

# **AMSTRAD A L'ÉCOLE**



**DANIEL NIELSEN  
ET AUGUSTIN GARCIA AMPUDIA**



**AMSTRAD**  
**A L'ÉCOLE**

## Connaissez-vous toute la collection Amstrad chez P.S.I. ?

---

### Pour les Amstrad CPC 464, 664 et 6128 :

---

#### Initiation

- La découverte de l'Amstrad — Daniel-Jean David
- Exercices en Basic pour Amstrad — Maurice Charbit

#### Programmation BASIC

- 102 programmes pour Amstrad — Jacques Deconchat
- Super jeux Amstrad — Jean-François Sehan
- Amstrad en famille — Jean-François Sehan
- Super générateur de caractères sur Amstrad — Jean-François Sehan
- Photographie sur Amstrad et Apple II — Pierrick Moigneau et Xavier de la Tullaye
- Amstrad en musique — Daniel Lemahieu

#### Maîtrise du BASIC

- Basic Amstrad, 1. Méthodes pratiques — Jacques Boisgontier et Bruno Césard
- Basic Amstrad, 2. Programmes et fichiers — Jacques Boisgontier
- Basic Plus, 80 routines sur Amstrad — Michel Martin
- Périphériques et fichiers sur Amstrad — Daniel-Jean David

#### Assembleur et Pascal

- Assembleur de l'Amstrad — Marcel Henrot
- Graphisme en assembleur sur Amstrad CPC — Francis Piérot
- Création et animations graphiques sur Amstrad CPC — Gilles Fouchard et Jean-Yves Corre
- Trois étapes vers l'intelligence artificielle sur Amstrad CPC — René Descamps
- Turbo Pascal sur Amstrad — Pierre Brandeis et Frédéric Blanc

#### Système

- Clefs pour Amstrad, 1. Système de base — Daniel Martin
- CPM Plus sur Amstrad 6128 et 8256 — Yvon Dargery
- Clefs pour Amstrad, 2. Système disque — Daniel Martin et Philippe Jadoul

#### A paraître :

- Intelligence artificielle : langage et formes sur Amstrad — Thierry Lévy-Abégnolli et Olivier Magnan
- Clefs pour Amstrad 8256 — Eric Baumarti

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>er</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**

# **AMSTRAD A L'ÉCOLE**

**DANIEL NIELSEN  
ET AUGUSTIN GARCIA AMPUDIA**



**ÉDITIONS DU P.S.I.  
1986**

**Daniel Nielsen**, directeur d'école, travaille depuis quatre 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.

**Augustin Garcia Ampudia**, 21 ans, étudie l'informatique à l'université Pierre et Marie Curie (Paris VI).

Par ailleurs, il collabore à plusieurs revues spécialisées "micro-informatique". Spécialiste des langages Basic et Pascal, il sévit sur les micro-ordinateurs familiaux, ne dédaignant pas pour autant les standards 16 et 32 bits.

En septembre 1981, il avait acheