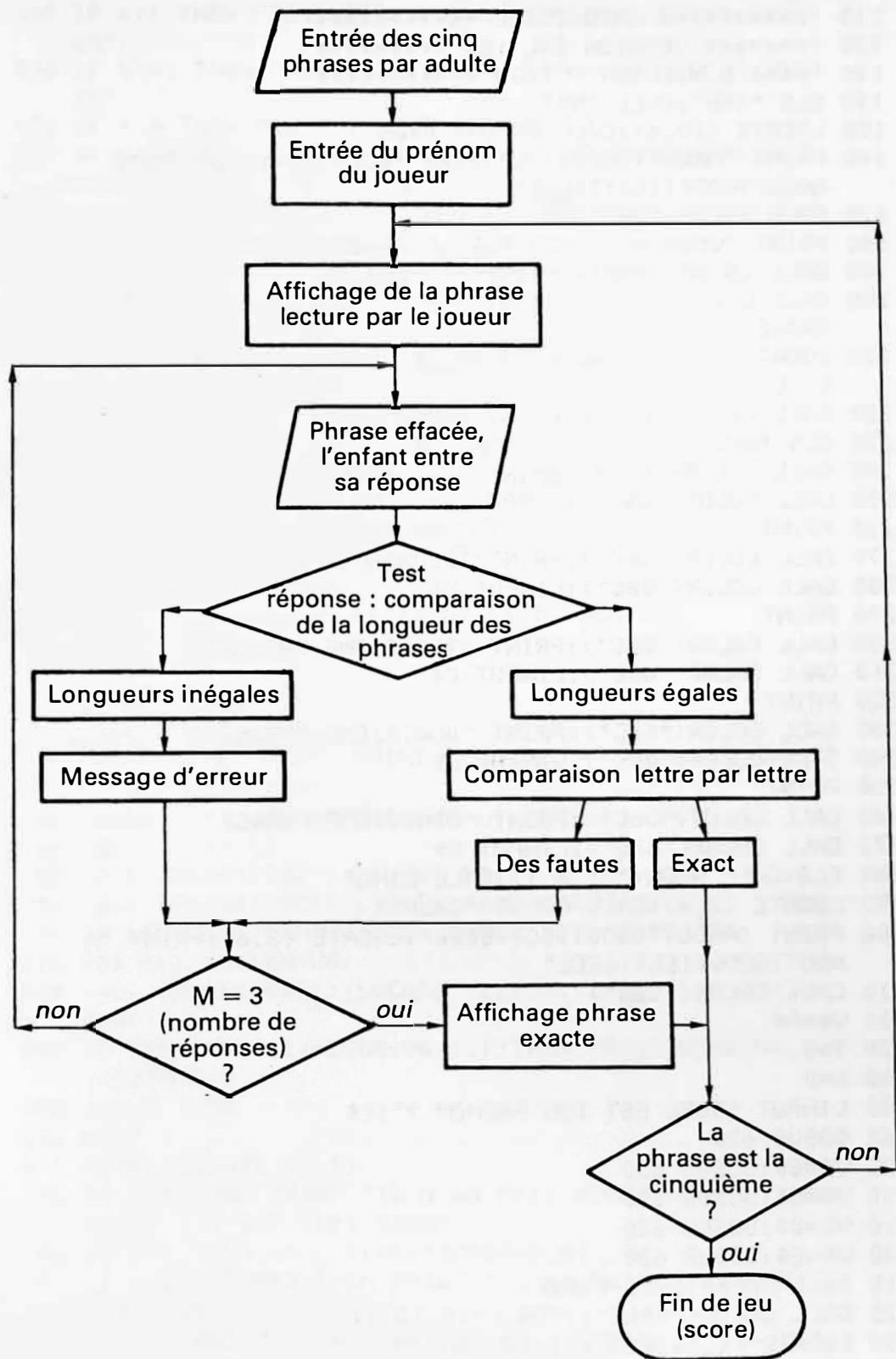
Certains programmes d'autodictée utilisent des phrases inscrites en DATA. Ici, l'entrée des phrases en début de programme, si elle présente quelques inconvénients techniques, donne une plus grande liberté d'adaptation au niveau des enfants, par exemple :

- phrases très courtes, étudiées en lecture, en C.P. ;
- phrases d'une dictée déjà faite, ou à faire ensuite, et qui comportent certaines difficultés particulières, en C.M.

Il serait aussi possible de constituer un fichier de phrases, sur cassette ou disquette, et de « piocher » dans celui-ci ce qui réduirait le temps de mise en route du programme.

Enfin, dans une école, lorsque plusieurs enfants utilisent le programme en « libre service », il est utile de garder une trace de leurs erreurs, à l'aide de l'imprimante, par exemple :

```
791 OPEN#1,« 20 »,OUTPUT
792 PRINT#1,Z$,V$
793 CLOSE #1
```



```
100 ON BREAK ERROR:ON ERROR 1140
110 !***** AUTODICTEE *****
120 !***** VERSION EXL 100 *****
130 !**** D.NIELSEN - 1983 *****
140 CLS "CBB":CALL INIT
150 LOCATE (10,6):CALL COLOR("0YLH")
160 PRINT "AAUUTTOODDIICCTTEEEE":LOCATE (11,6):PRINT "
AAUUTTOODDIICCTTEEEE"
170 CALL COLOR("0RB")
180 PRINT "VOUS ALLEZ ENTRER VOTRE TEXTE"
190 CALL COLOR("0MB"):PRINT "ATTENTION"
200 CALL COLOR("0RB"):PRINT "DEUX LIGNES MAXIMUM PAR P
HRASE !"
210 LOCATE (20,1):PRINT "POUR COMMENCER,TAPEZ LA TOUCH
E 'C'"
220 CALL KEY1(A,B):IF A<>67 OR B<>1 THEN 220
230 CLS "BCC"
240 CALL COLOR("0bC"):PRINT "PREMIERE PHRASE"
250 CALL COLOR("0BC"):LINPUT A$
260 PRINT
270 CALL COLOR("0bC"):PRINT "DEUXIEME PHRASE"
280 CALL COLOR("0BC"):LINPUT B$
290 PRINT
300 CALL COLOR("0bC"):PRINT "TROISIEME PHRASE"
310 CALL COLOR("0BC"):LINPUT C$
320 PRINT
330 CALL COLOR("0bC"):PRINT "QUATRIEME PHRASE"
340 CALL COLOR("0BC"):LINPUT D$
350 PRINT
360 CALL COLOR("0bC"):PRINT "CINQUIEME PHRASE"
370 CALL COLOR("0BC"):LINPUT E$
380 CLS:CALL HRON("C",4,1):CALL LINOF
390 LOCATE (2,6):CALL COLOR("0bLH")
400 PRINT "AAUUTTOODDIICCTTEEEE":LOCATE (3,6):PRINT "A
AUUTTOODDIICCTTEEEE"
410 CALL COLOR("0BC")
420 V$=A$
430 T=0
440 X=0
450 LINPUT "QUEL EST TON PRENOM ?";Z$
460 GOSUB 630
470 V$=B$:GOSUB 630
480 V$=C$:GOSUB 630
490 V$=D$:GOSUB 630
500 V$=E$:GOSUB 630
510 CALL HROFF:CALL LINOF
520 CALL COLOR("0RLH"):FOR L=10 TO 11
530 LOCATE (L,4):FOR K=1 TO LEN(Z$)
540 PRINT RPT$(SEG$(Z$,K,1),2);
```

```
550 NEXT K:NEXT L:CALL COLOR("0BC"):LOCATE (15,1)
560 IF X>1 THEN PRINT "TON SCORE TOTAL EST DE";X;"FAUT
ES"
570 IF X<=1 THEN PRINT "TON SCORE TOTAL EST DE";X;"FAU
TE"
580 IF X<5 THEN PRINT "C'EST TRES BIEN"
590 IF X>=5 AND X<=10 THEN PRINT "C'EST UN BON RESULTA
T"
600 IF X>10 THEN PRINT "IL FAUDRA ETRE PLUS ATTENTIF L
A PROCHAINE FOIS"
610 PRINT:PRINT "APPELLE UN CAMARADE","POUR FAIRE L'EX
ERCICE"
620 PAUSE 10:GOTO 380
630 CLS
640 M=0
650 T=0
660 PRINT "TU VAS LIRE CETTE PHRASE":PRINT "LORSQUE TU
LA CONNAITRAS BIEN,"
670 PRINT "APPUIE SUR 'C'"
680 LOCATE (12,1):CALL COLOR("0bB")
690 PRINT V$&RPT$(" ",80-LEN(V$));
700 CALL COLOR("0BC")
710 CALL KEY1(A,B):IF A<>67 OR B<>1 THEN 710
720 X=X+T
730 T=0
740 IF M=3 THEN 950
750 CLS:LOCATE (8,1)
760 CALL COLOR("0RB"):PRINT "TAPE LE TEXTE,SANS OUBLIE
R LA PONTUATION"
770 PRINT "      ET APPUIE SUR LA TOUCHE 'PRET'      "
780 LOCATE (19,1)
790 CALL COLOR("0BC"):LINPUT M$
800 CALL COLOR("0RC"):LOCATE (8,1):PRINT RPT$(" ",80)
810 IF LEN(M$)<>LEN(V$)THEN GOSUB 1020:GOTO 740
820 FOR I=1 TO LEN(V$)
830 CALL COLOR("0bC"):LOCATE (12+INT(I/40),I+1-40*INT(
I/40))
840 IF SEG$(M$,I,1)=SEG$(V$,I,1)THEN PRINT SEG$(M$,I,1
):GOTO 860
850 GOSUB 1150
860 NEXT I
870 M=M+1:LOCATE (6,1)
880 IF T=0 THEN PRINT "TU N'AS FAIT AUCUNE FAUTE,":Z$:
PRINT " C'EST TRES BIEN"
890 IF T=0 THEN CALL PLAY("DOREREDORE",5)
900 IF T=1 AND M<3 THEN PRINT "TU AS FAIT";T;"FAUTE.":GOTO 920
910 IF T>1 AND M<3 THEN PRINT "TU AS FAIT";T;"FAUTES."
920 IF T>=1 AND M<3 THEN PRINT "TU VAS POUVOIR RECOMME
NCER LE TEXTE."
```

```

930 PAUSE 5
940 IF T>0 AND M<3 THEN 720
950 IF T>=1 AND M=3 THEN PRINT "VOICI LE TEXTE EXACT :
  :CALL COLOR("0RB")
960 IF T>=1 AND M=3 THEN PRINT V$;RPT$(" ",80-LEN(V$))
  :CALL COLOR("0BC")
970 IF T>1 AND M=3 THEN PRINT "TU AS FAIT";T;"FAUTES."
980 IF T=1 AND M=3 THEN PRINT "TU AS FAIT UNE FAUTE."
990 X=X+T
1000 PAUSE 6
1010 RETURN
1020 PRINT "TU AS FAIT UNE ERREUR : TON TEXTE N'A"
1030 PRINT "PAS LE MEME NOMBRE DE LETTRES QUE LE"
1040 PRINT "PREMIER. IL FAUT RECOMMENCER."
1050 K=LEN(V$)-LEN(M$)
1060 IF K>1 THEN PRINT "IL MANQUE";K;"LETTRES DANS TON
  TEXTE."
1070 IF K=1 THEN PRINT "IL MANQUE UNE LETTRE DANS TON T
  EXTE."
1080 IF K<-1 THEN PRINT "TON TEXTE A";-K;"LETTRES DE TR
  OP."
1090 IF K=-1 THEN PRINT "TON TEXTE A UNE LETTRE DE TROP
  "
1100 PAUSE 5
1110 T=T+1
1120 M=M+1
1130 RETURN
1140 CALL HROFF:CLS "BCC":END
1150 CALL COLOR("0RR"):PRINT " ";:CALL COLOR("0RC"):T=T
  +1
1160 CALL PLAY("SIDO",25):RETURN
1170 SUB INIT:CALL HROFF:CALL LINOF
1180 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)
1190 SUBEND
1200 SUB LINOF
1210 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
  50850):SUBEND
1220 SUB PLAY(NO$,L)
1230 FOR I=1 TO LEN(NO$)STEP 2
1240 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)
1250 PER=VAL(SEG$("199176158149133118105",K,3))
1260 CALL POKE(258,PER,132):PAUSE L/50:CALL POKE(259,45
  )
1270 NEXT:SUBEND

```

# **SOUTIEN**

## **Thème**

Jeu de questions à réponses multiples (deux séries de quatre questions), les sujets de ces questions-réponses sont entrés par l'adulte en début de programme.

## **Niveau**

A partir de 6 ans – toutes matières.

## **Intérêt pédagogique**

### **Connaissances**

- jeu de contrôle ;
- exercice de révision de notions déjà abordées. Vérification des acquis.

### **Savoir-faire**

- choix entre plusieurs réponses (élimination des réponses déjà données).

### **Déroulement**

L'adulte (enseignant ou parent) va d'abord entrer les huit courtes questions et leurs réponses. Ces éléments sont groupés en deux séries de quatre.

Ce programme permet d'aborder tous les domaines :

- **Français** : grammaire, conjugaison, vocabulaire, orthographe, etc.
- **Maths** : opérations, calcul mental.
- **Eveil** : évaluation des connaissances en histoire, en géographie, sciences, etc.

Ensuite, l'enfant va travailler sur ces données.

Quatre réponses sont inscrites, chacune précédée d'un numéro. Une question apparaît, le joueur doit entrer le numéro de la réponse qu'il pense être juste. Une musique lui indique son résultat. En cas d'erreur, la bonne réponse est affichée.

Le prénom du joueur et son score sont affichés en haut de l'écran.

Il en va de même pour les trois questions suivantes. Ensuite, une nouvelle série de quatre réponses apparaît, et le jeu continue.

A la fin, le score total est indiqué et commenté. Le joueur est invité à aller chercher un camarade pour le remplacer.

## Commentaires

<b>140 à 230</b>	Présentation du programme.
<b>240 à 480</b>	Entrée des données.
<b>490 à 560</b>	Présentation joueur.
<b>570 à 710</b>	Jeu.
<b>720 à 840</b>	Score – fin de jeu.
<b>850 à 870</b>	Réponse exacte.
<b>880 à 920</b>	Réponse fausse.
<b>930 à 1010</b>	Place aléatoire des réponses.
<b>1020 à 1150</b>	Gestion réponse.
<b>1170 à 1200</b>	SUB INIT.
<b>1210 à 1260</b>	SUB PLAY (joue un air de musique).

## Particularités techniques

Afin d'éviter une accoutumance à ce logiciel (au bout de trois ou quatre utilisations, l'enfant sait vite que la réponse à la 1<sup>re</sup> question est celle de la 2<sup>e</sup> ligne, et ainsi de suite), un nombre aléatoire détermine la place des réponses.

Dans sa version actuelle, le programme ne doit pas être interrompu (par l'appui de CTL puis C) sous peine de devoir retaper les questions et les réponses.

Enfin, ces données ne peuvent être sauvegardées sur cassette.

## Modifications

Les commentaires du score final (lignes 770 à 790) ainsi que la ligne 820 peuvent être considérés comme trop rigoureux si ce programme est utilisé à la maison.

## Adaptations

Un tel programme peut aussi être utilisé entre adultes sous forme de jeu pour tester les connaissances de vos amis sur les résultats du tournoi des cinq nations ou des Jeux Olympiques !

Si vous voulez éviter d'avoir à taper les données lors de chaque nouveau chargement, il faut supprimer les lignes 240 à 400 et entrer :

240 Q\$(1)=« votre question »  
 250 R\$(1)=« sa réponse »  
 260 etc...

Vous devrez ensuite sauvegarder sur cassette ce nouveau programme.

Si les instructions d'entrées-sorties de fichiers n'ont plus de secret pour vous, vous pouvez modifier ces lignes afin d'avoir, à côté du programme de base « soutien », un fichier sur cassette, disquette ou cartouche CRAM contenant différentes séries de questions-réponses.

D'autres séries de quatre questions et réponses peuvent être ajoutées : il suffit de modifier la ligne 670 :

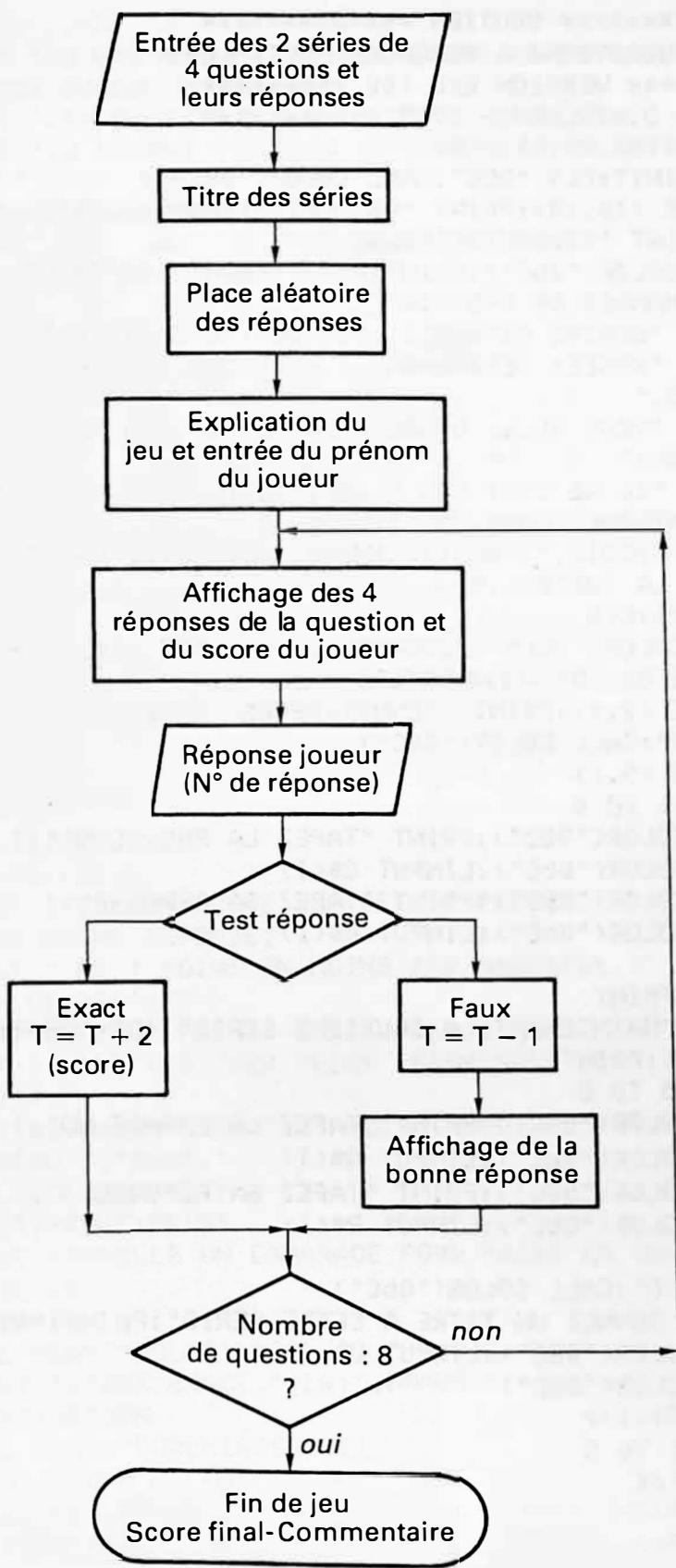
670 IF JEU=13 ... (*exemple pour quatre séries*)

et de redimensionner les tableaux Y,Q\$ et R\$ à la ligne 140.

Il faut aussi prévoir d'entrer ces nouvelles séries en début de programme, ce qui devient très long ! En fait, il n'est intéressant d'ajouter des séries que si l'on travaille à partir d'un fichier de questions enregistrées.

Enfin, on peut donner une nouvelle possibilité de réponse en cas d'erreur, en modifiant les lignes 600 à 660.

Tel qu'il est présenté, ce programme a été conçu pour être utilisé par des enseignants n'ayant aucune notion de programmation.



```
100 !***** SOUTIEN *****
110 !*** QUESTIONS A REPONSES MULTIPLES
120 !***** VERSION EXL 100 *****
130 !**** D.NIELSEN - 1985 *****
140 DIM Q$(8),R$(8),Y(8)
150 CALL INIT:CLS "BCC":CALL COLOR("ORLH")
160 LOCATE (10,10):PRINT "SSOOUUTTIIEENN":LOCATE (11,1
0):PRINT "SSOOUUTTIIEENN"
170 CALL COLOR("0bc"):PRINT:PRINT:PRINT "CE PROGRAMME
VA DEMANDER DE CHOISIR"
180 PRINT "QUATRE REPONSES POUR QUATRE QUESTIONS"
190 PRINT "POSEES SEPAREMENT. IL Y A DEUX SERIES DE QUE
STIONS."
200 PRINT "VOUS ALLEZ D'ABORD ENTRER LES DONNEES. ATT
ENTION :"
210 PRINT "IL NE DOIT PAS Y AVOIR DEUX REPONSES":PRINT
"IDENTIQUES DANS UNE ";
220 PRINT "SERIE.":PRINT:PRINT:PRINT "TAPEZ UNE TOUCH
POUR LA SUITE..."
230 A$=KEY$:CLS
240 CALL COLOR("0blh"):LOCATE (1,1):PRINT "EENNTRREEE
E DDEESS DDOONNNNEEEESS"
250 LOCATE (2,1):PRINT "EENNTRREEE DDEESS DDOONNNN
EEEESS":CALL COLOR("0bc")
260 LOCATE (5,1)
270 FOR I=1 TO 4
280 CALL COLOR("0bc"):PRINT "TAPEZ LA PHRASE NO";I
290 CALL COLOR("0bc"):LINPUT Q$(I)
300 CALL COLOR("0bc"):PRINT "TAPEZ SA REPONSE"
310 CALL COLOR("0bc"):LINPUT R$(I)
320 NEXT I
330 PRINT:PRINT
340 PRINT "MAINTENANT, LA DEUXIEME SERIE", "DE QUESTIONS
":PRINT:PRINT
350 FOR I=5 TO 8
360 CALL COLOR("0bc"):PRINT "TAPEZ LA PHRASE NO";I
370 CALL COLOR("0bc"):LINPUT Q$(I)
380 CALL COLOR("0bc"):PRINT "TAPEZ SA REPONSE"
390 CALL COLOR("0bc"):LINPUT R$(I)
400 NEXT I
410 CLS "BCC":CALL COLOR("0bc")
420 PRINT "DONNEZ UN TITRE A CETTE SERIE":PRINT:PRINT
430 CALL COLOR("0bc"):LINPUT U$
440 CALL COLOR("0bc")
450 C=LEN(Q$(1))
460 FOR I=1 TO C
470 RANDOMIZE
480 NEXT I
490 L,M,N,P=0
```

```
500 CLS
510 PRINT "TU VAS AVOIR LE CHOIX ENTRE QUATRE", "REPONSES POUR CHAQUE QUESTION."
520 PRINT "LES QUESTIONS SONT ECRITES EN VERT."
530 PRINT "TU DEVRAS INDIQUER LE NUMERO DE LA BONNE REPONSE."
540 PRINT:PRINT:PRINT
550 LINPUT "QUEL EST TON PRENOM ?"; Z$
560 CLS
570 T,V=0
580 JEU=1:GOSUB 930
590 !1ere serie
600 FOR I=JEU TO JEU+3
610 GOSUB 1030
620 CALL COLOR("0GB"):LOCATE (18,1):PRINT Q$(I):CALL COLOR("0BC")
630 PRINT:PRINT:INPUT "TA REPONSE ?";REP:REP=REP+JEU-1

640 IF Y<REP)=I THEN GOSUB 850 ELSE GOSUB 880
650 PAUSE 5
660 CLS:NEXT I
670 IF JEU=5 THEN 720
680 !2eme serie
690 JEU=JEU+4
700 GOSUB 930
710 GOTO 600
720 !fin de jeu
730 LOCATE (12,1)
740 PRINT Z$;" ,TON SCORE EST DE";T;"POINTS.", "(2 POINTS PAR BONNE REPONSE,"
750 PRINT " ET 1 POINT EN MOINS PAR MAUVAISE.)"
760 CALL COLOR("0RB")
770 IF T>=15 THEN PRINT "C'EST TRES BIEN."
780 IF T<15 AND T>8 THEN PRINT "BIEN,MAIS SOIS PLUS ATTENTIF."
790 IF T<=8 THEN PRINT "IL FAUDRA REVOIR LE TRAVAIL CORRESPONDANT."
800 CALL COLOR("0BC")
810 PRINT:PRINT:PRINT
820 PRINT "APPELLE UN CAMARADE POUR FAIRE CE JEU"
830 PAUSE 15
840 GOTO 450
850 CALL PLAY("SOLASILASO",2)
860 PRINT "C'EST EXACT,";Z$;";BRAVO."
870 T=T+2:RETURN
880 CALL PLAY("DODOMISOSO",2)
890 PRINT "TU N'AS PAS TROUVE,";Z$
900 PRINT "LA BONNE REPONSE ETAIT :";:CALL COLOR("0RB")
):PRINT R$(I)
```

```
910 T=T-1
920 CALL COLOR("0bC"):PAUSE 5:RETURN
930 !place des reponses
940 FOR I=JEU TO JEU+3
950 Y(I)=INTRND(4)+JEU-1
960 IF I=JEU THEN 1000
970 FOR K=JEU TO I-1
980 IF Y(I)=Y(K)THEN 950
990 NEXT K
1000 NEXT I
1010 RETURN
1020 !affichage des reponses
1030 CLS:LOCATE (1,1):CALL COLOR("0bC"):PRINT U$
1040 PRINT:PRINT TAB(10);Z$;" =";T;"POINT";
1050 IF T<>0 AND T<>1 THEN PRINT "S" ELSE PRINT
1060 FOR K=1 TO 4
1070 LOCATE (K*2+5,10)
1080 AF=K+JEU-1
1090 CALL COLOR("0bC")
1100 PRINT R$(Y(AF))&" ?"
1110 CALL COLOR("0"&SEG$("RbMG",K,1)&"B")
1120 LOCATE (K*2+5,7):PRINT K;
1130 NEXT K
1140 CALL COLOR("0bC")
1150 RETURN
1160 END
1170 SUB INIT
1180 CALL POKE(50580,162,5,45,162,136,45,10):CALL EXEC(
      50580)
1190 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

1200 SUBEND
1210 SUB PLAY(NO$,L)
1220 FOR I=1 TO LEN(NO$)STEP 2
1230 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)
1240 P=VAL(SEG$("199176158149133118105",K,3))
1250 CALL POKE(258,P,132):PAUSE L/50:CALL POKE(259,45)
1260 NEXT:SUBEND
```

# **SAFARI**

## **Thème**

Exercice d'entraînement au calcul rapide sous forme de jeu.

## **Niveau**

A partir de 6 ans.

## **Intérêt pédagogique**

### **Connaissances**

- vérification de la connaissance des tables d'addition ou de multiplication et du mécanisme opératoire ;
- entraînement au calcul mental.

### **Savoir-faire**

- rapidité de calcul ;
- habileté de commande au clavier.

## **Déroulement**

C'est une course de vitesse avec un lion. « Tu es sorti de ta voiture pour photographier un lion endormi. Il vient de se réveiller et te regarde. Tu vas devoir répondre aux opérations pro-

posées. Si tu réponds bien, tu avances vers ta voiture. Si tu te trompes, ou si tu ne réponds pas assez vite, le lion vient vers toi pour te manger. »

A chaque bonne réponse, le temps de réponse diminue.

Le jeu s'arrête lorsque le joueur a regagné sa voiture ou a été mangé par le lion.

## Commentaires

- |                    |   |
|--------------------|---|
| <b>130 à 350</b>   | Présentation du jeu.  |
| <b>360 à 420</b>   | Niveau du jeu.  |
| <b>440 à 470</b>   | Le nombre de lettres du prénom du joueur est utilisé pour initialiser les nombres aléatoires. |
| <b>480 à 700</b>   | Affichage du jeu et test des réponses.  |
| <b>710 à 770</b>   | Réponse fausse. Le lion se rapproche (770).   |
| <b>780</b>         | Fin si le lion a gagné.   |
| <b>790 à 840</b>   | Continuer la partie.  |
| <b>850 à 880</b>   | Son spécial si le lion est proche ou s'il gagne.  |
| <b>890 à 960</b>   | Bonne réponse.  |
| <b>970 à 1110</b>  | Fabrique les dessins du lion, de la voiture et du bonhomme.                                   |
| <b>1180 à 1280</b> | Fin de jeu « gagné ! ».   |
| <b>1290 à 1370</b> | Fin de jeu « perdu ! ».   |
| <b>1380 à 1420</b> | Affichage du rectangle chronomètre.   |
| <b>1430 à 1500</b> | Jeu chronométré.  |

Les trois niveaux de jeu correspondent à :

- 1 = Résultat < 9
- 2 = Résultat < 19
- 3 = Résultat < 39

## Particularités techniques

Le graphisme des trois dessins est réalisé par des caractères graphiques reprogrammés. Il est assez long à taper mais il pimente cet exercice de calcul mental d'une course contre la montre.

Le temps de réponse (variable S) diminue à chaque réponse exacte : 6 secondes au début, 3 secondes lorsque le joueur est presque arrivé à sa voiture. Il est visualisé par un chronomètre analogique qui se raccourcit progressivement.

## Modifications

- 480**                « Temps de réponse de base ». Il peut être allongé en ajoutant une ou des notes ligne 1480.
- 570 à 580**      {
- 600 à 610**      {
- 630 à 640**      }       Modifient les niveaux de jeu.

## Adaptations

En modifiant les lignes de niveau de jeu (570 à 640) et la ligne 1440, on peut facilement transposer ce jeu en des exercices de soustraction, de multiplication ou de division.

Exemple pour la multiplication :

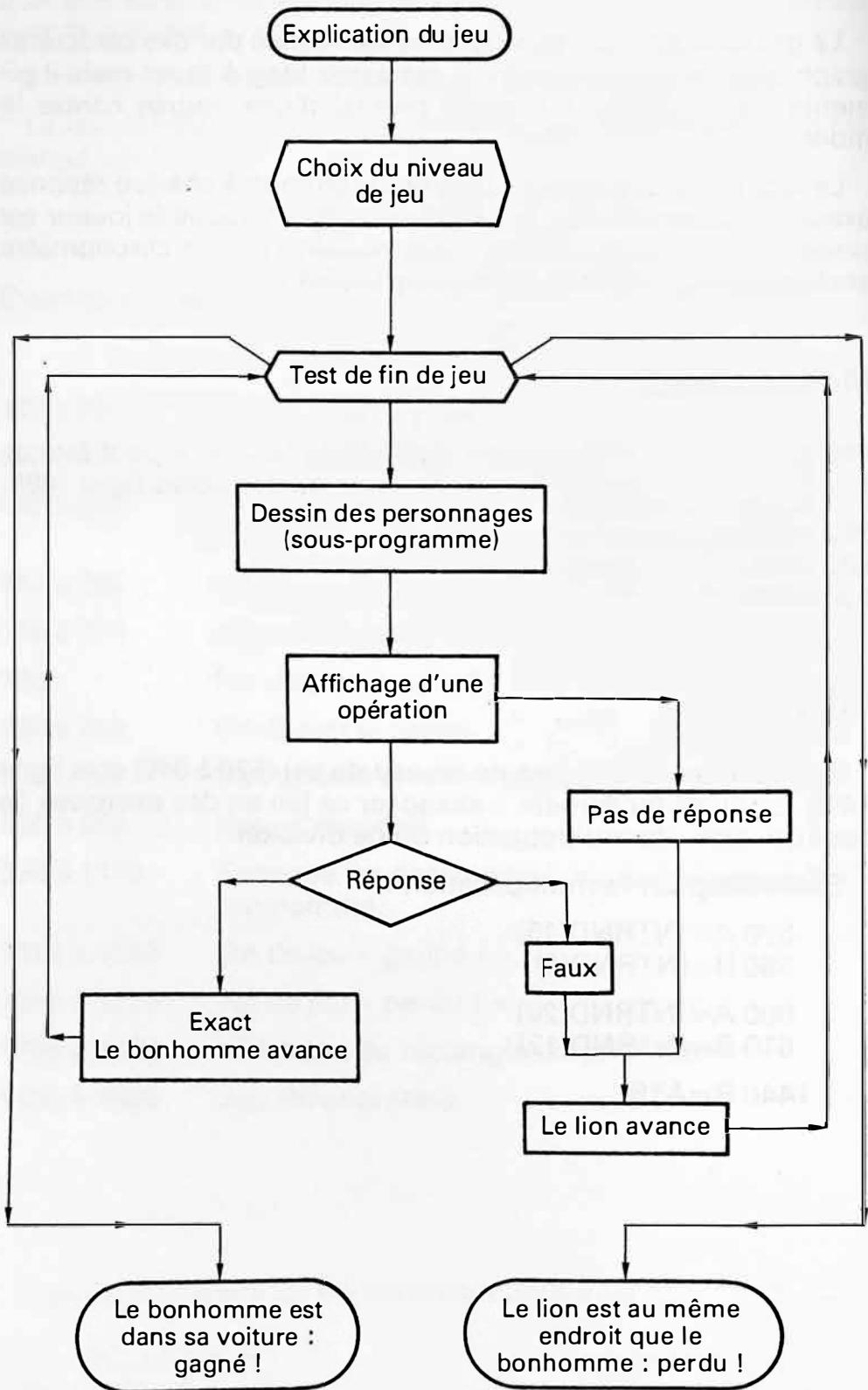
570 A=INTRND(15)

580 B=INTRND(9)

600 A=INTRND(20)

610 B=INTRND(12)

1440 R=A\*B ...



```
100 !***** SAFARI *****
110 !***** VERSION EXL 100 *****
120 !**** D.NIELSEN - 1985 *****
130 CALL INIT:CLS "BCC":CALL COLOR("0bHL")
140 LOCATE (6,14):PRINT "SSAAFFAARRII":LOCATE (7,14):P
RINT "SSAAFFAARRII"
150 CALL COLOR("0BC")
160 LOCATE (10,6):PRINT "COURSE DE VITESSE AVEC UN LIO
N":PRINT:PRINT
170 PRINT "Seule ta rapidite a trouver les reponses"
180 PRINT "aux operations proposees peut te sauver."
190 PRINT
200 CALL COLOR("0GB"):LOCATE (22,5):PRINT "Appuie une
touche pour continuer"
210 CALL KEY1(A,B):IF A=0 OR B<>1 THEN 210
220 CLS
230 P=19:M=1
240 CALL COLOR("1RC"):GOSUB 970
250 CALL COLOR("0BC"):PRINT "Tu es sorti de ta voiture
pour"
260 PRINT "photographier un lion endormi.":PRINT "Il v
ient de se reveiller."
270 PRINT "Tu vas devoir repondre aux","operations pro
posees."
280 PRINT "Si tu reponds bien,tu avances","vers ta voi
ture."
290 PRINT "Si tu te trompes,le lion","vient vers toi p
our te manger."
300 CALL COLOR("0RB"):LOCATE (18,15):PRINT "ATTENTION:
"
310 PRINT "A chaque operation,le temps de reponse":LOC
ATE (20,15)
320 PRINT "diminue."
330 CALL COLOR("0BW"):PRINT:PRINT "APPUIE UNE TOUCHE P
OUR COMMENCER";
340 CALL KEY1(A,B):IF A=0 OR B<>1 THEN 340
350 CALL COLOR("0BC"):CLS "BCC"
360 !**** JEU *****
370 LOCATE (5,1):PRINT "NIVEAU DE JEU"
380 PRINT:PRINT "TRES FACILE ..... 1"
390 PRINT:PRINT " FACILE ..... 2"
400 PRINT:PRINT "PLUS DIFFICILE .. 3"
410 PRINT
420 PRINT "Ton choix ..... ";ACCEPT VALIDATE("123")
SIZE(1)NULL(1),CH
430 PRINT
440 PRINT "Quel est ton prenom ?";
450 ACCEPT VALIDATE(ALPHA)SIZE(15),N$
460 T=LEN(N$):FOR I=1 TO T
470 RANDOMIZE:NEXT I
480 S=38
```

```
490 V=0
500 CALL COLOR("1RC"):CLS:GOSUB 1120
510 GOSUB 1390
520 IF M=P-2 THEN GOSUB 1300
530 IF P=37 THEN GOSUB 1180
540 V=V+1
550 CALL COLOR("0")
560 ON CH GOTO 570,600,630
570 A=INT(9*RND)
580 B=INTRND(9-A)
590 GOTO 660
600 A=INT(19*RND)
610 B=INTRND(19-A)
620 GOTO 660
630 A=INT(39*RND)
640 B=INTRND(39-A)
650 GOTO 660
660 CALL COLOR("0bLH"):LOCATE (20,5):CALL DBL(A,Z$):PR
INT Z$;"++";
670 CALL DBL(B,Z$):PRINT Z$;"==";
680 LOCATE (21,5):CALL DBL(A,Z$):PRINT Z$;"++";
690 CALL DBL(B,Z$):PRINT Z$;"=="
700 GOTO 1430
710 CALL COLOR("0RLH"):LOCATE (20,20):CALL DBL(REP,Z$)
:PRINT Z$
720 LOCATE (21,20):PRINT Z$
730 LOCATE (20,25):PRINT "FFAAUUXX !!!"
740 LOCATE (21,25):PRINT "FFAAUUXX !!!"
750 CALL COLOR("0RBF"):LOCATE (22,20):PRINT "ATTENTION
AU LION !";
760 CALL COLOR("0bC")
770 M=M+2
780 IF M>10 THEN 850
790 FOR I=1 TO 5
800 CALL PLAY("SILASOFAMIREDOLAMIDOLA",.4,5-I)
810 NEXT
820 CALL PLAY("LA",10,3)
830 PAUSE 5
840 GOTO 500
850 FOR I=1 TO 10
860 CALL PLAY("SOSODO",5,I):NEXT I
880 GOTO 500
890 CALL COLOR("0bLH"):LOCATE (20,20):CALL DBL(REP,Z$)
:PRINT Z$
900 LOCATE (21,20):PRINT Z$
910 LOCATE (20,25):PRINT "OOUFF !!!"
920 LOCATE (21,25):PRINT "OOUFF !!!"
930 P=P+2:S=S-2
940 CALL PLAY("SILASOFAMIREDOLAMIDOLA",5,2)
950 PAUSE 5
```

```

960 GOTO 500
970 RESTORE 970:FOR I=33 TO 43:READ A$:CALL CHAR(I,A$)
:NEXT
980 DATA 0000000003070F1F1F1F,00000000000000000000,000
000C0E0F0F8FCFEFF
990 DATA 070F1F3F6FCF8F8F8783,FFFFFFFFFFFF8F8F0,FFF
FFFFFFFFF070703
1000 DATA FFFEFCE0F0FCFEE0E0C0,01010100000000000000,E0E
0E0C0C0C000000000
1010 DATA 0303030101010000000,C0C0C080808000000000
1020 FOR I=44 TO 47:READ A$:CALL CHAR(I,A$):NEXT
1030 FOR I=58 TO 61:READ A$:CALL CHAR(I,A$):NEXT
1040 DATA 0000000000000000FFFF,000003060C183060E0FF,000
OFF381010101010FF
1050 DATA 0000F4606060607C7EFF,FFFFFFF1F0703010000,FFF
FFFFFFFFFF0E0C0800000
1060 DATA FFFFFFFF1F0703010000,FFFFFFF0E0C0800000
1070 FOR I=62 TO 63:READ A$:CALL CHAR(I,A$):NEXT:FOR I=
91 TO 95:READ A$
1080 CALL CHAR(I,A$):NEXT
1090 DATA 00000103070F0703017F,000080C0E0F0E0C080FE,7F6
76767670707070C0C
1100 DATA 000000000000000000000000
1110 DATA FEE6E6E6E0E0E03030,0C0C0C0C7C7C00000000,303
030303E3E00000000
1120 LOCATE (5,M):PRINT " !%"":LOCATE (6,M):PRINT "$%&"
":LOCATE (7,M)
1130 PRINT "()*+"
1140 LOCATE (6,35):PRINT ",-./":LOCATE (7,35):PRINT "::;
<="
1150 IF P=38 OR M=P-2 THEN RETURN
1160 LOCATE (5,P):PRINT ">?":LOCATE (6,P):PRINT "'$":LO
CATE (7,P):PRINT "'_"
1170 RETURN
1180 !GAGNE
1190 FOR I=10 TO 21:LOCATE (I,1):PRINT RPT$( " ",40);:NE
XT
1200 CALL COLOR("ORLH"):LOCATE (15,1):FOR I=1 TO LEN(N$)
)
1210 PRINT RPT$(SEG$(N$,I,1),2);:NEXT
1220 LOCATE (16,1):FOR I=1 TO LEN(N$)
1230 PRINT RPT$(SEG$(N$,I,1),2);:NEXT
1240 LOCATE (17,7):PRINT "CC'eesstt ggaagggnnee !!!"
1250 LOCATE (18,7):PRINT "CC'eesstt ggaagggnnee !!!"
1260 CALL COLOR("OR"):LOCATE (20,11)
1270 PRINT "En";V;"operations."
1280 GOSUB 1580:CALL HROFF:END
1290 PAUSE 5
1300 !PERDU
1310 FOR I=10 TO 21:LOCATE (I,1):PRINT RPT$( " ",40);:NE
XT

```

```

1320 CALL COLOR("0RLH"):FOR K=15 TO 16:LOCATE (K,5):FOR
    I=1 TO LEN(N$)
1330 PRINT RPT$(SEG$(N$,I,1),2);:NEXT:NEXT
1331 LOCATE (18,5):PRINT "Auuu rreevvooiirr"
1335 LOCATE (19,5):PRINT "Auuu rreevvooiirr "
1340 CALL COLOR("0b"):PRINT:PRINT
1350 PRINT "Tu as perdu, le lion t'a mangé(e)"
1360 CALL PLAY("FAMIFAMIFAREMIDOREREFASOFAMIREDOMIFAM
    IMIRERERERE",3,1)
1370 GOSUB 1580:CALL HROFF:END
1380 PAUSE 5
1390 !CHRONO
1400 LOCATE (15,1):PRINT CHR$(9);RPT$(CHR$(4),S);CHR$(6
    )
1410 PRINT CHR$(3);:LOCATE (16,S+2):PRINT CHR$(3)
1420 PRINT CHR$(8);RPT$(CHR$(4),S);CHR$(7):RETURN
1430 !JEU
1440 R=A+B:CALL COLOR("0I")
1450 LOCATE (16,2):FOR I=1 TO S
1460 PRINT " ";
1470 CALL KEY1(2,Y):IF Y<>0 AND (Z>47 AND Z<58)THEN REP
    $=CHR$(Z):GOTO 1510
1480 CALL PLAY("DODODO",2,1)
1490 NEXT I
1500 GOTO 750
1510 IF R>9 THEN 1530
1520 REP=VAL(REP$):IF REP=R THEN 890 ELSE 710
1530 D2=VAL(REP$)
1540 RP$=KEY$:IF RP$<"0"OR RP$>"9"THEN 1540
1550 UNIT=VAL(RP$)
1560 REP=D2*10+UNIT
1570 IF REP=R THEN 890 ELSE 710
1580 END
1600 SUB INIT
1610 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
    50850)
1620 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

1630 SUBEND
1640 SUB PLAY(N0$,L,D)
1650 FOR I=1 TO LEN(N0$)STEP 2
1660 K=POS("DO.RE.MI.FA.SO.LA.SI .",SEG$(N0$,I,2),1)
1670 P=VAL(SEG$("199176158149133118105",K,3))
1680 CALL POKE(258,P,D+128):PAUSE L/50
1690 NEXT:CALL POKE(259,0):SUBEND
1700 SUB DBL(A,Z$)
1710 K$=STR$(A):Z$=" "
1720 FOR I=1 TO LEN(K$)
1730 Z$=Z$&RPT$(SEG$(K$,I,1),2)
1740 NEXT I
1750 SUBEND

```

# **LECTURE RAPIDE**

## **Thème**

Test de lecture globale de mots : dix mots s'inscrivent rapidement (l'un après l'autre) puis douze mots s'affichent ensemble sur l'écran. Il faut trouver les deux intrus en répondant à l'aide du clavier ou de la manette de jeu.

## **Niveau**

A partir de 7 ans (fin de Cours élémentaire 1<sup>re</sup> année et au-delà).

## **Intérêt pédagogique**

### **Connaissances**

- reconnaissance globale de mots ;
- entraînement à la lecture silencieuse ;
- utilisation d'informations recueillies rapidement à la suite d'une activité.

### **Savoir-faire**

- mémoire visuelle ;
- durée d'attention ;
- déplacement dans les mots à l'aide de la manette.

## Déroulement

Après entrée du prénom du joueur, 10 mots vont s'inscrire l'un après l'autre, au centre de l'écran, en double taille pendant une seconde.

Puis une liste de 12 mots (les 10 précédents, plus 2 « intrus ») s'affiche dans des cadres. Le joueur déplace un curseur (qui provoque l'inversion des couleurs dans le cadre du mot visé) à l'aide des flèches ou de la manette, et appuie la barre espace ou le bouton de tir de la manette lorsqu'il pense avoir trouvé un intrus.

Le joueur doit ainsi indiquer les 2 intrus, en pointant leur case avec ce curseur. Au bout de trois erreurs, les deux mots à trouver sont affichés.

Le jeu reprend ensuite de la même manière avec trois autres listes de mots.

A la fin du jeu, le score de l'enfant (bonnes réponses) est affiché.

Les listes de mots de ce programme ont été établies après plusieurs essais de façon à tester l'aptitude des enfants à une lecture rapide et globale. Ces listes sont de difficulté croissante :

1. Noms d'animaux : 10 animaux de la ferme + 2 intrus (animaux d'Afrique).
2. Verbes : 10 verbes correspondant à des expressions du visage + 2 verbes de mouvement.
3. Nombres écrits en lettres : de 1 à 12.
4. Prénoms.

Dans les deux premières listes, une association d'idées peut guider l'enfant dans sa recherche ; dans les deux suivantes, il devra compter seulement sur sa mémoire visuelle.

## Commentaires

- |                  |   |
|------------------|---|
| <b>140 à 210</b> | Initialisation et présentation du jeu. Définition de séquences musicales.               |
| <b>220 à 310</b> | Affichage des 10 mots (les mots sont lus dans les lignes DATA par une boucle FOR NEXT). |
| <b>320 à 460</b> | Affichage des mots.   |
| <b>470 à 500</b> | Réponse (test du nombre de réponses puis vérification de la réponse).                   |

<b>510 à 540</b>	Réponse fausse.
<b>550</b>	Envoie au jeu suivant si les deux bonnes réponses sont trouvées.
<b>560</b>	Le programme va chercher en 950 les mots intrus selon le numéro de la liste (F) et les affiche.
<b>580</b>	Teste le nombre de jeux et envoie au jeu suivant.
<b>590 à 670</b>	Fin du jeu. Affichage du score et séquence musicale.
<b>680 à 890</b>	Liste des mots en DATA. Pour chaque jeu, il y a deux listes : – la liste des 10 mots ; – la liste des 12 mots dans un ordre différent.
<b>900 à 1140</b>	Déterminent pour chaque jeu les cases contenant les bonnes réponses.
<b>1150 à 1200</b>	Déterminent et affichent les deux « intrus », s'ils n'ont pas été trouvés tous les deux.
<b>1210 à 1240</b>	SUB INIT.
<b>1250 à 1300</b>	SUB PLAY ; joue un air musical.
<b>1310 à 1370</b>	SUB DBLPRT ; écrit en double taille.
<b>1380 à 1420</b>	SUB BOXPRT ; écrit dans un cadre.
<b>1430 à 1550</b>	SUB GETKEY ; assure la gestion du gros curseur.
<b>1560 à 1580</b>	SUB NORMAL ; retour en écriture normale.

## Particularités techniques

L'entrée des réponses est effectuée grâce au sous-programme spécialisé GETKEY. Il s'occupe de changer le curseur de place, d'éviter la sortie de l'écran, et d'afficher les couleurs des mots.

## Modifications

La temporisation d'affichage des 10 mots lus séparément est fixée à la ligne 290 (une seconde dans le listing). Elle peut être adaptée. Mais l'utilisation de ce programme avec de nombreux

enfants a montré que donner un temps plus long de lecture modifiait les buts de ce jeu (voir « lecture débutant »).

Les séquences musicales peuvent être retirées si l'ordinateur est utilisé dans une classe, pour éviter de perturber les voisins !

Afin de simplifier le programme, les listes de mots lues en DATA sont doubles, pour un jeu. On pourrait imaginer de lire en une seule fois les 12 mots, de tirer les 2 intrus au sort, et d'afficher les autres. Le jeu serait ainsi différent à chaque fois.

Il est très utile, dans le cadre d'une classe, de garder une trace écrite des résultats des enfants, bonnes réponses mais aussi erreurs, car on peut avoir 8 bonnes réponses sans erreur et 8 bonnes réponses après 8 réponses fausses.

Si l'on dispose d'une imprimante, il faut alors ajouter :

```
631 OPEN #1,« 20 »,OUTPUT
632 PRINT#1,P$,TR;« BONNES REPONSES »;RF;« REPONSES FAUSSES »
633 CLOSE#1:END
```

## Adaptations

Il est bien sûr possible de changer les listes de mots en DATA. C'est d'ailleurs nécessaire assez souvent car les enfants retiennent vite ces mots.

On peut envisager de faire travailler les enfants sur des types de mots bien précis :

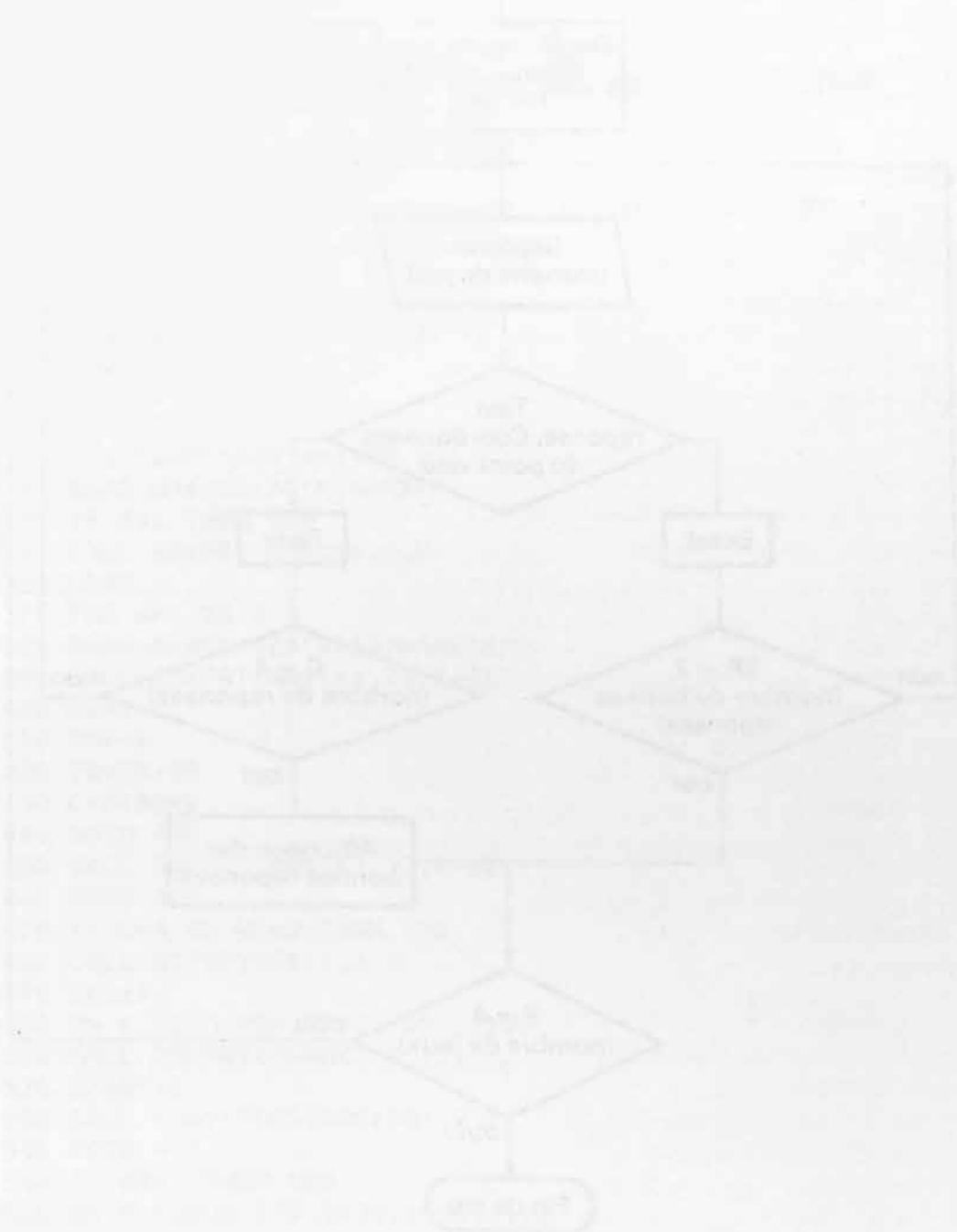
- pronoms (possessifs, relatifs) ;
- déterminants ;
- verbes conjugués ;
- homophones ;
- listes de mots d'une dictée, d'un texte étudié... ;
- mots nouveaux qu'ils ne connaissent pas (vocabulaire) ;
- etc.

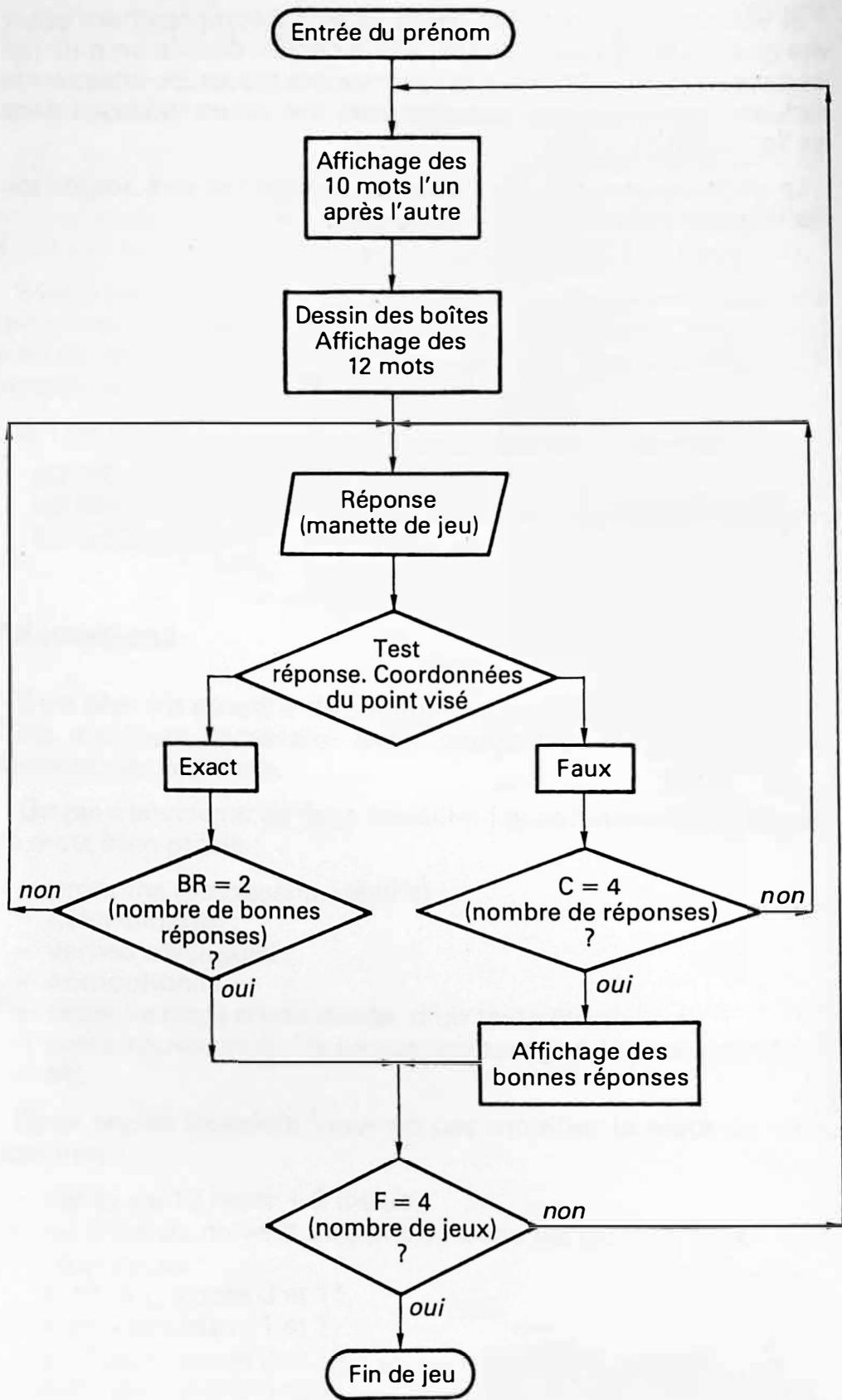
Deux règles toutefois (pour ne pas modifier le reste du programme) :

- séries de 10 mots + 2 intrus ;
- les 2 intrus doivent être placés dans les mêmes cases que pour ce jeu :
  - 1<sup>er</sup> jeu : cases 3 et 11,
  - 2<sup>e</sup> jeu : cases 1 et 7,
  - 3<sup>e</sup> jeu : cases 3 et 10,
  - 4<sup>e</sup> jeu : cases 6 et 8.

Si vos enfants sont des petits futés, ils comprendront assez vite que, quel que soit le mot, il faut choisir dans le premier jeu les cases 3 et 11, etc. Il vous faudra alors placer les intrus autre part, en changeant les coordonnées des cases-réponses dans les lignes 950 à 1140.

Le programme « lecture débutant » propose une adaptation réalisée pour de jeunes lecteurs (6 et 7 ans).





```
100 !***** TEST LECTURE RAPIDE *****
110 !***** VERSION EXL 10 *****
120 !***** D.NIELSEN - 1985 *****
130 !***** *****
140 DIM AA$(10),Z$(12)
150 CALL INIT:CLS "BCC":CALL NORMAL
160 B$="FAFAFASO":C$="LASO"
170 D$="FALASOSO":E$="FA"
180 CALL PLAY("DOREMIFASOLASIDO",3,1)
190 CALL COLOR("0bC"):CALL DBLPRT("TEST LECTURE RAPIDE
",1,7)
200 CALL COLOR("0RC"):LOCATE (12,7)
210 PRINT "Quel est ton prenom ?":ACCEPT BEEP VALIDAT
E(ALPHA),P$
220 F,RF=0
230 CLS:CALL COLOR("0b")
240 CALL PLAY("SOMISOSOFAMIREDOREMIMIREDOREDO",4,3)
250 NMOT=10
260 FOR I=1 TO NMOT
270 READ MOT$
280 CALL DBLPRT(MOT$,14,12)
290 PAUSE 1
300 CLS
310 NEXT I
320 CLS "bCC":FOR K=1 TO 10
330 READ AA$(K):Z$(K)=AA$(K)
340 IF K>6 THEN 450
350 CALL BOXPRT(AA$(K),3,K)
360 NEXT K
370 FOR K=1 TO 2
380 READ AA$(K):Z$(K+10)=AA$(K)
390 CALL BOXPRT(AA$(K),23,K+4)
400 NEXT K
410 F=F+1
420 TR=TR+BR
430 C=0:BR=0
440 GOTO 470
450 CALL BOXPRT(AA$(K),23,K-6)
460 GOTO 360
470 IF C=4 OR BR=2 THEN 550
480 CALL GETKEY(Z$(),X,Y)
490 C=C+1
500 ON F GOTO 950,1000,1050,1100
510 CALL DBLPRT("FAUX !",15,1)
520 RF=RF+1
530 CALL PLAY("DOSIDOSIDO",7,2)
540 GOTO 470
550 IF BR=2 THEN 580
560 ON F GOSUB 970,1020,1070,1120
```

```
570 PAUSE 5
580 IF F<4 THEN 230
590 TR=TR+BR
600 CLS:PRINT
610 CALL DBLPRT(P$&" TU AS",1,12)
620 CALL DBLPRT("TROUVE "&STR$(TR),1,14)
630 CALL DBLPRT("MOTS NOUVEAUX",1,16)
640 FOR I=1 TO 3
650 CALL PLAY(B$&C$&D$&E$,3,1)
660 NEXT
670 RUN
680 DATA LAPIN,CHEVAL,PIGEON,VACHE,COQ
690 DATA POULE,CANARD,ANE,MOUTON,DINDE
700 DATA DINDE,CANARD,SINGE,CHEVAL
710 DATA VACHE,MOUTON,POULE,COQ,LAPIN
720 DATA PIGEON,LION,ANE
730 DATA PLEURER,RIRE,SOUPIRER,CHANTER
740 DATA GRIMACER,SOUPIRER,GROGNER
750 DATA CRIER,SIFFLER,RICANER
760 DATA COURIR,CRIER,GRIMACER,SOUPIRER
770 DATA PLEURER,GROGNER,SAUTER,CHANTER
780 DATA RICANER,SIFFLER,RIRE,SOURIRE
790 DATA DEUX,SIX,UN,HUIT,QUATRE,CINQ
800 DATA NEUF,DOUZE,SEPT,ONZE
810 DATA UN,DEUX,TROIS,QUATRE,CINQ,SIX
820 DATA SEPT,HUIT,NEUF,DIX,ONZE,DOUZE
830 DATA JACQUES,PIERRE,ELISABETH
840 DATA ANNIE,CLAUDE,PHILIPPE,NATHALIE
850 DATA MARTINE,PASCAL,CHRISTINE
860 DATA PIERRE,NATHALIE,CLAUDE,ANNIE
870 DATA JACQUES,PAUL,CHRISTINE
880 DATA SANDRINE,MARTINE,ELISABETH
890 DATA PASCAL,PHILIPPE
900 CALL DBLPRT("BRAVO!",15,1)
910 CALL PLAY("SOFAMIFASO",1,1)
920 BR=BR+1
930 C=C+1
940 GOTO 470
950 IF Y=3 OR Y=11 THEN 900
960 GOTO 510
970 Y=3:GOSUB 1150
980 Y=11:GOSUB 1190
990 RETURN
1000 IF Y=1 OR Y=7 THEN 900
1010 GOTO 510
1020 Y=1:GOSUB 1150
1030 Y=7:GOSUB 1190
1040 RETURN
1050 IF Y=3 OR Y=10 THEN 900
```

```
1060 GOTO 510
1070 Y=3:GOSUB 1150
1080 Y=10:GOSUB 1190
1090 RETURN
1100 IF Y=6 OR Y=8 THEN 900
1110 GOTO 510
1120 Y=6:GOSUB 1150
1130 Y=8:GOSUB 1190
1140 RETURN
1150 CLS:CALL COLOR("0RC")
1160 PRINT "LES MOTS NOUVEAUX ETAIENT":CALL COLOR("0bB"
    )
1170 CALL BOXPRT(Z$(Y),3,4):CALL NORMAL
1180 RETURN
1190 CALL COLOR("0bB"):CALL BOXPRT(Z$(Y),23,4)
1200 CALL NORMAL:RETURN
1210 SUB INIT
1220 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
    50850)
1230 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

1240 SUBEND
1250 SUB PLAY(N0$,L,D)
1260 FOR I=1 TO LEN(N0$)STEP 2
1270 K=POS("DC.RE.MI.FA.SO.LA.SI.",SEG$(N0$,I,2),1)
1280 P=VAL(SEG$("199176158149133118105",K,3))
1290 CALL POKE(258,P,D+128):PAUSE L/50
1300 NEXT:CALL POKE(259,0):SUBEND
1310 SUB DBLPRT(Z$,X,Y)
1320 CALL COLOR("0LH")
1330 K$="":FOR I=1 TO LEN(Z$)
1340 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
1350 LOCATE (Y,X):PRINT K$
1360 LOCATE (Y+1,X):PRINT K$
1370 CALL COLOR("0"):SUBEND
1380 SUB BOXPRT(Z$,X,Y)
1390 LOCATE (Y*3,X):PRINT CHR$(9);RPT$(CHR$(4),11);CHR$(
    6)
1400 LOCATE (Y*3+1,X):PRINT CHR$(3);Z$:RPT$" ",11-LEN(
    Z$);CHR$(3)
1410 LOCATE (Y*3+2,X):PRINT CHR$(8);RPT$(CHR$(4),11);CH
    R$(7)
1420 SUBEND
1430 SUB GETKEY(Z$(),X,Y)
1440 X=3:Y=1
1450 CALL COLOR("0RB"):CALL BOXPRT(Z$(Y),X,Y+6*(Y>6))
1460 CALL KEY1(A,B)
1470 IF A=13 OR A=32 THEN 1550
1480 IF A<128 OR A>131 THEN 1460
```

```
1490 CALL COLOR("0bc"):CALL BOXPRT(Z$(Y),X,Y+6*(Y>6))
1500 IF A=128 AND Y>1 THEN Y=Y-1:IF Y=6 THEN X=X-20
1510 IF A=129 AND X<10 THEN X=X+20:Y=Y+6
1520 IF A=130 AND Y<12 THEN Y=Y+1:IF Y=7 THEN X=X+20
1530 IF A=131 AND X>10 THEN X=X-20:Y=Y-6
1540 GOTO .1450
1550 CALL COLOR("0bc"):CALL BOXPRT(Z$(Y),X,Y+6*(Y>6)):S
    UBEND
1560 SUB NORMAL
1570 CALL COLOR("0bc")
1580 SUBEND
```

# **LECTURE DEBUTANT**

## **Thème**

Acquisition de mots nouveaux ou recherche de mots connus sous forme de test de lecture rapide adapté aux plus jeunes.

## **Niveau**

Cours préparatoire et début de Cours élémentaire 1<sup>re</sup> année.  
A partir de 5 ans.

## **Intérêt pédagogique**

## **Connaissances**

- apprentissage de la lecture ;
- perception visuelle et globale de mots connus.

## **Savoir-faire**

- mémoire ;
- activités de discrimination visuelle (identifier et distinguer des signes graphiques différents).

## Déroulement

Le déroulement de ce programme est identique à celui de « lecture rapide ».

Voici les différences :

- Les listes ne sont formées que de 5 mots et 1 intrus.
- Chaque mot de la première liste est affiché durant trois secondes.
- Les mots sont entrés en minuscules (qui sont affichés en double taille) car, au début de l'apprentissage de la lecture, les enfants ne sont pas familiarisés avec le graphisme des lettres majuscules.

Les mots entrés en DATA ne sont qu'un exemple. Ils ont été définis en tout début d'année scolaire par des institutrices de Cours préparatoire, et doivent évoluer en cours d'année (voir « adaptations »).

Le choix des intrus effectué ici porte sur des mots nouveaux, inconnus des enfants, mais dont le graphisme est proche de celui d'un des mots de la première ligne (de-deux, papa-pépé, elle-élue, fait-four).

## Commentaires

Voir le thème « lecture rapide ». Le programme est organisé rigoureusement de la même manière. Seules différences :

- |                  |   |
|------------------|---|
| <b>250</b>       | Cinq mots au lieu de dix.                                   |
| <b>320 à 420</b> | Le dessin des six boîtes se fait en deux fois trois boîtes. |
| <b>520</b>       | Une seule bonne réponse, donc une seule zone à définir.     |

## Modifications

Elles sont les mêmes que pour « lecture rapide » :

- durée de la temporisation de l'affichage des mots de la première liste, ligne 290 ;
- nombre de réponses possibles : on peut considérer, surtout si ce programme sert à l'apprentissage de mots nou-

veaux, que l'enfant doit trouver de lui-même la bonne réponse, quel que soit le nombre de réponses.

Il faut alors supprimer les lignes :

430  
510  
570 à 590  
790 à 800  
830 à 840  
870 à 880  
910 à 960

et modifier :

500 GOTO 440  
760 IF F<4 THEN 230  
761 GOTO 560  
570 PRINT P\$;« AU REVOIR »

## Adaptations

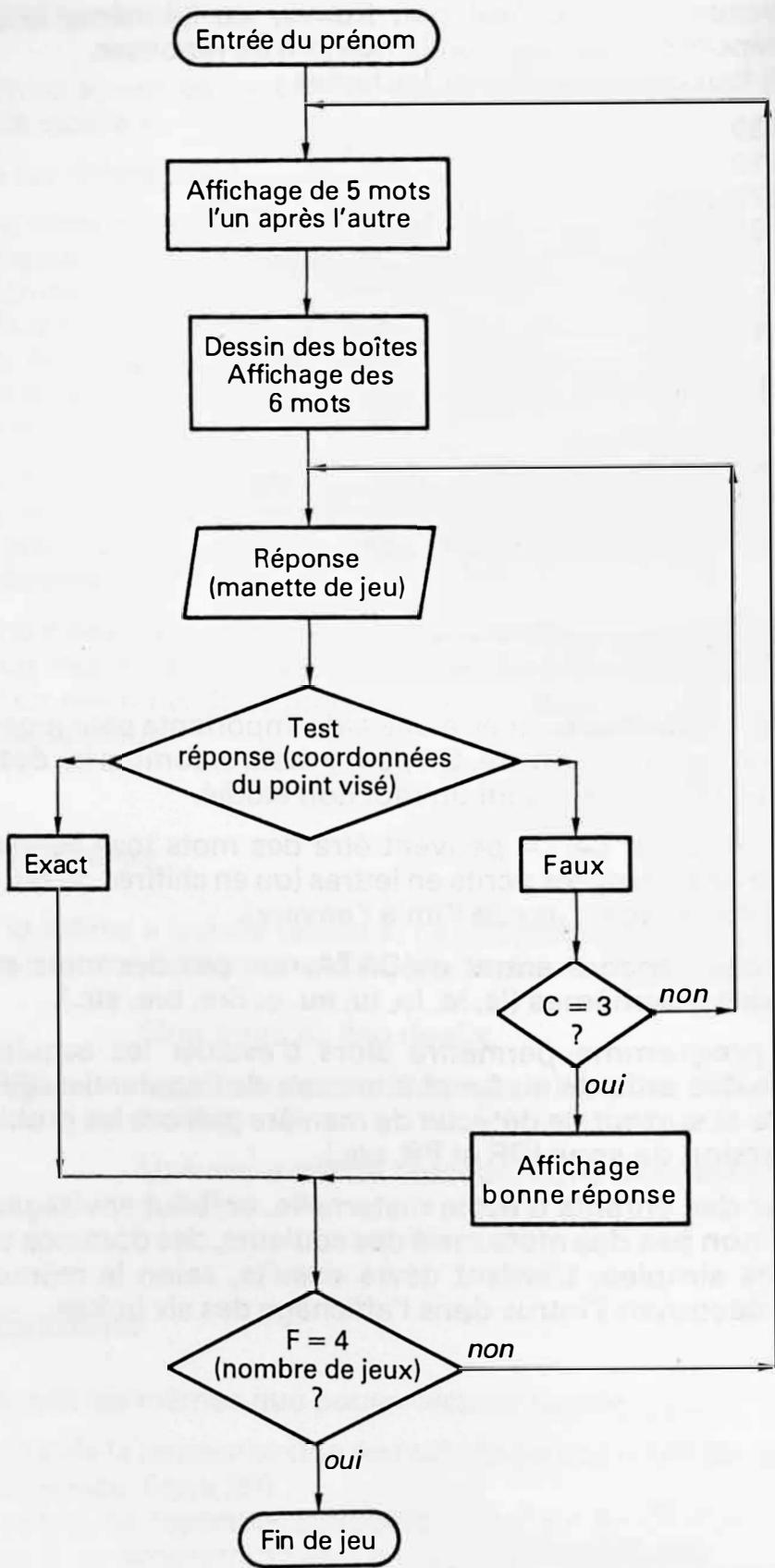
Ce programme peut être une aide importante pour juger l'apprentissage de la lecture. On peut y mettre, comme ici, des mots connus, l'intrus étant lui un mot non étudié.

Les mots en DATA peuvent être des mots tous connus, ou encore des nombres écrits en lettres (ou en chiffres, ce qui serait alors une sorte de jeu de Kim à l'envers).

On peut encore entrer en DATA non pas des mots entiers mais des graphèmes (le, la, lo, lu, au, o, dre, bre, etc.).

Ce programme permettra alors d'évaluer les acquisitions réelles des enfants au fur et à mesure de l'apprentissage de la lecture et surtout de détecter de manière précoce les problèmes d'inversion de sons (DR et PR, etc.).

Pour des enfants d'école maternelle, on peut envisager d'afficher non pas des mots mais des couleurs, des dominos ou des dessins simples. L'enfant devra ensuite, selon le même procédé, découvrir l'intrus dans l'affichage des six boîtes.



```
100 !***** TEST LECTURE DEBUTANT *****
110 !***** VERSION EXL 100 *****
120 !***** D.NIELSEN - 1985 *****
130 !*****
140 DIM AA$(6),Z$(6)
150 CALL INIT:CLS "BCC":CALL COLOR("0BC")
160 B$="FAFAFASO":C$="LASO"
170 D$="FALASOSO":E$="FA"
180 CALL PLAY("DOREMI FASOLASIDO",3,1)
190 CALL COLOR("0bC"):CALL DBLPRT("LECTURE DEBUTANT",1,7)
200 CALL COLOR("0RC"):LOCATE (12,7)
210 PRINT "Quel est ton prenom ?":ACCEPT BEEP VALIDATE(1,ALPHA),P$
220 F,RF=0
230 CLS:CALL COLOR("0b")
240 CALL PLAY("SOMISOSOFAMIREDOREMIMIREDOREDO",2,0)
250 NMOT=5
260 FOR I=1 TO NMOT
270 READ MOT$
280 CALL DBLPRT(MOT$,14,12)
290 PAUSE 3
300 CLS
310 NEXT I
320 CLS "bCC":FOR K=1 TO 6
330 READ AA$(K):Z$(K)=AA$(K)
340 IF K>3 THEN 410
350 CALL BOXPRT(AA$(K),3,K)
360 NEXT K
370 F=F+1
380 TR=TR+BR
390 C=0:BR=0
400 GOTO 430
410 CALL BOXPRT(AA$(K),23,K-3)
420 GOTO 360
430 IF C=3 OR BR=1 THEN 510
440 CALL GETKEY(Z$(),X,Y)
450 C=C+1
460 ON F GOTO 770,810,850,890
470 CALL DBLPRT("FAUX !",15,1)
480 RF=RF+1
490 CALL PLAY("DOSIDOSIDO",7,2)
500 GOTO 430
510 IF BR=1 THEN 540
520 ON F GOSUB 790,830,870,910
530 PAUSE 5
540 IF F<4 THEN 230
550 TR=TR+BR
560 CLS:PRINT
```

```

570 CALL DBLPRT(P$&" TU AS",1,12)
580 CALL DBLPRT("TROUVE "&STR$(TR),1,14)
590 CALL DBLPRT("MOTS NOUVEAUX",1,16)
600 FOR I=1 TO 3
610 CALL PLAY(B$&C$&D$&E$,3,1)
620 NEXT
630 RUN
640 DATA un,une,le,de,il
650 DATA un,le,deux,il,de,une
660 DATA maman,papa,petit,chat,fait
670 DATA papa,fait,chat,maman,pepe
680 DATA petit
690 DATA dans,est,il,avec,elle
700 DATA elue,avec,elle,il,dans,est
710 DATA fait,elle,de,est,et
720 DATA et,fait,de,four,est,elle
730 CALL DBLPRT("BRAVO!",15,1)
740 CALL PLAY("SOFAMIFASO",1,1)
750 BR=BR+1
760 GOTO 430
770 IF Y=3 THEN 730
780 GOTO 470
790 Y=3:GOSUB 930
800 RETURN
810 IF Y=5 THEN 730
820 GOTO 470
830 Y=5:GOSUB 930
840 RETURN
850 IF Y=1 THEN 730
860 GOTO 470
870 Y=1:GOSUB 930
880 RETURN
890 IF Y=4 THEN 730
900 GOTO 470
910 Y=4:GOSUB 930
920 RETURN
930 CLS:CALL COLOR("0RC")
940 PRINT "le mot nouveau etait":CALL COLOR("0bB")
950 CALL BOXPRT(Z$(Y),3,4):CALL COLOR("0BC")
960 RETURN
970 SUB INIT
980 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
      50850)
990 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

1000 SUBEND
1010 SUB PLAY(NO$,L,D)
1020 FOR I=1 TO LEN(NO$)STEP 2
1030 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)

```

```
1040 P=VAL(SEG$("199176158149133118105",K,3))
1050 CALL POKE(258,P,D+128):PAUSE L/50
1060 NEXT:CALL POKE(259,0):SUBEND
1070 SUB DBLPRT(Z$,X,Y)
1080 CALL COLOR("0LH")
1090 K$="" :FOR I=1 TO LEN(Z$)
1100 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
1110 LOCATE (Y,X):PRINT K$
1120 LOCATE (Y+1,X):PRINT K$
1130 CALL COLOR("0"):SUBEND
1140 SUB BOXPRT(Z$,X,Y)
1150 LOCATE (Y*4,X):PRINT CHR$(9);RPT$(CHR$(4),11);CHR$(6)
1160 LOCATE (Y*4+1,X):PRINT CHR$(3);Z$;RPT$(" ",11-LEN(Z$));CHR$(3)
1170 LOCATE (Y*4+2,X):PRINT CHR$(8);RPT$(CHR$(4),11);CHR$(7)
1180 SUBEND
1190 SUB GETKEY(Z$(),X,Y)
1200 X=3:Y=1
1210 CALL COLOR("0RB"):CALL BOXPRT(Z$(Y),X,Y+3*(Y>3))
1220 CALL KEY1(A,B)
1230 IF A=13 OR A=32 THEN 1310
1240 IF A<128 OR A>131 THEN 1220
1250 CALL COLOR("0bC"):CALL BOXPRT(Z$(Y),X,Y+3*(Y>3))
1260 IF A=128 AND Y>1 THEN Y=Y-1:IF Y=3 THEN X=X-20
1270 IF A=129 AND X<10 THEN X=X+20:Y=Y+3
1280 IF A=130 AND Y<6 THEN Y=Y+1:IF Y=4 THEN X=X+20
1290 IF A=131 AND X>10 THEN X=X-20:Y=Y-3
1300 GOTO 1210
1310 CALL COLOR("0bC"):CALL BOXPRT(Z$(Y),X,Y+3*(Y>3)):SUBEND
```



# **PLURIEL DES NOMS**

## **Thème**

Didact d'apprentissage ou de révision du pluriel des noms, utilisant les exceptions les plus courantes.

## **Niveau**

A partir du Cours élémentaire 1<sup>re</sup> année (7 ans).

## **Intérêt pédagogique**

### **Connaissances**

- vérification des acquis en grammaire et orthographe ;
- étude ou révision des mots faisant exception à la règle générale.

### **Savoir-faire**

- en cas d'erreur, lecture de la règle d'exception et application directe au mot demandé.

## **Déroulement**

Après enregistrement du prénom du joueur, l'écran affiche un nom au singulier, accompagné de son article indéfini, et demande le pluriel de ce nom. L'enfant doit entrer « des ... ».

Si la réponse est bonne, on passe à un autre nom (10 questions se succèdent ainsi avant la fin du jeu).

Si elle est fausse, deux cas se présentent :

- **Règle générale** : une autre réponse est demandée sans précision particulière.
- **Cas d'exception** : la règle d'accord du pluriel s'inscrit sur l'écran pendant 15 secondes.

L'enfant propose une autre réponse : le programme repart comme après la première réponse (exact ou faux).

## Commentaires

<b>140 à 210</b>	Présentation du programme et enregistrement du prénom du joueur.
<b>220 à 240</b>	Lecture des mots en DATA.
<b>250 à 320</b>	Tirage au sort d'un mot.
<b>330 à 410</b>	Affichage du mot choisi.
<b>420 à 550</b>	Vérification des exceptions.
<b>560 à 590</b>	Contrôle de la réponse (cas général).
<b>600 à 680</b>	Réponse exacte et fin du tour de jeu.
<b>690 à 890</b>	DATA des mots.
<b>900 à 1010</b>	Contrôle de la réponse pour les mots en S, Z et X.
<b>1020 à 1170</b>	Contrôle de la réponse pour les mots en AL.
<b>1180 à 1260</b>	Contrôle de la réponse pour les mots en OU.
<b>1270 à 1350</b>	Contrôle de la réponse pour les mots en AIL.
<b>1360 à 1390</b>	Fin du jeu.
<b>1400 à 1430</b>	SUB INIT.
<b>1440 à 1490</b>	SUB PLAY (joue un air de musique).
<b>1500 à 1560</b>	SUB DBLPRT (écrit en double taille).
<b>1570 à 1590</b>	SUB NORMAL (retour en écriture normale).
<b>1600 à 1640</b>	SUB UPPER (transforme minuscules en majuscules).

## Particularités techniques

Le nom proposé est choisi au hasard dans une liste en DATA. Il est ensuite comparé (lignes 280 à 310) aux autres noms déjà proposés afin que le même nom ne revienne pas plusieurs fois dans l'exercice.

Lorsque la réponse est donnée, le nom proposé est comparé aux différentes exceptions à la règle générale, afin de déterminer la bonne réponse.

Les exceptions sont inscrites en DATA, elles peuvent ainsi servir à la comparaison avec le nom donné (lignes 420 à 550) et aussi à l'affichage de la règle en cas de réponse erronée (instruction RESTORE pour pouvoir relire le bloc DATA).

## Modifications

Ligne 210 : pour éviter la répétition du même jeu pour une même personne (c'est utile en classe pour voir la progression d'un élève sur un exercice donné mais à la longue il faut renouveler les tirages au sort), on peut supprimer le paramètre apparaissant derrière RANDOMIZE (voir les conseils techniques généraux dans l'introduction).

On peut augmenter le nombre de mots à tirer au sort en DATA (lignes 690 à 890) en changeant seulement la valeur du paramètre de la fonction INTRND de la ligne 270.

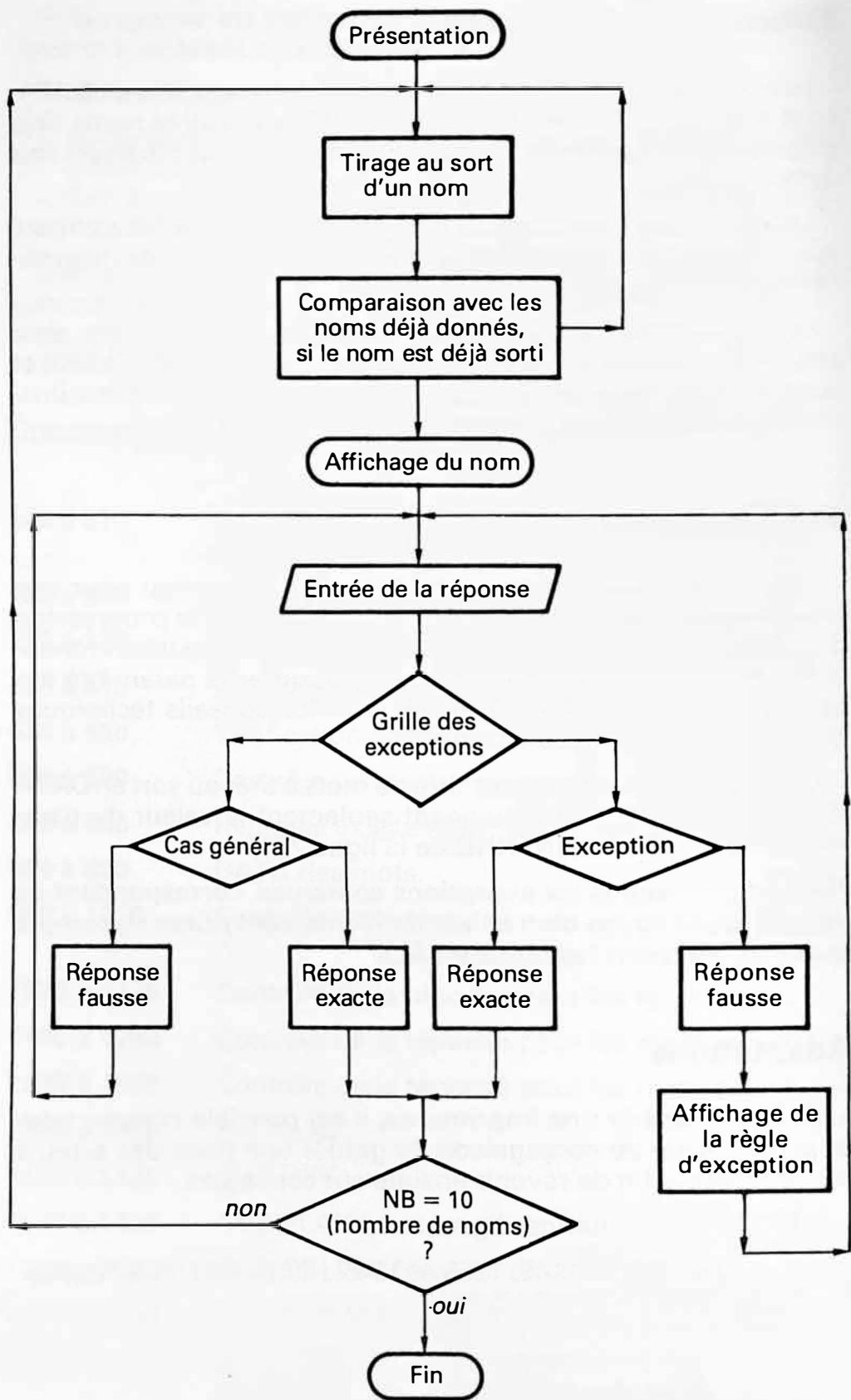
*Remarque :* seules les exceptions courantes, correspondant au vocabulaire moyen d'un enfant de 10 ans, sont prises en compte (notamment pour les noms en AL).

## Adaptations

Si l'on possède une imprimante, il est possible comme pour le programme de conjugaison de garder une trace des erreurs rencontrées, afin de revenir ensuite sur ces règles.

Il faut alors ajouter ces lignes :

```
585, 945, 1005, 1145, 1255 et 1345 : CALL PRT(PR$,N$,P$)
1700 SUB PRT(PR$,N$,P$)
1710 OPEN#1,« 20 »,OUTPUT
1720 PRINT #1,PR$,N$,P$
1730 CLOSE #1 :SUBEND
```



```
100 !***** LE PLURIEL DES NOMS *****
110 !***** VERSION EXL 100 *****
120 !***** D.NIELSEN - 1985 *****
130 !*****
140 DIM Z$(6),M$(30),A$(15),T(30)
150 CALL INIT:CLS "BCC":CALL NORMAL
160 NB=1:V=0
170 CLS "BCC":RESTORE
180 CALL DBLPRT("LE PLURIEL DES NOMS","ORB",1,12)
190 LOCATE (15,5):PRINT "Quel est ton prenom ?";
200 ACCEPT BEEP VALIDATE(ALPHA),PR$
210 RANDOMIZE LEN(PR$)
220 READ N$:IF N$="*"THEN 250
230 V=V+1:M$(V)=N$
240 GOTO 220
250 !*** TIRAGE AU SORT
260 NB=NB+1
270 FLG=0:T(NB)=INTRND(30)
280 IF NB=2 THEN 320
290 FOR I=1 TO NB-1
300 IF T(NB)=T(I)THEN FLG=1:I=NB
310 NEXT I:IF FLG=1 THEN 270
320 Z=T(NB):N$=M$(Z)
330 !*** PLURIEL
340 PAUSE 8:CLS:CALL DBLPRT(N$,"ORB",5,5)
350 LOCATE (10,5):PRINT "Ecris ce mot au pluriel"
360 CALL COLOR("0b"):PRINT "ATTENTION,N'OUBLIE PAS L'ARTICLE"
370 CALL NORMAL:LOCATE (15,5):LINPUT "? ";P$:CALL UPPE
R(P$)
380 L=LEN(N$):P=LEN(P$)
390 IF SEG$(P$,1,3)="DES"THEN 420
400 PRINT PR$;" Tu as oublié l'article au pluriel"
410 GOTO 330
420 !*** GRILLE DES EXCEPTIONS
430 IF SEG$(N$,L,1)="Z"OR SEG$(N$,L,1)="S"OR SEG$(N$,L
,1)="X"THEN 900
440 IF SEG$(N$,L-1,2)="EU"OR SEG$(N$,L-1,2)="AU"THEN 9
60
450 IF SEG$(N$,L-1,2)="AL"THEN 1020
460 RESTORE 800
470 FLG=0:FOR I=1 TO 7
480 READ EXC$
490 IF N$=EXC$THEN FLG=1:I=8
500 NEXT I:IF FLG=1 THEN 1180
510 RESTORE 830
520 FLG=0:FOR I=1 TO 8
530 READ EXC$
540 IF N$=EXC$THEN FLG=1:I=9
```

```

550 NEXT I:IF FLG=1 THEN 1270
560 !*** CONTROLE RESULTAT
570 IF SEG$(P$,P-2,3)=SEG$(N$,L-1,2)&"S"THEN 600
580 CALL COLOR("0RB"):PRINT "Tu te trompes,";PR$:PRINT
    "La regle est simple."
590 PRINT "Recommence.":CALL NORMAL:GOTO 330
600 !*** RESULTAT EXACT
610 PRINT:PRINT:CALL PLAY("DOMISO",5,1):CALL PLAY("DO"
    ,5,0):CALL PLAY("SO",10,1)
620 CALL PLAY("DO",7,0)
630 CALL COLOR("0Wb"):PRINT "C'est exact ,";PR$
640 CALL NORMAL:IF NB=11 THEN 1360
650 PRINT "Voici un autre mot"
660 RESTORE
670 N$=D$(NB)
680 GOTO 250
690 DATA UN CHAT,UN CHEVAL,UN CANARD
700 DATA UN CHEVEU,UN BAL,UN CHIEN
710 DATA UNE JOURNEE,UN PRIX
720 DATA UNE FOURRURE,UN TROU,UN BIJOU
730 DATA UN FOUR,UNE SOURIS,UN VITRAIL
740 DATA UN NEZ,UN CHACAL,UN OBJECTIF
750 DATA UNE NOIX,UN PINGOUIN,UN GAZ
760 DATA UN TRAVAIL,UN RELAIS,UN CHOU
770 DATA UNE SERRURE,UN PUIT,UN RECITAL
780 DATA UN JOUJOU,UNE TISANE,UN VERRE
790 DATA *
800 DATA UN BIJOU,UN JOUJOU,UN CHOU
810 DATA UN GENOU,UN HIBOU,UN CAILLOU
820 DATA UN POU
830 DATA UN VITRAIL,UN TRAVAIL,UN EMAIL
840 DATA UN BAIL,UN SOUPIRAIL
850 DATA UN CORAIL,UN EVENTAIL
860 DATA UN FERMAIL
870 DATA UN BAL,UN FESTIVAL,UN CARNAVAL
880 DATA UN REGAL,UN CEREMONIAL
890 DATA UN RECITAL,UN CHACAL
900 !*** MOTS EN S,Z,X
910 IF SEG$(N$,L-2,3)=SEG$(P$,P-2,3)THEN 600
920 PRINT "Les mots terminees au singulier par"
930 PRINT "S,Z ou X gardent la meme forme":PRINT "au p
luriel."
940 PRINT:PRINT "EXEMPLE : Un bois , des bois"
950 GOTO 330
960 !*** MOTS EN AU ET EU
970 IF SEG$(P$,P-2,3)=SEG$(N$,L-1,2)&"X"THEN 600
980 PRINT "Le mots terminees au singulier par AU ou EAU
    ou EU ou OEU prennent";
990 PRINT " un X":PRINT "au pluriel.":PRINT

```

```

1000 PRINT "EXEMPLE :Un adieu , des adieux"
1010 GOTO 330
1020 !*** MOTS EN AL
1030 RESTORE 870
1040 FLG=0:FOR I=1 TO 7
1050 READ EXC$
1060 IF N$=EXC$THEN FLG=1:I=8
1070 NEXT I:IF FLG=1 THEN 1160
1080 IF SEG$(P$,P-2,3)=SEG$(N$,L-1,1)&"UX"THEN 600
1090 RESTORE 870
1100 PRINT "Les mots terminees au singulier par AL"
1110 PRINT "ont un pluriel en AUX sauf :"
1120 FOR I=1 TO 6:READ EXC$:PRINT EXC$;" , ";
1130 NEXT I:READ EXC$:PRINT EXC$
1140 PRINT "qui ont un pluriel en ALS."
1150 GOTO 330
1160 IF SEG$(P$,P-2,3)=SEG$(N$,L-1,2)&"S"THEN 600
1170 GOTO 1090
1180 !*** MOTS EN OU
1190 IF SEG$(P$,P-2,3)=SEG$(N$,L-1,2)&"X"THEN 600
1200 RESTORE 800
1210 PRINT "Les mots suivants :"
1220 FOR I=1 TO 6:READ EXC$
1230 PRINT EXC$;" , ";
1240 NEXT I:READ EXC$:PRINT EXC$
1250 PRINT "prennent un X au pluriel."
1260 GOTO 330
1270 !*** MOTS EN AIL
1280 IF SEG$(P$,P-2,3)=SEG$(N$,L-2,1)&"UX"THEN 600
1290 RESTORE 830
1300 PRINT "Les mots suivants :"
1310 FOR I=1 TO 7
1320 READ EXC$:PRINT EXC$;" , ";
1330 NEXT I:READ EXC$:PRINT EXC$
1340 PRINT "ont un pluriel en AUX."
1350 GOTO 330
1360 !*** FIN
1370 CLS:CALL DBLPRT("C'EST FINI","0RB",5,12):CALL DBLP
    RT(PR$,"0YB",5,14)
1380 CALL DBLPRT("AU REVOIR !","0bB",5,16)
1390 END
1400 SUB INIT
1410 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
    50850)
1420 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)
1430 SUBEND
1440 SUB PLAY(NO$,L,D)
1450 FOR I=1 TO LEN(NO$)STEP 2
1460 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)

```

```
1470 P=VAL(SEG$("199176158149133118105",K,3))
1480 CALL POKE(258,P,D+128):PAUSE L/50
1490 NEXT:CALL POKE(259,0):SUBEND
1500 SUB DBLPRT(Z$,C$,X,Y)
1510 CALL COLOR(C$):CALL COLOR("OLH")
1520 K$="" :FOR I=1 TO LEN(Z$)
1530 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
1540 LOCATE (Y,X):PRINT K$
1550 LOCATE (Y+1,X):PRINT K$
1560 CALL NORMAL:SUBEND
1570 SUB NORMAL
1580 CALL COLOR("OBC")
1590 SUBEND
1600 SUB UPPER(Z$)
1610 K$="" :FOR I=1 TO LEN(Z$)
1620 S=ASC(SEG$(Z$,I,1)):IF S>96 THEN S=S-32
1630 K$=K$&CHR$(S):NEXT:Z$=K$
1640 SUBEND
```

# **CONJUGAISON**

## **Thème**

Didact d'apprentissage ou de révision des verbes du 1<sup>er</sup> et du 2<sup>e</sup> groupe au présent, imparfait, futur, passé simple de l'indicatif.

## **Niveau**

A partir du Cours élémentaire 2<sup>e</sup> année (8 ans).

## **Intérêt pédagogique**

### **Connaissances**

- vérification des acquis en conjugaison et révision des verbes-type du 1<sup>er</sup> et du 2<sup>e</sup> groupe.

### **Savoir-faire**

- en cas d'erreur, utilisation de la conjugaison-type affichée, reconnaissance du radical et de la terminaison pour formuler une nouvelle réponse au verbe demandé.

## **Déroulement**

Après enregistrement du prénom du joueur, l'écran affiche un verbe à l'infinitif et demande sa conjugaison à l'un des quatre temps et pour une personne donnée. L'enfant entre sa réponse.

Si elle est bonne, on passe à un autre verbe (10 questions se succèdent ainsi avant la fin de l'exercice).

Si elle est fausse, sur l'écran apparaît la conjugaison du verbe chanter ou du verbe finir au temps demandé et à toutes les personnes.

L'enfant doit ensuite proposer une autre réponse. Si celle-ci est encore mauvaise, la conjugaison-type est de nouveau affichée, ainsi de suite.

## Commentaires

- |                    |  |
|--------------------|--|
| <b>140 à 190</b>   | Présentation du jeu et saisie du prénom du joueur. |
| <b>200 à 240</b>   | Mémorisation des verbes des blocs DATA.            |
| <b>250 à 270</b>   | Mémorisation des personnes.                        |
| <b>280 à 300</b>   | Mémorisation des temps.                            |
| <b>310 à 400</b>   | Tirage au sort d'un verbe.                         |
| <b>410 à 580</b>   | Entrée de la réponse de l'enfant.                  |
| <b>590 à 610</b>   | Test de la réponse.                                |
| <b>620 à 750</b>   | Réponse fausse. Affichage de la conjugaison-type.  |
| <b>760 à 790</b>   | Nouvelle entrée de réponse.                        |
| <b>800 à 870</b>   | Bonne réponse, et test de suite du jeu.            |
| <b>880 à 910</b>   | Fin du jeu.  |
| <b>920 à 1020</b>  | DATA verbes.                                       |
| <b>1030 à 1080</b> | DATA personnes.                                    |
| <b>1090 à 1120</b> | DATA temps.  |
| <b>1130 à 1210</b> | DATA terminaisons.                                 |
| <b>1220 à 1250</b> | SUB INIT.  |
| <b>1260 à 1310</b> | SUB PLAY (air de musique).                         |
| <b>1320 à 1380</b> | SUB DBLPRT (écriture double taille).               |
| <b>1390 à 1410</b> | SUB NORMAL (écriture normale).                     |
| <b>1420 à 1560</b> | SUB UPPER (passage en majuscules).                 |

## Particularités techniques

Le choix du verbe (30), de la personne (6), et du temps (4) est obtenu au hasard à partir d'une séquence aléatoire initialisée par la longueur du prénom du joueur. Il y a donc 720 possibilités d'affichage différentes.

Les lignes 480 à 570 placent le pointeur de données DATA sur la ligne correspondant aux terminaisons du verbe et du temps du verbe demandés.

La conjugaison-type, en cas d'erreur, est affichée par une boucle FOR NEXT aux lignes 700 à 730.

## Modifications

Ligne 210 : on peut supprimer l'expression suivant RANDOMIZE afin que les verbes soient véritablement choisis au hasard et non plus en fonction de la longueur du prénom du joueur.

Lignes 220 à 240 : la lecture des verbes dans les DATA ne se fait pas grâce à une boucle FOR NEXT mais plutôt par une boucle du type WHILE WEND, jusqu'à ce que le pointeur trouve le symbole « \* ». De cette façon, on peut ajouter sans difficulté d'autres verbes (il faudra toutefois revoir les dimensionnements de tableaux de la ligne 140 si le nombre de verbes devient trop important). Il suffira de modifier le paramètre de la fonction INTRND de la ligne 320 pour reconnaître les nouveaux verbes dans le tirage au sort.

Attention toutefois aux verbes en GER (exemple : manger), en OYER, UYER, ELER et en ETER pour lesquels la conjugaison est irrégulière et qui nécessiteraient des sous-programmes particuliers.

Il n'y a pas de verbes commençant par une voyelle dans ce programme, mais vous pouvez en ajouter si toutefois vous prévoyez le cas de la première personne du singulier qui se nommera « J' » et non plus « JE ».

## Adaptations

On peut modifier les temps, faire par exemple un programme sur le subjonctif (plus de 10 ans), en changeant simplement les lignes de DATA 1090 à 1210.

Un programme sur les temps composés peut permettre de traiter aussi les verbes du 3<sup>e</sup> groupe, mais amènerait une modification plus profonde :

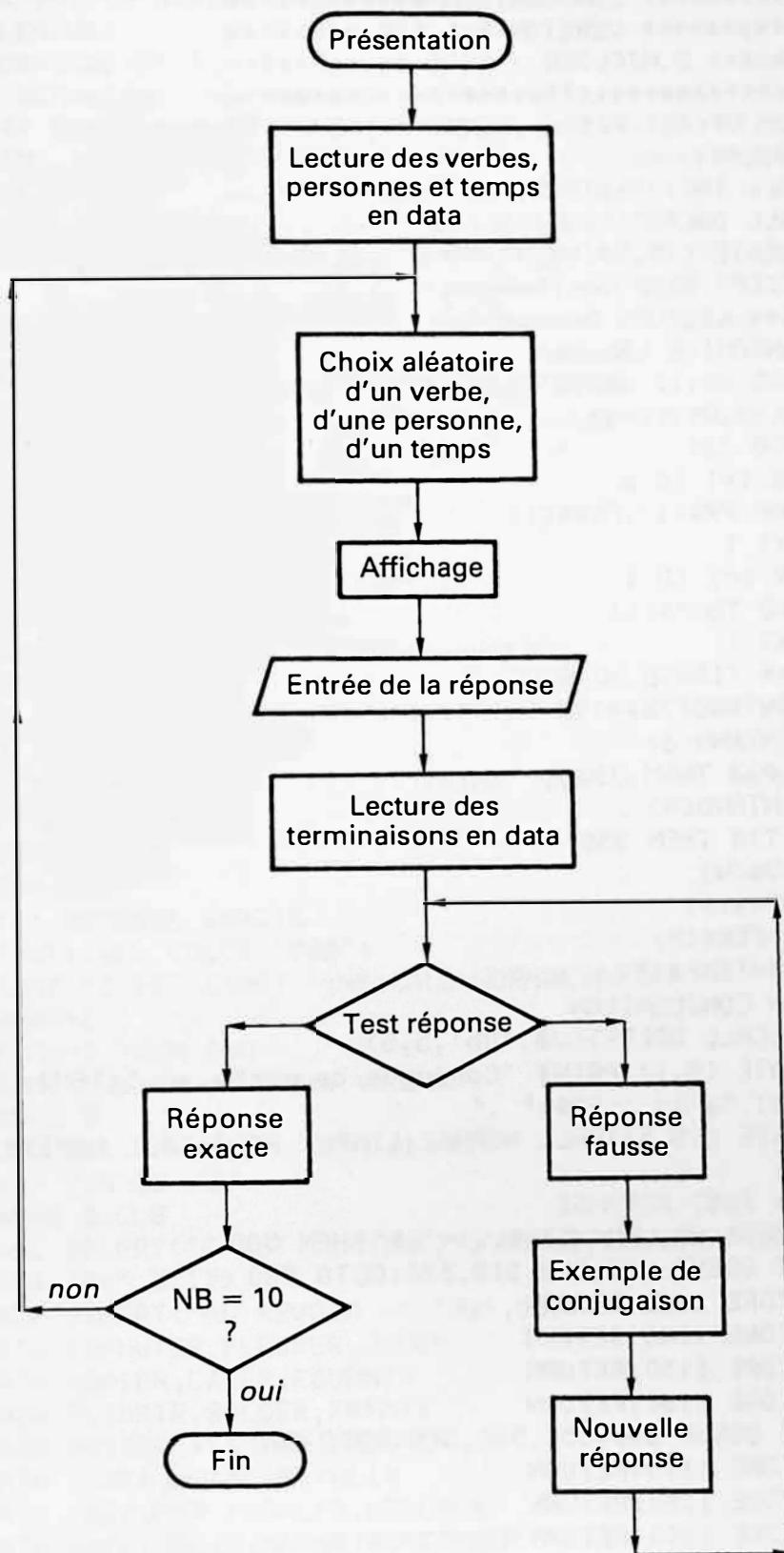
- lignes 610 à 720 ;
- conjugaison de l'auxiliaire et lecture des participes passés en DATA.

L'enfant ne pouvant sortir du programme qu'en trouvant la bonne réponse, il est inutile de comptabiliser le score. Par contre, et surtout dans une école, il est très utile de garder une trace écrite des erreurs.

Si l'on dispose d'une imprimante, il faut alors ajouter :

```
771 OPEN#1, « 20 »,OUTPUT  
772 PRINT#1,N$;«ERREUR AU VERBE»;V$;PER$;«DU»;TEMP$  
773 PRINT#1,« REPONSE »;REP$  
774 CLOSE#1
```

ce qui permet de connaître les difficultés rencontrées lors de l'exercice et d'y remédier ensuite.



```

100 !***** CONJUGAISON *****
110 !***** VERSION EXL 100 *****
120 !***** D.NIELSEN - 1985 *****
130 !*****
140 DIM V$(40),PR$(6),TERM$(6),TEMP$(4),PER$(6)
150 NB,L=0
160 CALL INIT:RESTORE:CLS "BCC"
170 CALL DBLPRT("CONJUGAISON","OR",7,12)
180 LOCATE (15,5):PRINT "Quel est ton prenom ?";
190 ACCEPT BEEP VALIDATE(ALPHA),N$
200 !*** LECTURE DES DATAS
210 RANDOMIZE LEN(N$)
220 READ W$:IF W$="*"THEN 250
230 L=L+1:V$(L)=W$
240 GOTO 220
250 FOR I=1 TO 6
260 READ PR$(I),PER$(I)
270 NEXT I
280 FOR I=1 TO 4
290 READ TEMP$(I)
300 NEXT I
310 !*** TIRAGE AU SORT
320 W=INTRND(28)
330 P=INTRND(6)
340 IF P>6 THEN 330
350 T=INTRND(4)
360 IF T>4 THEN 350
370 W$=V$(W)
380 P$=PR$(P)
390 PE$=PER$(P)
400 TEMP$=TEMP$(T)
410 !*** CONJUGAISON
420 CLS:CALL DBLPRT(W$,"0b",5,5)
430 LOCATE (8,1):PRINT "Conjugue ce verbe au ";TEMP$
440 PRINT "a la ";PE$;" ."
450 LOCATE (15,5):CALL NORMAL:LINPUT REP$:CALL UPPER(R
EP$)
460 !*** TEST REPONSE
470 IF SEG$(W$,LEN(W$)-1,2)="IR"THEN 530
480 ON T GOSUB 490,500,510,520:GOTO 580
490 RESTORE 1130:RETURN
500 RESTORE 1140:RETURN
510 RESTORE 1150:RETURN
520 RESTORE 1160:RETURN
530 ON T GOSUB 540,550,560,570:GOTO 580
540 RESTORE 1170:RETURN
550 RESTORE 1180:RETURN
560 RESTORE 1200:RETURN
570 RESTORE 1210:RETURN

```

```

580 FOR I=1 TO 6:READ TERM$(I):NEXT I
590 L=LEN(W$)
600 RAD$=SEG$(W$,1,L-2)
610 IF REP$=P$&" "&RAD$&TERM$(P)THEN 800
620 !*** REPONSE FAUSSE
630 PRINT:CALL COLOR("0RB"):PRINT "Tu te trompes,";N$:
CALL NORMAL
640 PAUSE 3
650 FOR I=14 TO 22:LOCATE (I,1):PRINT RPT$(" ",39);:NE
XT
660 LOCATE (14,5):PRINT "Voici un exemple de conjugais
on"
670 PRINT "d'un verbe semblable au ";TEM$
680 IF SEG$(W$,LEN(W$)-1,2)="IR"THEN EX$="FIN" ELSE EX
$="CHANT"
690 CALL COLOR("0b")
700 FOR I=1 TO 6
710 LOCATE (15+I,10)
720 PRINT PR$(I); " ";EX$;TERM$(I)
730 NEXT I
740 CALL NORMAL:PRINT "As-tu trouve ?";
750 PAUSE 10
760 FOR I=13 TO 22:LOCATE (I,1):PRINT RPT$(" ",39);:NE
XT
770 CALL NORMAL:LOCATE (13,1):PRINT "Quelle est ta nou
uelle reponse ?"
780 LINPUT REP$
790 GOTO 610
800 !*** REPONSE EXACTE
810 PRINT:CALL COLOR("0GB")
820 PRINT "C'EST EXACT ";N$:CALL NORMAL:PRINT
830 NB=NB+1
840 IF NB=9 THEN 880
850 PRINT "Voici un autre verbe."
860 PAUSE 5
870 GOTO 310
880 !*** FIN DU JEU
890 PAUSE 5:CLS
900 CALL DBLPRT("C'EST TERMINE","","0R",5,15):CALL DBLPR
T(N$,"0b",5,17)
910 CALL DBLPRT("AU REVOIR !","0W",5,19)
920 DATA CHAHUTER,PLEURER,JOUER
930 DATA COPIER,CREER,FOURNIR
940 DATA FLEURIR,SOLDER,FREMIR
950 DATA FRISER,FLECHIR,FROTTER
960 DATA LOUER,NOIRCIR,PALIR
970 DATA PARFUMER,PEDALER,MESURER
980 DATA RACCOUCIR,NOURRIR,REFUSER
990 DATA RONFLER,REUNIR,REUSSIR

```

```

1000 DATA SERRER,PUNIR,BONDIR
1010 DATA VIEILLIR,TRAHIR,CONVERTIR
1020 DATA *
1030 DATA JE,1ere personne du singulier
1040 DATA TU,2eme personne du singulier
1050 DATA IL,3eme personne du singulier
1060 DATA NOUS,1ere personne du pluriel
1070 DATA VOUS,2eme personne du pluriel
1080 DATA ILS,3eme personne du pluriel
1090 DATA Present de l'indicatif
1100 DATA Imperfait de l'indicatif
1110 DATA Futur simple
1120 DATA Passe simple
1130 DATA E,ES,E,ONS,EZ,ENT
1140 DATA AIS,AIS,AIT,IONS,IEZ,AIENT
1150 DATA ERAI,ERAS,ERA,ERONS,EREZ,ERONT
1160 DATA AI,AS,A,AMES,ATES,ERENT
1170 DATA IS,IS,IT,ISSONS,ISSEZ,ISSENT
1180 DATA ISSAIS,ISSAIS,ISSAIT,ISSIONS
1190 DATA ISSIEZ,ISSAIENT
1200 DATA IRAI,IRAS,IRA,IRONS,IRES,IRONT
1210 DATA IS,IS,IT,IMES,ITES,IRENT
1220 SUB INIT
1230 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(50850)
1240 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

1250 SUBEND
1260 SUB PLAY(N0$,L,D)
1270 FOR I=1 TO LEN(N0$)STEP 2
1280 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(N0$,I,2),1)
1290 P=VAL(SEG$("199176158149133118105",K,3))
1300 CALL POKE(258,P,D+128):PAUSE L/50
1310 NEXT:CALL POKE(259,0):SUBEND
1320 SUB DBLPRT(Z$,C$,X,Y)
1330 CALL COLOR(C$):CALL COLOR("0LH")
1340 K$="" :FOR I=1 TO LEN(Z$)
1350 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
1360 LOCATE (Y,X):PRINT K$
1370 LOCATE (Y+1,X):PRINT K$
1380 CALL NORMAL:SUBEND
1390 SUB NORMAL
1400 CALL COLOR("0BC")
1410 SUBEND
1420 SUB UPPER(Z$)
1430 K$="" :FOR K=1 TO LEN(Z$)
1440 S=ASC(SEG$(Z$,K,1)):IF S>90 THEN S=S-32
1450 K$=K$&CHR$(S):NEXT:Z$=K$
1460 SUBEND

```

# **ADDITION**

## **Thème**

Didact d'apprentissage de la technique opératoire de l'addition (trois options : nombres à un, deux, trois chiffres).

## **Niveau**

A partir de 6 ans (pour la 1<sup>re</sup> option).

## **Intérêt pédagogique**

### **Connaissances**

- élaboration de la technique opératoire de l'addition avec retenues ;
- réinvestissement d'acquis : notions d'unités, dizaines et centaines, table d'addition.

### **Savoir-faire**

- apprentissage (ou rappel) d'une présentation écrite de l'opération ;
- mémorisation des opérations élémentaires (tables) et utilisation dans l'opération en cours.

## Déroulement

Après entrée du prénom du joueur, trois niveaux d'opérations lui sont présentés : additions de nombres à un, deux, ou trois chiffres.

Deux nombres, selon le niveau choisi, s'inscrivent en double grandeur sur le haut de l'écran. La moitié basse de l'écran est réservée à la décomposition de l'opération en addition de nombres simples et aux messages de réponses. Les chiffres s'afficheront dans l'opération inscrite en haut au fur et à mesure des réponses.

EXL 100 demande au joueur d'indiquer par quels chiffres doit commencer l'addition (unités, dizaines, centaines).

Puis l'opération s'effectue petit à petit : l'enfant additionne les unités, puis les dizaines, puis les centaines. Pour chaque opération simple, en cas d'erreur, une table d'addition lui est proposée, afin qu'il trouve de lui-même la bonne réponse.

Si le résultat de l'addition simple comporte deux chiffres, le programme explique à l'enfant le principe de la retenue et la décomposition en dizaines et unités (ou centaines et dizaines). Cette retenue est visualisée sur l'opération affichée en haut de l'écran et ajoutée dans l'opération simple suivante. Ainsi, l'enfant exécute l'opération de la même manière que sur son cahier, mais aucune erreur ne pourra être écrite dans l'opération elle-même, car elle est immédiatement détectée et doit être corrigée.

Quatre additions sont proposées selon le même principe dans ce programme.

## Commentaires

- |                  |  |
|------------------|--|
| <b>140 à 250</b> | Initialisation et présentation du jeu.   |
| <b>280 à 310</b> | Les chiffres de chaque nombre sont tirés au sort séparément.   |
| <b>330 à 370</b> | Affichage de l'opération.  |
| <b>380</b>       | EXL 100 autorise l'assignation d'une valeur à plusieurs variables. Ici, RED et RC sont mises à zéro simultanément. |
| <b>390 à 480</b> | Par où commencer l'opération ?   |

<b>500 à 640</b>	Addition des unités. <i>Lignes 500 à 520 : Réponse et test de la réponse. En cas de réponse fausse, envoie au sous-programme de table d'addition puis redemande une nouvelle réponse.</i>
	<i>Ligne 550 : Affichage de la réponse si elle ne comporte qu'un seul chiffre.</i>
	<i>Lignes 580 à 600 : Décomposition du résultat en unités et dizaines.</i>
	<i>Lignes 620 à 630 : Affichage des unités et de la retenue.</i>
<b>650 à 830</b>	Addition des dizaines (même déroulement que les unités).
<b>840 à 940</b>	Addition des centaines.
<b>950 à 990</b>	Fin d'opération et présentation de l'opération suivante.
<b>1000 à 1100</b>	Table d'addition : elle est affichée pendant dix secondes en cas de réponse fausse pour une des additions simples : la variable T prend la valeur du chiffre (unités, dizaines ou centaines) du nombre supérieur A.
<b>1110 à 1130</b>	Fin du jeu.
<b>1130 à 1140</b>	Fin du programme. L'écran est débarrassé de ses fenêtres par CALL HROFF.
<b>1150 à 1180</b>	Initialisation générale.
<b>1190 à 1250</b>	Ecriture en double taille.
<b>1260 à 1280</b>	Ecriture normale.
<b>1290 à 1310</b>	Effacement des fenêtres.

## Particularités techniques

Les deux nombres sont définis aux lignes 290 à 310. En fait, pour faciliter la décomposition pour les opérations simples, on tire au sort séparément les chiffres (AC,AD,AU et BC,BD,BU).

L'affichage en double taille (lignes 360 et 370) permet de placer les chiffres les uns juste à côté des autres. Le signe opérateur est placé à l'extrême gauche de l'opération, en fonction du nombre de chiffres du résultat.

Le sous-programme DBLPRT est utilisé pour afficher les chiffres en double taille à un endroit donné de l'écran. L'instruction LOCATE est également très utilisée, notamment pour placer les retenues ou les tables d'addition.

Les lignes 660 et 850, au début de l'addition des dizaines et des centaines, testent le choix (CH) effectué au début sur le niveau des opérations et envoient éventuellement à la fin du jeu.

Pour les mêmes raisons, les lignes 560 et 750 modifient l'affichage de la retenue si l'opération ne comporte que des unités, ou des unités et des dizaines.

## Modifications

Ce programme est destiné à apprendre à réaliser et à présenter une addition. L'enfant ne peut passer à l'addition des dizaines que s'il a correctement effectué celle des unités, même s'il doit recommencer dix fois !

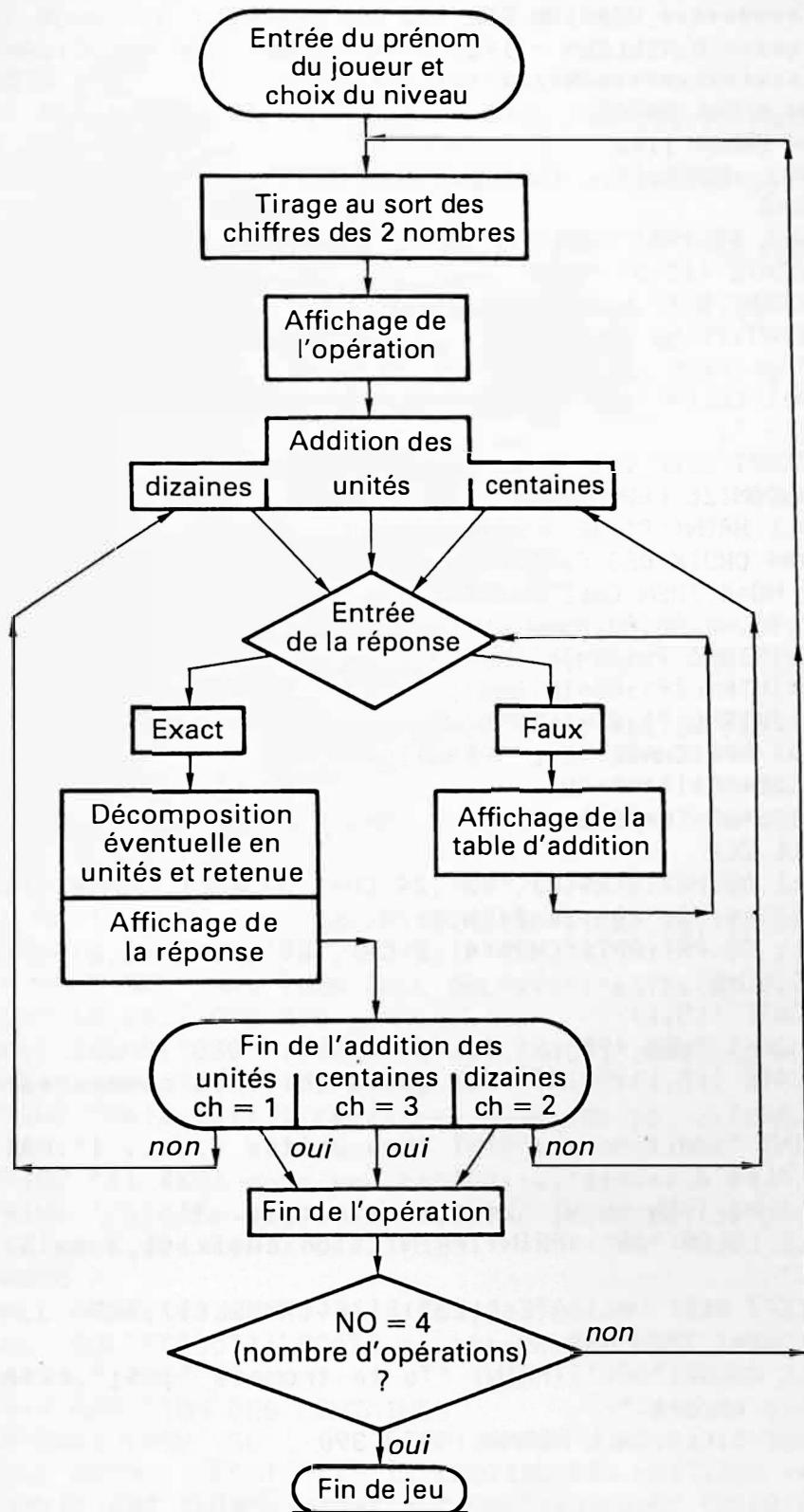
On peut transformer ce programme en un exercice de contrôle, en n'acceptant que deux réponses au maximum, par exemple. Il faut ajouter dans chacune des trois séries d'opérations simples un « compteur » et afficher un score à la fin du jeu.

## Adaptations

« Addition » est un programme complet où l'opération est décomposée en séquences simples.

Le mécanisme opératoire (lignes 280 à 990) peut être incorporé à d'autres programmes sur des applications de l'addition (factures, comptes, etc.).

Il serait aussi possible de moduler les opérations suivantes en fonction des résultats : une addition de nombres à deux chiffres pour commencer, puis selon le nombre d'erreurs, une addition de nombres à trois chiffres, ou retour à une addition à un seul chiffre.



```

100 !***** ADDITION *****
110 !***** VERSION EXL 100 *****
120 !***** D.NIELSEN - 1985 *****
130 !***** *****
140 ON BREAK ERROR
150 ON ERROR 1140
160 CALL HROFF:CALL INIT:CLS "BCC"
170 NO=0
180 CALL DBLPRT("ADDITION","0b",12,12)
190 LOCATE (15,5):PRINT "Quel est ton prenom ?";
200 ACCEPT BEEP VALIDATE(ALPHA),N$
210 PRINT:PRINT "Additions avec des nombres a UN,DEUX"
; "ou TROIS chiffres ?"
220 CALL COLOR("OR"):PRINT "Indique ton choix (1,2 ou
3) : ";
230 ACCEPT BEEP VALIDATE("123")NULL(1)SIZE(1),CH
240 RANDOMIZE LEN(N$)*5
250 CALL HRON("C",12,1):CALL INIT
260 !*** CHOIX DES CHIFFRES
270 IF NO=4 THEN CALL HROFF:GOTO 1110
280 AC,BC,AD,BD,AU,BU=0:ON CH GOTO 310,300,290
290 AC=INTRND(9):BC=INTRND(9)
300 AD=INTRND(9):BD=INTRND(9)
310 AU=INTRND(9):BU=INTRND(9)
320 !*** AFFICHAGE DE L'OPERATION
330 A=100*AC+10*AD+AU
340 B=100*BC+10*BD+BU
350 CALL CLR
360 CALL DBLPRT(STR$(A),"0B",24-CH*2,3):CALL DBLPRT("+
"&STR$(B),"0B",20-2*CH,6)
370 CALL DBLPRT(RPT$(CHR$(4),2+CH),"0B",20-CH*2,8)
380 RED,RC=0
390 LOCATE (15,1)
400 IF CH=1 THEN 490
410 LOCATE (15,1):PRINT "Par quels chiffres commences-
tu ton"
420 PRINT "addition ::PRINT "Les unites ..... 1":PRI
NT "Les dizaines .... 2"
430 IF CH=3 THEN PRINT "Les centaines ... 3"
440 CALL COLOR("OR"):PRINT:PRINT "Ton choix (1,2 ou 3)
: ";
450 ACCEPT BEEP VALIDATE("123")SIZE(1)NULL(1),DEB
460 IF DEB=1 THEN 490
470 CALL COLOR("ORB"):PRINT "Tu te trompes ";N$;,"refl
eches encore."
480 PAUSE 5:CLS:CALL NORMAL:GOTO 390
490 !*** ADDITION DES UNITES
500 CLS:PRINT "Que font";AU;"unite(s)":PRINT TAB(9);"+"
;BU;"unites ?"

```

```

510 INPUT RU
520 IF RU=AU+BU THEN 550
530 T=AU:GOSUB 1000
540 GOTO 490
550 IF RU<10 THEN CALL DBLPRT(STR$(RU),"0b",22,10):GOT
0 650
560 IF RU>9 AND CH=1 THEN CALL DBLPRT("1"&STR$(RU-10),
"0b",20,10):GOTO 650
570 CALL COLOR("0GB"):PRINT "C'EST EXACT,";N$:CALL NOR
MAL
580 PRINT "Cela fait 1 dizaine, plus";RU-10;"unites"
590 PRINT "Il faut donc ecrire";RU-10;"dans la"
600 PRINT "colonne des unites,et 1 au dessus de","la c
olonne des dizaines."
610 PAUSE 7
620 CALL DBLPRT(STR$(RU-10),"0b",22,10)
630 CALL COLOR("0b"):LOCATE (1,20):PRINT "+1"
640 PAUSE 7
650 !*** ADDITION DES DIZAINES
660 IF CH=1 THEN 950
670 CLS:CALL NORMAL:LOCATE (15,1)
680 PRINT "Que font";AD;"dizaine(s) +":PRINT TAB(9);BD
;"dizaine(s)"
690 RED=0:IF RU>9 THEN PRINT "plus 1 dizaine de retenu
e":RED=1
700 INPUT RD
710 IF RD=AD+BD+RED THEN 740
720 T=AD:GOSUB 1000
730 GOTO 650
740 IF RD<10 THEN CALL DBLPRT(STR$(RD),"0b",20,10):GOT
0 840
750 IF RD>9 AND CH=2 THEN CALL DBLPRT("1"&STR$(RD-10),
"0b",18,10):GOTO 840
760 CALL COLOR("0GB"):PRINT "C'EST EXACT,";N$:CALL NOR
MAL
770 PRINT "Cela fait 1 centaine, plus";RD-10;"dizaine(
s)"
780 PRINT "Il faut donc ecrire";RD-10;"dans la"
790 PRINT "colonne des dizaines,et 1 au dessus de","la c
olonne des centaines."
800 PAUSE 7
810 CALL DBLPRT(STR$(RD-10),"0b",20,10)
820 CALL COLOR("0b"):LOCATE (1,18):PRINT "+1"
830 PAUSE 7
840 !*** ADDITION DES CENTAINES
850 IF CH=2 THEN 950
860 CALL NORMAL:LOCATE (15,1):CLS
870 PRINT "Que font";AC;"centaine(s) +":PRINT TAB(9);B
C;"centaine(s)"

```

```

880 RC=0:IF RD>9 THEN PRINT "plus 1 centaine de retenue"
   e":RC=1
890 INPUT CC
900 IF CC=AC+BC+RC THEN 930
910 T=AC:GOSUB 1000
920 GOTO 840
930 IF CC<10 THEN CALL DBLPRT(STR$(CC),"0b",18,10):GOT
   0 950
940 CALL DBLPRT("1"&STR$(CC-10),"0b",16,10)
950 !*** BRAVO
960 CLS:CALL DBLPRT("BRAVO "&N$,"0B",1,15)
970 PRINT:PRINT:PRINT "Voici maintenant une autre opération"
980 NO=N0+1:PAUSE 7
990 GOTO 270
1000 !*** TABLE D'ADDITION
1010 CLS:LOCATE (13,1)
1020 CALL COLOR("0BR"):PRINT "TU TE TROMPES, ";N$
1030 PRINT "REЛИS LA TABLE D'ADDITION":PAUSE 4
1040 CLS:CALL NORMAL
1050 FOR I=1 TO 10
1060 PRINT T;"+";I;"=";T+I,
1070 NEXT I
1080 CALL COLOR("0b"):PRINT:PRINT "AS-TU TROUVE LA BONNE REPONSE ,";N$;" ?"
1090 PAUSE 10
1100 CALL NORMAL:LOCATE (15,1):RETURN
1110 CLS:CALL DBLPRT("C'EST TERMINE.", "0B",1,12)
1120 CALL DBLPRT("AU REVOIR,"&N$,"0b",1,15)
1130 END
1140 CALL HROFF:CALL NORMAL:LOCATE (1,1):RETURN 1110
1150 SUB INIT
1160 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(50850)
1170 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)
1180 SUBEND
1190 SUB DBLPRT(Z$,C$,X,Y)
1200 CALL COLOR(C$):CALL COLOR("0LH")
1210 K$="" :FOR I=1 TO LEN(Z$)
1220 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
1230 LOCATE (Y,X):PRINT K$
1240 LOCATE (Y+1,X):PRINT K$
1250 CALL NORMAL:SUBEND
1260 SUB NORMAL
1270 CALL COLOR("0BC")
1280 SUBEND
1290 SUB CLR
1300 FOR I=1 TO 22:LOCATE (I,1):PRINT RPT$(" ",40):NEXT
1310 SUBEND

```

# **SOUSTRACTION**

## **Thème**

Didact d'apprentissage de la technique opératoire de la soustraction (trois options : nombres à un, deux ou trois chiffres).

## **Niveau**

A partir de 6 ans (pour la 1<sup>re</sup> option).

## **Intérêt pédagogique**

### **Connaissances**

- comparaison de chiffres et de nombres ;
- élaboration de la technique opératoire de la soustraction ;
- mécanisme de la « retenue » ;
- réinvestissement d'acquis : notions d'unités, de dizaines, de centaines, tables d'addition.

### **Savoir-faire**

- apprentissage (ou rappel) d'une présentation écrite de l'opération ;
- mémorisation des opérations élémentaires et utilisation dans l'opération en cours.

## Déroulement

Le joueur entre son prénom, puis son choix de niveau d'opération (avec des nombres à un, deux, trois chiffres).

Une première opération s'inscrit sur la moitié haute de l'écran en chiffres de double taille. Chaque chiffre étant tiré au sort entre 1 et 9, il est possible d'obtenir une soustraction « impossible » à faire (le premier nombre est plus petit que le second). Si tel est le cas, l'ordinateur demande au joueur s'il peut faire cette soustraction. Une autre opération est alors proposée.

Il faut ensuite indiquer par quels chiffres la soustraction doit commencer (unités, dizaines, centaines).

L'opération débute alors par la soustraction des unités. Deux cas peuvent se présenter :

- **AU > BU** (AU est le chiffre des unités du nombre A, BU celui du nombre B) : la soustraction est possible et l'enfant entre sa réponse.
- **AU < BU** : le joueur tape « N » pour indiquer qu'il ne peut pas faire cette soustraction. Le programme propose alors de prendre une dizaine et la décomposition de cette retenue est expliquée en bas de l'écran et affichée dans l'opération. La soustraction devient alors possible et le résultat est demandé.

En cas d'erreur dans cette soustraction simple, c'est une table d'addition (du chiffre de B) qui est proposée. L'enfant n'a donc pas directement la réponse, mais doit utiliser les renseignements de cette table pour la découvrir.

En fonction du niveau choisi, le programme continue par la soustraction des dizaines, puis des centaines, selon le même principe. Il rappelle la retenue utilisée dans la soustraction précédente s'il y a lieu.

Le programme propose quatre opérations à la suite.

## Commentaires

<b>140 à 240</b>	Initialisation et présentation du programme.
<b>260 à 300</b>	Tirage au sort des chiffres des deux nombres.
<b>310 à 360</b>	Affichage de l'opération.

- 380 à 470** Sens de l'opération.  
*Ligne 380 : Si A<B (soustraction impossible), le programme va en 770.*  
*Lignes 400 - 470 : Demandent au joueur par où il veut commencer.*
- 490 à 500** Affectent aux variables N1 et N2 (chiffres de l'opération), T\$ et V\$ (noms des unités), X (place où la réponse doit être affichée) les valeurs correspondant à la soustraction des unités. L'opération elle-même est effectuée en sous-programme (GOSUB 870).
- 520 à 530** Si CH (choix du niveau) est 2 ou 3, affectent aux variables les valeurs correspondant à la soustraction des dizaines.
- 550 à 560** Si CH est 3, affectent aux variables les valeurs correspondant à la soustraction des centaines.
- 570 à 580** Fin d'opération.
- 590** Envoie à une autre opération.
- 600** Fin du programme.
- 610 à 710** Affichage de la table d'addition en cas d'erreur de calcul mental.
- 720 à 740** Fin du programme.
- 770 à 860** Interroge l'enfant si l'opération est impossible A<B.
- 870 à 990** Opération.  
*Ligne 880 : S'il y a une retenue de l'opération précédente, envoie au sous-programme 1100. La réponse est entrée ligne 920 : si elle est « N » (pour « NON »), le programme va en ligne 1000 (retenue).*  
*Ligne 950 : Convertit la chaîne REP\$ en sa valeur numérique si l'opération est possible.*  
*Ligne 960 : Si la réponse est bonne, envoie en 980.*  
*Ligne 970 : Réponse fausse. Envoie au sous-programme d'affichage de la table d'addition puis attend une nouvelle réponse (GOSUB 620).*
- 1000 à 1090** Retenue. Même déroulement, mais en prenant une unité supérieure : décomposition de la retenue.

- |                    |   |
|--------------------|---|
| <b>1100 à 1150</b> | Affichage et explications de la nouvelle opération en cas de retenue dans la précédente soustraction. |
| <b>1160 à 1190</b> | Initialisation générale.  |
| <b>1200 à 1260</b> | Ecriture double taille.   |
| <b>1270 à 1290</b> | Ecriture normale.   |
| <b>1300 à 1320</b> | Effacement des fenêtres.  |

## Particularités techniques

Une soustraction de nombres à deux ou trois chiffres étant, en fait, formée de deux ou trois soustractions simples, la décomposition de chacune des opérations se fait à l'intérieur d'un programme unique (870 à 990) dans lequel les variables sont affectées de la valeur correspondant à la soustraction en cours.

Chaque résultat exact (chiffre et retenue) s'inscrit dans l'opération elle-même, après avoir été expliqué et contrôlé dans la fenêtre de travail (lignes du bas de l'écran).

## Modifications

Ce programme est lui aussi destiné à APPRENDRE à effectuer une soustraction.

Pour l'utiliser comme contrôle de connaissances, les propositions faites dans addition sont encore valables (installation d'un compteur d'entrées). Les lignes 400 à 470 peuvent alors être supprimées.

Si l'on veut éviter l'affichage d'opérations impossibles, il suffit de supprimer les lignes 770 à 860 et de modifier la ligne :

770 IF A<B THEN 260

Il faut toutefois noter que la comparaison de deux nombres et la reconnaissance des cas dans lesquels on ne peut pas faire une soustraction n'est pas un exercice inutile.

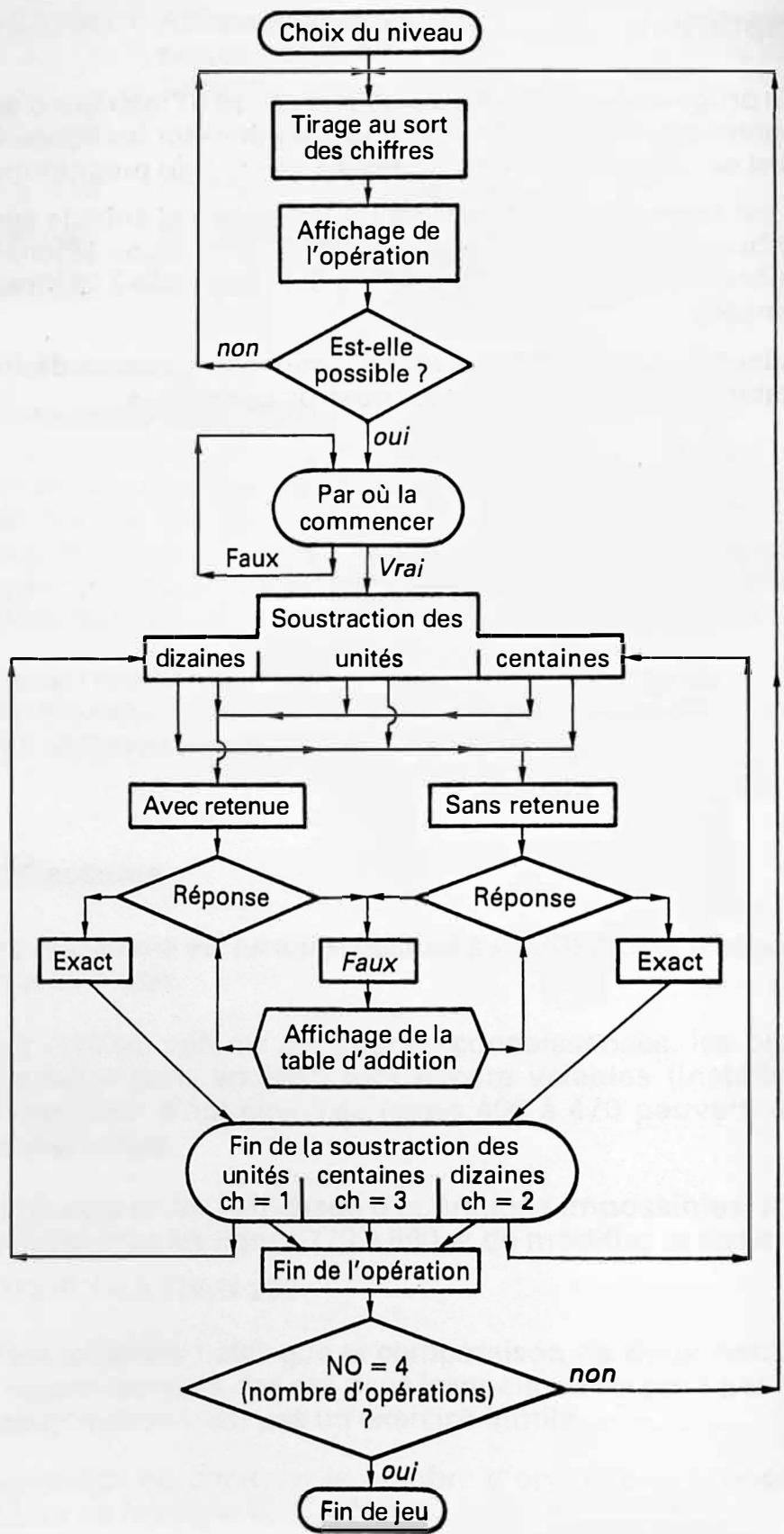
Augmenter ou diminuer le nombre d'opérations proposées au joueur se fait ligne 590 (IF NO<4).

## Adaptations

Ce programme peut, lui aussi, être utilisé à l'intérieur d'autres programmes de raisonnement (en supprimant les lignes 490 à 560 et en affectant aux variables les valeurs du programme).

Il est bien sûr possible de faire travailler les enfants sur des nombres plus grands (quatre ou cinq chiffres) ou afficher des nombres de niveau différent (A=3 chiffres, B=2 chiffres par exemple).

Adapter ce programme à l'utilisation de nombres décimaux ne devrait pas non plus poser de gros problèmes.



```

100 !***** SOUSTRACTION *****
110 !***** VERSION EXL 100 *****
120 !**** D.NIELSEN - 1985 *****
130 !***** *****
140 ON BREAK ERROR
150 ON ERROR 750
160 CLS "BCC":CALL HROFF:CALL INIT
170 NO=0:A$(1)="unite":A$(2)="dizaine":A$(3)="centaine"
   :"Z$="SOUSTRACTION"
180 CALL DBLPRT(ZZ$,"0b",8,12)
190 LOCATE (15,5):PRINT "Quel est ton prenom ?";
200 ACCEPT BEEP VALIDATE(ALPHA),N$
210 PRINT:PRINT "Soustractions avec des nombres a UN,D
   EUX";"ou TROIS chiffres ?"
220 CALL COLOR("OR"):PRINT "Indique ton choix (1,2 ou
   3) : ";
230 ACCEPT BEEP VALIDATE("123")NULL(1)SIZE(1),CH
240 RANDOMIZE LEN(N$)*5
250 !*** CHOIX DES CHIFFRES
260 CALL HRON("C",14,1):CALL INIT:CLS
270 AC,BC,AD,BD,AU,BU=0:ON CH GOTO 300,290,280
280 AC=INTRND(9):BC=INTRND(9)
290 AD=INTRND(9):BD=INTRND(9)
300 AU=INTRND(9):BU=INTRND(9)
310 !*** AFFICHAGE DE L'OPERATION
320 A=100*AC+10*AD+AU
330 B=100*BC+10*BD+BU
340 CALL CLR
350 CALL DBLPRT(STR$(A),"0B",24-CH*2,3):CALL DBLPRT("-
   "&STR$(B),"0B",20-2*CH,6)
360 CALL DBLPRT(RPT$(CHR$(4),2+CH),"0B",20-CH*2,9)
370 LOCATE (15,1)
380 IF A<B THEN 770
390 IF CH=1 THEN 480
400 LOCATE (15,1):PRINT "Par quels chiffres commences-
   tu ta"
410 PRINT "soustraction":PRINT "Les unites ..... 1":P
   RINT "Les dizaines .... 2"
420 IF CH=3 THEN PRINT "Les centaines ... 3"
430 CALL COLOR("OR"):PRINT:PRINT "Ton choix (1,2 ou 3)
   : ";
440 ACCEPT BEEP VALIDATE("123")SIZE(1)NULL(1),DEB
450 IF DEB=1 THEN 480
460 CALL COLOR("ORB"):PRINT "Tu te trompes ";N$;" ,refl
   echis encore."
470 PAUSE 5:CLS:CALL NORMAL:GOTO 370
480 !***
490 N1=AU:N2=BU:X=22
500 T$=A$(1):V$=A$(2):GOSUB 870

```

```

510 IF CH=1 THEN 570
520 N1=AD:N2=BD:X=20
530 T$=A$(2):V$=A$(3):GOSUB 870
540 IF CH=2 THEN 570
550 N1=AC:N2=BC:X=18
560 T$=A$(3):GOSUB 870
570 CLS:CALL DBLPRT("BRAVO "&N$,"OR",1,15)
580 NO=NO+1:PAUSE 3
590 IF NO<4 THEN 270
600 CALL HROFF:GOTO 720
610 !*** TABLE D'ADDITION
620 CLS:LOCATE (15,1)
630 CALL COLOR("0B"):PRINT "TU TE TROMPES," ;N$
640 PRINT "RELIS LA TABLE D'ADDITION":PAUSE 4
650 CLS:CALL NORMAL
660 FOR I=1 TO 10
670 PRINT N2;"+" ;I ;"=" ;N2+I ,
680 NEXT I
690 CALL COLOR("0b"):PRINT:PRINT "AS-TU TROUVE LA BONNE
E REPONSE ,";N$;" ?"
700 PAUSE 10
710 CLS:CALL NORMAL:LOCATE (15,1):RETURN
720 CLS:CALL DBLPRT("C'EST TERMINE.", "0B", 1, 12)
730 CALL DBLPRT("AU REVOIR," &N$, "0b", 1, 15)
740 END
750 CALL HROFF:CALL NORMAL:RETURN 720
760 CALL HROFF:CALL NORMAL:LOCATE (1,1):RETURN 720
770 !*** SOUSTRACTION IMPOSSIBLE
780 PRINT "Est-ce que tu peux faire cette soustrac-
    tion," ;N$
790 PRINT:PRINT "1 pour oui, 2 pour non ?";
800 ACCEPT BEEP VALIDATE("12")SIZE(1)NULL(1),I
810 IF I=2 THEN 840
820 PRINT:PRINT "ATTENTION," ;N$; ",regarde":PRINT "bien
    les deux nombres."
830 PRINT "Reflechis encore.":GOTO 780
840 PRINT "Tu as raison," ;N$; ",nous":PRINT "allons donc
    faire une autre"
850 PRINT "operation.":PAUSE 5
860 GOTO 270
870 !*** SOUSTRACTION PAR UNITES
880 CLS:IF RET=1 THEN GOSUB 1100
890 PRINT "Que font";N1;T$;:IF N1>1 THEN PRINT "s";
900 PRINT "-";N2;T$;:IF N2>1 THEN PRINT "s";
910 PRINT "Si tu ne peux pas faire cette soustrac-
    tion, tapes <N>"
920 ACCEPT BEEP VALIDATE("0123456789N")SIZE(2)NULL("N"
    ),REP$
930 IF REP$="N"AND N1<N2 THEN 1000

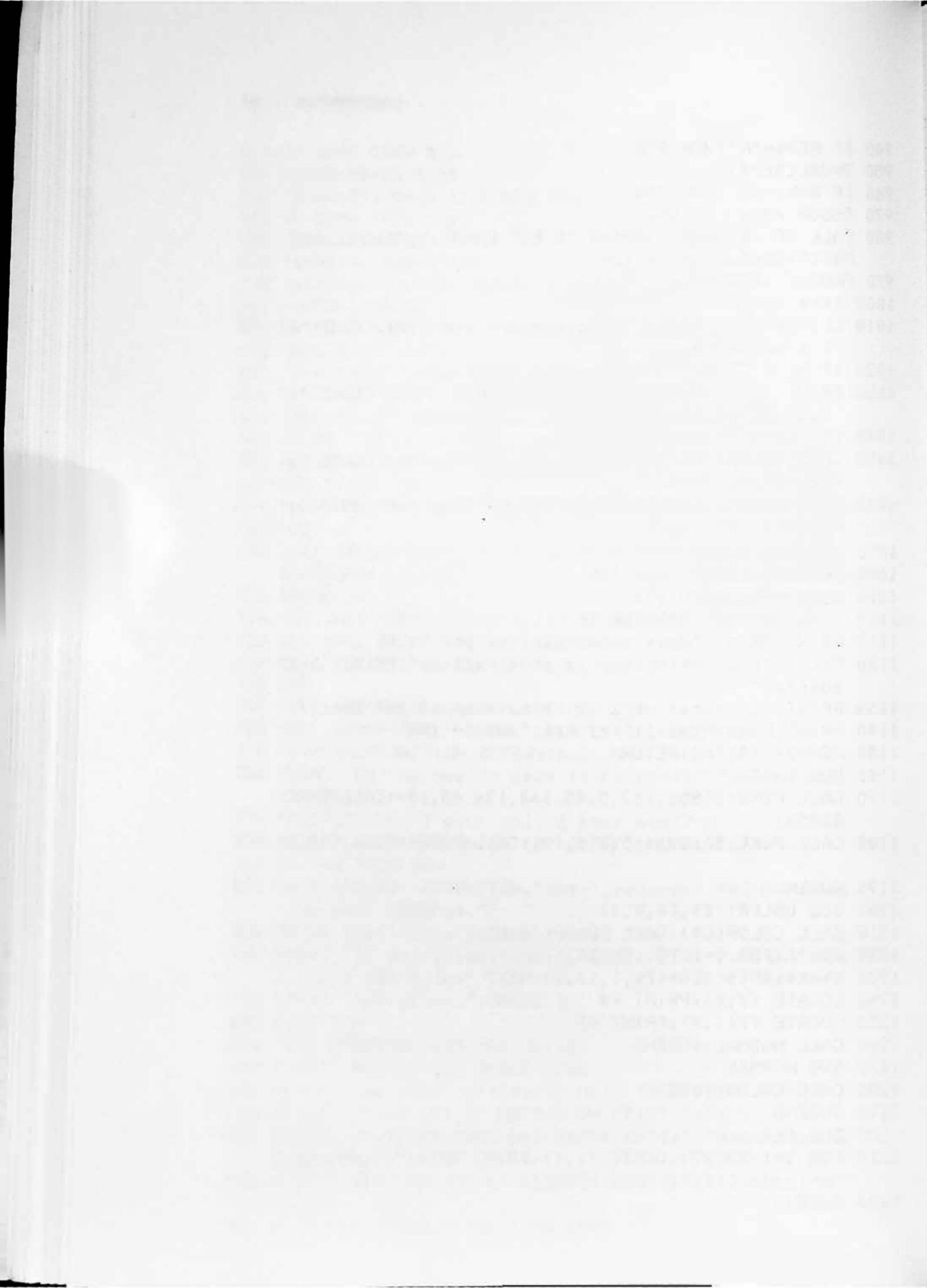
```

```

940 IF REP$="N"THEN 970
950 R=VAL(REP$)
960 IF R=N1-N2 THEN 980
970 GOSUB 620:GOTO 890
980 CALL COLOR("0BR"):PRINT "C'EST EXACT,";N$:CALL DBL
    PRT(STR$(R),"0b",X,12)
990 PAUSE 5:RETURN
1000 !*** RETENUE
1010 CLS:PRINT "Il faut donc prendre une ";V$:PRINT "et
    1'ajouter aux";N1;T$;
1020 IF N1>1 THEN PRINT "s" ELSE PRINT
1030 PRINT "Cela fait";N1+10;T$;:IF N1>1 THEN PRINT "s"
    ELSE PRINT
1040 RET=1:PAUSE 5:CLS
1050 CALL COLOR("0b"):LOCATE (2,X):PRINT "1":LOCATE (8,
    X-2):PRINT "+1"
1060 CALL NORMAL:LOCATE (15,1):PRINT "Que font";N1+10;"-
";N2;" ?"
1070 INPUT R
1080 IF R=N1+10-N2 THEN 980
1090 GOSUB 620:GOTO 1060
1100 !*** RETENUE PRECEDENTE
1110 PRINT:PRINT "Nous avons utilise une ";T$
1120 PRINT "pour effectuer la soustraction":PRINT "precedente."
1130 PRINT "L'operation a faire maintenant est donc:"
1140 PRINT N1;"-";N2+1;" et non ";N1;"-";N2
1150 N2=N2+1:RET=0:RETURN
1160 SUB INIT
1170 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
    50850)
1180 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

1190 SUBEND
1200 SUB DBLPRT(Z$,C$,X,Y)
1210 CALL COLOR(C$):CALL COLOR("0LH")
1220 K$="":FOR I=1 TO LEN(Z$)
1230 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
1240 LOCATE (Y,X):PRINT K$
1250 LOCATE (Y+1,X):PRINT K$
1260 CALL NORMAL:SUBEND
1270 SUB NORMAL
1280 CALL COLOR("0BC")
1290 SUBEND
1300 SUB CLR
1310 FOR I=1 TO 22:LOCATE (I,1):PRINT RPT$(" ",40):NEXT
1320 SUBEND

```



# **MULTIPLICATION**

## **Thème**

Didact d'apprentissage de la technique opératoire de la multiplication (trois options : multiplication par un nombre à un, deux, trois chiffres).

## **Niveau**

A partir de 8 ans (Cours élémentaire 2<sup>e</sup> année).

## **Intérêt pédagogique**

### **Connaissances**

- élaboration et décomposition de la technique opératoire de la multiplication par un nombre à un, deux, trois chiffres ;
- réinvestissement d'acquis : notions d'unités, de classe, technique de l'addition.

### **Savoir-faire**

- apprentissage (ou rappel) d'une présentation écrite de la multiplication ;
- mémorisation des opérations élémentaires (tables d'addition et de multiplication) et utilisation dans les séquences opératoires.

## Déroulement

Le joueur note son prénom, puis son choix de niveau : le premier nombre A est toujours un nombre de trois chiffres, mais il peut être multiplié par un nombre B formé de un, deux, trois chiffres.

L'opération s'inscrit, en double grandeur, sur la moitié haute de l'écran. La décomposition de l'opération en multiplications simples se fera en bas de l'écran.

EXL 100 propose d'abord de multiplier A par le chiffre des unités de B (BU). L'enfant exécute ce travail en trois multiplications simples successives. A chaque fois, s'il y a lieu, la retenue est expliquée et ajoutée à l'opération suivante.

Si la réponse est fausse, une table de multiplication s'affiche pendant dix secondes et une nouvelle réponse est demandée.

Si la réponse est bonne, elle s'inscrit (avec la retenue) dans l'opération en haut.

Selon le niveau choisi, EXL 100 demande ensuite de multiplier A par le chiffre des dizaines de B, puis par celui des centaines. Chaque séquence est décomposée en trois multiplications simples, comme pour les unités. Les chiffres s'inscrivent dans l'opération sur une 2<sup>e</sup> et une 3<sup>e</sup> ligne.

Enfin, le programme demande à l'enfant d'additionner ces deux ou trois lignes de résultats (si l'on a choisi le niveau 2 ou 3) pour obtenir la réponse finale.

Quatre opérations sont proposées dans ce programme.

## Commentaires

- |                  |   |
|------------------|---|
| <b>100 à 240</b> | Initialisation et présentation du programme.<br><i>Lignes 175 et 176 :</i> Entrée des noms des unités qui seront réutilisées par la suite.  |
| <b>250 à 305</b> | Tirage au sort des chiffres des deux nombres.   |
| <b>310 à 360</b> | Affichage de l'opération.<br><i>Ligne 360 :</i> Trace un trait sous les deux nombres.   |
| <b>390 à 545</b> | Multiplication par le chiffre des unités.<br><i>Lignes 401-402, 403-404, 405-510 :</i> Décomposition en multiplications simples effectuées par le même sous-programme en 4000. Ces lignes attribuent aux variables du sous-programme les valeurs correspondant à l'opé- |

ration en cours (voir particularités techniques).

*Lignes 403 - 405 - 530* : Au retour du sous-programme, donnent un nom précis à chaque réponse (pour l'addition finale).

- 550 à 860** Même déroulement pour la multiplication des dizaines.
- 870 à 1140** Multiplication des centaines.  
*Ligne 1140* : Envoie à l'addition ligne 5000.
- 3000 à 3080** Affichage de la table de multiplication par N2 en cas d'erreur de calcul.
- 4000 à 4200** Multiplication simple : affichage et réponse. Si la réponse est bonne, le programme continue en 4100. Sinon, il inscrit la table de multiplication (GOSUB 3000) et demande une nouvelle réponse.
- 4110 à 4200** Explication de la retenue.  
*Ligne 4100* : S'il y a une retenue de l'opération précédente, envoie en 4500.
- 4150 à 4200** Affichage de la réponse. Fin du sous-programme « opération simple ».
- 4500 à 4600** Retenue précédente : demande de l'ajouter au résultat de la multiplication simple.
- 5000 à 5580** Addition des multiplications précédentes.  
*Lignes 5030 à 5100* : Unités.  
*Lignes 5110 à 5200* : Addition des dizaines.  
*Lignes 5210 à 5300* : Addition des centaines.  
*Lignes 5310 à 5400* : Addition des milliers.  
*Lignes 5410 à 5510* : Addition des dizaines de milliers.  
*Lignes 5515 à 5580* : Addition des centaines de milliers.
- 5600 à 5650** Fin d'opération. Retour à une autre opération ou fin du jeu.
- 5700 à 5720** Erreur d'addition.
- 10000 à 10020** Fin du programme.
- 11600 à 11900** Initialisation générale.
- 12000 à 12600** Ecriture double taille.
- 12700 à 12900** Ecriture normale.
- 13000 à 13200** Effacement des fenêtres.

## Particularités techniques

Les multiplications simples sont toutes effectuées dans le même sous-programme (4000-4320) dont les variables prennent des valeurs différentes selon l'opération.

Compte tenu du nombre important de variables dans ce programme, voici leur sens :

<b>CH</b>	Choix du niveau.
<b>AC - AD - AU</b>	Centaines, dizaines, unités du nombre A.
<b>BC - BD - BU</b>	Centaines, dizaines, unités du nombre B.
<b>V\$</b>	Nom de l'unité du résultat de la multiplication.
<b>T\$</b>	Nom de l'unité supérieure.
<b>X et Y</b>	Coordonnées pour l'affichage du résultat.
<b>R1</b>	Réponse de l'enfant.
<b>R2</b>	Réponse de la multiplication + retenue.

Si **R1** (ou **R2**) est formé de deux chiffres, EXL 100 le décompose en :

<b>S</b>	Chiffre de droite (celui qui sera affiché dans l'opération).
<b>RET</b>	Chiffre de gauche (valeur de la retenue).
<b>STA,STB, STC,etc.</b>	Après chaque opération simple, la variable S est gardée en mémoire sous un nom différent. Chaque chiffre des lignes de multiplication peut ainsi être utilisé séparément dans l'addition.

Dans le cas maximal d'une multiplication par un nombre à trois chiffres, voici l'affichage de l'opération avec les noms des variables (un par chiffre) :

			AC	AD	AU
	X		BC	BD	BU
			STD	STC	STB
			STH	STG	STF
	STL	STK	STJ	STI	STA
	TF	TE	TD	TC	TB
					TA

## Modifications

Ce programme visualise une présentation claire de la multiplication et l'enfant y est guidé pas à pas.

Les erreurs sont immédiatement détectées dans chaque étape et seules les réponses exactes peuvent être inscrites dans l'opération. Il est donc très complet.

On peut, toutefois, ajouter, en cas d'erreur d'addition (de retenue, ou dans l'addition finale) l'affichage d'une table d'addition (présentée en deux parties côté à côté, comme la table de multiplication pour des raisons de place sur l'écran).

On peut regretter que le nom des unités (unités, dizaines, centaines, etc.) étant écrit au pluriel car convenant 8 fois sur 10, on risque de voir afficher « 0 dizaines » ou « 1 dizaines » par exemple. Afin d'éviter cette faute d'orthographe sur l'écran, deux solutions :

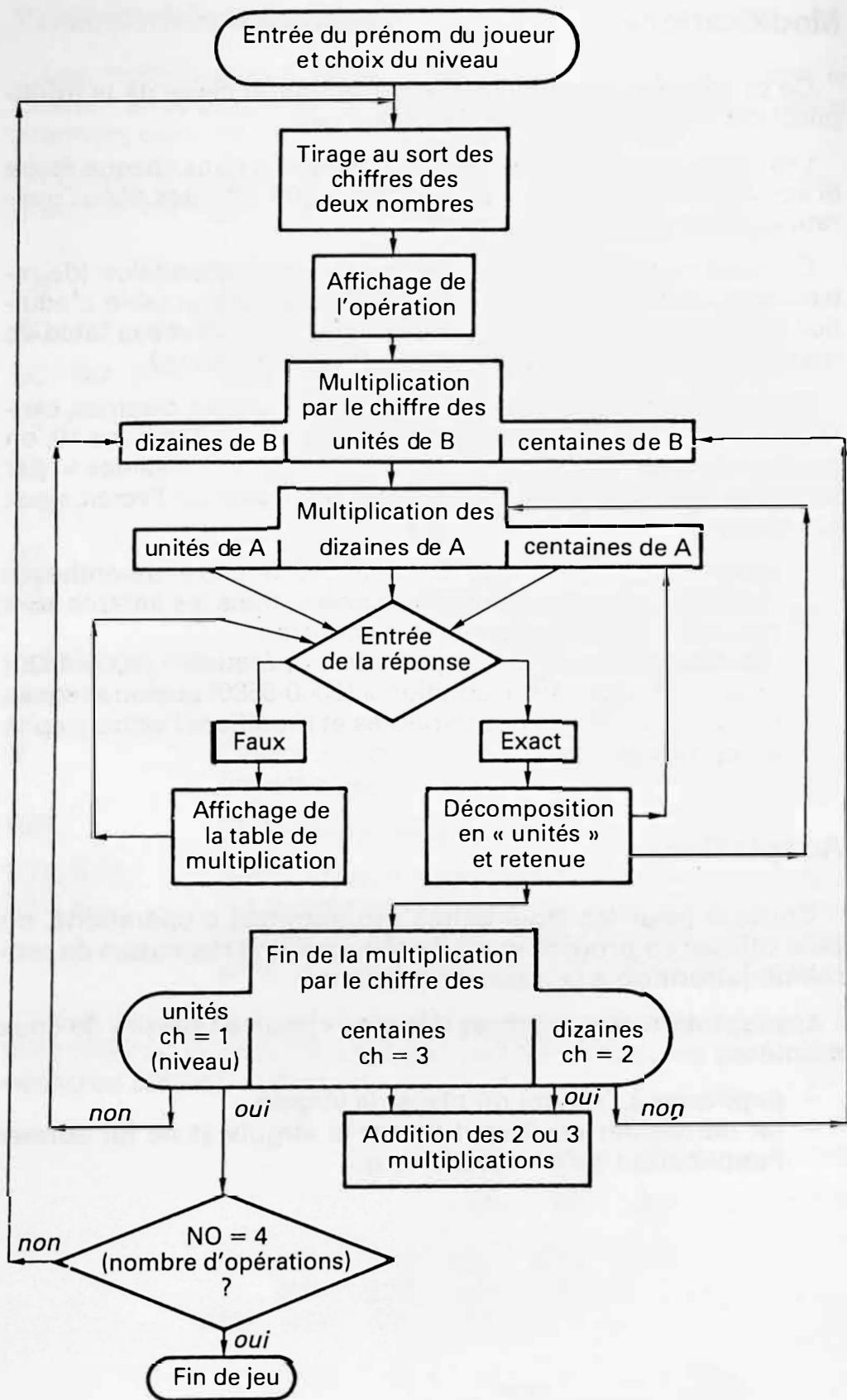
- écrire dans les lignes 31 à 36 le S entre parenthèses UNITE(S), comme dans « addition », mais les enfants sont peu sensibles à cette forme d'écriture ;
- ajouter dans le sous-programme « opération » (4000-4320) et dans la séquence « addition » (5000-5580) quelques lignes testant les valeurs des variables et modifiant l'orthographe en conséquence.

## Adaptations

Comme pour les trois autres programmes d'opérations, on peut utiliser ce programme à l'intérieur d'une résolution de problème (attention à la place en mémoire).

L'adaptation aux nombres décimaux pourrait se faire de deux manières :

- expliquer à l'enfant où placer la virgule ;
- lui demander où il veut placer la virgule et ne lui donner l'explication qu'en cas d'erreur.



```

100 !***** MULTIPLICATION *****
110 !***** VERSION EXL 100 *****
120 !***** D.NIELSEN - 1985 *****
130 !*****
140 ON BREAK ERROR
150 ON ERROR 10000
160 CLS "BCC":CALL HROFF:CALL INIT:CALL COLOR("0BC"):D
    IM A$(6)
170 NO=0:FOR I=1 TO 6:READ A$(I):NEXT:Z2$="MULTIPLICAT
    ION"
175 DATA unites,dizaines,centaines,milliers
176 DATA dizaines de milliers,centaines de milliers
180 CALL DBLPRT(Z2$,"0b",6,12)
190 LOCATE (15,5):PRINT "Quel est ton prenom ?";
200 ACCEPT BEEP VALIDATE(ALPHA)NULL("EXEL"),N$
210 PRINT:PRINT "Multiplications par un nombre a UN,DE
    UX";"ou TROIS chiffres ?"
220 CALL COLOR("0R"):PRINT "Indique ton choix (1,2 ou
    3) : ";
230 ACCEPT BEEP VALIDATE("123")NULL(1)SIZE(1),CH
240 RANDOMIZE LEN(N$)*7
250 !*** CHOIX DES CHIFFRES
260 CALL HRON("C",18,1):CALL INIT:CLS
270 AC,BC,AD,BD,AU,BU=0:ON CH GOTO 300,290,280
280 BC=INTRND(9)
290 BD=INTRND(9)
300 BU=INTRND(9)
305 AU=INTRND(9):AD=INTRND(9):AC=INTRND(9)
310 !*** AFFICHAGE DE L'OPERATION
320 A=100*AC+10*AD+AU
330 B=100*BC+10*BD+BU
340 CALL CLR
350 CALL DBLPRT(STR$(A),"0B",18,2):CALL DBLPRT("x "&ST
    R$(B),"0B",20-2*CH,4)
360 CALL DBLPRT(RPT$(CHR$(4),2+CH),"0B",20-CH*2,6)
370 LOCATE (19,1)
390 IF CH=1 THEN 401
400 PRINT "Tu vas commencer par multiplier";A:PRINT "p
    ar";BU;"unites.":PAUSE 5
401 X=22:Y=8:N1=AU:N2=BU
402 T$=A$(2):V$=A$(1):GOSUB 4000
403 STA=S:X=20:N1=AD
404 T$=A$(3):V$=A$(2):GOSUB 4000
405 STB=S:X=18:N1=AC
510 T$=A$(4):V$=A$(3):GOSUB 4000
530 STC=S:STD=RET
540 IF RET<>0 THEN CALL DBLPRT(STR$(RET),"0R",X-2,Y)
542 IF CH=1 THEN 5600
545 LOCATE (1,1):PRINT RPT$(" ",40)

```

```

550 CLS:LOCATE (19,1):PRINT "Tu vas maintenant multipl
ier";A,,,"par";BD;A$(2)
560 RET=0
570 PAUSE 5:X=20:Y=10:N1=AU:N2=BD
620 T$=A$(3):V$=A$(2):GOSUB 4000
640 STE=S:X=18:N1=AD
700 T$=A$(4):V$=A$(3):GOSUB 4000
730 STF=S:X=16:N1=AC
810 T$=A$(5):V$=A$(4):GOSUB 4000
830 STG=S:STH=RET
840 IF RET<>0 THEN CALL DBLPRT(STR$(RET),"0R",X-2,Y)
845 LOCATE (1,1):PRINT RPT$(" ",40)
850 RET=0
860 IF CH=2 THEN 5000
870 CLS:LOCATE (19,1):PRINT "Tu vas maintenant multipl
ier";A,,,"par";BC;A$(3)
880 PAUSE 5:X=18:Y=12:N1=AU:N2=BC
920 T$=A$(4):V$=A$(3):GOSUB 4000
940 STI=S:X=16:N1=AD
1010 T$=A$(5):V$=A$(4):GOSUB 4000
1030 STJ=S:X=14:N1=AC
1110 T$=A$(6):V$=A$(5):GOSUB 4000
1120 IF RET<>0 THEN CALL DBLPRT(STR$(RET),"0R",X-2,Y)
1130 STK=S:STL=RET
1137 LOCATE (1,1):PRINT RPT$(" ",40)
1140 GOTO 5000
3000 !*** TABLE DE MULTIPLICATION
3010 CLS:LOCATE (19,1):CALL COLOR("0RB"):PRINT "Tu te t
rompes,";N$;"."
3015 CALL COLOR("0BC"):PRINT:PRINT "Relis la table de m
ultiplication."
3020 PAUSE 7:CLS:LOCATE (19,1)
3040 FOR I=1 TO 10
3050 K=INT((I-1)/3):LOCATE (K+19,12*(I-1-3*K)+1)
3060 PRINT I;"x";N2;"=";I*N2;",";
3070 PAUSE 1:NEXT
3080 PAUSE 2:RETURN
4000 !*** OPERATION PAR UNITES
4005 CLS:LOCATE (19,1):CALL COLOR("0BC")
4010 PRINT "Opération";N1;"x";N2;" ?";
4020 ACCEPT BEEP VALIDATE(DIGIT)NULL(0),R1
4030 IF R1<>N1*N2 THEN GOSUB 3000:GOTO 4000
4100 IF RET<>0 THEN GOSUB 4500
4110 IF R1<10 THEN CALL DBLPRT(STR$(R1),"0b",X,Y):S=R1:
GOTO 4200
4115 RET=INT(R1/10):S=R1-(10*RET)
4120 CLS:PRINT "Cela fait";RET;T$;" plus";S;V$#
4130 PRINT "Il faut donc écrire";S;"dans la colonne"
4135 PRINT "des ";V$;" et ajouter";RET

```

```

4140 PRINT "a la multiplication des ";T$:PAUSE 5
4150 CALL DBLPRT(STR$(S),"0b",X,Y):LOCATE (1,X-1):PRINT
    RET
4200 PAUSE 8:RETURN
4500 !*** RETENUE
4505 CLS:LOCATE (19,1):PRINT "Il faut maintenant ajoute
r la retenue de"
4506 PRINT "la multiplication precedente"
4510 PRINT "Que fait";R1;"+";RET;" ?";
4520 ACCEPT BEEP NULL(0)VALIDATE(DIGIT),R2
4530 IF R2=R1+RET THEN 4600
4540 CALL COLOR("0RB"):PRINT "Tu te trompes,";N$;".":CA
LL COLOR("0BC")
4550 PRINT "Recommence.":PAUSE 3:GOTO 4500
4600 R1=R2:RET=0:RETURN
5000 !*** ADDITION
5005 CALL DBLPRT(RPT$(CHR$(4),3+CH),"0B",18-CH*2,14)
5010 LOCATE (19,1):CLS:PRINT "Il faut maintenant additi
onner les ";CH
5020 PRINT "multiplications."
5030 PRINT "D'abord,les ";A$(1);".":"Combien cela fait-
il ?";
5040 ACCEPT BEEP VALIDATE(DIGIT)NULL(0),TA
5050 IF TA<>STA THEN GOSUB 5700:GOTO 5030
5100 CALL DBLPRT(STR$(TA),"0B",22,16)
5110 CLS:LOCATE (19,1):PRINT "Ensuite les ";A$(2)
5120 PRINT STB;"+";STE;"= ?";
5130 ACCEPT BEEP VALIDATE(DIGIT)NULL(0),TB
5140 IF TB<>STE+STB THEN GOSUB 5700:GOTO 5120
5200 CALL DBLPRT(STR$(TB-10*INT(TB/10)),"0B",20,16)
5210 CLS:LOCATE (19,1):PRINT "Ensuite les ";A$(3)
5220 PRINT STC;"+";STF;"+";STI;"+";INT(TB/10);"(de rete
nue) = ?";
5230 ACCEPT BEEP VALIDATE(DIGIT)NULL(0),TC
5240 IF TC<>STC+STF+STI+INT(TB/10)THEN GOSUB 5700:GOTO
    5220
5300 CALL DBLPRT(STR$(TC-10*INT(TC/10)),"0B",18,16)
5310 CLS:LOCATE (19,1):PRINT "Ensuite les ";A$(4)
5320 PRINT STD;"+";STG;"+";STJ;"+";INT(TC/10);"(de ete
nue) = ?";
5330 ACCEPT BEEP VALIDATE(DIGIT)NULL(0),TD
5340 IF TD<>STD+STG+STJ+INT(TC/10)THEN GOSUB 5700:GOTO
    5320
5400 CALL DBLPRT(STR$(TD-10*INT(TD/10)),"0B",16,16)
5405 IF CH<>2 OR STH<>0 THEN 5410
5406 IF TD/10>=1 THEN CALL DBLPRT(STR$(INT(TD/10)),"0B"
    ,14,16)
5407 GOTO 5600
5410 CLS:LOCATE (19,1):PRINT "Ensuite les ";A$(5)

```

```

5420 PRINT STH;"+";STK;"+";INT(TD/10);"(de retenue) = ?
";
5430 ACCEPT BEEP VALIDATE(DIGIT)NULL(0),TE
5440 IF TE<>STH+STK+INT(TD/10)THEN GOSUB 5700:GOTO 5420
5500 CALL DBLPRT(STR$(TE-10*INT(TE/10)),"0B",14,16)
5510 IF STL=0 THEN CALL DBLPRT(STR$(INT(TE/10)),"0B",12
,16)
5515 IF CH=2 OR STL=0 THEN 5600
5520 CLS:LOCATE (19,1):PRINT "Enfin les ";A$(6)
5530 PRINT STL;"+";INT(TE/10);"(de retenue) = ?";
5540 ACCEPT BEEP VALIDATE(DIGIT)NULL(0),TF
5550 IF TF<>STL+INT(TE/10)THEN GOSUB 5700:GOTO 5520
5580 CALL DBLPRT(STR$(TF),"0B",12,16)
5600 CLS:LOCATE (19,1):CALL COLOR("0BG")
5605 PRINT "TU AS TERMINE CETTE MULTIPLICATION":CALL CO
LOR("0BC"):PRINT
5610 NO=NO+1
5620 IF NO=4 THEN 10000
5630 PRINT:PRINT "Voici une autre operation...":PAUSE 5
5650 CLS:GOTO 270
5700 !*** ERREUR ADDITION
5710 PRINT:PRINT "TU TE TROMPES ,";N$;" !"
5715 PRINT "REFLECHIS ET RECOMMENCE CETTE ADDITION"
5720 PAUSE 5:CLS:RETURN
10000 CALL HROFF:CALL DBLPRT("C'EST TERMINE.,","0B",1,12)
10010 CALL DBLPRT("AU REVOIR,"&N$,"0b",1,15)
10020 END
10030 CALL NORMAL:LOCATE (1,1):RETURN 10000
11600 SUB INIT
11700 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
50850)
11800 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)
11900 SUBEND
12000 SUB DBLPRT(Z$,C$,X,Y)
12100 CALL COLOR(C$):CALL COLOR("0LH")
12200 K$="":FOR I=1 TO LEN(Z$)
12300 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
12400 LOCATE (Y,X):PRINT K$
12500 LOCATE (Y+1,X):PRINT K$
12600 CALL NORMAL:SUBEND
12700 SUB NORMAL
12800 CALL COLOR("0BC")
12900 SUBEND
13000 SUB CLR
13100 FOR I=1 TO 22:LOCATE (I,1):PRINT RPT$(" ",40):NEXT
13200 SUBEND

```

# **DIVISION**

## **Thème**

Didact d'apprentissage de la technique opératoire de la division (cinq options différentes).

## **Niveau**

A partir du Cours moyen 1<sup>re</sup> année. 9 ans et plus.

## **Intérêt pédagogique**

### **Connaissances**

- élaboration et décomposition de la technique opératoire de la division par un nombre à un ou deux chiffres ;
- notion de multiple, de diviseur ;
- réorganisation des acquis sur l'addition, la multiplication, la soustraction et l'ordre des nombres.

### **Savoir-faire**

- apprentissage (ou rappel) d'une présentation écrite de la division ;
- mémorisation des opérations élémentaires et utilisation dans les séquences opératoires.

## Déroulement

Le programme propose, pour quatre divisions, un découpage en opérations simples, comme pour les trois autres opérations.

Il y a ici cinq choix de niveau :

1. Dividende (B) compris entre 1 et 70.  
Diviseur (A) compris entre 1 et 8.
2. B compris entre 1 et 140.  
A compris entre 1 et 12.
3. B compris entre 1 et 210.  
A compris entre 1 et 16.
4. B compris entre 1 et 280.  
A compris entre 1 et 20.
5. B compris entre 1 et 350.  
A compris entre 1 et 24.

L'opération, selon le niveau choisi, s'affiche, en double grandeur, en haut de l'écran. EXL 100 demande à l'enfant par quel chiffre de B la division doit être commencée.

Si le nombre B est supérieur à 99, le programme demande à l'enfant s'il peut diviser les centaines par A. Si c'est possible, il doit calculer le quotient entier, puis le reste.

Sinon, EXL 100 propose de réunir les centaines et les dizaines et demande le quotient du nombre obtenu par A, puis le reste. La division continue sur le même principe : le reste du quotient précédent est ajouté à l'opération suivante.

En cas de réponse fausse dans une de ces recherches de quotient simple, sur la moitié basse de l'écran, la partie de B qui est divisée est représentée en rangées de A « bâtons » (plus un reste éventuel), ce qui visualise le mécanisme de la division, ou partage en parts égales. L'enfant doit alors compter le nombre de rangées (quotient entier) et le reste, puis donner une nouvelle réponse à l'opération demandée.

Chaque bonne réponse (quotient et reste) s'affiche à sa place dans l'opération en haut de l'écran.

## Commentaires

<b>10 à 65</b>	Initialisation.
<b>70 à 200</b>	Choix du niveau.
<b>210 à 290</b>	Tirage au sort des nombres A et B.

- 300 à 500** Décomposition de A et B en centaines, dizaines, unités et affichage de l'opération.  
*Lignes 350 à 400* : Dessinent les deux traits matérialisant l'opération.
- 505 à 790** Division des centaines.  
*Ligne 505* : Envoie au sous-programme (ligne 2500) pour déterminer par où l'opération doit être commencée.  
*Ligne 510* : Demande si la division est possible.  
*Lignes 600 à 660* : Entrée du quotient entier. Test réponse et message de réponse fausse.  
*Lignes 700 à 790* : Entrée du reste et affichage du résultat.
- 800 à 1090** Division des dizaines.  
*Lignes 800 à 840* : Réunissent les centaines et les dizaines si la division des centaines était impossible.  
*Lignes 850 à 910* : Entrée du quotient entier (division des centaines et des dizaines ou du reste de la division des centaines plus les dizaines). Test réponse et message d'erreur.  
*Lignes 1000 à 1090* : Entrée du reste et affichage.
- 1100 à 1255** Division des unités (plus reste de la division des dizaines).  
*Lignes 1100 à 1170* : Entrée du quotient entier. Test réponse et message de réponse fausse.  
*Lignes 1200 à 1255* : Entrée du reste et affichage.
- 1290 à 1330** Fin d'opération.
- 2000 à 2020** Fin du jeu.
- 2500 à 2920** Sous-programme début d'opération : demande par où l'opération doit être commencée et teste la réponse en fonction du nombre B.
- 3000 à 3150** Erreur dans le quotient : affichent en bas de l'écran le nombre à diviser (V), représenté par des « étoiles » en rangées de A (diviseur) bâtons, plus le reste.
- 4000 à 4040** Evitent une erreur lorsque le nombre de rangées d'étoiles dépasse 5. Un texte indique les mêmes valeurs et remplace ainsi le schéma qui sortirait de l'écran.

**10000 à 10030** Initialisation générale.

**10040 à 10090** Jouent une mélodie.

**10100 à 10160** Ecriture double taille.

## Particularités techniques

A l'inverse des trois autres programmes d'opérations, ce sont les deux nombres qui sont tirés au sort, et non leurs chiffres. Ils sont ensuite décomposés en centaines, dizaines, unités.

En cas d'erreur de calcul, on aurait pu afficher une table de multiplication, mais il a semblé plus concret de « matérialiser » ce que représente la division sous forme de rangées de points (lignes 3000 à 3150) :

**3010 à 3060** Affichage de ces rangées égales par une double boucle :

FOR I = I to A nombre total de points à afficher  
FOR J = I to INT(V/A) : nombre de rangées

**3070 à 3090** Affichage du reste.

Les lignes 4000 à 4040 interviennent lorsque le nombre de rangées est trop grand (sortie de l'écran) afin d'éviter une interruption du déroulement du programme pour cause d'erreur.

## Modifications

Le choix des nombres est tiré au sort, en fonction du niveau des opérations (N), dans les lignes 280 à 290. Il peut évidemment être modifié, en particulier pour B qui peut être choisi entre 1 et 999 en écrivant :

$$B = \text{INT}((19 \times Y + 1) \times N \times 10)$$

La valeur du diviseur est limitée par l'affichage des rangées de points, en cas d'erreur de calcul : 40 points maximum sur une rangée.

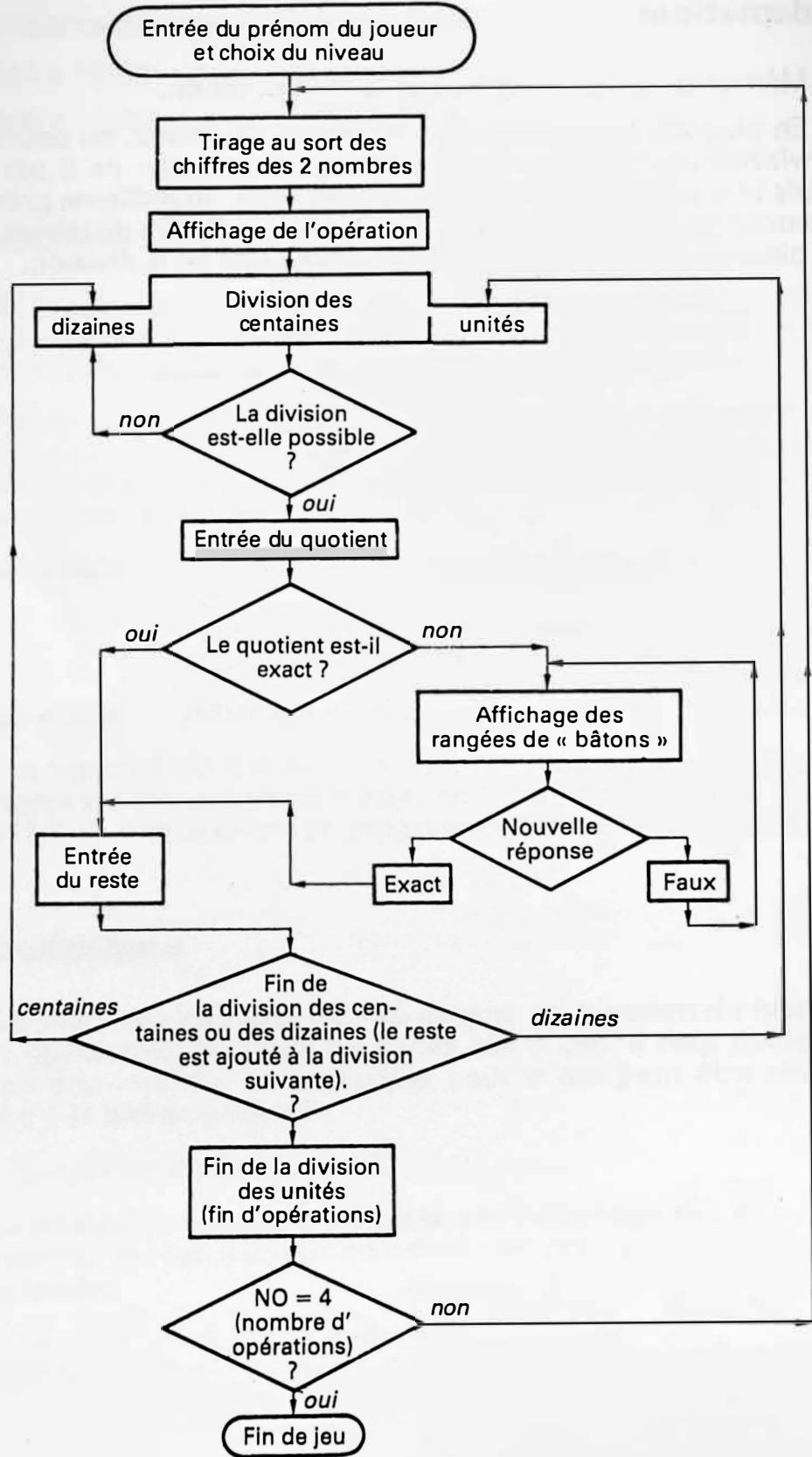
Il ne faut donc pas que A dépasse 40, faute de quoi cet affichage serait perturbé. La plus grande valeur de A serait donc :

$$A = \text{INT}(Y \times N \times 8) + 1$$

## Adaptations

Mêmes possibilités que pour la multiplication.

En plus de l'adaptation aux nombres décimaux, on pourrait envisager de calculer, non pas le quotient entier de B par A, mais le quotient à un dizième, un centième, un millième près. Il faudrait modifier l'affichage des chiffres de B, afin de laisser de la place entre le chiffre des unités et la barre de la division.



```

2 !***** DIVISION *****
3 !***** VERSION EXL 100 *****
4 !***** D.NIELSEN - 1985 *****
5 !***** *****
7 CALL HROFF:CALL INIT:CLS "BCC":CALL COLOR("0RC")
20 ON ERROR 4000
40 CALL DBLPRT("DIVISION",5,10)
60 LOCATE (14,5):PRINT "Quel est ton prenom ? ";
65 ACCEPT BEEP VALIDATE(ALPHA)NULL("INCONNU"),N$
70 CLS:CALL COLOR("0b"):LOCATE (1,1):PRINT "Niveau de
   difficulte":PRINT
100 PRINT "FACILE ..... 1":PRINT
110 PRINT "ASSEZ FACILE .... 2":PRINT
120 PRINT "MOYEN ..... 3":PRINT
130 PRINT "ASSEZ DIFFICILE .. 4":PRINT
140 PRINT "DIFFICILE ..... 5":PRINT
200 CALL COLOR("0R"):PRINT "Ton choix ? ";
210 ACCEPT BEEP VALIDATE("12345")NULL(3),N
230 W,X=0
240 RANDOMIZE LEN(N$)
275 Y=RND
280 A=INT(Y*4+N*4)
290 B=INT((6*Y+1)*N*10)
295 !AFFICHAGE DES NOMBRES
300 W=W+1
310 IF W>5 THEN 2000
330 CLS:CALL COLOR("1b"):CALL HRON("C",13,1):CALL INIT

350 FOR I=1 TO 11
360 LOCATE (I,17):PRINT "Z"
370 NEXT
380 FOR I=1 TO 5
390 LOCATE (4,I+17):PRINT "Z"
400 NEXT
410 CALL DBLPRT(STR$(A),19,1)
420 IF B>99 THEN BC=INT(B/100) ELSE BC=0
430 BD=INT((B-BC*100)/10)
440 BU=B-BC*100-BD*10
450 IF B>100 THEN 500
460 CALL DBLPRT(STR$(B),9,1)
470 GOTO 505
500 CALL DBLPRT(STR$(B),7,1)
505 GOSUB 2500
507 IF B<99 THEN 800
510 LOCATE (16,5):PRINT "PEUX-TU DIVISER";BC;"PAR";A;;
   ? (O/N)":PRINT C$
520 ACCEPT BEEP VALIDATE("ON")BEEP NULL("0"),P$
540 IF P$="0"AND BC/A=1 THEN 600
550 IF P$="N"AND BC/A<1 THEN 800

```

```

560 PRINT
570 PRINT "REFLECHIS BIEN,";N$;" TU TE TROMPES ."
580 PAUSE 5
590 CLS:GOTO 510
600 CLS:LOCATE (16,1):PRINT "Avec";BC;,"combien peux-tu faire"
605 PRINT "de rangees de";A;" ?";
615 ACCEPT BEEP VALIDATE(DIGIT)NULL(BC/A+1),RC
620 IF RC=INT(BC/A)THEN 700
630 PRINT "Ce n'est pas cela.Regarde bien :"
640 BC=V
650 GOSUB 3000
660 GOTO 600
700 CALL DBLPRT(STR$(RC),19,7)
710 LOCATE (16,1):CLS:PRINT "Combien reste t'il ? ";
715 ACCEPT BEEP VALIDATE(DIGIT)NULL(BC),SC
720 IF SC=BC-(INT(BC/A)*A)THEN 750
730 PRINT:PRINT "CE N'EST PAS CELA,";N$;".":PRINT "Donne une autre reponse"
740 CALL PLAY("DODOREREDOREDORE",2,3):PAUSE 3:CLS:LOCATE (16,1):GOTO 710
750 CLS:LOCATE (16,1):PRINT "C'EST EXACT !":CALL PLAY("DOREMIDOREMIDOREMI",1,0)
760 CALL DBLPRT(STR$(SC),8,4)
770 V=SC*10+BD
780 CALL DBLPRT(STR$(BD),10,7)
790 GOTO 850
800 V=BC*10+BD:IF BC=0 THEN 850
805 LOCATE (16,1):CLS:PRINT "Tu vas maintenant reunir ensemble ";"les";BC;
810 PRINT "centaines et les";BD;"dizaines"
820 PRINT "Cela fait";V;"dizaines."
830 NL=3
840 PAUSE 5
850 CLS:LOCATE (16,1)
870 PRINT "Avec";V;"combien peux-tu faire de ";"rangees de ";A;" ?";
875 ACCEPT BEEP VALIDATE(DIGIT)NULL(V),RD
880 IF RD=INT(V/A)THEN 1000
890 PRINT N$&" CE N'EST PAS CELA ,REGARDE BIEN"
900 GOSUB 3000
910 GOTO 850
1000 CALL DBLPRT(STR$(RC*10+RD),19,7)
1010 LOCATE (16,1):CLS:PRINT "Combien reste-t'il ? ";
1015 ACCEPT BEEP VALIDATE(DIGIT)NULL(V),SD
1020 IF SD=V-INT(V/A)*A THEN 1050
1030 PRINT "CE N'EST PAS CELA,"&N$&","DONNE UNE AUTRE RE PONSE":PAUSE 3
1040 CLS:LOCATE (16,1):GOTO 1010

```

```

1050 PRINT "C'EST EXACT !"
1070 IF SD<10 THEN CALL DBLPRT(STR$(SD*10+BU),9,7-NL)
1080 IF SD>9 THEN CALL DBLPRT(STR$(SD*10+BU),7,7-NL)
1090 IF SD=0 THEN CALL DBLPRT(STR$(SD*10+BU),11,7-NL)
1100 V=SD*10+BU
1120 CLS:LOCATE (16,1):PRINT "Avec";V;,"combien peux-tu
 faire de"
1130 PRINT "rangees de";A;" ? ";
1135 ACCEPT BEEP VALIDATE(DIGIT)NULL(V),RU
1140 IF RU=INT(V/A)THEN 1200
1150 PRINT N$;,"ce n'est pas cela.";"Regarde bien ."
1160 GOSUB 3000
1170 GOTO 1120
1200 CALL DBLPRT(STR$(RC*100+RD*10+RU),19,7)
1210 CLS:LOCATE (16,1):PRINT "Combien reste-t'il ? ";
1215 ACCEPT BEEP VALIDATE(DIGIT)NULL(V),SU
1220 IF SU=V-INT(V/A)*A THEN 1250
1230 PRINT "CE N'EST PAS CELA,";N$;,"DONNE UNE AUTRE RE
 PONSE.":PAUSE 5
1235 CALL PLAY("DODOREREDOREDORE",1,3)
1240 CLS:GOTO 1210
1250 PRINT "C'EST EXACT":CALL PLAY("DOREMIDOREMIDOREMI"
 ,1,0)
1255 CALL DBLPRT(STR$(SU),11+2*(SU>10),10-NL)
1290 CLS:LOCATE (16,1):PRINT "BONNE OPERATION,";N$
1305 X=X+1
1310 IF W<4 THEN PRINT "VOICI MAINTENANT UNE AUTRE DIVI
 SION ..."
1320 PAUSE 5
1330 GOTO 275
2000 CLS:CALL DBLPRT("C'EST TERMINE",1,16)
2020 RUN
2500 CLS:LOCATE (16,1)
2510 PRINT "PAR OU COMMENCES-TU LA DIVISION ? "
2520 PRINT "1:Les unites":PRINT "2:Les dizaines"
2540 IF B>99 THEN PRINT "3:Les centaines";
2560 PRINT "TAPE LE NUMERO DE TON CHOIX : ";
2565 A$=KEY$:IF A$<"1"OR A$>"3"THEN 2565 ELSE CH=VAL(A$
 )
2570 ON CH GOTO 2600,2700,2800
2600 PRINT "TU TE TROMPES,";N$;,"REFLECHIS ENCORE"
2605 PAUSE 5
2610 GOTO 2500
2700 IF B<100 THEN 2900
2710 GOTO 2600
2800 IF B>99 THEN 2900
2810 GOTO 2600
2900 PRINT "C'EST EXACT,";N$
2910 PAUSE 5

```

```

2920 CLS:LOCATE (16,1):RETURN
3000 PAUSE 5
3005 LOCATE (16,1):CLS
3010 FOR I=1 TO A
3020 FOR J=1 TO INT(V/A)
3030 LOCATE (17+J,I):PRINT CHR$(42)
3040 NEXT J
3060 NEXT I
3070 FOR J=1 TO V-INT(V/A)*A
3080 LOCATE (18+INT(V/A),J):PRINT CHR$(42)
3090 NEXT J
3110 LOCATE (16,1):CALL COLOR("0B"):PRINT "Combien y a
t'il de rangs de";A;" ?";
3115 A$=KEY$:IF A$<"0" OR A$>"9" THEN 3115 ELSE T=VAL(A$)

3120 IF T<>INT(V/A)THEN 3005
3130 PRINT "Maintenant recommence l'opération"
3140 PAUSE 5
3150 CLS:LOCATE (16,1):RETURN
4000 CALL ERR(ER1,ER2,ER3,ER4)
4004 CLS
4005 IF ER1<>29 OR ER4<>3030 THEN 2020
4010 LOCATE (16,1):PRINT "ON A";INT(V/A);;"RANG";:IF INT
(V/A)>1 THEN PRINT "S";
4020 PRINT " DE";A
4030 PRINT " IL RESTE";V-INT(V/A)*A
4035 PAUSE 5:CLS
4040 RETURN 3110
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10025 CALL CHAR(90,"oooooooooooooooooooo")
10030 SUBEND
10040 SUB PLAY(N0$,L,D)
10050 FOR I=1 TO LEN(N0$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(N0$,I,2),1)
10070 P=VAL(SEG$("199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,X,Y)
10110 CALL COLOR("0LH")
10120 K$="" :FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("0"):SUBEND

```

# **RAVITAILLEMENT EN VOL**

## **Thème**

Découverte d'un nombre aléatoire compris entre 1 et 200 par encadrements successifs sous forme de jeu.

## **Niveau**

A partir de 7 ans.

## **Intérêt pédagogique**

### **Connaissances**

- exploration de la suite des nombres naturels de 1 à 200.

### **Savoir-faire**

- déduction ;
- mémoire ;
- raisonnement : recherche d'une technique permettant de trouver la solution en utilisant le moins de réponses possibles.

## Déroulement

« Ravitailler un avion en vol, c'est lui donner du carburant sans qu'il soit obligé d'atterrir. Pour cela, il faut que l'avion ravitailleur soit à la même altitude que l'avion de chasse. »

Cette altitude est fixée par un nombre aléatoire compris entre 1 et 200.

Le joueur est le pilote de l'avion ravitailleur. Il doit indiquer à l'EXL 100 l'altitude qu'il a choisie. Si cette altitude est plus élevée que celle de l'avion de chasse, l'avion ravitailleur se dessine sur l'écran au-dessus de l'avion de chasse. Si elle est trop basse, l'avion se dessine en dessous. Si l'altitude proposée est la bonne, une ligne relie les deux avions.

La position de l'avion ravitailleur indique donc, pour chaque coup, si le nombre proposé est trop petit ou trop grand.

A partir de la 10<sup>e</sup> altitude, un compteur indique le nombre de coups restant à jouer.

Si la bonne altitude n'est pas trouvée en 15 réponses, le joueur a perdu.

## Commentaires

- |                      |   |
|----------------------|---|
| <b>20 à 70</b>       | Définition du dessin de l'avion.  |
| <b>100 à 270</b>     | Présentation du jeu sur l'écran.  |
| <b>500 à 520</b>     | Choix de la bonne altitude par l'EXL 100.                                       |
| <b>900 à 950</b>     | Affichage de l'avion de chasse (en bleu).                                       |
| <b>1000 à 1100</b>   | Test de l'altitude proposée par le joueur et affichage de l'avion ravitailleur. |
| <b>1110 à 1220</b>   | Message d'erreur. Compteur de coups restant à jouer (si plus de 10 réponses).   |
| <b>2000 à 2030</b>   | Affichage réponse exacte.   |
| <b>3000 à 3010</b>   | Affichage « perdu ».  |
| <b>4000 à 4035</b>   | Proposent de rejouer ou d'arrêter.  |
| <b>10000 à 10030</b> | Initialisation générale.  |
| <b>10040 à 10090</b> | Jouent une mélodie.   |
| <b>10100 à 10160</b> | Ecriture double taille.   |

## Particularités techniques

Les deux avions sont dessinés à partir de caractères personnalisés (lignes 35 à 70).

Afin que la solution ne soit pas toujours la même, la bonne altitude est déterminée dans une séquence aléatoire, à partir de la première altitude proposée. C'est pourquoi cette première réponse est toujours trop grande (il faut donc au moins deux réponses pour gagner).

## Modifications

Lignes 510-520 : le nombre aléatoire choisi par l'EXL 100 comme altitude à trouver peut être déterminé par d'autres formules.

*Exemple :*

B = INTRND (200)

L'avion de chasse peut être ainsi à n'importe quelle altitude entre 1 et 200, mais la séquence de nombres aléatoires sera toujours la même, si elle n'est pas initialisée lors d'une instruction RANDOMIZE.

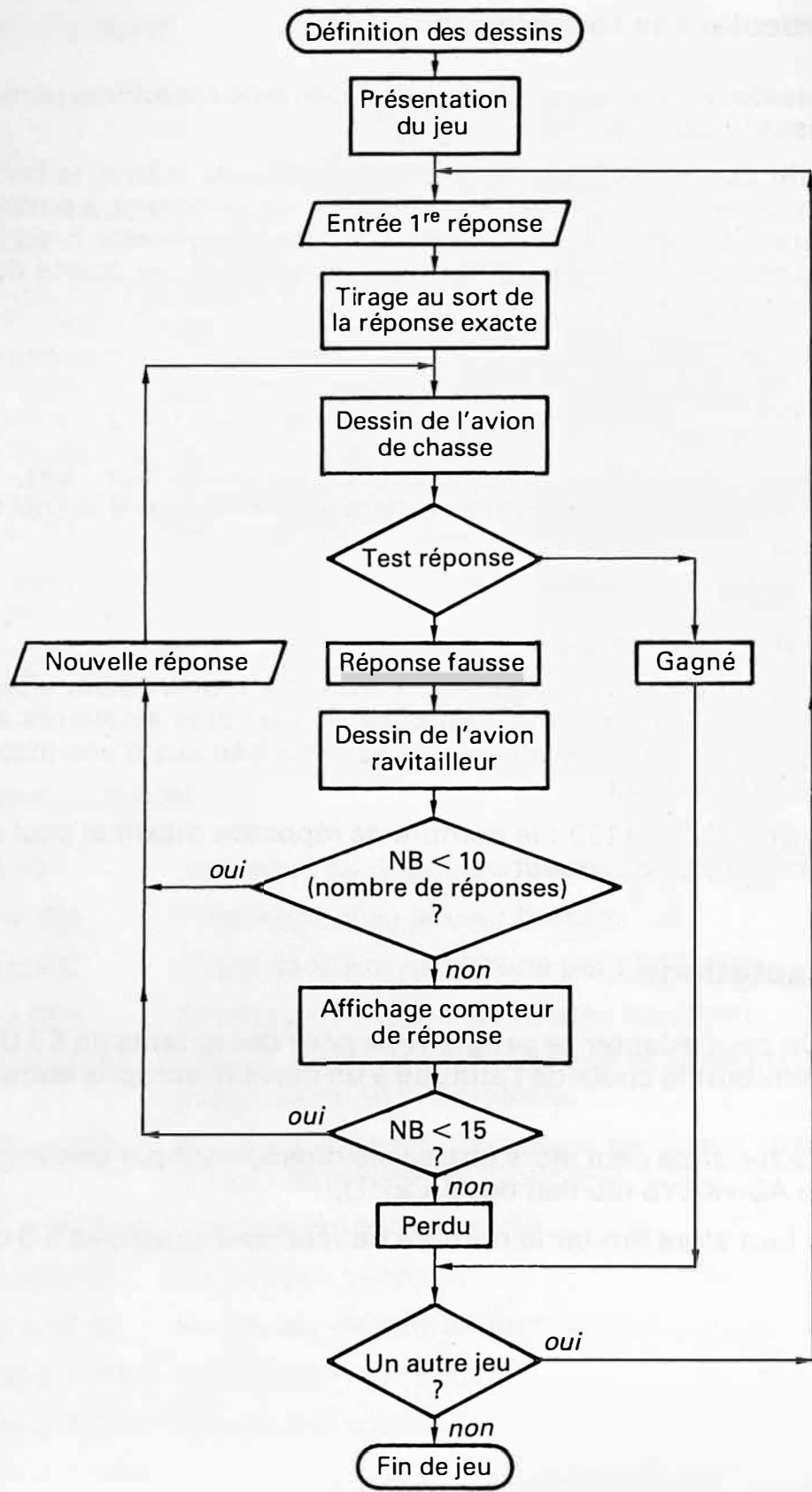
Lignes 1110-1130 : le nombre de réponses maximal peut être ajusté à l'âge du joueur.

## Adaptations

On peut adapter ce programme pour des enfants de 5 à 6 ans en limitant le choix de l'altitude à un nombre compris entre 1 et 9.

La réponse peut alors être saisie directement par une instruction A\$=KEY\$ (au lieu de ACCEPT).

Il faut alors limiter le nombre de réponses possibles à 5 ou 6.



```

2 !***** RAVITAILLEMENT EN VOL *****
3 !***** VERSION EXL 100 *****
4 !*** D.NIELSEN - 1985 *****
5 !***** *****
7 CALL INIT:CLS "BCC":CALL COLOR("0BC")
20 ON BREAK ERROR:ON ERROR 4000
35 CALL CHAR(48,"00000000060F1E3C0000")
40 CALL CHAR(49,"00001F36FF7F00000000")
45 CALL CHAR(50,"78F0FFAAFFFF0780000")
50 CALL CHAR(51,"0307FFBEFCFC00000000")
55 CALL CHAR(52,"3C1E0F07030000000000")
60 G$(1)=" 0"
65 G$(2)="123"
70 G$(3)=" 4"
100 CLS:CALL DBLPRT("RAVITAILLEMENT","0B",6,10)
110 CALL DBLPRT("EN VOL","0B",13,15)
120 PAUSE 5
130 CLS:LOCATE (5,5)
140 PRINT "RAVITAILLER UN AVION EN VOL, C'EST LUI DON
NER DU CARBURANT SANS ";
142 PRINT "QU'IL SOIT";:PRINT "OBLIGE D'ATTERRIR."
150 PRINT "POUR CELA, IL FAUT QUE L'AVION RAV
ITAILLEUR (en rouge) ";
151 PRINT "SOIT A LA MEME ALTITUDE QUE L'AVION DE CHA
SSE (en bleu)"
155 PAUSE 20
157 CLS:LOCATE (5,5)
210 PRINT "L'ALTITUDE EST COMPRISE ENTRE 0 ET 200 MET
RES"
220 PRINT "TU PILOTES L'AVION RAVITAILLEUR ET TU DOI
S DONNER TON ALTITUDE"
230 PRINT "Si ton altitude est trop elevee, ton avi
on (en rouge) monte au ";
232 PRINT "dessus de";:PRINT "l'avion de chasse"
240 PRINT "Si ton altitude est trop basse, ton avi
on (en rouge) descend ";
242 PRINT "en dessous de";:PRINT "l'avion de chasse"
250 PRINT "TU DOIS TROUVER L'ALTITUDE EXACTE"
260 PRINT:PRINT "SI TU ES PRET TAPE SUR 'C' ET 'PRET'
";
265 ACCEPT VALIDATE("C")SIZE(1)NULL("C"),P$
270 IF P$<>"C"THEN 260
500 C=0
505 CLS:PRINT "QUELLE ALTITUDE PROPOSES-TU ?"
506 ACCEPT VALIDATE(DIGIT)SIZE(3)NULL(0),A
510 IF A>200 OR A<1 THEN 500
520 IF A>100 THEN B=INT(RND*(A/10))*8+1 ELSE B=INT(RND
*(A/10))*15+1
900 CALL HRON("G",17,1)
901 LOCATE (1,1):FOR I=1 TO 3
910 LOCATE (6+I,5):CALL COLOR("1BC"):PRINT G$(I):CALL
COLOR("0BC")
930 NEXT I
950 LOCATE (18,1)
1000 ! ***** JEU *****
1005 IF A>B THEN Y=2
1010 IF A<B THEN Y=1

```

```

1015 IF A=B THEN Y=6
1030 FOR I=1 TO 3
1040 LOCATE (Y+I,20):CALL COLOR("1RC"):PRINT G$(I):CALL
    COLOR("0BC")
1050 NEXT I
1090 IF A=B THEN 2000
1100 C=C+1
1110 IF C>10 THEN LOCATE (22,1):PRINT "DEPECHE TOI, TU
    N'AS PLUS QUE";15-C;
1111 IF C>10 THEN PRINT " ALTITUDES A PROPOSER"
1115 IF C=14 THEN LOCATE (22,1):PRINT "C'EST TA DERNIER
    E CHANCE, TU N'AS PLUS ";
1116 IF C=14 THEN PRINT "QU'UNE ALTITUDE A PROPOSER"
1120 IF C=15 THEN 3000
1130 IF C<11 THEN 1210
1140 PAUSE 5
1210 CLS:LOCATE (20,1):PRINT "CE N'EST PAS LA BONNE ALT
    ITUDE. QUELLE";
1211 PRINT " NOUVELLE ALTITUDE PROPOSES-TU ? ";
1215 ACCEPT VALIDATE(DIGIT)SIZE(3)NULL(0),A
1220 GOTO 900
2000 !***** GAGNE *****
2005 LOCATE (22,1):CLS:PRINT "GAGNE, C'EST LA BONNE ALT
    ITUDE, TU PEUX"
2010 PRINT "RAÙTAILLER L'AVION":PRINT "TU AS GAGNE EN"
    ;C;" REPONSES"
2030 GOTO 4000
3000 !***** PERDU *****
3005 LOCATE (22,1):CLS:PRINT "PERDÜ, C'EST TROP TARD.L'
    AVION DE CHASSE N'A ";
3010 PRINT "PLUS DE CARBURANT"
4000 !***** FIN DU JEU *****
4001 PAUSE 10:CALL HROFF
4010 PRINT "UNE AUTRE PARTIE O/N";
4015 ACCEPT VALIDATE("ON")SIZE(1)NULL("C"),T$
4030 IF T$="O"THEN 500
4035 CLS:END
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
    50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10030 SUBEND
10040 SUB PLAY(N0$,L,D)
10050 FOR I=1 TO LEN(N0$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(N0$,I,2),1)
10070 P=VAL(SEG$("199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,C$,X,Y)
10105 CALL COLOR(C$)
10110 CALL COLOR("0LH")
10120 K$="" :FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("0BC"):SUBEND

```

# **LE TRAIN CAPRICIEUX**

## **Thème**

Jeu de questions à réponses multiples sous forme animée. Il s'agit d'accrocher à une locomotive, contenant un élément-question, le wagon contenant le bon élément-réponse, parmi cinq wagons dessinés sur l'écran.

## **Niveau**

En fonction des questions utilisées dans ces exercices.

Le système de réponse n'utilisant que les deux touches ↓ et ↑ pour déplacer la locomotive, ce jeu est accessible dès 4 ans.

## **Intérêt pédagogique**

### **Connaissances**

- selon les questions-réponses, les thèmes liés qui sont introduits au début du programme, il peut s'agir de révision, de contrôle, ou d'acquisitions nouvelles en tous domaines (français, maths, éveil).

### **Savoir-faire**

- rapidité ;
- attention ;
- esprit de décision ;
- globalisation d'une réponse.

## Déroulement

L'enseignant (ou le parent) entre d'abord pour le premier jeu le texte qu'il veut voir s'inscrire dans la locomotive puis cinq réponses possibles, dont une seule est exacte (EXL 100 vous précise quelle réponse doit être exacte). On entre ensuite les données du 2<sup>e</sup>, du 3<sup>e</sup>, puis du 4<sup>e</sup> jeu.

C'est alors au tour des enfants de venir. Après une explication succincte du jeu, l'écran se vide.

Une locomotive se dessine à gauche avec, à l'intérieur, l'élément-question n° 1, puis à droite cinq wagons de couleurs différentes avec les cinq éléments-réponses.

Les wagons avancent lentement vers la locomotive. L'enfant, en utilisant les touches ↑ et ↓, doit faire monter ou descendre la locomotive pour accrocher le bon wagon. Mais la locomotive est capricieuse, elle change parfois de voie toute seule, il faut être attentif jusqu'au bout et la remettre à sa bonne place !

Lorsque les wagons arrivent au niveau de la locomotive, EXL 100 affiche « EXACT » ou donne la bonne réponse selon le wagon qui a été accroché.

Le programme recommence ensuite au dessin de la locomotive avec le 2<sup>e</sup> jeu... etc., jusqu'au 4<sup>e</sup>.

## Commentaires

- |                    |   |
|--------------------|---|
| <b>10 à 46</b>     | Initialisation.   |
| <b>49 à 90</b>     | Entrée des données.   |
| <b>330 à 390</b>   | Définition des caractères personnalisés.  |
| <b>400 à 460</b>   | Présentation du jeu pour l'enfant.  |
| <b>500 à 900</b>   | Jeu : pour chacun des quatre jeux, on attribue aux variables V\$ (question), R\$(I) (réponses) et BR (numéro de la bonne réponse), la valeur qui correspond au jeu, puis le jeu est réalisé dans un sous-programme à partir de la ligne 1100. |
| <b>910 à 930</b>   | Le jeu recommence pour un autre joueur.   |
| <b>1100 à 1250</b> | Dessin de la locomotive sur trois lignes : la taille de la locomotive est ajustée à la longueur de la question (V\$) : lignes 1130 à 1145, 1170 à 1180,   |

- 1220 à 1240** (avec toutefois un minimum de quatre cases).
- 1300 à 1570** Dessin des cinq wagons. Les wagons sont aussi dessinés sur trois lignes de la même manière que la locomotive. Ils sont définis successivement à l'intérieur d'une boucle FOR K = 1 to 5 (ligne 1305)... NEXT K (ligne 1570).
- 1600** Envoie au sous-programme d'affichage de la locomotive.
- 1610 à 1690** Affichage des cinq wagons de couleurs différentes et de leurs réponses.
- 1700 à 1770** Déplacement de la locomotive et des wagons. L'ordinateur teste les entrées en permanence. Si l'une des touches ↑ ou ↓ est appuyée, sa valeur est testée (ligne 1720) et la variable T (qui détermine la hauteur de la locomotive sur l'écran) est modifiée en conséquence (lignes 1730 et 1750). La locomotive est alors redessinée à sa nouvelle place. Si aucune touche n'est appuyée, la variable DEP, qui détermine la place des wagons, augmente (ligne 1770) et les wagons avancent. Pour chaque « pas en avant » des wagons, la ligne 1715 renvoie au sous-programme de test de bonne réponse.
- 3000 à 3120** Sous-programme Affichage de la locomotive. La chaîne L\$(4) est formée de carrés de la couleur de fond cyan (lignes 3010 à 3050). Son rôle est d'effacer le précédent dessin de la locomotive, au fur et à mesure que le nouveau se forme.
- 4000 à 4540** Test bonne réponse. Si les wagons sont au niveau de la locomotive, il y a deux cas :
  - la locomotive a accroché le bon wagon (T=BR) lignes 4100 à 4120 : exact.
  - la locomotive a accroché un mauvais wagon : lignes 4500 à 4540 : la bonne réponse est affichée.
- 10000 à 10030** Initialisation.
- 10040 à 10090** Jouent une mélodie.
- 10100 à 10160** Ecriture double taille.

## Particularités techniques

Le jeu lui-même est exécuté par un sous-programme (GOSUB 1100) dont l'instruction RETURN est placée dans le test de réponse (lignes 4120 et 4540).

Au cours du déplacement de la locomotive et des wagons, chaque modification de place des wagons est testée (ligne 1715 GOTO 4000). Si les wagons ne sont pas arrivés au niveau de la locomotive, le jeu reprend à la ligne suivante (4030 GOTO 1720). Si un des wagons est accroché à la locomotive, la réponse est évaluée et le sous-programme se termine.

La locomotive et les wagons ont été dessinés en caractères personnalisés. Dans le dessin des wagons, un caractère de couleur du fond a été ajouté derrière chaque ligne, ce qui sert de « gomme » pour effacer les anciens éléments lorsque les wagons avancent.

## Modifications

Une temporisation du déplacement des wagons (ligne 1760) peut être ajoutée afin qu'ils avancent moins vite.

Lignes 1585-1590 : elles ne servent à rien actuellement. Après utilisation du programme par des enfants, il a semblé que le jeu était un peu lent lorsque le joueur avait correctement placé sa locomotive. C'est pourquoi le programme garde en mémoire les déplacements des jeux précédents (à partir du 2<sup>e</sup>) et les exécute tout seul, ce qui donne ce caractère « capricieux » et anime le jeu en obligeant à une plus grande attention.

Pour de très jeunes enfants, il suffit d'écrire ligne 1585 LOC(I) = 0 (au lieu de LOC(C) qui ne veut rien dire) pour redonner à ces deux lignes leur rôle, qui est d'effacer de la mémoire tous les déplacements précédents. Ainsi la locomotive ne bougera plus toute seule.

## Adaptations

Voici, à titre d'exemple, quelques exercices qui peuvent être étudiés dans ce programme .

## • Français

### *Conjugaison*

- Loco : un pronom (je, tu, il, etc.).
- Wagons : cinq conjugaisons dont une seule convient.

### *Orthographe*

- Loco : « la ».
- Wagons : « chat », « chatte », « chaton », « chats », « chattes ».

## • Maths

### *Calcul mental*

- Loco :  $6 \times 5$ .
- Wagons :  $32 - 30 - 28 - 35 - 40$ .

### *Evaluation*

- Loco :  $120 \times 36$ .
- Wagons :  $4212 - 4320 - 5415 - 4328 - 425$ .  
(l'enfant ne peut effectuer cette opération mentalement, mais doit deviner que la réponse se termine par 0, c'est un travail de raisonnement très utile).

## • Eveil

### *Histoire*

- Loco : Louis XIV.
- Wagon : 12<sup>e</sup> S, 15<sup>e</sup> S, 17<sup>e</sup> S, 19<sup>e</sup> S, 20<sup>e</sup> S, (Quand a-t-il vécu ?).

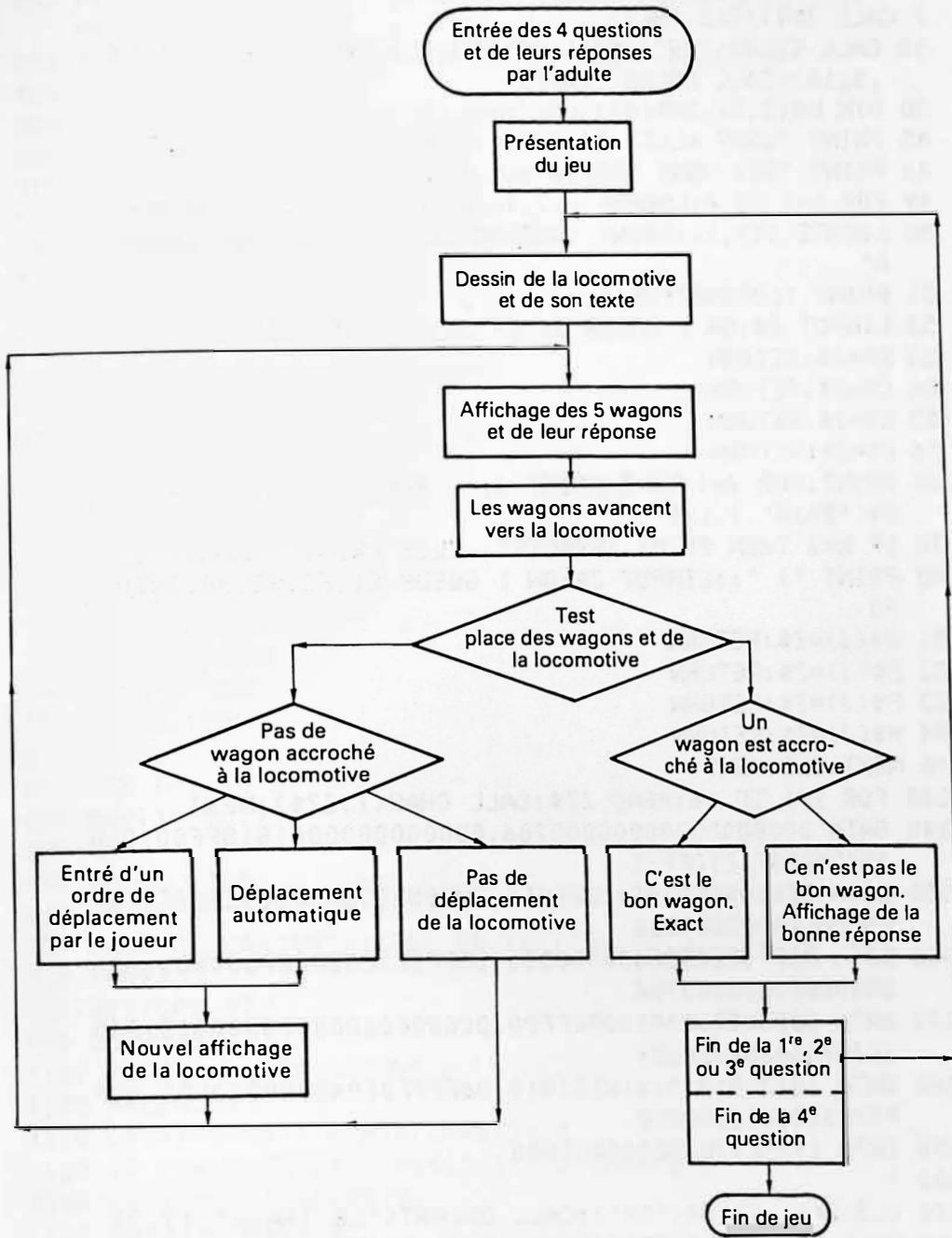
### *Géographie*

- Loco : Seine.
- Wagon : Marne, Saône, Somme, Dordogne, Cher (un seul est un affluent de la Seine).

Vous pouvez inventer vous-même de très nombreux autres sujets. Mais il faut respecter un impératif : le texte dans la locomotive et dans les wagons ne doit pas dépasser huit cases,

sinon le train aura une allure de mille-pattes ! Pour des questions et des réponses plus longues, il est préférable d'utiliser le programme « Soutien ».

Enfin, pour de très jeunes enfants, on pourrait, en adaptant le programme, remplacer le texte par une couleur : l'enfant devrait alors accrocher le wagon de la même couleur que la locomotive.



```

2 !***** LE TRAIN CAPRICIEUX *****
3 !***** VERSION EXL 100 *****
4 !***** D.NIELSEN - 1985 *****
5 !***** *****
6 DIM R$(5),B$(5),D$(5),F$(5),H$(5)
7 CALL INIT:CLS "BCC"
10 CALL COLOR("OR"):CALL DBLPRT("LE TRAIN CAPRICIEUX"
    ,5,10):CALL COLOR("OBC")
30 DIM W$(3,5),IN$(40),LOC(40)
45 PRINT "VOUS ALLEZ D'ABORD ENTRER LES QUESTIONS"
46 PRINT "QUI VONT SERVIR AU JEU."
49 FOR I=1 TO 4:LOCATE (17,1):PRINT I;"e JEU":PRINT
50 LOCATE (19,1):PRINT "QUE VOULEZ VOUS ECRIRE DANS L
    A"
51 PRINT "LOCOMOTIVE ";
52 LINPUT Z$ :ON I GOSUB 53,54,55,56:GOTO 60
53 A$=Z$ :RETURN
54 C$=Z$ :RETURN
55 E$=Z$ :RETURN
56 G$=Z$ :RETURN
60 PRINT:FOR J=1 TO 5:PRINT J;"e REPONSE (" ;:K=VAL(SE
    G$("2514",I,1))
70 IF K=J THEN PRINT "EXACTE"; ELSE PRINT "FAUSSE";
80 PRINT ") ";:LINPUT Z$ :ON I GOSUB 81,82,83,84:GOTO
    90
81 B$(J)=Z$ :RETURN
82 D$(J)=Z$ :RETURN
83 F$(J)=Z$ :RETURN
84 H$(J)=Z$ :RETURN
90 NEXT:CLS:NEXT
330 FOR I=1 TO 16:READ ZZ$:CALL CHAR(I,ZZ$):NEXT
340 DATA 0000000000000000706,000000000001818FF00,000
    00000003E1C1CFF7E
350 DATA 0E0E0E0E0E0E1EBEFEFE,303030303030303E3E3E,FEF
    F9F0E040000000000
360 DATA 00FF3E1C080000000000,3EFF3E1C080000000000,000
    00000000000000706
370 DATA 0000000000000000FF00,0000000000000000F818,060
    606060606060686
380 DATA 181818181818181819,86FF9F0E040000000000,00F
    F0000000000000000
390 DATA 19FFF9702000000000000
400 !
410 CLS:CALL COLOR("OR"):CALL DBLPRT("LE TRAIN",12,3)
415 CALL DBLPRT("CAPRICIEUX",12,6):PRINT:PRINT
420 CALL COLOR("OBC"):PRINT "TU VAS VOIR APPARAITRE UN
    E LOCOMOTIVE"
425 PRINT "A GAUCHE DE L'ECRAN ET,A DROITE,5 WAGONS"
440 PRINT "LES WAGONS VONT AVANCER VERS LA","LOCOMOTIV
    E.TU DOIS FAIRE MONTER OU"

```

```
450 PRINT "DESCENDRE LA LOCOMOTIVE POUR ACCROCHER"
455 PRINT "LE WAGON QUI CONTIENT LA BONNE REPONSE."
456 CALL COLOR("0YB"):PRINT:PRINT "ATTENTION, LA LOCOMO
TIVE CHANGE PARFOIS"
457 PRINT "TOUTE SEULE DE VOIE. TU DOIS LA REPLACER COR
RECTEMENT." :PRINT
458 PRINT:PRINT "appuie une touche pour commencer"
460 ZZ$=KEY$
500 !JEU
505 CLS.
510 V$=A$
520 FOR I=1 TO 5
530 R$(I)=B$(I)
540 NEXT
550 BR=2
560 GOSUB 1100
600 V$=C$
610 FOR I=1 TO 5
620 R$(I)=D$(I)
630 NEXT
640 BR=5
650 GOSUB 1100
700 V$=E$
710 FOR I=1 TO 5
720 R$(I)=F$(I)
730 NEXT
740 BR=1
750 GOSUB 1100
800 V$=G$
810 FOR I=1 TO 5
820 R$(I)=H$(I)
830 NEXT
850 GOSUB 1100
860 BR=4
900 CALL COLOR("0R"):CALL DBLPRT("AU REVOIR",12,1)
910 PAUSE 5
920 RESTORE 920
930 GOTO 400
1100 !DESSIN LOCO+WAGONS
1105 P=LEN(V$)
1110 L$(1)=CHR$(1)&RPT$(CHR$(2),4)
1120 IF P>4 THEN L$(1)=L$(1)&RPT$(CHR$(2),P-4)
1150 L$(1)=L$(1)&CHR$(3)
1160 L$(2)=CHR$(4) & ""
1170 IF P>4 THEN L$(2)=L$(2)&RPT$(" ",P-4)
1190 L$(2)=L$(2)&CHR$(5)
1200 L$(3)=CHR$(6)&RPT$(CHR$(7),4)
1210 IF P>4 THEN L$(3)=L$(3)&RPT$(CHR$(7),P-4)
1250 L$(3)=L$(3)&CHR$(8)
```

```

1300 !DESSIN WAGONS
1305 FOR K=1 TO 5:L=LEN(R$(K))
1400 W$(1,K)=CHR$(9)&RPT$(CHR$(10),4)
1405 IF L>4 THEN W$(1,K)=W$(1,K)&RPT$(CHR$(10),L-4)
1440 W$(1,K)=W$(1,K)&CHR$(11)&" "
1500 W$(2,K)=CHR$(12)&" "
1505 IF L>4 THEN W$(2,K)=W$(2,K)&RPT$(" ",L-4)
1508 W$(2,K)=W$(2,K)&CHR$(13)&" "
1510 W$(3,K)=CHR$(14)&RPT$(CHR$(15),4)
1520 IF L>4 THEN W$(3,K)=W$(3,K)&RPT$(CHR$(15),L-4)
1560 W$(3,K)=W$(3,K)&CHR$(16)&" "
1570 NEXT
1580 T=3:DEP=0
1582 IF C=0 THEN 1600
1585 FOR I=1 TO C:LOC(C)=0:NEXT
1595 C=0
1600 GOSUB 3000
1610 !AFFICHAGE WAGON
1615 RESTORE 1620
1620 DATA "1RC","1bC","1YC","1MC","1GC"
1630 FOR K=1 TO 5
1640 R=26-DEP
1650 READ S$:CALL COLOR(S$)
1660 FOR J=1 TO 3
1670 LOCATE (K*4+J-3,R):PRINT W$(J,K)
1680 NEXT
1685 LOCATE (K*4-1,R+1):CALL COLOR("0B"):PRINT R$(K);
1690 NEXT
1700 !DEPLACEMENT LOCO ET WAGONS
1710 CALL KEY1(A,B):IN$(C)=CHR$(-A*(B=1))
1715 GOTO 4000
1720 IF IN$(C)<>""THEN LOC(C)=A
1730 IF LOC(C)=130 THEN T=T+1:GOSUB 3000
1750 IF LOC(C)=128 THEN T=T-1:GOSUB 3000
1770 DEP=DEP+1:C=C+1:GOTO 1610
3000 !AFFICHAGE LOCOMOTIVE
3005 IF T<=0 OR T>6 THEN RETURN
3010 L$(4)="
3015 IF P>4 THEN L$(4)=L$(4)&RPT$(" ",P-4)
3050 L$(4)=L$(4)&" "
3060 CALL COLOR("1BC"):FOR J=1 TO 3
3070 LOCATE (T*4+J-3,1):PRINT L$(J);
3080 IF T>1 THEN LOCATE (T*4+J-7,1):PRINT L$(4)
3090 IF T<5 THEN LOCATE (T*4+J+1,1):PRINT L$(4)
3100 NEXT
3110 LOCATE (T*4-1,2):CALL COLOR("0RC"):PRINT V$
3120 RETURN
4000 !TEST BONNE REPONSE
4005 IF P>4 THEN PR=P-4 ELSE PR=0

```

```
4010 IF T=BR AND R=7+PR THEN 4100
4020 IF R=7+PR AND T<>BR THEN 4500
4030 GOTO 1720
4100 LOCATE (22,20):CALL COLOR("0BL"):PRINT "EEXXAACCTT
!!"
4110 CALL PLAY("DOMISOSIREFALA",1,0):PAUSE 5:CALL COLOR
("0B")
4120 CLS:LOCATE (1,1):RETURN
4500 LOCATE (21,1):CALL COLOR("0RL"):PRINT "FFAAUUXX !
!"
4505 CALL PLAY("DODODODOMIDOMIDOMIDOSI",1,3)
4510 LOCATE (20,22):CALL COLOR("0BC"):PRINT "LA BONNE R
EPONSE EST :" :PRINT:PRINT
4520 PRINT:PRINT:CALL COLOR("0b"):CALL DBLPRT(R$(BR),1,
21)
4530 PAUSE 10
4540 CALL COLOR("0B"):CLS:LOCATE (1,1):RETURN
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10030 SUBEND
10040 SUB PLAY(N0$,L,D)
10050 FOR I=1 TO LEN(N0$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(N0$,I,2),1)
10070 P=VAL(SEG$("199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,X,Y)
10110 CALL COLOR("0LH")
10120 K$="" :FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("0"):SUBEND
```



# VILLES DE FRANCE

## Thème

Interrogation sur treize villes de France indiquées sur une carte de France par des points clignotants.

## Niveau

A partir de 8 ans (Cours élémentaire 2<sup>e</sup> année).

## Intérêt pédagogique

### Connaissances

- contrôle des acquis en géographie ;
- place d'une ville sur la carte ;
- notion d'échelle.

### Savoir-faire

- utilisation d'une carte ;
- recherche sur des documents ;
- comparaison d'une liste à une situation.

## Déroulement

Ce programme a été réalisé sur une idée de Jacques Boisgontier (dans *30 programmes pour ORIC 1*) adaptée et développée sur EXL 100.

Après entrée du prénom du joueur, une carte de France se dessine sur la moitié basse de l'écran. Dans cette carte, une étoile représentant une ville clignote pendant quelques secondes avant de s'afficher complètement.

La liste des treize villes du jeu s'inscrit en haut de l'écran et le joueur doit taper le nom de la ville marquée sur l'écran.

En cas de mauvaise réponse, le nom exact est donné. Ainsi de suite pour neuf autres interrogations.

A la fin du jeu, EXL 100 donne au joueur son score.

Lorsque le programme est utilisé pour la première fois par des enfants, il est intéressant de leur donner en plus une carte de France (carte routière, par exemple). Le jeu servira alors de support à un travail de recherche sur document et le score final n'aura plus d'importance.

## Commentaires

<b>5 à 28</b>	Initialisation et présentation.
<b>30 à 170</b>	Coordonnées des différents points de la carte de France.
<b>172 à 199</b>	Coordonnées et noms des treize villes du jeu.
<b>200 à 260</b>	Dessin de la carte de France.
<b>300 à 330</b>	Lecture des villes en DATA.
<b>450 à 470</b>	Initialisation de RND (nombre aléatoire).
<b>500 à 565</b>	Choix aléatoire d'une ville et clignotement de l'étoile sur la carte de France.
<b>570 à 604</b>	Affichage de la liste des villes et question.
<b>620 à 700</b>	Test de la réponse.
<b>800 à 840</b>	Fin de jeu. Affichage du score.
<b>10000 à 10030</b>	Initialisation générale.
<b>10100 à 10160</b>	Ecriture double taille.

## Particularités techniques

La carte de France est dessinée à partir de segments de droites. Les coordonnées (mode graphique) des points sont lues en DATA (ligne 220), ces derniers étant marqués sur l'écran (ligne 230) puis reliés entre eux en ordre successif par un segment de droite (ligne 250), jusqu'au retour au point de départ.

Les villes sont localisées par un point sur la carte. Mais un point seul étant trop difficile à lire en haute résolution, il a fallu dessiner une étoile autour de ce point et la faire clignoter pour que l'enfant le voit (lignes 520 à 565).

Les coordonnées des villes et leur nom sont lus (lignes 300-330) par une boucle GOTO 300 ce qui permet d'en modifier le nombre sans changer le programme.

## Modifications

Il est possible de donner au joueur une ou plusieurs chances supplémentaires en permettant plusieurs réponses, en ajoutant :

ligne 505 : REP = 0  
 ligne 622 : REP = REP + 1  
 ligne 625 : IF REP > 3 THEN 600

On peut aussi changer, augmenter, diminuer le nombre de villes en DATA afin de diversifier l'interrogation (voir adaptations).

Les villes sont tirées au sort (ligne 510) par un nombre aléatoire. Il est donc possible que la même ville soit choisie plusieurs fois. Si vous préférez avoir des villes toujours différentes, quelques lignes supplémentaires permettent de comparer le nombre tiré au sort à ses précédents (voir programme « Construis une phrase »).

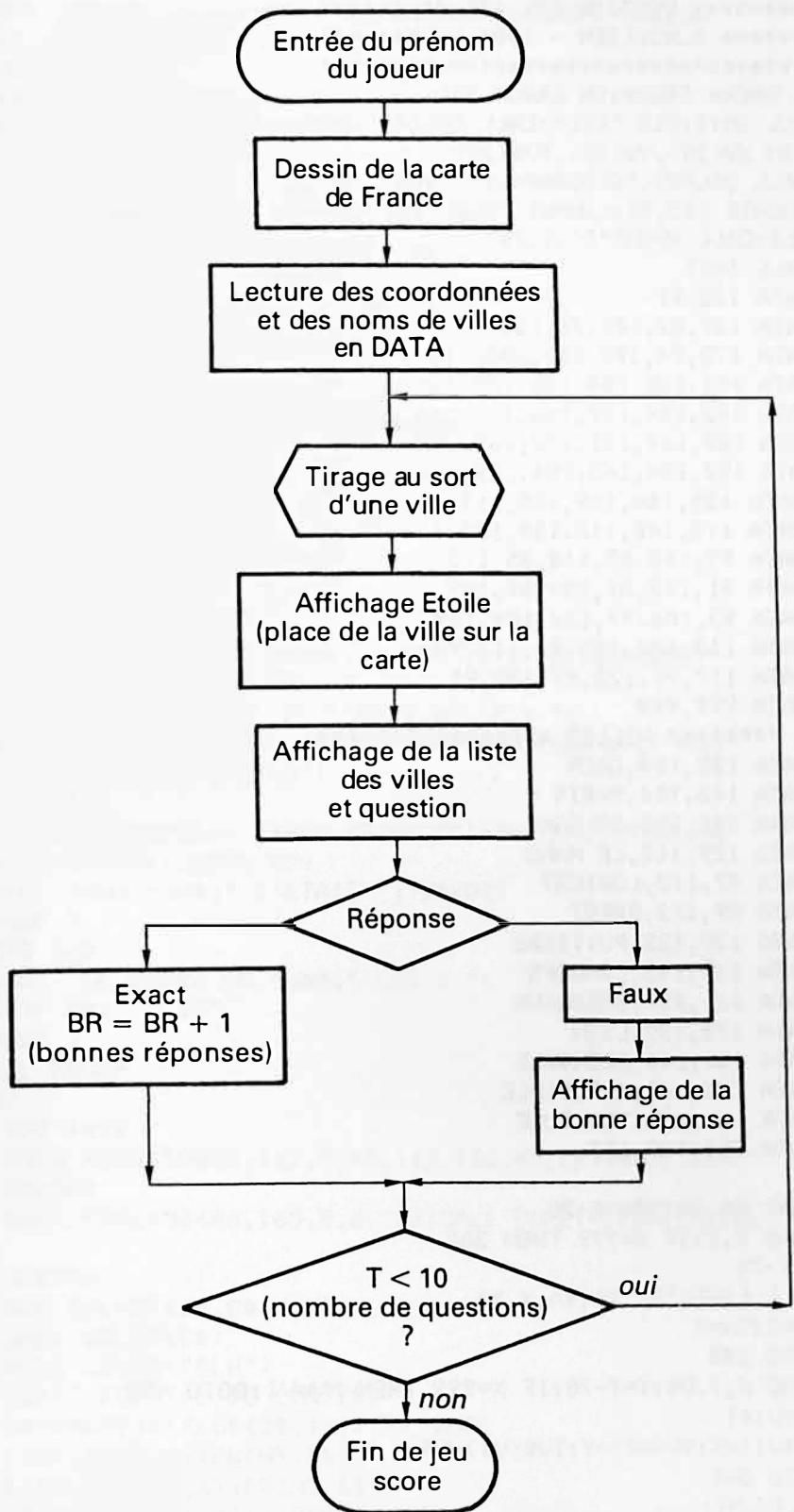
## Adaptations

Vous pouvez ajouter d'autres villes au programme, vous pouvez aussi interroger vos enfants et vos amis sur les fleuves, ou les montagnes, ou les ports, etc.

EXL 100 permet, en effet, d'obtenir d'autres dessins sur l'écran très facilement. Voici deux exemples :

- Vous décalquez sur un transparent la carte de France du programme.
  - Vous y placez d'autres villes qui seront matérialisées par un point.
  - Vous y dessinez les fleuves ou les montagnes qui seront matérialisés par des segments de droites à la suite les uns des autres.

Ensuite, à vous de programmer !



```

2 !***** VILLES DE FRANCE *****
3 !***** VERSION EXL 100 *****
4 !***** D.NIELSEN - 1985 *****
5 !***** *****
6 ON BREAK ERROR:ON ERROR 830
7 CALL INIT:CLS "BCC":CALL COLOR("0BC")
18 DIM XV(20),YV(20),TV$(20)
20 CALL DBLPRT("GEOGRAPHIE","ORG",10,10)
25 LOCATE (15,5):LINPUT "QUEL EST TON PRENOM ? ";N$
27 CLS:CALL HRON("G",1,12)
28 CALL INIT
30 DATA 133,91
40 DATA 137,82,141,76,153,85
50 DATA 173,94,197,100,193,115
60 DATA 181,130,189,130,189,139
70 DATA 193,154,197,160,197,166
80 DATA 189,169,181,172,165,172
90 DATA 157,184,145,184,129,181
100 DATA 125,184,109,175,113,157
110 DATA 113,148,113,139,105,127
120 DATA 93,118,85,118,85,115
130 DATA 81,112,81,109,89,109
140 DATA 93,106,97,106,109,109
150 DATA 113,106,109,97,113,94
160 DATA 117,97,125,97,133,91
170 DATA 999,999
172 ! ***** VILLES *****
175 DATA 125,103,CAEN
176 DATA 145,106,PARIS
177 DATA 141,115,ORLEANS
178 DATA 129,115,LE MANS
179 DATA 97,118,LORIENT
180 DATA 89,112,BREST
181 DATA 129,133,POITIERS
182 DATA 137,145,LIMOGES
183 DATA 121,157,BORDEAUX
184 DATA 173,139,LYON
185 DATA 181,148,GRENOBLE
186 DATA 181,169,MARSEILLE
187 DATA 141,172,TOULOUSE
199 DATA 999,999,222
200 !
220 READ XA,YA:YA=YA-70
240 READ X,Y:IF X=999 THEN 300
245 Y=Y-70
250 CALL LINE("B",XA,YA,X,Y)
251 XA=X:YA=Y
260 GOTO 240
300 READ X,Y,V$":Y=Y-70:IF X=999 THEN NV=VI:GOTO 450
310 VI=VI+1
320 XV(VI)=X:YV(VI)=Y:TV$(VI)=V$
330 GOTO 300
450 L=LEN(N$)
460 FOR I=1 TO L

```

```

470 Y=RND:NEXT I
480 PRINT "DONNE UN NOMBRE QUELCONQUE:";:ACCEPT VALIDATE(DIGIT)SIZE(4)NULL(12),Q
490 RANDOMIZE Q
500 ! ***** INTERROGATION *****
505 LOCATE (20,1):CLS:F=0
507 T=T+1:IF T=10 THEN 800
510 VI=INT(RND*NV)+1
520 CALL PLOT("R",XV(VI),YV(VI))
521 CALL PLOT("R",XV(VI)-1,YV(VI)+1)
522 CALL PLOT("R",XV(VI)-1,YV(VI)-1)
523 CALL PLOT("R",XV(VI)+1,YV(VI)-1)
524 CALL PLOT("R",XV(VI)+1,YV(VI)+1)
525 IF F=5 THEN 570
530 PAUSE .5
540 CALL PLOT("B",XV(VI),YV(VI))
541 CALL PLOT("B",XV(VI)-1,YV(VI)+1)
542 CALL PLOT("B",XV(VI)-1,YV(VI)-1)
543 CALL PLOT("B",XV(VI)+1,YV(VI)-1)
544 CALL PLOT("B",XV(VI)+1,YV(VI)+1)
550 PAUSE .5
560 F=F+1
565 GOTO 520
570 LOCATE (13,1):PRINT TV$(1);:FOR II=2 TO 13:PRINT " -";TV$(II);:NEXT II:PRINT
600 PRINT:PRINT "QUELLE EST CETTE VILLE ? ";
601 CALL COLOR("ORC"):ACCEPT VALIDATE(UALPHA)NULL("AAA"),V$:CALL COLOR("OBC")
604 PRINT
620 PRINT:IF V$=TV$(VI)THEN PRINT "C'EST EXACT ";N$:PA
USE 3:BR=BR+1:GOTO 500
630 PRINT "NON ";N$;" C'ETAIT ";TV$(VI)
640 PAUSE 3
700 GOTO 500
800 PRINT "LE SCORE DE ";N$;" EST : ";
810 PRINT BR;" / 10"
820 PAUSE 3
830 CALL HROFF
840 END
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10030 SUBEND
10100 SUB DBLPRT(Z$,C$,X,Y)
10105 CALL COLOR(C$)
10110 CALL COLOR("OLH")
10120 K$="":FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("OBC"):SUBEND

```



# **CONSTRUCTION DE PHRASES**

## **Thème**

Exercice de construction de phrases à partir d'éléments présentés dans le désordre.

## **Niveau**

A partir de 8 ans (Cours élémentaire 2<sup>e</sup> année).

## **Intérêt pédagogique**

## **Connaissances**

- grammaire ;
- éléments d'une phrase (groupe sujet, verbe, groupe complément) ;
- éléments du groupe nominal : déterminant, nom, expansions adjectifs, complément de nom.

## **Savoir-faire**

- synthèse d'une phrase ;
- copie d'un texte (orthographe).

## Déroulement

Après la présentation du jeu, les éléments d'une phrase apparaissent dans le désordre. Le joueur doit taper la phrase sur le clavier en remettant les éléments dans le bon ordre et sans faire de fautes de copie.

Si la réponse est bonne, on passe à une autre phrase.

Sinon, le joueur est invité à recommencer deux autres fois. En cas d'erreur à la troisième réponse, la bonne phrase est affichée.

Le jeu comporte quatre phrases différentes.

## Commentaires

- |                    |  |
|--------------------|--|
| <b>40 à 135</b>    | Initialisation et présentation du jeu.   |
| <b>200 à 240</b>   | Effacement de la mémoire des phrases précédentes.  |
| <b>290 à 350</b>   | Lecture de la phrase et décomposition en éléments.   |
| <b>500 à 540</b>   | Initialisation du jeu.   |
| <b>600 à 660</b>   | Tirage au sort d'un élément : le numéro d'un élément est tiré au sort (ligne 620). Il est comparé aux numéros déjà sortis (lignes 630 à 650).<br>S'il n'a pas déjà été utilisé, il est validé, sinon, on recommence le tirage au sort. |
| <b>800 à 860</b>   | Affichage de chaque élément en une des trois couleurs définies (rouge, vert ou jaune).<br>Cet affichage se fait sur une ligne et à une place aléatoire déterminée par X (ligne 820).   |
| <b>1000 à 1030</b> | Entrée de la réponse.  |
| <b>1040</b>        | Test réponse.  |
| <b>1050 à 1080</b> | Réponse fausse « Recommence ».   |
| <b>1090 à 1120</b> | Affichage de la bonne phrase (trois réponses erronées).  |
| <b>1200</b>        | Affichage réponse exacte.  |
| <b>1300 à 1320</b> | Test nombre de jeux.   |

- 1500 à 1510** Fin de jeu.
- 10000 à 10030** Initialisation générale.
- 10040 à 10090** Jouent une mélodie.
- 10100 à 10160** Ecriture double taille.
- 2000 à 2310** Entrée des quatre phrases en DATA.

## Particularités techniques

La phrase à reconstituer est formée de deux éléments (P\$ et R\$) car un seul élément dépasserait une ligne de DATA.

Elle est décomposée en éléments (quel que soit leur nombre) dans les lignes 310 à 350. EXL 100 considère alors chaque espace entre deux éléments comme séparateur, ce qui oblige impérativement à terminer chaque phrase en DATA par un espace, faute de quoi le dernier élément serait oublié. Bien sûr, on élimine cet espace pour la comparaison avec la réponse donnée (ligne 1040).

L'affichage des éléments est déterminé par Y, nombre aléatoire. Il est donc différent à chaque jeu. Pour donner une impression d'affichage en désordre, la place de chaque élément sur une ligne est aussi définie par un autre nombre aléatoire X qui déplace cet élément de la gauche à la droite de l'écran (affichage prévu pour des mots ayant 10 lettres au maximum).

Les lignes 220 à 240 servent pour la deuxième phrase et les suivantes, à effacer de la mémoire les éléments des phrases précédentes (pour le cas où la nouvelle phrase serait plus courte que la précédente).

EXL 100 comparant la réponse à la phrase de base, lettre à lettre, un espace en moins ou en trop amènerait un message d'erreur.

## Modifications

Il est possible de donner plus ou moins de réponses pour une phrase en modifiant la ligne 1080. On peut aussi augmenter ou diminuer le nombre de phrases du jeu (variable F, ligne 1320) en aménageant les lignes de DATA.

Il peut être intéressant d'ajouter un compteur d'erreurs :

```
40 : TR = 0
1075 : TR = TR + 1
1505 : PRINT « TU AS DONNE »;TR;« REPONSES
FAUSSES »
```

On peut aussi lister sur une imprimante les réponses fausses des enfants afin de connaître les erreurs rencontrées et y remédier par la suite.

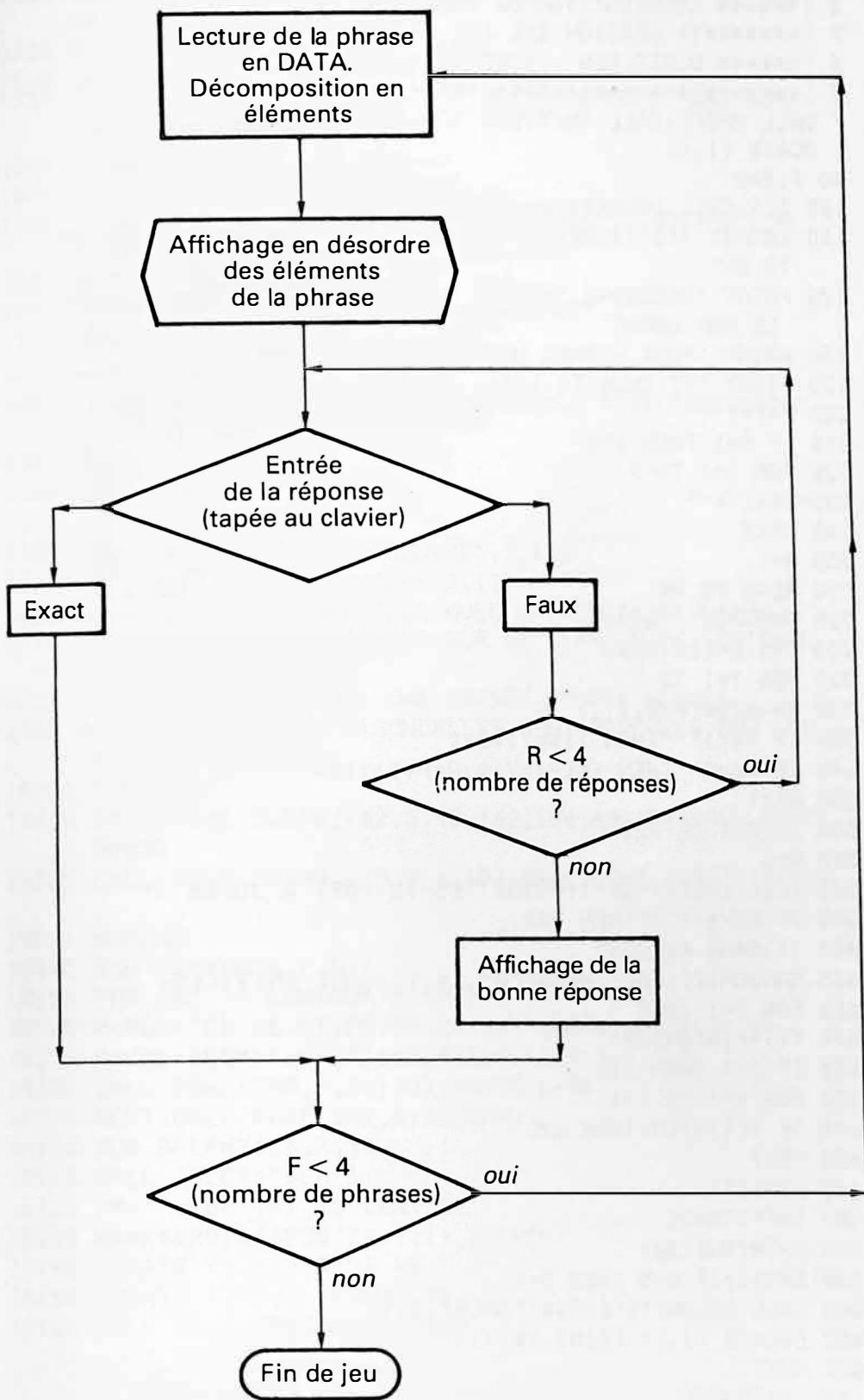
## Adaptations

Un autre programme « J'écris tout seul » vous propose une adaptation pour les enfants de 6 et 7 ans, utilisant les manettes de jeu et n'obligant pas l'enfant à taper la phrase.

Il est bien sûr possible de changer les phrases en DATA, toutefois il faut respecter certaines normes si l'on ne veut pas être obligé de transformer le programme :

- compte tenu de l'écran, chaque phrase ne doit pas dépasser 10 éléments ;
- chaque phrase doit se terminer par un espace (voir particularités techniques) ;
- vous devez veiller à ce que vos phrases ne puissent pas s'écrire dans un autre ordre (complément circonstanciel qui peut être placé avant ou après le verbe), EXL n'acceptant comme bonne réponse que votre phrase.

Il serait possible, comme pour « Autodictée » de rentrer les phrases en début de programme, ce qui permet de les adapter au travail du moment, mais oblige de les taper à chaque nouveau chargement (pensez aussi à placer en ligne 1510 GOTO 100).



```

2 !***** CONSTRUCTION DE PHRASE *****
3 !***** VERSION EXL 100 *****
4 !***** D.NIELSEN - 1985 *****
5 !***** *****
7 CALL HROFF:CALL INIT:CLS "CBB":CALL COLOR("0RB"):L
OCATE (1,1)
40 F,S=0
100 CLS:CALL DBLPRT("LA PHRASE",11,12)
110 LOCATE (15,1):PRINT "TU VAS VOIR APPARAÎTRE DES MO
TS EN"
120 PRINT "DESORDRE.":PRINT "TU DOIS LES REPLACER DANS
LE BON ORDRE"
130 PRINT "POUR FORMER UNE PHRASE CORRECTE ET"
135 PRINT "ET ENSUITE TAPER CETTE PHRASE."
200 F=F+1
210 IF F=1 THEN 290
220 FOR I=1 TO P
230 Y$(I)=""
240 NEXT
250 P=1
290 READ P$,R$
300 PH$=P$&" "&R$&" "
303 P=1:L=LEN(PH$)
310 FOR I=1 TO L
320 X$=SEG$(PH$,I,1)
330 IF X$<>" "THEN Y1$=Y1$&X$
340 IF X$=" "THEN Y$(P)=Y1$:P=P+1:Y1$=""
350 NEXT
500 !DEBUT DE JEU
510 R=0
520 CALL COLOR("0R"):PRINT "ES-TU PRÊT À JOUER ?"
540 IF KEY$<>"0"THEN 540
600 !TIRAGE AU SORT
605 RANDOMIZE:CALL HRON("B",19,1):CALL INIT:CLS
610 FOR I=1 TO P
620 Y(I)=INTRND(P)
625 IF I=1 THEN 660
630 FOR K=1 TO I-1
640 IF Y(I)=Y(K)THEN 620
650 NEXT
660 Y1=Y(I)
800 !AFFICHAGE
820 X=INTRND(28)
830 S=S+1:IF S=5 THEN S=1
840 CALL COLOR("0"&SEG$("GMCR",S,1))
850 LOCATE (I,X):PRINT Y$(Y1)
860 NEXT
1000 !REPONSE

```

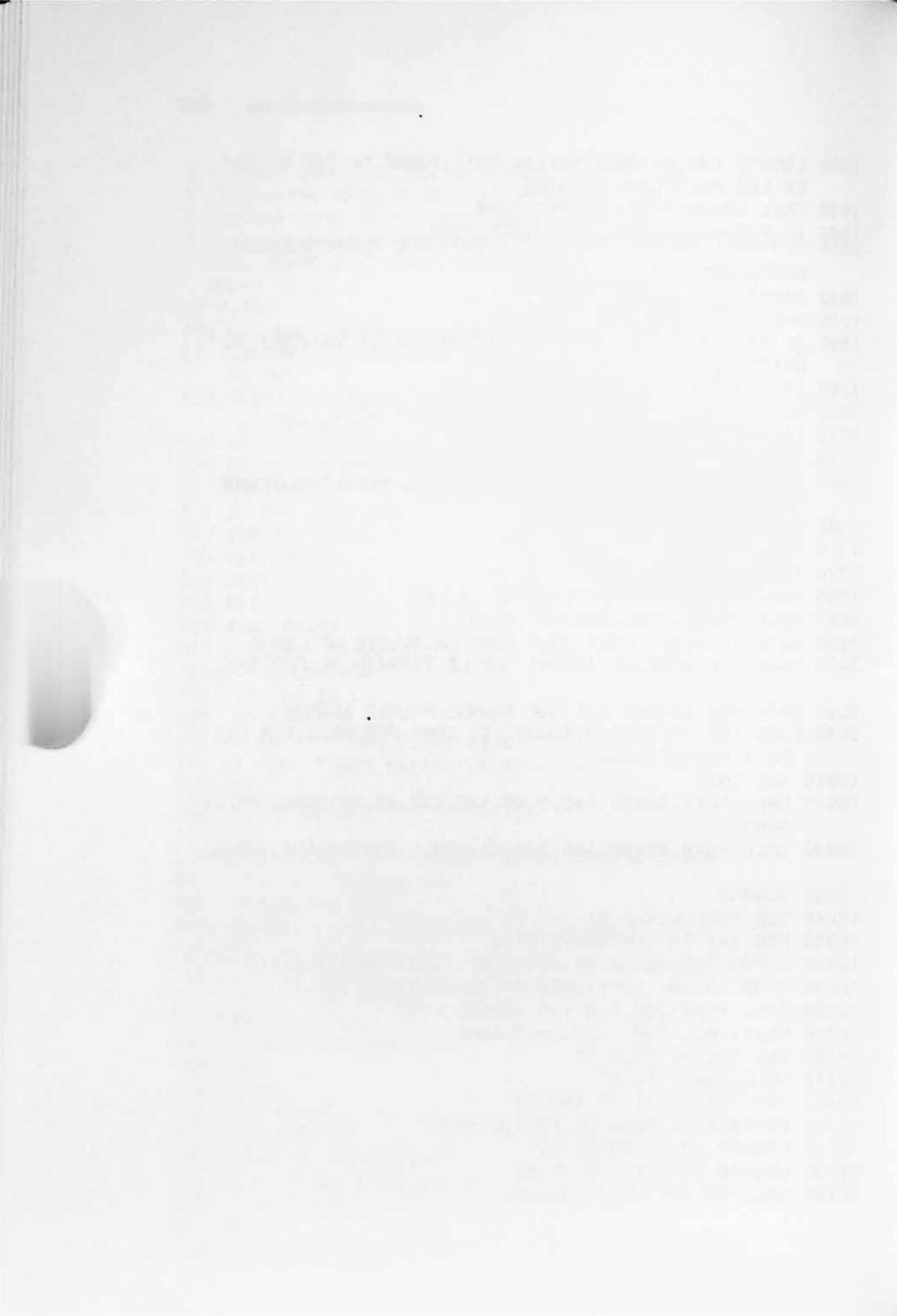
```

1020 LOCATE (20,5):CALL COLOR("OW"):PRINT "A TOI DE TAP
ER LES MOTS DANS L'ORDRE"
1030 CALL COLOR("OR") :LINPUT REP$
1040 IF REP$=SEG$(PH$,1,1-1)THEN 1200
1050 CLS:CALL DBLPRT(*FAUX!,1,20):CALL PLAY("DODODOMI
DOMIDODO",1,3)
1060 PAUSE 5
1070 R=R+1
1080 IF R<4 THEN CALL DBLPRT("RECOMMENCE",1,20):CALL PL
AY("DO",5,0):PAUSE 3
1085 IF R<4 THEN 1030
1090 CLS:PRINT "LA PHRASE CORRECTE ETAIT :"
1110 CALL COLOR("OR") :PRINT PH$
1120 GOTO 1300
1200 CLS:CALL DBLPRT("BRAVO",1,20):CALL PLAY("DODOMIDOF
ADOSODOLADOSISI",1,0)
1300 PAUSE 5:CLS
1310 CALL HROFF
1320 IF F<>4 THEN 200
1500 CALL DBLPRT("C'EST TERMINE",5,10)
1510 CALL DBLPRT("AU REVOIR",9,13)
2000 DATA LE PETIT CHAT JOUE AVEC LA,PELOTE DE LAINE
2010 DATA LES GARCONS JOUENT SUR LE TERRAIN,DE FOOTBALL

2020 DATA MES SOEURS ONT UNE GRANDE,POUPEE BLONDE
2030 DATA LES PETITES HIRONDELLES SONT PERCHEES,SUR LES
    FILS ELECTRIQUES
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
    50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10030 SUBEND
10040 SUB PLAY(NO$,L,D)
10050 FOR I=1 TO LEN(NO$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)
10070 P=VAL(SEG$("199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,X,Y)
10110 CALL COLOR("OLH")
10120 K$="" :FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("OGB") :SUBEND

```



# J'ÉCRIS TOUT SEUL

## Thème

Exercice d'assemblage des différents éléments d'une phrase, présentés dans le désordre.

## Niveau

6-7 ans (Cours préparatoire et Cours élémentaire 1<sup>re</sup> année).

## Intérêt pédagogique

### Connaissances

- lecture de mots, de phrases ;
- grammaire : découverte intuitive de la phrase correcte ;
- reconnaissance des éléments de la phrase.

### Savoir-faire

- synthèse orale de la phrase ;
- raisonnement par élimination des éléments déjà utilisés.

## Déroulement

Après explication du jeu (qu'il faudra lire aux plus petits qui ne pourront déchiffrer seuls le texte de présentation), les éléments d'une phrase apparaissent dans le désordre, dans des cases jaunes.

Le joueur doit illuminer la case contenant l'élément n° 1 de la phrase avec la manette de jeu.

S'il se trompe, un message et une musique le lui signalent, il peut recommencer.

Lorsqu'il a pointé la bonne case, le mot s'inscrit au bas de l'écran.

Puis il doit indiquer le 2<sup>e</sup> élément, ainsi de suite. Au fur et à mesure des bonnes réponses, la phrase apparaît en bas de l'écran.

Trois phrases différentes sont proposées dans ce jeu.

Le déroulement est assez proche de celui du programme « Construis une phrase ». La principale différence réside dans l'utilisation de la manette pour donner la réponse, au lieu de taper la phrase. Bien sûr, la copie de mots par l'intermédiaire du clavier a son intérêt pour les plus grands, mais l'exercice serait trop long pour les petits.

Comme pour « Lecture C.P. », les mots qui apparaissent sur l'écran sont entrés en minuscules car leur graphisme est celui des livres de lecture, donc plus lisible pour l'enfant.

## Commentaires

- |                  |   |
|------------------|---|
| <b>10 à 130</b>  | Initialisation et présentation du jeu.  |
| <b>200 à 240</b> | Effacement de la mémoire des phrases précédentes.   |
| <b>290 à 350</b> | Lecture de la phrase et décomposition en éléments.  |
| <b>500 à 540</b> | Initialisation du jeu.  |
| <b>600 à 660</b> | Tirage au sort d'un élément : le numéro d'un élément est tiré au sort (ligne 620). Il est comparé aux numéros déjà sortis (lignes 630 à |

- 650). S'il n'a pas déjà été utilisé, il est validé, sinon on recommence le tirage au sort.
- 700 à 710** Dessin du rectangle dans lequel le mot sera inscrit.
- 800 à 830** Affichage de l'élément de phrase en double grandeur dans la case.
- 840** Boucle (retour à 610) pour afficher les différents éléments de la phrase.
- 900 à 920** Question.
- 930** Réponse et test de la réponse.  
*Ligne 930 : Entrée de la réponse.*
- 1500 à 1530** Bonne réponse : affichage de « EXACT » : une séquence musicale est jouée : les éléments de la phrase déjà trouvés sont affichés.  
*Ligne 1530 : Renvoie à la ligne 910 pour la réponse suivante.*
- 1600 à 1630** Phrase complète - Test nombre de jeux.
- 1650 à 1680** Fin de jeu.
- 2000 à 2030** Réponse fausse - Affichage et musique différente - Retour à la ligne 930 pour une autre réponse.
- 3000 à 3200** Phrases du jeu en DATA.
- 10000 à 10030** Initialisation générale.
- 10040 à 10090** Jouent une mélodie.
- 10100 à 10160** Ecriture double taille.
- 10170 à 10210** Ecriture encadrée.
- 10220 à 10340** Gestion du joystick et des flèches.

## Particularités techniques

Jusqu'à la ligne 660, le programme est très proche de « Construis une phrase ». Seules différences :

- la phrase, plus courte, est lue en un seul DATA ;
- le principe de découpage de la phrase en éléments (lignes 310 à 350) implique que P sera plus grand que le nombre d'éléments. On a ajouté la variable NM (nombre de mots) = P-1, qui correspond donc au nombre d'éléments.

Lors de l'affichage des éléments, chaque case (I) (une à dix au maximum) est affectée du numéro d'ordre de l'élément dans la phrase Y(I).

C'est ce qui permet de tester ensuite la valeur de la réponse ligne 1000 : Y(Y1) = I

**REP** : numéro de la case visée ;  
**I** : numéro de l'élément demandé.

Les enfants de 6 ans ayant parfois du mal à déchiffrer les messages EXACT ou FAUX, les deux séquences musicales, d'octave et de durée de notes différentes leur sont une aide précieuse pour juger leur réponse.

## Modifications

La temporisation (ligne 1525) qui permet à l'enfant de lire en bas de l'écran les éléments déjà trouvés de la phrase peut être augmentée si nécessaire.

Le programme ne permet de passer à l'élément suivant de la phrase que lorsque l'on a trouvé la bonne réponse (l'instruction NEXT I est placée ligne 1530 après le message exact), ainsi l'enfant peut recommencer autant de fois que cela est nécessaire.

Pour limiter le nombre de réponses, il faudrait ajouter un compteur de « coups » qui enverrait à une ligne affichant l'élément à trouver au bout de deux ou trois essais (dans le sous-programme « Réponse fausse » lignes 2000 à 2030).

On pourrait ainsi augmenter le nombre de phrases, mais un enfant de 6 ans se fatigue vite sur un tel programme et l'expérience prouve que trois phrases correspondent à une durée acceptable.

## Adaptations

Ce programme permet d'entrer vos propres phrases en DATA à la place des trois qui sont proposées, à condition de respecter quelques règles :

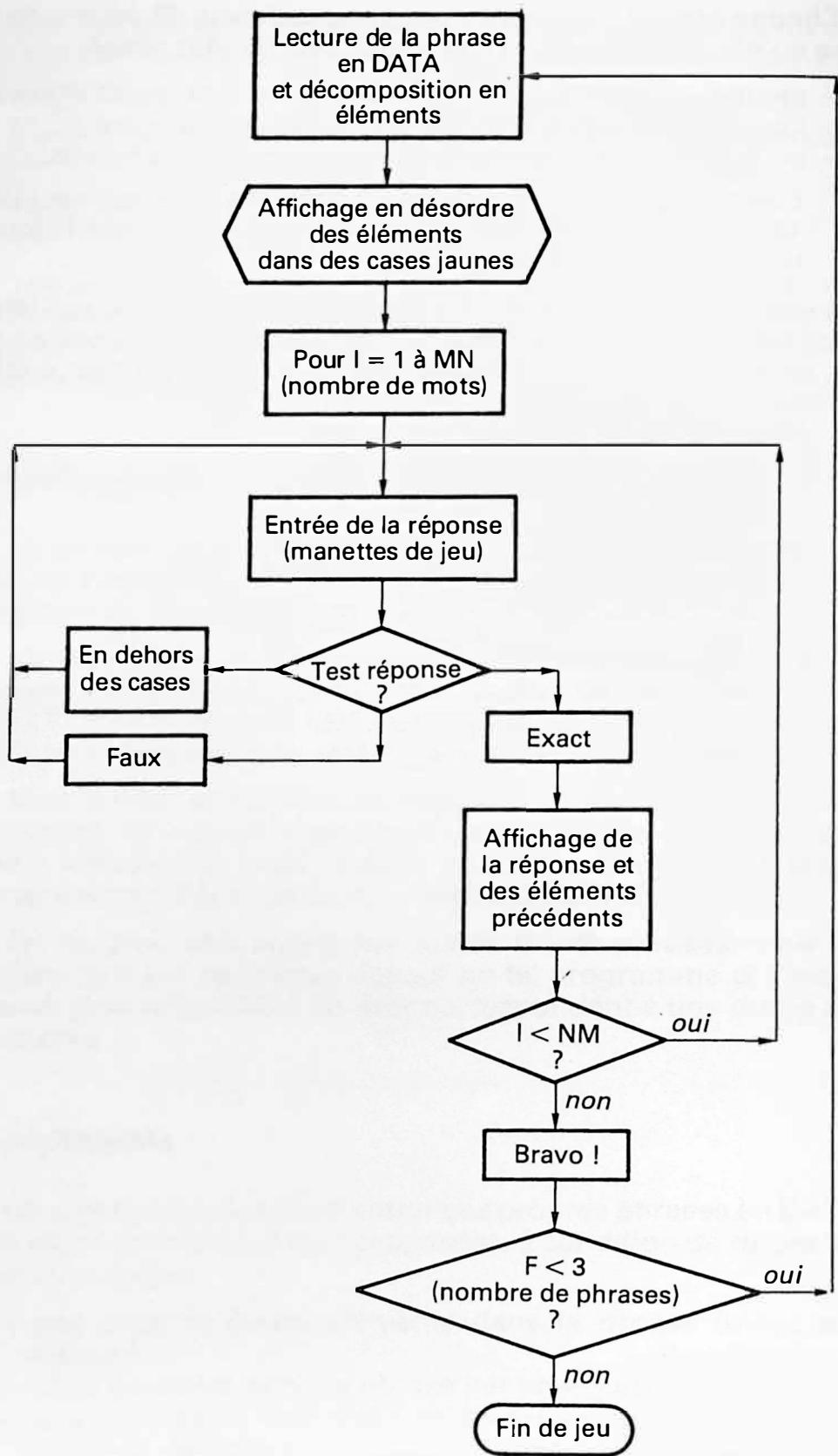
- pas plus de douze éléments dans la phrase (place sur l'écran) ;
- vous devez terminer la phrase par un espace ;
- si vous utilisez des mots de plus de cinq lettres, il faut agrandir les boîtes.

Chaque phrase commence par une majuscule. Si vous tenez à ce qu'elle finisse aussi par un point, rien de plus simple :

- ajoutez un point juste derrière le dernier mot (puis l'espace final), il apparaîtra avec ce mot, dans la même case ;
- ou bien encore, séparez-le du dernier mot par un espace.

Exemple : rue . “ . Il formera un élément à lui tout seul (le 12<sup>e</sup> au maximum !) dans une boîte et l'enfant devra l'ajouter à la suite des éléments.

Le programme suivant « Reconstitution de textes », sur une base technique très proche de « J'écris tout seul », propose pour les enfants plus grands un exercice dont le but est tout à fait différent.



```

2 !***** J'ÉCRIS TOUT SEUL *****
3 !***** VERSION EXL 100 *****
4 !***** D.NIELSEN - 1985 *****
5 !***** *****
6 DIM Z$(13),Y$(13)
7 CALL HROFF:CALL INIT:CLS "CBB":CALL COLOR("ORB")
40 F,S=0
100 CALL DBLPRT("J'écris tout seul",3,12):CALL COLOR("OG")
110 LOCATE (15,5):PRINT "tu vas voir apparaitre des mots en","desordre."
120 PRINT "tu dois les replacer dans le bon ordre pour former une phrase ";
125 PRINT "correcte."
130 PRINT:PRINT "pour cela, il faut choisir les mots en manipulant la manette"
140 PRINT
200 F=F+1
210 IF F=1 THEN 290
220 FOR I=1 TO P
230 Y$(I)=""
240 NEXT
260 REP$=""
290 READ PH$:PH$=PH$&" "
305 P=1:L=LEN(PH$)
310 FOR I=1 TO L
320 X$=SEG$(PH$,I,1)
330 IF X$<>" "THEN Y1$=Y1$&X$
340 IF X$="" "THEN Y$(P)=Y1$ :P=P+1:Y1$=""
350 NEXT
360 NM=P-1
500 !DEBUT DE JEU
510 R=0
520 CALL COLOR("OR"):PRINT "ES-TU PRET A JOUER ?"
540 RANDOMIZE:IF KEY$<>"0"THEN 540
600 !TIRAGE AU SORT
605 CLS:CALL HRON("B",19,1):CALL INIT
610 FOR I=1 TO NM
620 Y(I)=INTRND(NM)
625 IF I=1 THEN 660
630 FOR K=1 TO I-1
640 IF Y(I)=Y(K)THEN 620
650 NEXT
660 Y1=Y(I):Z$(I)=Y$(Y1)
700 !DESSIN BOITE
710 CALL COLOR("OCB"):CALL BOXPRT(Y$(Y1),3-20*(I)5),I+5*(I)5)
840 NEXT
900 !REPONSE

```

```

905 FOR I=1 TO NM
910 CLS:LOCATE (20,1):CALL COLOR("0R"):PRINT "Quel est
    le mot No";I;"de"
920 PRINT "la phrase ?"
930 CALL GETKEY(Z$(),X,Y1)
1000 IF Y(Y1)=I THEN 1500 ELSE 2000
1400 IF I=9 THEN 1500
1410 GOTO 2000
1450 IF I=10 THEN 1500
1460 GOTO 2000
1500 CLS:CALL COLOR("0G"):CALL DBLPRT("EXACT",1,20)
1505 CALL PLAY("DOREMI FASOLASI",1,0)
1510 REP$=REP$&Y$(I)& "
1520 CALL COLOR("0C"):PRINT REP$;
1525 PAUSE 5
1530 NEXT
1600 CLS:CALL DBLPRT("BRAVO",6,20)
1610 PAUSE 10
1620 CALL HROFF:CALL INIT
1630 IF FK3 THEN 200
1650 !FIN DE JEU
1660 CLS:CALL HROFF:CALL COLOR("0R"):CALL DBLPRT("AU RE
    VOIR",5,15)
1680 GOTO 9999
2000 !REPONSE FAUSSE
2010 CALL PLAY("LAMIDOLA",2,4)
2020 CALL COLOR("0R"):CLS:CALL DBLPRT("FAUX ,recommence
    ",1,20)
2030 GOTO 930
3000 DATA Elle est dans la rue
3010 DATA Le bebe est dans son lit
3020 DATA Le petit chat joue avec elle
9999 END
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
    50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10030 SUBEND
10040 SUB PLAY(NO$,L,D)
10050 FOR I=1 TO LEN(NO$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)
10070 P=VAL(SEG$("199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,X,Y)
10110 CALL COLOR("0LH")
10120 K$="":FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT

```

```
10140 LOCATE (Y,X):PRINT K$  
10150 LOCATE (Y+1,X):PRINT K$  
10160 CALL COLOR("0"):SUBEND  
10170 SUB BOXPRT(Z$,X,Y)  
10180 LOCATE (Y*3,X):PRINT CHR$(9);RPT$(CHR$(4),11);CHR$  
    (6)  
10190 LOCATE (Y*3+1,X):PRINT CHR$(3);Z$;RPT$(" ",11-LEN(  
    Z$));CHR$(3)  
10200 LOCATE (Y*3+2,X):PRINT CHR$(8);RPT$(CHR$(4),11);CH  
    R$(7)  
10210 SUBEND  
10220 SUB GETKEY(Z$(),X,Y)  
10230 X=3:Y=1  
10240 CALL COLOR("0BC"):CALL BOXPRT(Z$(Y),X,Y+5*(Y>5))  
10250 CALL KEY1(A,B)  
10260 IF A=13 OR A=32 THEN 10340  
10270 IF A<128 OR A>131 THEN 10250  
10280 CALL COLOR("0CB"):CALL BOXPRT(Z$(Y),X,Y+5*(Y>5))  
10290 IF A=128 AND Y>1 THEN IF Z$(Y-1)<>"THEN Y=Y-1:IF  
    Y=5 THEN X=X-20  
10300 IF A=129 AND X<10 THEN IF Z$(Y+5)<>"THEN X=X+20:Y  
    =Y+5  
10310 IF A=130 AND Y<12 THEN IF Z$(Y+1)<>"THEN Y=Y+1:IF  
    Y=6 THEN X=X+20  
10320 IF A=131 AND X>10 THEN IF Z$(Y-5)<>" "THEN X=X-20:  
    Y=Y-5  
10330 GOTO 10240  
10340 CALL COLOR("0CB"):CALL BOXPRT(Z$(Y),X,Y+5*(Y>5)):S  
    UBEND
```



# **RECONSTITUTION DE TEXTES**

## **Thème**

Exercice de reconstitution d'un texte à partir d'éléments présentés dans le désordre.

## **Niveau**

A partir de 7 ans (avec des textes simples). A partir de 9 ans pour les textes proposés dans le programme.

## **Intérêt pédagogique**

### **Connaissances**

- imprégnation de textes choisis parmi les œuvres de grands auteurs ;
- découverte des structures grammaticales d'un texte.

### **Savoir-faire**

- recherche par élimination, lecture d'indices (ponctuation) ;
- synthèse orale d'un texte ;
- logique.

## Déroulement

Après explication de l'exercice, sept parties d'un texte s'affichent en désordre, dans des cases sur l'écran. Le joueur doit ensuite sélectionner avec la manette les éléments du texte, les uns après les autres, dans l'ordre correct du texte initial.

Une séquence musicale et un message signalent les erreurs et le joueur doit recommencer.

Si la case pointée est la bonne, une autre séquence musicale est jouée et le texte s'affiche petit à petit, dans le bon ordre, en bas de l'écran.

Trois textes d'auteurs sont ainsi proposés, dans ce programme.

## Commentaires

Ce programme a été réalisé sur la base de « J'écris tout seul ». Voici les modifications effectuées :

- 3000 à 3070 }    3080 à 3150 }    3160 à 3230 }** Les 7 parties des textes sont entrées séparément, mais dans l'ordre exact, en DATA.
- 300 à 330** Lecture des différentes parties d'un texte en DATA.
- 705** La case où s'inscrit une partie du texte est définie en fonction de la longueur de cette partie (LEN(Y\$)).
- 900 à 1000** Réponse et test réponse. La réponse (case visée) est déterminée à partir de la coordonnée verticale du point visé.

## Particularités techniques

Les textes sont partagés en sept éléments maximum pour une question de mise en place sur l'écran : c'est le nombre limite de lignes qui peuvent être affichées à la fois dans le désordre en haut de l'écran (deux lignes par élément en comprenant le dessin de la boîte) et dans le bon ordre en fonction des réponses en bas de l'écran.

Toutefois, le principe de lecture des parties de texte en DATA (lignes 300-330) permet d'entrer éventuellement des textes plus courts (ne pas oublier la ligne DATA « ZZZZ » qui sert de séparateur entre les textes).

## Modifications

On peut introduire une limite au nombre de réponses en ajoutant un compteur d'erreurs dans le sous-programme « Réponse fausse » lignes 2000 à 2030.

Le nombre de textes est fixé ligne 1630 : des textes supplémentaires pourront être ajoutés en modifiant cette ligne et... en les entrant en DATA !

## Adaptations

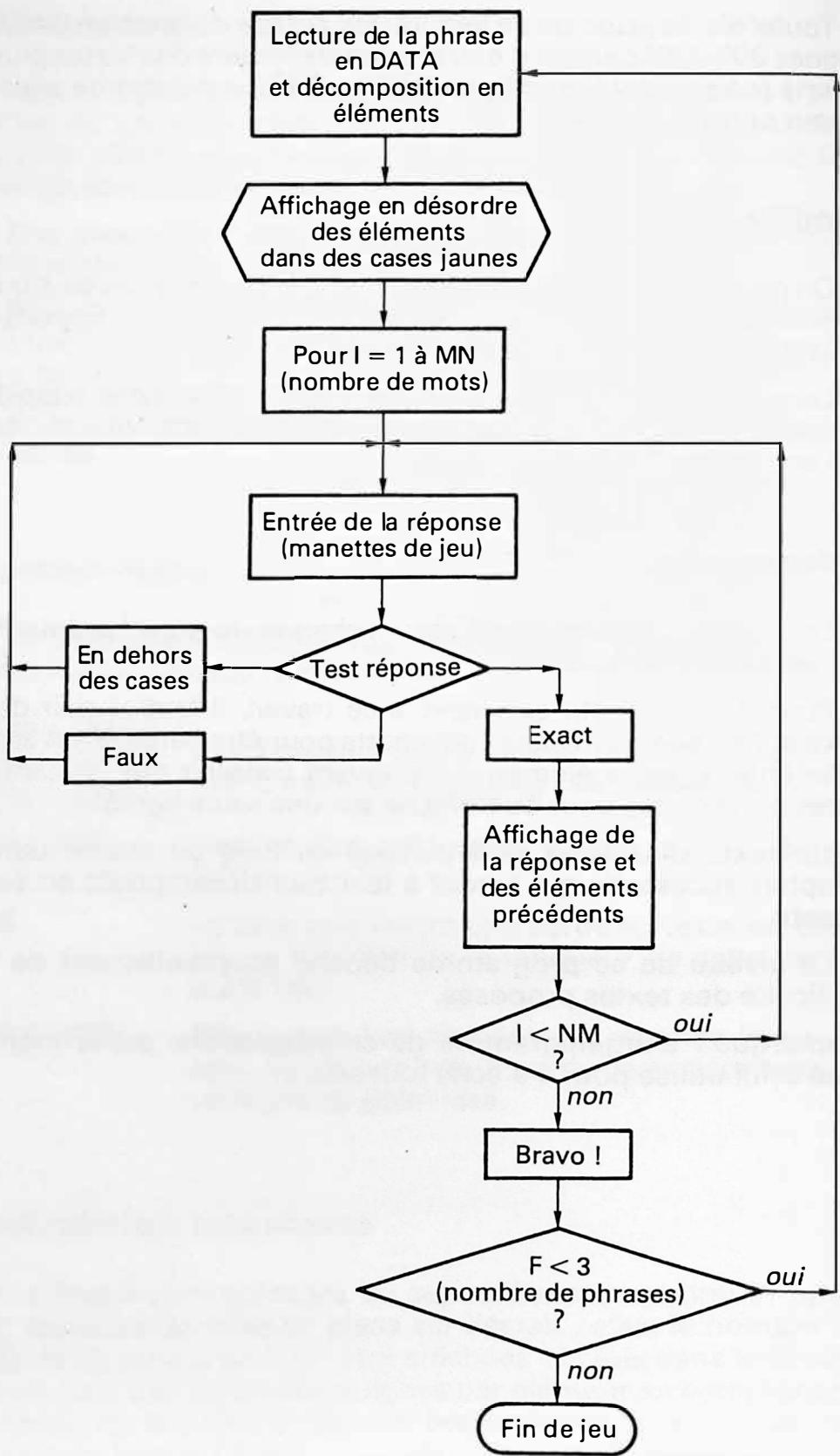
Les textes entrés en DATA seront changés lorsque les enfants connaîtront bien ceux-là.

Pour donner toute sa valeur à ce travail, il faut choisir des textes d'auteurs, suffisamment courts pour être partagés en sept éléments (chaque élément ne pouvant contenir que 38 caractères ou espaces pour être affiché sur une seule ligne).

Un texte plus long sera partagé en trois ou quatre paragraphes successifs qui seront à leur tour décomposés en éléments.

Le niveau de ce programme dépend essentiellement de la difficulté des textes proposés.

*Remarque :* L'organigramme de ce programme est le même que celui utilisé pour « J'écris tout seul ».



```
2 !*** RECONSTITUTION DE TEXTES *****
3 !***** VERSION EXL 100 *****
4 !***** D.NIELSEN - 1985 *****
5 !*****
6 DIM Z$(13),Y$(13)
7 CALL HROFF:CALL INIT:CLS "CBB":CALL COLOR("ORB")
40 F,S=0
100 CALL DBLPRT("RECONSTITUTION TEXTE",1,8):CALL COLOR
    ("OG")
110 LOCATE (15,5):PRINT "tu vas voir apparaitre des mo
rceaux de"
115 PRINT "texte en desordre ."
120 PRINT "tu dois les replacer dans le bon ordre pou
r former une phrase ";
125 PRINT "correcte."
130 PRINT:PRINT "pour cela, il faut choisir les elemen
ts avec la manette"
140 PRINT
200 F=F+1
210 IF F=1 THEN 290
220 FOR I=1 TO P
230 Y$(I)=""
240 NEXT
260 REP$=""
290 P=0
300 READ Y1$:IF Y1$="ZZZZ"THEN 500
310 P=P+1
320 Y$(P)=Y1$
330 GOTO 300
500 !DEBUT DE JEU
510 R=0
520 CALL COLOR("OW"):PRINT "ES-TU PRET A JOUER ?"
540 RANDOMIZE:IF KEY$<>"0"THEN 540
600 !TIRAGE AU SORT
605 CLS:CALL HRON("B",16,1):CALL INIT
610 FOR I=1 TO P
620 Y(I)=INTRND(P)
625 IF I=1 THEN 660
630 FOR K=1 TO I-1
640 IF Y(I)=Y(K)THEN 620
650 NEXT
660 Y1=Y(I):Z$(I)=Y$(Y1)
700 !AFFICHAGE PHRASE
705 L=LEN(Y$(Y1))
710 COL$="0"&SEG$("CWYCbMY",I-INT(I/7),1):CALL COLOR(C
OL$)
720 LOCATE (2*I-1,1):PRINT Y$(Y1)
840 NEXT
900 !REPONSE
```

```
905 CLS:FOR I=1 TO P
910 LOCATE (17,1):CALL COLOR("0C"):PRINT "Quel est l'e
lement No";I;"?"
930 CALL GETKEY(Z$(),X,Y1)
1000 IF Y(Y1)=I THEN 1500 ELSE 2000
1500 CLS:CALL COLOR("0G"):CALL DBLPRT("EXACT",1,20)
1505 CALL PLAY("DOREMIFASOLASI",1,0)
1510 LOCATE (18,1)
1520 CALL COLOR("0CB")
1522 FOR K=1 TO I
1523 FOR L=1 TO LEN(Y$(K)):PRINT SEG$(Y$(K),L,1);:NEXT
1524 PRINT " ";:NEXT
1525 PAUSE 5
1530 NEXT
1600 CLS:CALL DBLPRT("BRAVO",6,20)
1610 PAUSE 10
1620 CALL HROFF:CALL INIT
1630 IF F<3 THEN 200
1650 !FIN DE JEU
1660 CLS:CALL HROFF:CALL COLOR("0R"):CALL DBLPRT("AU RE
VOIR",5,15)
1680 GOTO 9999
2000 !REPONSE FAUSSE
2010 CALL PLAY("LAMIDOLA",2,4)
2020 CALL COLOR("0Y"):CLS:CALL DBLPRT("FAUX ,recommence
",1,20)
2030 GOTO 930
3000 DATA MA VIE EST MONOTONE.
3010 DATA "JE CHASSE LES POULES,"
3020 DATA LES HOMMES ME CHASSENT.
3030 DATA TOUTES LES POULES SE RESSEMBLENT
3040 DATA ET TOUS LES HOMMES SE RESSEMBLENT.
3050 DATA "MAIS SI TU M'APPRIVOISES,"
3060 DATA MA VIE SERA COMME ENSOLEILLEE.
3070 DATA ZZZZ
3080 DATA C'EST UN PETIT PASSEREAU
3090 DATA "A ROBE GRISE,"
3100 DATA "D'UN GRIS VELOUTE ET PROFOND."
3110 DATA "LORSQUE LE VENT REMUE LA RAMILLE"
3120 DATA "QUI LE PORTE,"
3130 DATA "IL OSCILLE DOUCEMENT,"
3140 DATA "SUR SES PATTES."
3150 DATA ZZZZ
3160 DATA "DEHORS,"
3170 DATA "LA TEMPETE S'ETAIT LEVEE;"
3180 DATA "LE VENT D'OUEST GEMISSAIT"
3190 DATA "DANS LA CHEMINEE."
3200 DATA "ON L'ENTENDAIT RODER DANS LES COULOIRS"
3210 DATA "COMME UN GRAND CHIEN INQUIET"
```

```
3220 DATA "QUI PLEURE."
3230 DATA ZZZZ
9999 CALL HROFF:END
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10030 SUBEND
10040 SUB PLAY(NO$,L,D)
10050 FOR I=1 TO LEN(NO$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)
10070 P=VAL(SEG$("199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,X,Y)
10110 CALL COLOR("OLH")
10120 K$="" :FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("0"):SUBEND
10220 SUB GETKEY(Z$(),X,Y)
10230 X=1:Y=1
10240 LOCATE (2*Y-1,1)
10245 COL$="0"&SEG$("CWYCBMY",Y-INT(Y/7),1):CALL COLOR(C
OL$&"BI"):PRINT Z$(Y)
10250 CALL KEY1(A,B)
10260 IF A=13 OR A=32 THEN 10340
10270 IF A<128 OR A>131 THEN 10250
10271 PAUSE .05
10280 CALL COLOR(COL$):LOCATE (2*Y-1,1):PRINT Z$(Y)
10290 IF A=128 AND Y>1 THEN IF Z$(Y-1)<>"THEN Y=Y-1
10310 IF A=130 AND Y<8 THEN IF Z$(Y+1)<>"THEN Y=Y+1
10330 GOTO 10240
10340 CALL COLOR(COL$):LOCATE (2*Y-1,1):PRINT Z$(Y):SUBE
ND
```



# **COCKTAIL MAGIQUE**

## **Thème**

Programme de travail sur les fractions en composant des cocktails.

## **Niveau**

A partir de 10 ans (Cours moyen 2<sup>e</sup> année - 6<sup>e</sup> - 5<sup>e</sup>).

## **Intérêt pédagogique**

## **Connaissances**

- utilisation de fractions ;
- mécanisme de la proportionnalité et de la règle de trois ;
- travail sur les volumes.

## **Savoir-faire**

- entraînement au calcul mental ;
- habileté du barman !

## **Déroulement**

Il s'agit de composer un cocktail (sans alcool !) pour lequel les proportions sont affichées, dans une carafe dont le volume est déterminé de façon aléatoire. Le joueur doit indiquer le volume de chacun des jus ou sirops qui composent ce cocktail.

En cas d'erreur de calcul, le programme guide le « futur barman » afin qu'il parvienne à la bonne réponse.

Enfin, il doit indiquer quel volume d'eau doit être ajouté pour compléter ce cocktail.

Chaque réponse est matérialisée par une couleur différente dans la carafe dessinée sur l'écran.

## Commentaires

- |                      |  |
|----------------------|--|
| <b>10 à 80</b>       | Présentation du programme.   |
| <b>100 à 160</b>     | Choix du cocktail.   |
| <b>170 à 195</b>     | Choix aléatoire du volume de la carafe (10 - 20 - 30 - 40 - 50 - 60 ... 100 cl).   |
| <b>200 à 470</b>     | Dessin de la carafe et indication des proportions.   |
| <b>500 à 760</b>     | Calcul du volume de chaque composant (1 à 3).  |
| <b>1000 à 1070</b>   | Visualisation dans la carafe de chaque bonne réponse.  |
| <b>1200 à 1480</b>   | Calcul du complément d'eau et visualisation de la réponse donnée, même si elle est fausse (la carafe « déborde » si la réponse est trop grande). |
| <b>1500 à 1540</b>   | Fin de programme. Retour possible à un autre calcul avec un cocktail différent et un nouveau volume de carafe.                                   |
| <b>1700 à 1750</b>   | Sous-programme « la carafe déborde » en cas de complément d'eau trop important, puis effacement de cette visualisation.                          |
| <b>2000 à 2020</b>   |  |
| <b>2100 à 2120</b>   |  |
| <b>2200 à 2220</b>   |  |
| <b>2300 à 2320</b>   |  |
| <b>2400 à 2420</b>   |  |
| <b>2500 à 2520</b>   |  |
| <b>10000 à 10030</b> | Initialisation générale.   |
- Liste des proportions des cocktails et couleurs des composants.

- 10040 à 10090** Jouent une mélodie.
- 10100 à 10160** Ecriture double taille.
- 10170 à 10180** Ecriture normale.
- 10200 à 10280** Remplissage du verre.
- 10300 à 10332** Effacement d'écran.

## Particularités techniques

Les proportions des cocktails sont inscrites en DATA. Le choix du cocktail renvoie à une ligne (310 à 370) contenant une instruction RESTORE qui permet la lecture des proportions de ce cocktail.

Les trois calculs de volume des jus de fruits ou des sirops sont obtenus par le même programme à l'intérieur d'une boucle FOR... NEXT (lignes 510 à 1070). A l'intérieur de ce calcul, si la réponse n'est pas bonne, le programme décompose la fraction en une multiplication et une division entière, demande chaque calcul et revient en arrière si la réponse n'est toujours pas la bonne (lignes 570 à 760).

Le joueur ne peut donc sortir du programme qu'en ayant trouvé la bonne réponse (d'où l'inutilité d'un score à afficher).

L'initialisation de la séquence aléatoire qui détermine le volume de la carafe est obtenue en utilisant à la fois la longueur du prénom du joueur et le choix du cocktail (ligne 180), ce qui permet de varier les calculs.

## Modifications

Quelques aménagements à ce programme permettraient de prendre en compte des réponses décimales.

Ne tenir compte que de la partie entière de la division donne en effet des résultats trop « arrondis » si le volume tiré au sort est faible. Mais d'autre part, un travail sur les nombres décimaux ne peut plus se faire en calcul mental à 10 ans !

A chacun donc de choisir l'adaptation qui correspond à la tranche d'âge des enfants qui travailleront sur ce programme.

## Adaptations

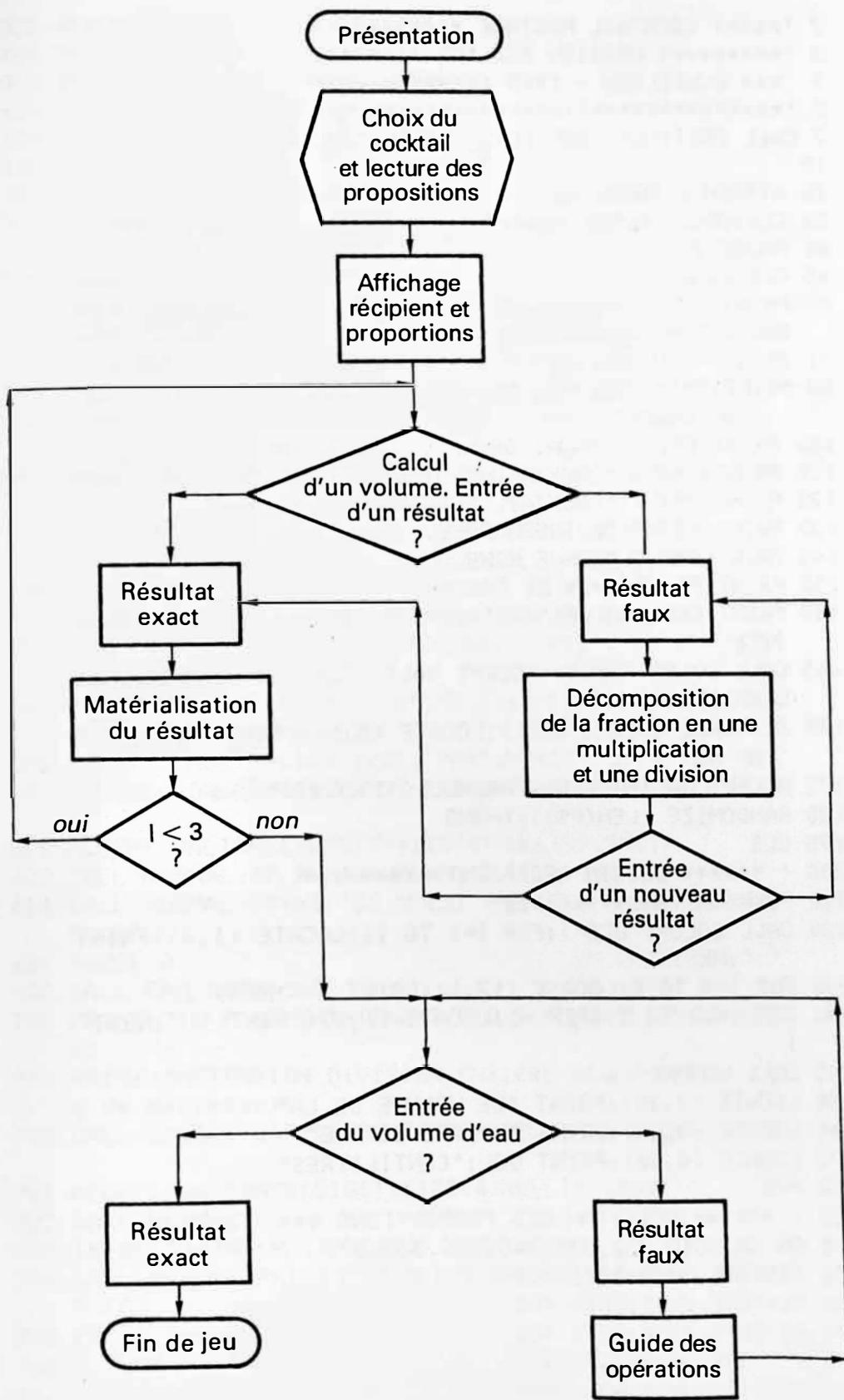
L'auteur décline toute responsabilité dans les proportions de ces cocktails inventés pour le programme !

Vous pouvez modifier les lignes 2000-2520 en y composant vos propres cocktails.

*Attention deux impératifs :*

- trois composants + de l'eau ;
- ne pas utiliser de fractions plus petites que 1/10 (car le joueur aura des problèmes si le volume choisi est 10 cl).

Enfin, si vous voulez faire connaître à vos amis vos boissons préférées, toutes les variations d'Alexandra, de Manhattan, de Bloody Mary, etc., peuvent y être notées. A eux d'avoir le bon tour de main !



```
2 !***** COCKTAIL MAGIQUE *****
3 !***** VERSION EXL 100 *****
4 !*** D.NIELSEN - 1985 *****
5 !***** *****
7 CALL INIT:CLS "BCC":CALL COLOR("0BC")
10 !
20 ATTACH LIQUID
50 CLS:CALL DBLPRT("COCKTAIL MAGIQUE","0RC",4,10)
60 PAUSE 2
65 CLS:LOCATE (2,5)
70 PRINT "UN COCKTAIL EST UN MELANGE DE PLUSIEURS BOIS-
    SONS DANS UN VERRE";
71 PRINT " OU UNE CARAFE"
80 PRINT:PRINT "TU VAS DEVOIR PREPARER UN COCKTAIL DE
    TON CHOIX"
100 PRINT:PRINT "HAWAI DRINK.....1"
110 PRINT:PRINT "COCKTAIL EXOTIQUE....2"
120 PRINT:PRINT "COCKTAIL ESQUIMAUX...3"
130 PRINT:PRINT "BOISSON ROSE.....4"
140 PRINT:PRINT "BONNE MINE.....5"
150 PRINT:PRINT "VIN DE FRAISE.....6"
160 PRINT:CALL COLOR("0RC"):PRINT "QUEL EST TON CHOIX
    ? ";
165 CALL COLOR("0BC"):ACCEPT VALIDATE(DIGIT)SIZE(1)NUL-
    L(1),CH
170 CLS:CALL COLOR("0BC"):LOCATE (5,1):PRINT "QUEL EST
    TON NOM ? ";
172 ACCEPT VALIDATE(UALPHA)NULL("INCONNU"),P$
180 RANDOMIZE (LEN(P$)):Y=RND
195 CLS
200 ! ***** DESSIN RECIPIENT *****
210 VOL=(INT(RND*9)+1)*10
220 CALL COLOR("0CB"):FOR I=1 TO 11:LOCATE (I,4):PRINT
    " ":"NEXT I
230 FOR I=4 TO 8:LOCATE (12,I):PRINT " ":"NEXT I
240 FOR I=11 TO 1 STEP -1:LOCATE (I,8):PRINT " ":"NEXT
    I
245 CALL NORMAL
250 LOCATE (2,20):PRINT "LE VOLUME DE LA"
260 LOCATE (3,20):PRINT "CARAFE EST DE"
270 LOCATE (4,20):PRINT VOL;"CENTILITRES"
280 F=0
300 ! ***** NOTATION DES PROPORTIONS ***
310 ON CH GOTO 320,330,340,350,360,370
320 RESTORE 2000:GOTO 400
330 RESTORE 2100:GOTO 400
340 RESTORE 2200:GOTO 400
350 RESTORE 2300:GOTO 400
360 RESTORE 2400:GOTO 400
```

```

370 RESTORE 2500
400 FOR I=1 TO 3
410 READ NU(I),DE(I),A$(I),COU$(I)
420 LOCATE (4+2*I,20)
430 PRINT A$(I)
440 NEXT I
470 LOCATE (12,20):PRINT "ET DE L'EAU GAZEUSE"
500 !***** CALCUL DES VOLUMES *****
510 FOR I=1 TO 3
520 LOCATE (14,1)
530 PRINT "COMBIEN DOIS-TU VERSER DE ";A$(I);"?"
540 PRINT:PRINT "IL FAUT QUE TU CALCULES CE QUE
    REPRESENTE ";NU(I);
542 PRINT "/";DE(I); DE ";VOL;" CENTILITRES"
545 PRINT "ATTENTION, IL FAUT QUE TU ARRONDISSES TON
    RESULTAT AU PREMIER ";
546 PRINT "ENTIER","INFERIEUR (pas de nombre decimal)"

550 CALL COLOR("ORC"):PRINT "QUELLE EST TA REPONSE ";
551 ACCEPT VALIDATE(DIGIT)SIZE(4)NULL(0),REP(I)
552 CALL NORMAL
560 IF REP(I)=INT((VOL*NU(I))/DE(I))THEN 1000
570 CALL FAC:PRINT "TU TE TROMPES, ";P$;". VOICI","COM
    MENT TU DOIS CALCULER"
580 PRINT "POUR PRENDRE LES";NU(I);"/";DE(I);DE";VOL;
    "c1,";IL FAUT D'ABORD";
590 PRINT " MULTIPLIER";VOL;"PAR";NU(I)
600 CALL COLOR("ORC"):PRINT "COMBIEN CELA FAIT-IL ? ";

601 ACCEPT VALIDATE(DIGIT)SIZE(4)NULL(0),RN(I)
602 CALL NORMAL:IF RN(I)=VOL*NU(I)THEN 700
610 CALL NORMAL:PRINT "CE N'EST PAS CELA ";P$;";RECOMM
    ENCE"
620 PAUSE 2
630 CALL FAC:GOTO 580
700 PRINT "IL FAUT MAINTENANT DIVISER";RN(I);"PAR";DE(
    I)
710 PRINT "ATTENTION DIVISION ENTIERE SEULEMENT (pa
    s de decimaux)"
720 CALL COLOR("ORC"):PRINT "COMBIEN CELA FAIT-IL ? ";

721 ACCEPT VALIDATE(DIGIT)SIZE(4)NULL(0),RD(I)
722 CALL NORMAL
730 IF RD(I)=INT(RN(I)/DE(I))THEN 1000
740 CALL NORMAL:PRINT "CE N'EST PAS CELA ";P$;";RECOMM
    ENCE"
750 PAUSE 5
760 CALL FAC:GOTO 700
1000 ! ***** RESULTAT EXACT *****

```

```

1004 CALL FAC
1005 REP(I)=INT((VOL*NU(I))/DE(I))
1010 PRINT "C'EST EXACT, ";P$
1020 H=INT((VOL*NU(I))/DE(I)):IF H>2 THEN H=H-1
1021 PAUSE 2
1030 CALL LIQUID(H,VOL,COU$(I))
1040 F=F+H
1050 PAUSE 5
1060 CALL FAC
1070 NEXT I
1200 ! ***** COMPLEMENT D'EAU *****
1210 LOCATE (15,1):CALL COLOR("ORC"):PRINT "COMBIEN D'E
    AU GAZEUSE FAUT-IL ";
1211 PRINT "RAJOUTER ? ";
1212 ACCEPT VALIDATE(DIGIT)SIZE(4)NULL(0),R
1220 CALL LIQUID(H,VOL,"W")
1230 IF R<VOL-(REP(1)+REP(2)+REP(3))THEN CALL FAC:LOCAT
    E (15,1):PRINT "CE";
1231 IF R>VOL-(REP(1)+REP(2)+REP(3))THEN PRINT " N'EST
    PAS ASSEZ, "&P$:GOTO 1300
1240 IF R>VOL-(REP(1)+REP(2)+REP(3))THEN GOSUB 1700
1250 IF R=VOL-(REP(1)+REP(2)+REP(3))THEN 1500
1300 PRINT "IL FAUT D'ABORD ADDITIONNER LES VOLUMES DES
    3 JUS OU SIROPS VERSES"
1350 PRINT REP(1);"+";REP(2);"+";REP(3)
1360 CALL COLOR("ORC"):PRINT "COMBIEN CELA FAIT-IL ? ";

1361 ACCEPT VALIDATE(DIGIT)SIZE(4)NULL(0),TR
1370 IF TR=REP(1)+REP(2)+REP(3)THEN 1450
1380 CALL NORMAL:PRINT "CE N'EST PAS CELA ";P$;,RECOMM
    ENCE"
1390 PAUSE 2
1400 CALL FAC:GOTO 1250
1450 CALL COLOR("ORC"):PRINT "MAINTENANT COMBIEN D'EAU
    GAZEUSE FAUT-IL";
1451 PRINT "RAJOUTER ? ";
1452 ACCEPT VALIDATE(DIGIT)SIZE(4)NULL(0),R
1453 CALL NORMAL
1480 GOTO 1220
1500 CALL FAC:PRINT "C'EST EXACT, ";P$;" TU PEUX DEVENI
    R"," UN VRAI BARMAN"
1505 PRINT
1510 CALL COLOR("ORC"):PRINT "UN AUTRE COCKTAIL (O/N) ?
    ";
1511 ACCEPT VALIDATE("ON")SIZE(1)NULL("N"),REC$
1520 IF REC$<>"0"THEN PRINT "AU-REVOIR":PAUSE 1:CLS:END

1530 PAUSE 2
1540 CALL NORMAL:CLS:RELEASE LIQUID:GOTO 10

```

```

1700 !
1725 CALL FAC
1730 LOCATE (20,5):PRINT "CA DEBORDE....."
1740 PAUSE 4
1750 CALL FAC:RETURN
2000 DATA 1,3,JUS D'ANANAS,Y
2010 DATA 1,5,LAIT DE COCO,W
2020 DATA 1,4,SIROP DE SUCRE,C
2100 DATA 1,5,JUS DE MANGUE,Y
2110 DATA 2,8,JUS DE PAPAYE,G
2120 DATA 1,10,JUS DE GOYAVE,b
2200 DATA 2,5,CITRON PRESSE,Y
2210 DATA 1,10,SIROP DE MENTHE,G
2220 DATA 1,3,GLACE PILLEE,C
2300 DATA 1,3,JUS DE FRAMBOISES,M
2310 DATA 1,4,SIROP DE SUCRE,W
2320 DATA 1,10,CREME FRAICHE,W
2400 DATA 3,5,JUS DE TOMATE,R
2410 DATA 1,4,JUS DE CAROTTES,Y
2420 DATA 1,10,JUS DE CELERI,W
2500 DATA 2,5,JUS DE FRAISE,R
2510 DATA 2,10,JUS DE CITRON,Y
2520 DATA 1,4,SIROP DE SUCRE,W
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
      50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10030 SUBEND
10040 SUB PLAY(NO$,L,D)
10050 FOR I=1 TO LEN(NO$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)
10070 P=VAL(SEG$( "199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,C$,X,Y)
10105 CALL COLOR(C$)
10110 CALL COLOR("0LH")
10120 K$="" :FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("0BC"):SUBEND
10170 SUB NORMAL
10180 CALL COLOR("0BC"):SUBEND
10200 SUB LIQUID(H,VOL,COLO$)
10203 CALL FAC
10205 IF TT=0 THEN TT=12
10206 H=INT((H*10)/VOL)+1

```

```
10210 FOR I=TT-1 TO TT-H STEP -1
10220 CALL COLOR("OC"&COL0$)
10230 LOCATE (I,5)
10240 PRINT RPT$(" ",3)
10250 NExt I
10260 TT=TT-.4
10270 CALL NORMAL
10280 SUBEND
10300 SUB FAC
10310 FOR I=14 TO 21:LOCATE (I,1)
10320 PRINT RPT$(" ",40);
10330 NEXT I
10331 PRINT RPT$(" ",35);:LOCATE (14,1)
10332 SUBEND
```

# **MASSE ET VOLUME**

## **Thème**

Il s'agit de calculer le volume d'un cube, puis la masse de ce cube rempli d'eau. La masse est vérifiée en mettant des poids marqués sur une balance.

## **Niveau**

A partir de 10 ans (Cours moyen 1<sup>re</sup> année et Cours moyen 2<sup>e</sup> année) et collège.

## **Intérêt pédagogique**

### **Connaissances**

- calcul du volume d'un cube (référence à l'aire d'une de ses faces et à sa hauteur) ;
- connaissance des unités de mesure de volume et de mesure des masses ;
- approche de la notion de masse volumique.

### **Savoir-faire**

- calcul mental ;
- utilisation progressive de poids marqués pour déterminer la masse d'un objet.

## Déroulement

Un cube se dessine sur l'écran. La longueur du côté de ce cube est déterminée par un nombre aléatoire : nombre entier entre 1 et 10. Le joueur doit calculer le volume de ce cube. S'il connaît la formule et est doué pour le calcul mental, il peut donner directement la réponse.

S'il ne sait pas comment faire, ou si sa réponse est erronée, le programme l'invite à calculer d'abord la surface de la base du cube, puis à multiplier cette surface par la hauteur pour trouver le volume. Il faut noter que, si le côté du cube est supérieur à 5 cm, cette deuxième opération ne relève plus tellement du calcul mental et l'enfant aura besoin de poser cette multiplication par écrit.

Ensuite, une balance se dessine sur la moitié haute de l'écran : sur le plateau de gauche se trouve notre cube rempli d'eau, sur celui de droite, rien pour l'instant. Donc le plateau de gauche est plus bas et l'aiguille entre les deux plateaux penche vers la gauche.

Dans une série de huit cases jaunes, des valeurs de poids marqués s'inscrivent au bas de l'écran (500 g - 200 g - 100 g - 50 g - 20 g - 10 g - 5 g - 1 g). Le joueur doit donner la lettre correspondant au poids qu'il veut mettre dans la balance.

Lorsque la masse exacte du cube est trouvée, les deux plateaux sont à la même hauteur.

Si l'on met trop de poids sur le plateau de droite, celui-ci baisse et il faut recommencer la pesée.

Le programme rappelle aussi la notion de masse volumique : « 1 cm<sup>3</sup> d'eau pèse 1 g, donc... ».

## Commentaires

- |                  |  |
|------------------|--|
| <b>7 à 101</b>   | Initialisation et présentation du jeu.   |
| <b>160</b>       | Envoie en 1500 pour effacer de la mémoire les valeurs des poids sur le plateau au cours du jeu précédent.                                      |
| <b>195 à 236</b> | Volume du cube. Le cube est dessiné et affiché par un sous-programme ligne 200 : GOSUB 2000.<br><i>Ligne 220 : Valeur de C = Côté du cube.</i> |

- 240** Test réponse.
- 250 à 440** Décomposition du calcul du volume en cas de réponse fausse.
- 500 à 530** Calcul du volume : bonne réponse.
- 540 à 550** Dessin de la balance à l'intérieur d'un sous-programme :
- GOSUB 3000 : définition du plateau gauche, définition du plateau droit, affichage des plateaux (la variable H détermine la place des plateaux ligne 540).
- 600 à 650** Présentation de la recherche de la masse du cube. Les poids sont dessinés en sous-programme (ligne 4000).
- 700 à 768** Entrée des réponses.
- Lignes 700 à 708* : En fonction de la case visée, le programme va prendre dans les lignes 5000 à 5700 la valeur de PD (poids) et le dessin des poids correspondants.
- Lignes 710 à 740* : Déterminent la valeur de H, c'est-à-dire la hauteur des plateaux l'un par rapport à l'autre (en fonction des poids placés sur le plateau).
- Ligne 750* : La balance est à nouveau affichée en tenant compte des modifications apportées par la réponse.
- Lignes 765-768* : La valeur de PD est inscrite au-dessus de la balance.
- 775 à 810** Test de fin de recherche de poids.
- 900 à 915** Gagné !
- 920 à 965** Fin de jeu.
- 1000 à 1040** Affichage poids trop élevé.
- Ligne 1020* : Envoie en 1500 pour effacer les valeurs précédentes de PD.
- Ligne 1040* : Retour à 540.
- 2000 à 2100** Dessin du cube par des lignes noires.
- 3000 à 3590** Définition du dessin du plateau de gauche.  
Définition du dessin du plateau de droite.
- 4000 à 4070** Dessin des poids marqués.
- 5000 à 5730** Valeur des poids

- 10000 à 10030** Initialisation générale.
- 10040 à 10090** Jouent une mélodie.
- 10100 à 10160** Ecriture en double taille.
- 10170 à 10190** Ecriture normale.
- 10200 à 10250** Dessin d'un poids.
- 10300 à 10390** Dessin des plateaux.
- 10400 à 10630** Dessin du cube.
- 10700 à 10780** Effacement d'un plateau.

## Particularités techniques

Le choix des poids se fait avec la lettre indiquée sur le poids.

Huit zones sont affichées pour le dessin des poids marqués (on ne peut utiliser que huit cases au maximum, en raison de la place, c'est pourquoi il n'y a que huit poids et que le poids de 2 g a disparu !).

L'instruction ON... GOSUB envoie chercher (dans les lignes 5000 à 5700) les éléments de réponse et d'affichage correspondant au choix du joueur.

Ensuite, la balance est redessinée : les lignes 3510-3520 permettent d'effacer le précédent dessin de la balance sans effacer le reste de l'écran.

## Modifications

Dans le programme présenté, le joueur peut utiliser chaque poids autant de fois qu'il le souhaite. On peut imaginer un jeu qui corresponde exactement à la boîte de poids marqués. Il faut limiter leur utilisation :

une	fois	pour	500 g
une	fois	pour	200 g
deux	fois	pour	100 g
une	fois	pour	50 g
une	fois	pour	20 g
deux	fois	pour	10 g
deux	fois	pour	5 g
cinq	fois	pour	1 g (pour compenser l'absence du poids de 2 g).

Il faut alors ajouter dans chaque série (5000 - 5100 - 5200 etc.) un compteur.

*Exemple :*

5002 IF A=1 THEN 1200

5003 A=A+1

1200 LOCATE:PRINT« IL N'Y A PLUS CE POIDS DANS LA  
BOITE »

1210 GOTO 700

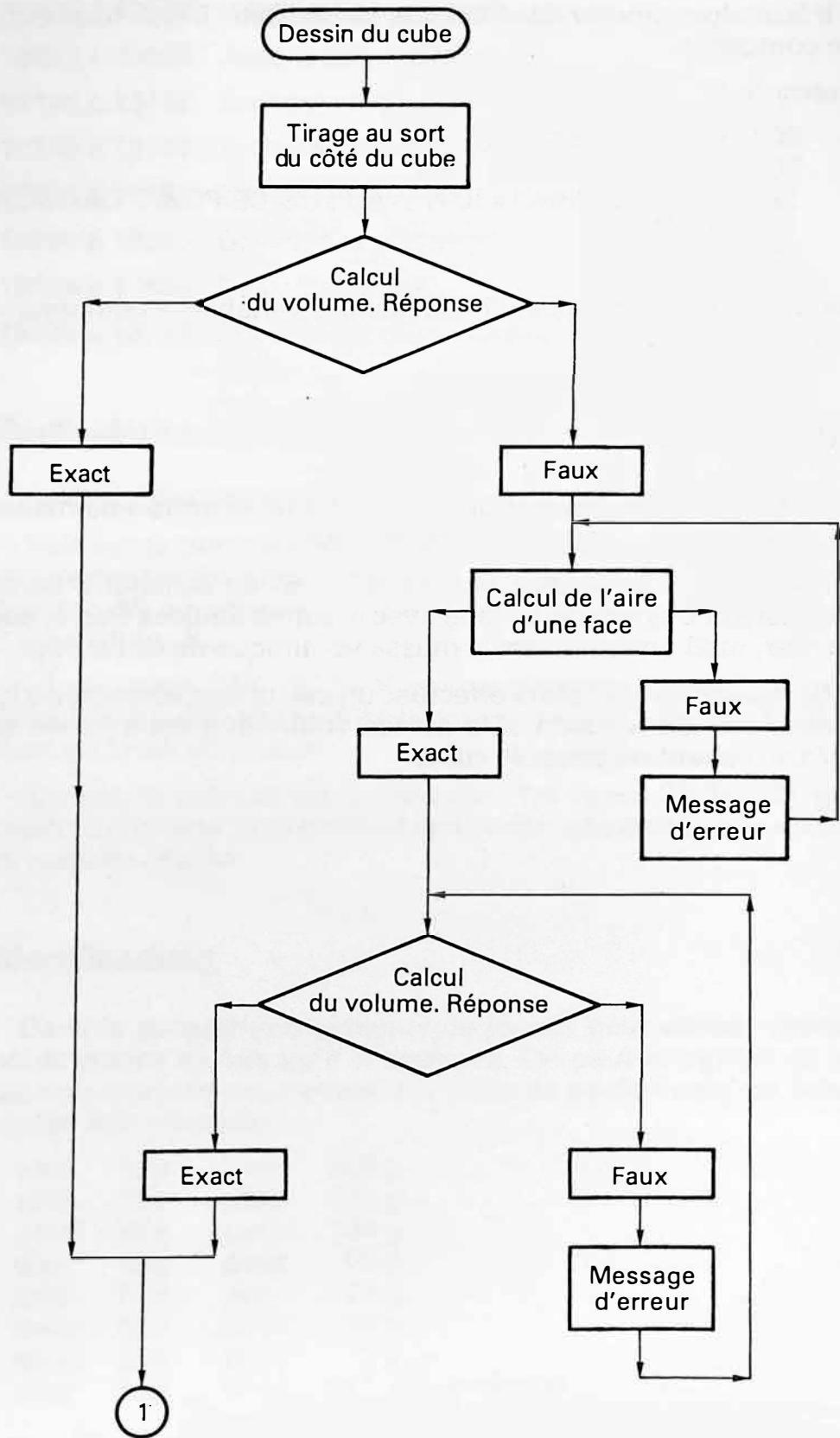
Ne pas oublier non plus d'initialiser ces variables « compteur » au début du jeu.

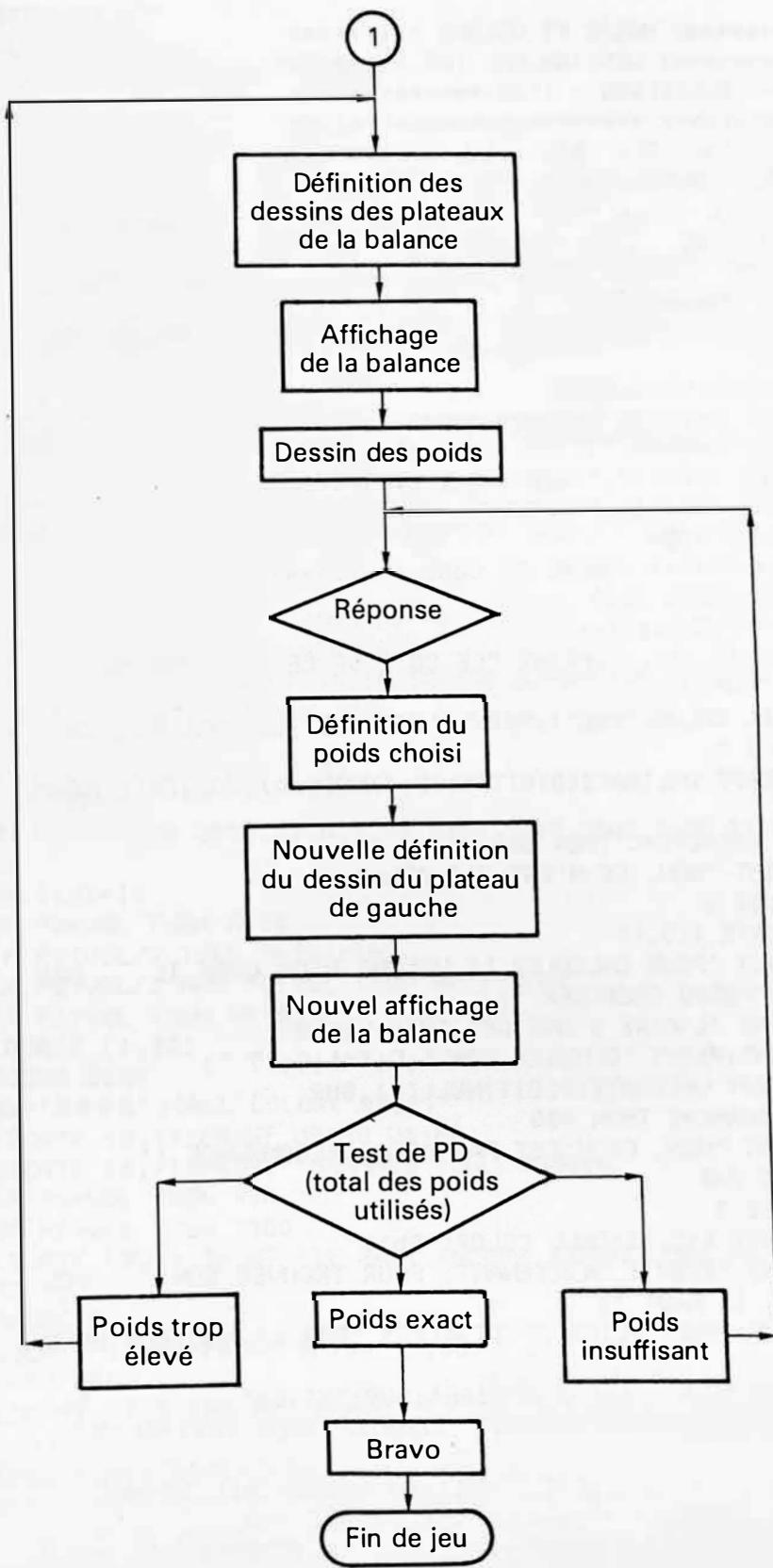
## Adaptations

Ce programme permet une approche de la notion de masse volumique avec un liquide simple : l'eau.

Pour des enfants plus grands (6<sup>e</sup> et 5<sup>e</sup> de collège) il serait intéressant de remplir le cube avec d'autres liquides (huile, eau de mer, etc.) en indiquant la masse volumique de ce liquide.

Le joueur devrait alors effectuer un calcul supplémentaire (et même une conversion si la masse volumique est affichée en kg/dm<sup>3</sup>) avant de peser le cube.





```

2 !***** MASSE ET VOLUME *****
3 !***** VERSION EXL 100 *****
4 !*** D.NIELSEN - 1985 *****
5 !***** *****
7 CALL INIT:CLS "BCC":CALL COLOR("0BC")
40 CALL DBLPRT("MASSE ET VOLUME","OR",5,8)
50 LOCATE (13,1):PRINT "TU VAS DEVOIR CALCULER LE VOL
UME D'UN CUBE, SA MASSE";
51 PRINT " REMPLI D'EAU, PUIS", "VERIFIER CETTE MASSE
SUR UNE BALANCE"
60 PRINT:PRINT "LORSQUE TU ES PRET APPUIE SUR UNE TOU
CHE"
70 I$=KEY$:RANDOMIZE
80 CALL CHAR(48,"0000000000FFFFF0000")
90 CALL CHAR(49,"111111111111110000")
100 CALL CHAR(50,"00000000111111110000")
101 CALL CHAR(51,"FFFFFFFFFFFFFFFFFF")
160 GOSUB 1500
195 !***** VOLUME DU CUBE *****
200 CLS:GOSUB 2000
220 C=INT(RND*10)+1
230 LOCATE (16,1):PRINT "LE COTE DE CE CUBE MESURE ";C
;" CM"
235 CALL COLOR("0RC"):PRINT "QUEL EST SON VOLUME, EN c
m3 ? ";
236 ACCEPT VALIDATE(DIGIT)SIZE(6)NULL(0),VOL:CALL NORM
AL
240 IF VOL=C*C*C THEN 500
250 PRINT "NON, CE N'EST PAS CELA"
260 PAUSE 3
270 LOCATE (13,1)
271 PRINT "POUR CALCULER LE VOLUME D'UN CUBE IL FAU
T D'ABORD CHERCHER ";
272 PRINT "L'AIRE D'UNE DE", "SES FACES"
280 PRINT:PRINT "COMBIEN FONT";C;"X";C;"? ";
290 ACCEPT VALIDATE(DIGIT)NULL(0),SUR
300 IF SUR=C*C THEN 400
310 PRINT "NON, CE N'EST PAS CELA, RECOMMENCE !"
320 GOTO 260
400 PAUSE 2
404 LOCATE (13,1):CALL COLOR("0b")
405 PRINT "BIEN ! MAINTENANT, POUR TROUVER SON VOL
UME, IL FAUT ";
406 PRINT "MULTIPLIER CETTE AIRE", "PAR LA HAUTEUR DU C
UBE"
410 PRINT:PRINT "C'EST A DIRE";SUR;"X";C;" "
420 CALL COLOR("0RC"):PRINT "COMBIEN CELA FAIT-IL ? ";
425 ACCEPT VALIDATE(DIGIT)NULL(0),VOLUM:CALL NORMAL
430 IF VOLUM=SUR*C THEN VOL=VOLUM:GOTO 500
440 PRINT "NON, CE N'EST PAS CELA, RECOMMENCE !"

```

```

450 GOTO 400
500 PAUSE 2
510 LOCATE (22,1):PRINT "BRAVO !"
520 PAUSE 5
530 CLS
540 H=16
550 GOSUB 3000
600 ! ***** RECHERCHE DES POIDS *****
610 LOCATE (2,1):PRINT "LE CUBE EST REMPLI D'EAU QUELLE
E EST SA MASSE ? ";
615 PRINT "POUR LE TROUVER, TU VAS PESER CE CUBE"
620 PRINT "VOICI LES POIDS DONT TU DISPOSES"
630 GOSUB 4000
635 PAUSE 5
640 LOCATE (2,1):PRINT "TAPE LA LETTRE CORRESPONDANT A
U POIDS QUE TU VEUX";
641 PRINT " UTILISER "
650 PRINT "ATTENTION TU NE POURRAS PAS ENLEVER LES POI
DS DE LA BALANCE "
655 IF PD=0 THEN 700
656 LOCATE (6,1):PRINT USING U$;PD
660 LOCATE (6,7):PRINT "GRAMMES"
700 LOCATE (5,22):ACCEPT VALIDATE("ABCDEFGH")NULL("I")
,REP$:IF REP$="I"THEN 700
702 REP=ASC(REP$)
706 CALL PLAY("REMIFA",1,0)
707 REP=REP-64
708 ON REP GOSUB 5000,5100,5200,5300,5400,5500,5600,57
00
710 H=10:H1=10
711 IF PD=VOL THEN H=10
720 IF PD<VOL/2 THEN H=12:H1=8
730 IF PD>VOL/2 AND PD<VOL THEN H=11:H1=9
740 IF PD>VOL THEN H=-8
745 LOCATE (1,22)
750 GOSUB 3000
760 U$="###":CALL COLOR("ORC")
765 LOCATE (6,1):PRINT USING U$;PD
768 LOCATE (6,7):PRINT "GRAMMES":CALL NORMAL
775 IF PD=VOL THEN 900
780 IF PD>VOL THEN 1000
790 LOCATE (7,1):PRINT "IL FAUT ENCORE AJOUTER DES POI
DS ";
800 PAUSE 5
805 LOCATE (7,1):PRINT RPT$( " ",33);
810 GOTO 700
900 LOCATE (7,1):PRINT "BRAVO !
905 CALL PLAY("DOREMIFASOLASI",1,0):PAUSE 3
910 CLS:PRINT "UN cm3 D'EAU PESE UN GRAMME DONC";VOL;"cm3","PESE";

```

```

911 IF VOL>1 THEN PRINT "NT";
915 PRINT VOL;"GRAMME";:IF VOL>1 THEN PRINT "S";
920 PAUSE 5
930 CLS
940 CALL COLOR("ORC")
941 PRINT "UNE AUTRE PARTIE 'O/N' ? ";
942 ACCEPT VALIDATE("ON")SIZE(1)NULL("N"),CH$
950 IF CH$="0"THEN PD=0:H=0:H1=8:H2=12:GOTO 160
955 CLS
960 CALL DBLPRT("AU REVOIR","0b",10,5)
965 PAUSE 2:CLS:END
1000 !***** POIDS TROP ELEVE *****
1001 CALL COLOR("OM"):CALL PLAY("DOMISOSILA",4,1)
1002 FOR I=1 TO 5:LOCATE (I,1):PRINT RPT$( " ",40);:NEXT

1005 LOCATE (1,1):CALL DBLPRT("C'EST TROP","OR",1,1):PR
INT:PRINT "RECOMMENCE"
1020 GOSUB 1500
1030 PAUSE 3
1040 CLS:GOTO 540
1500 FOR I=1 TO 2
1510 A$(I)="":B$(I)="":C$(I)="":D$(I)="":E$(I)="":F$=""
:G$="
1520 NEXT I
1530 PD=0:H=0:H1=8:H2=12:HP=0
1540 RETURN
2000 !***** DESSIN DU CUBE *****
2100 CALL CUBE
2100 RETURN
3000 !***** DESSIN DU PLATEAU *****
3500 !***** DEPLACEMENT DES PLATEAUX ***
3503 H2=H
3504 IF HK0 THEN H2=8:H1=12
3505 IF HP=0 THEN H1=8:H2=12:GOTO 3541
3510 CALL EFFPLAT(HP,6)
3520 CALL EFFPLAT(HP1,26)
3541 CALL PLAT(H2,6)
3550 CALL PLAT(H1,26)
3560 HP=H2:HP1=H1
3590 RETURN
4000 ! ***** DESSIN DES POIDS *****
4010 P$(0)="500":P$(1)="200":P$(2)="100":P$(3)="50":P$(4)="20":P$(5)="10"
4011 P$(6)=" 5":P$(7)=" 1"
4020 FOR I=0 TO 7
4030 CALL POIDS(21,1+I*5,I)
4040 LOCATE (22,1+I*5):PRINT P$(I)
4060 NEXT I
4070 RETURN
4990 !***** CHOIX DES POIDS *****
5000 PD=PD+500

```

```
5010 A$(1)="33"
5020 A$(2)="33"
5030 RETURN
5100 PD=PD+200
5110 B$(1)="3"
5120 B$(2)="3"
5130 RETURN
5200 PD=PD+100
5210 C$(1)="0"
5220 C$(2)="3"
5230 RETURN
5300 PD=PD+50
5310 D$(1)="1"
5320 D$(2)="1"
5330 RETURN
5400 PD=PD+20
5410 E$(1)="2"
5420 E$(2)="1"
5430 RETURN
5500 PD=PD+10
5510 F$="3"
5530 RETURN
5600 PD=PD+5
5630 RETURN
5700 PD=PD+1
5720 G$="2"
5730 RETURN
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
      50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10030 SUBEND
10040 SUB PLAY(NO$,L,D)
10050 FOR I=1 TO LEN(NO$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)
10070 P=VAL(SEG$("199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,C$,X,Y)
10105 CALL COLOR(C$)
10110 CALL COLOR("OLH")
10120 K$="" :FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("OBC"):SUBEND
10170 SUB NORMAL
10180 CALL COLOR("OBC")
10190 SUBEND
10200 SUB POIDS(X,Y,L)
```

```

10210 CALL COLOR("1RC")
10220 FOR I=X TO X-5 STEP -1
10225 LOCATE (I,Y)
10230 PRINT RPT$("3",4)
10240 NEXT I
10241 LOCATE (X-6,Y+1):PRINT "33"
10243 CALL DBLPRT(CHR$(65+L),"0b",Y+1,I+2)
10250 SUBEND
10300 SUB PLAT(H,DE)
10310 CALL CHAR(52,"F0F0F0F0F0F0F0F0F0F0")
10320 CALL CHAR(53,"FFFFFFFFF0000000000")
10330 CALL CHAR(54,"00FFFFFFF0000000000")
10335 CALL COLOR("1bC")
10340 LOCATE (H,DE):PRINT "4"      4"
10350 LOCATE (H+1,DE):PRINT "56666666665"
10352 IF DE<>6 THEN 10360
10355 CALL COLOR("1GC")
10356 LOCATE (H-1,DE+4):PRINT "33"
10357 LOCATE (H,DE+4):PRINT "33"
10360 CALL NORMAL
10390 SUBEND
10400 SUB CUBE
10410 CALL CHAR(55,"808080808080808080FF")
10420 CALL CHAR(56,"808080808080808080")
10430 CALL CHAR(57,"FF8080808080808080")
10440 CALL CHAR(58,"FF000000000000000000")
10450 CALL CHAR(59,"FF010101010101010101")
10460 CALL CHAR(60,"01010101010101010101")
10470 CALL CHAR(61,"010101010101010101FF")
10480 CALL CHAR(62,"000000000000000000FF")
10490 CALL CHAR(63,"01020204081010204080")
10500 CALL CHAR(65,"FF030505091111214181")
10510 CALL COLOR("1RC")
10520 LOCATE (11,15):PRINT "7";RPT$(">",8);"="
10530 FOR I=10 TO 5 STEP -1
10540 LOCATE (I,15):PRINT "8"      <
10550 NEXT I
10560 LOCATE (4,15):PRINT "9:::::::"; <
10570 LOCATE (3,15):PRINT "?      ? <"?
10580 LOCATE (2,16):PRINT "?      ?<"?
10590 LOCATE (1,17):PRINT "?:::::::A"
10600 J=24:FOR I=11 TO 9 STEP -1:J=J+1:LOCATE (I,J):PRINT
T "?":NEXT I
10620 FOR I=8 TO 5 STEP -1:LOCATE (I,J):PRINT "<":NEXT I

10630 CALL NORMAL:SUBEND
10700 SUB EFFPLAT(H,DE)
10710 FOR I=H-1 TO H+1
10720 LOCATE (I,DE):PRINT "
10730 NEXT I
10780 SUBEND

```

# KIM NOMBRES

## Thème

Sur le principe du jeu de Kim, reconnaissance d'un nombre qui a été enlevé dans une liste affichée.

## Niveau

Ce jeu est présenté en trois versions :

- **facile** : à partir de 6 ans.
- **moyen** : à partir de 8 ans.
- **difficile** : essayez ce programme vous-même et vous verrez !

## Intérêt pédagogique

### Connaissances

- lecture de nombres ;
- classement de nombres.

### Savoir-faire

- mémoire, et particulièrement mémoire visuelle.

## Déroulement

Après une présentation du jeu, une série de nombres tirés au sort s'affiche d'une manière irrégulière sur l'écran, ponctuée par une courte séquence musicale.

Puis, lorsque le joueur a suffisamment lu ces nombres, EXL 100 retire un nombre et inscrit de nouveau les autres. Le joueur doit retrouver le nombre disparu. Ensuite une nouvelle série, plus courte, apparaît... et cela jusqu'à ce qu'il n'y ait plus que quatre nombres dans la série.

Comme le jeu de Kim, ce jeu est évidemment plus facile à la fin qu'au début.

Le listing de ce programme étant relativement court, trois adaptations différentes vous sont proposées afin de vous aider à créer vos propres adaptations d'autres programmes.

- **1<sup>re</sup> version : difficile**

20 nombres tirés au sort entre 1 et 50. Dans la 2<sup>e</sup> liste, les nombres ne sont pas présentés dans le même ordre. 9 jeux.

- **2<sup>e</sup> version : moyen**

14 nombres tirés au sort entre 1 et 10. Nombres présentés aussi dans un ordre différent la 2<sup>e</sup> fois. 6 jeux.

- **3<sup>e</sup> version : facile**

10 nombres tirés au sort entre 1 et 10. Les nombres sont, les deux fois, dans le même ordre. 4 jeux.

## Commentaires

**10 à 135**      Initialisation et présentation du jeu.

**200 à 295**      Première séquence.

*Ligne 220 :* Un nombre est tiré au sort dans un encadrement différent selon le niveau.

*Ligne 230 :* La variable aléatoire X inscrit le nombre sur l'écran à une place chaque fois différente.

*Lignes 240-250 :* Pour chaque nombre, une fenêtre est déterminée sur 20 colonnes de large et 2 (ou 4 version 3) de haut. Le nombre s'inscrira dans cette fenêtre à une place déterminée par X. Ce qui donne une impression d'affichage en désordre, sans avoir le risque de faire effacer un nombre par un autre.

Sur la moitié gauche de l'écran, vont s'inscrire les 10 (ou 7 ou 5) premiers nombres, puis les suivants sur l'autre moitié.

*Ligne 270 :* Cette musique sert à temporiser l'affichage et à conserver l'attention du joueur tout entière sur l'écran.

*Lignes 290-295 :* C'est le joueur qui décide du moment où l'on passe à la suite.

### 300 à 420

Affichage de la liste moins un nombre.

*Ligne 320 :* Le nombre à retirer est tiré au sort. Pour les versions **difficile** et **moyen**, la place des nombres est modifiée : l'affichage commence par le bas de l'écran à droite.

Dans la version **facile**, les nombres sont affichés dans le même ordre, mais à une place différente à chaque ligne.

### 500 à 525

Réponse - Test de la réponse.

### 530 à 550

Affichage bonne réponse.

### 600 à 620

Affichage « gagné ».

### 630 à 660

Test suite de jeu en fonction du nombre de chiffres (N).

### 800 à 820

Affichage du score et fin de jeu.

### 10000 à 10030

Initialisation générale.

### 10040 à 10090

Jouent une mélodie.

### 10100 à 10160

Affichage double taille.

## Particularités techniques

La séquence aléatoire est initialisée dans la ligne 140.

Dans la première séquence, chaque nombre tiré au sort est affecté d'un numéro d'ordre A(l), ce qui permet de l'afficher de nouveau dans la deuxième séquence (le nombre retiré est remplacé par « ? », ligne 390).

## Modifications

Plusieurs variables peuvent être changées afin de transformer ce jeu ou l'adapter à des niveaux intermédiaires :

- N = quantité de nombres à afficher au départ ;
- nombre de jeux : ici  $N=N-2$  qui enlève deux nombres à chaque jeu donne 4 jeux (facile), 6 jeux (moyen), et 9 jeux (difficile).

On peut envisager aussi de donner une deuxième chance de réponse en cas d'erreur.

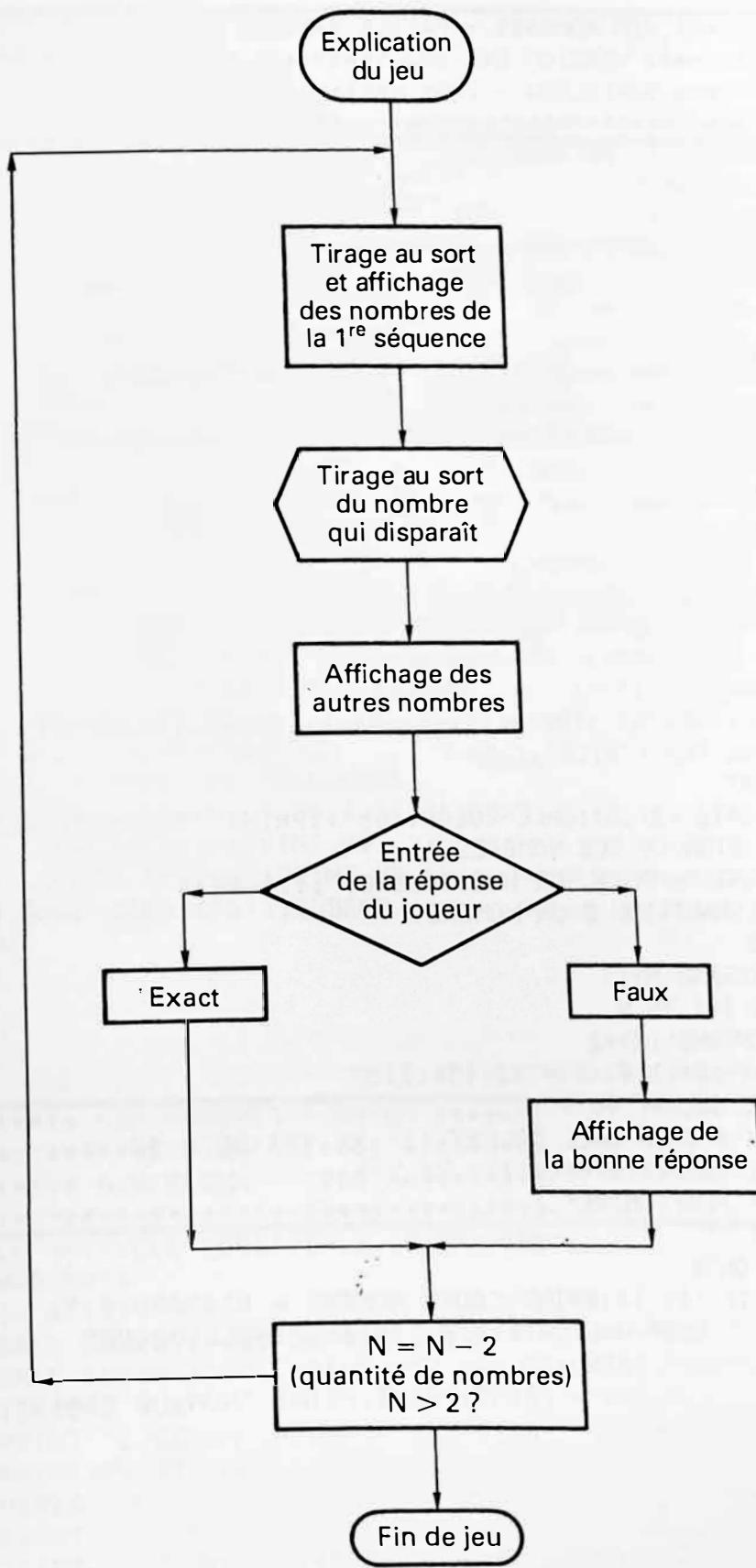
Il est aussi possible de faire disparaître deux nombres (au lieu d'un) dans la séquence tronquée et de n'en demander qu'un seul, sur les deux, comme réponse.

Des jeunes enfants (version facile) peuvent être gênés par l'affichage du même nombre plusieurs fois (ligne 220 : A(l) est déterminé par RND). Si vous voulez éviter cette répétition, vous pouvez ajouter entre les lignes 220 et 230 un petit programme qui va comparer le nombre A(l) à ceux déjà sortis (voir le programme « Construis une phrase »).

## Adaptations

Chacune des trois adaptations présentées peut être l'objet d'autres transformations. Par exemple, vous pouvez garder, pour la deuxième séquence, la même place pour chaque nombre en affectant à la variable X l'indice (l) dans la première séquence.

C'est à vous, en fonction du public à qui vous destinez ce programme, de l'adapter en conséquence.



---

```

2 !***** KIM NOMBRES - FACILE *****
3 !***** VERSION EXL 100 *****
4 !***** D.NIELSEN - 1985 *****
5 !***** *****
7 CALL INIT:CLS "CBB":CALL COLOR("0RB")
40 BR=0:N=10
50 DIM A(14)
100 CALL DBLPRT("JEU DE KIM",10,5)
110 CALL COLOR("0C"):PRINT "TU VAS VOIR DES NOMBRES S
    'ECRIRE SUR"
115 PRINT "L'ECRAN .REGARDE LES BIEN"
120 PRINT:PRINT:CALL COLOR("0M"):PRINT "ENSUITE LES NO
    MBRES VONT CHANGER DE"
125 PRINT "PLACE ET UN D'ENTRE EUX VA DISPARAIRE. A T
    OI DE LE RECONNAITRE."
130 PRINT:CALL COLOR("0M"):PRINT "ES-TU PRET ?";
135 I$=KEY$
200 !PREMIERE SEQUENCE
205 CLS:CALL COLOR("0Y")
210 FOR I=1 TO N
220 A(I)=INTRND(9)
230 X=INTRND(15)+1
240 CALL DBLPRT(STR$(A(I)),X-18*(I>6),I*3+15*(I>5))
270 CALL PLAY("SILA",1,0)
280 NEXT
290 LOCATE (21,5):CALL COLOR("0b"):PRINT "QUAND TU AUR
    AS BIEN LU CES NOMBRES,"
295 PRINT "APPUIE SUR UNE TOUCHE."::I$=KEY$
300 !DISPARITION D'UN NOMBRE.
310 CLS
320 F=INTRND(N-1)
330 FOR I=1 TO N
340 X=INTRND(14)+2
350 XX=X-18*(I>4):YY=I*3+15*(I>5)
380 CALL COLOR("0C")
390 IF I=F THEN CALL DBLPRT("?",XX,YY):GOTO 420
400 CALL DBLPRT(STR$(A(I)),XX,YY)
410 CALL PLAY("MIRE",1,0)
420 NEXT
500 !REPONSE
510 LOCATE (21,1):PRINT "QUEL NOMBRE A DISPARU ? ";
520 ACCEPT BEEP VALIDATE(DIGIT)SIZE(3)NULL(0),REP
525 IF REP=A(F)THEN 600
530 CALL COLOR("0R"):PRINT:PRINT:PRINT "NON, C'ETAIT";
    :CALL COLOR("0G")
540 PRINT A(F)
550 GOTO 630
600 !GAGNE
610 CLS:CALL COLOR("0R"):CALL DBLPRT("GAGNE!",15,15)

```

---

```

620 BR=BR+1
630 PAUSE 5
640 N=N-2
650 IF N=2 THEN 800
660 GOTO 200
800 CLS
810 CALL DBLPRT("C'EST TERMINE",5,10)
820 PRINT:PRINT:PRINT "TU AS TROUVE";BR;"NOMBRES CACHE
S"
830 END
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10030 SUBEND
10040 SUB PLAY(NO$,L,D)
10050 FOR I=1 TO LEN(NO$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,1,2),1)
10070 P=VAL(SEG$("199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,X,Y)
10110 CALL COLOR("OLH")
10120 K$="" :FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("0"):SUBEND

```

2 !\*\*\*\*\* KIM NOMBRES - MOYEN \*\*\*\*\*

3 !\*\*\*\*\* VERSION EXL 100 \*\*\*\*\*

4 !\*\*\*\*\* D.NIELSEN - 1985 \*\*\*\*\*

5 !\*\*\*\*\*

```

7 CALL INIT:CLS "CBB":CALL COLOR("ORB")
40 BR=0:N=14
50 DIM A(14)
100 CALL DBLPRT("JEU DE KIM",10,5)
110 CALL COLOR("OC"):PRINT "TU VAS VOIR DES NOMBRES S
'Ecrire SUR"
115 PRINT "L'ECRAN .REGARDE LES BIEN"
120 PRINT:PRINT:CALL COLOR("OM"):PRINT "ENSUITE LES NO
MBRES VONT CHANGER DE"
125 PRINT "PLACE ET UN D'ENTRE EUX VA DISPARAITRE. A T
OI DE LE RECONNAITRE."

```

```
130 PRINT:CALL COLOR("0M"):PRINT "ES-TU PRET ?";  
135 I$=KEY$  
200 !PREMIERE SEQUENCE  
205 CLS:CALL COLOR("0Y")  
210 FOR I=1 TO N  
220 A(I)=INTRND(20)  
230 X=INTRND(15)+1  
240 CALL DBLPRT(STR$(A(I)),X-18*(I>7),I*2+14*(I>7))  
270 CALL PLAY("SILA",1,0)  
280 NEXT  
290 LOCATE (21,5):CALL COLOR("0b"):PRINT "QUAND TU AUR  
AS BIEN LU CES NOMBRES,"  
295 PRINT "APPUIE SUR UNE TOUCHE.";:I$=KEY$  
300 !DISPARITION D'UN NOMBRE.  
310 CLS  
320 F=INTRND(N-1)  
330 FOR I=1 TO N  
340 X=INTRND(14)+2  
350 XX=X-18*(I>7):YY=I*2+14*(I>7)  
380 CALL COLOR("0C")  
390 IF I=F THEN CALL DBLPRT("?",XX,YY):GOTO 420  
400 CALL DBLPRT(STR$(A(I)),XX,YY)  
410 CALL PLAY("MIRE",1,0)  
420 NEXT  
500 !REPONSE  
510 LOCATE (21,1):PRINT "QUEL NOMBRE A DISPARU ? ";  
520 ACCEPT BEEP VALIDATE(DIGIT)SIZE(3)NULL(0),REP  
525 IF REP=A(F)THEN 600  
530 CALL COLOR("0R"):PRINT:PRINT:PRINT "NON, C'ETAIT";  
:CALL COLOR("0G")  
540 PRINT A(F)  
550 GOTO 630  
600 !GAGNE  
610 CLS:CALL COLOR("0R"):CALL DBLPRT("GAGNE!",15,15)  
620 BR=BR+1  
630 PAUSE 5  
640 N=N-2  
650 IF N=2 THEN 800  
660 GOTO 200  
800 CLS  
810 CALL DBLPRT("C'EST TERMINE",5,10)  
820 PRINT:PRINT:PRINT "TU AS TROUVE";BR;"NOMBRES CACHE  
S"  
830 END  
10000 SUB INIT  
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(  
50850)  
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)
```

```

10030 SUBEND
10040 SUB PLAY(NO$,L,D)
10050 FOR I=1 TO LEN(NO$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)
10070 P=VAL(SEG$("199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,X,Y)
10110 CALL COLOR("0LH")
10120 K$="":FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("0"):SUBEND

```

```

2 !***** KIM NOMBRES - DIFFICILE ***
3 !***** VERSION EXL 100 *****
4 !***** D.NIELSEN - 1985 *****
5 !***** *****
7 CALL INIT:CLS "CBB":CALL COLOR("ORB")
40 BR=0:N=20
50 DIM A(20)
100 CALL DBLPRT("JEU DE KIM",10,5)
110 CALL COLOR("OC"):PRINT "TU VAS VOIR DES NOMBRES S
 'Ecrire sur"
115 PRINT "L'ECRAN .REGARDE LES BIEN"
120 PRINT:PRINT:CALL COLOR("OM"):PRINT "ENSUITE LES NO
 MBRES VONT CHANGER DE"
125 PRINT "PLACE ET UN D'ENTRE EUX VA DISPARAITRE. A T
 OI DE LE RECONNAITRE."
130 PRINT:CALL COLOR("OM"):PRINT "ES-TU PRET ?";
135 I$=KEY$
200 !PREMIERE SEQUENCE
205 CLS:CALL COLOR("0Y")
210 FOR I=1 TO N
220 A(I)=INTRND(50)
230 X=INTRND(15)+1
240 CALL DBLPRT(STR$(A(I)),X-18*(I>10),I*2+20*(I>10))
270 CALL PLAY("SILA",1,0)
280 NEXT
290 LOCATE (22,5):CALL COLOR("0b")
295 PRINT "APPUIE SUR UNE TOUCHE QUAND TU ES PRET";:I$=
=KEY$
300 !DISPARITION D'UN NOMBRE.
310 CLS

```

```

320 F=INTRND(N-1)
330 FOR I=1 TO N
340 X=INTRND(14)+2
350 XX=X-18*(I>10):YY=I*2+20*(I>10)
380 CALL COLOR("0C")
390 IF I=F THEN CALL DBLPRT("?",XX,YY):GOTO 420
400 CALL DBLPRT(STR$(A(I)),XX,YY)
410 CALL PLAY("MIRE",1,0)
420 NEXT
500 !REPONSE
510 LOCATE (22,1):PRINT "QUEL NOMBRE A DISPARU ? ";
520 ACCEPT BEEP VALIDATE(DIGIT)SIZE(3)NULL(0),REP
525 IF REP=A(F)THEN 600
530 CALL COLOR("0R"):PRINT:PRINT:PRINT "NON, C'ETAIT";
:CALL COLOR("0G")
540 PRINT A(F)
550 GOTO 630
600 !GAGNE
610 CLS:CALL COLOR("0R"):CALL DBLPRT("GAGNE!",15,15)
620 BR=BR+1
630 PAUSE 5
640 N=N-2
650 IF N=2 THEN 800
660 GOTO 200
800 CLS
810 CALL DBLPRT("C'EST TERMINE",5,10)
820 PRINT:PRINT:PRINT "TU AS TROUVE";BR;"NOMBRES CACHE
S"
830 END
10000 SUB INIT
10010 CALL POKE(50850,162,5,45,162,136,45,10):CALL EXEC(
50850)
10020 CALL POKE(50688,165,8,6,10):CALL POKE(49156,198,0)

10030 SUBEND
10040 SUB PLAY(NO$,L,D)
10050 FOR I=1 TO LEN(NO$)STEP 2
10060 K=POS("DO.RE.MI.FA.SO.LA.SI.",SEG$(NO$,I,2),1)
10070 P=VAL(SEG$("199176158149133118105",K,3))
10080 CALL POKE(258,P,D+128):PAUSE L/50
10090 NEXT:CALL POKE(259,0):SUBEND
10100 SUB DBLPRT(Z$,X,Y)
10110 CALL COLOR("0LH")
10120 K$="":FOR I=1 TO LEN(Z$)
10130 K$=K$&RPT$(SEG$(Z$,I,1),2):NEXT
10140 LOCATE (Y,X):PRINT K$
10150 LOCATE (Y+1,X):PRINT K$
10160 CALL COLOR("0"):SUBEND

```

# Conclusion

Ces programmes ont tous été utilisés par des enfants d'âges et de niveaux scolaires différents.

Ce travail a permis un certain nombre de constatations :

- Les enfants ne considèrent pas cela comme une activité scolaire traditionnelle et prennent plaisir à travailler sur l'ordinateur.
- Les programmes adaptés à l'école stimulent l'enfant, favorisent une pédagogie de la réussite : l'ordinateur, c'est la possibilité d'essayer, de faire des erreurs sans être pénalisé, de les corriger. La méthode expérimentale y trouve toute sa valeur.

L'ordinateur fournit aussi un compte rendu du travail des élèves, des erreurs commises, des difficultés rencontrées, de la durée d'attention propre à chaque élève ; ce qui permet ensuite d'analyser ces résultats et de personnaliser enfin les leçons en classe.

Il apporte une nouvelle vision du travail des enfants. Ainsi, sur le programme de lecture rapide (avec des mots nouveaux à trouver), des enfants ayant de grosses difficultés au niveau du français ont eu d'aussi bons résultats que les meilleurs.

L'ordinateur doit ainsi aider, dans une école, à atténuer les différences d'origines culturelles ou sociales entre les enfants, principalement au niveau du soutien, en déchargeant l'enseignant de tâches répétitives dans cette forme de travail individualisé.

A la maison, ces programmes vous aideront à faire acquérir ou réviser à vos enfants des connaissances de base indispensables.

Il est important que chacun utilise les possibilités offertes dans les sections « modifications » et « adaptations » (et peut-être d'autres possibilités non définies ici !) car l'enseignement ne doit pas être banalisé par des « machines à apprendre ». Au contraire, ces programmes peuvent être un point de départ pour vous aider à résoudre des difficultés bien précises que vous pourrez rencontrer.

Les programmes purement didactiques auraient pu être assimilés à des jeux en ajoutant, comme on le voit beaucoup, une « sauce » autour du plat de résistance, selon le principe bien connu du « pendu » : une fusée qui perd du carburant à chaque mauvaise réponse, un lapin qui mange ou perd des carottes, etc. Mais alors, les listings auraient été encore plus longs... A vous de programmer maintenant, si vous le souhaitez, et si la place en mémoire de l'unité centrale seule vous le permet !

Un autre rôle de l'ordinateur à l'école, et non des moindres, réside dans l'apprentissage de la programmation par les enfants : obliger à établir un projet, le concevoir et l'écrire selon la logique informatique, le tester, rectifier les erreurs,... Toutes ces actions sont éminemment pédagogiques. Mais cela nécessite beaucoup de matériel et la formation de tous les enseignants.

Si l'entrée de l'ordinateur à l'école primaire est un des objectifs prioritaires du ministère de l'Education nationale pour les années 1985 à 1989, l'important est d'abord d'être convaincu de son utilité, puis de s'en servir, même si l'on ne connaît pas bien encore les bases de programmation, de faire travailler les enfants sur les programmes présentés ici, et, peu à peu, de s'investir dans le programme en le modifiant, en l'adaptant, puis enfin en créant ses propres programmes.

Les enfants connaîtront peut-être alors « cet éveil au phénomène socio-culturel et technologique que constitue l'informatique » encouragé par le ministère de l'Education nationale et le plan « Informatique pour tous ».

# Conseils de lecture

Pour programmer votre EXL 100 en jouant et utiliser le matériel Thomson pour apprendre ou enseigner, à l'école comme à la maison, P.S.I. vous propose une palette d'ouvrages utiles.

## Pour mieux programmer votre EXL 100

- **102 programmes pour EXL 100** – Jacques Deconchat (Editions du P.S.I.)  
Les bases de la programmation en cinq niveaux de difficulté pour apprendre l'Exelbasic en jouant.

## Pour apprendre ou enseigner avec Thomson

- **MO5 et T07/70 pour tout petits** – Daniel Nielsen (Editions du P.S.I.)  
Destiné aux enfants de 4 à 7 ans, cet ouvrage propose des programmes en BASIC de lecture et de calcul, aisément adaptables par les parents ou les enseignants.
- **Le Basic et l'école, tomes 1 et 2** – Jacques Gouet (Editions du P.S.I.)  
Destiné aux enseignants, aux parents et aux élèves, cet ouvrage indique comment réaliser des programmes d'E.A.O. en BASIC Microsoft, dans les classes de primaire.
- **MO5 et T07/70 à l'école** – Daniel Nielsen (Editions du P.S.I.)  
20 programmes BASIC proposent aux enfants de 6 à 12 ans d'aborder ou de réviser les matières du primaire : le calcul, le français et l'éveil.

- **MO5 et T07/70 pour réussir en CM1** – Daniel Nielsen (Editions du P.S.I.)  
Destiné aux élèves du Cours moyen 1<sup>re</sup> année, cet ouvrage propose d'aborder ou de réviser les matières de cette classe à l'aide de programmes BASIC facilement modifiables et adaptables.  
L'histoire, la géographie et les sciences et technologies remplacent « l'éveil » et proposent à l'enfant une initiation à l'informatique.
- **Destination collège** – Daniel Nielsen (Editions du P.S.I.)  
17 programmes en BASIC Thomson aident les élèves du CM2 à aborder en douceur le passage en Sixième.
- **Maths et Thomson en Sixième** – Jacques Deconchat (Editions du P.S.I.)  
Plus de 80 petits programmes en BASIC permettent à l'élève de Sixième de revoir ou de tester ses connaissances en mathématiques.

*A paraître :*

- **MO5 et T07/70 pour réussir en CE2** – Daniel Nielsen (Editions du P.S.I.)  
Aucune matière du Cours élémentaire 2<sup>e</sup> année n'est oubliée dans les programmes BASIC proposés aux élèves, à leurs parents et à leurs enseignants.
- **Logo à l'école** – Daniel Nielsen et Gérard Bossuet (Editions du P.S.I.)  
8 procédures pour se servir de la célèbre tortue et 10 procédures pour apprendre grâce à elle : un ouvrage clair et complet sur l'utilisation de Logo sur MO5 et T07/70 en primaire.
- **Maths et Thomson en Cinquième** – Jacques Deconchat (Editions du P.S.I.)  
Apprendre ou réviser de nombreux problèmes mathématiques de la classe de Cinquième à l'aide de l'ordinateur.

## Votre avis nous intéresse

- Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent livre.
- Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous adresserons votre demande à l'auteur qui ne manquera pas de vous répondre directement.

**- Ce livre vous donne-t-il toute satisfaction?**

---

---

---

---

**- Y a-t-il un aspect du problème que vous auriez aimé voir abordé?**

---

---

---

---

### Comment avez-vous eu connaissance de ce livre?

- |   |                                     |
|---|-------------------------------------|
| <input type="checkbox"/> publicité      | <input type="checkbox"/> cadeau     |
| <input type="checkbox"/> catalogue      | <input type="checkbox"/> librairie  |
| <input type="checkbox"/> boutique micro | <input type="checkbox"/> exposition |
| <input type="checkbox"/> autres         |                                     |

### Avez-vous déjà acquis des livres PSI?

lesquels? \_\_\_\_\_

qu'en pensez-vous? \_\_\_\_\_  
\_\_\_\_\_

Nom \_\_\_\_\_ Prénom \_\_\_\_\_ Age \_\_\_\_\_  
Adresse \_\_\_\_\_  
Profession \_\_\_\_\_  
Centre d'intérêt \_\_\_\_\_

## **CATALOGUE GRATUIT**

Vous pouvez obtenir un catalogue complet des ouvrages PSI, sur simple demande, ou en retournant cette page remplie à votre libraire, à votre boutique micro ou aux

**Editions du PSI  
BP 86  
77402 Lagny-sur-Marne Cedex**



# EXL 100 A L'ÉCOLE

**E**XL 100 à l'école" propose aux enseignants, aux parents et aux élèves, 21 programmes en BASIC pour aborder ou réviser les matières principales des classes de primaire : le calcul, le français et l'éveil.

**D**aniel Nielsen, directeur d'école, met ici en valeur le rôle de l'ordinateur en tant que soutien efficace du travail de l'enfant, à l'école comme à la maison. Les programmes suivent tous le schéma suivant : thème, niveau, intérêt pédagogique, déroulement du programme, commentaires, particularités techniques, modifications et adaptations, organigramme et listing.

**T**ous les programmes de cet ouvrage ont été testés par des enfants de 6 à 12 ans. Pour tirer profit au maximum de ce livre, il est préférable de connaître déjà un BASIC, et d'avoir assimilé les bases de la programmation informatique.



**EDITIONS DU P.S.I.**  
**BP 86 - 77402 LAGNY S/MARNE CEDEX - FRANCE**

**ISBN 2-86595-293-2**

**120 FF**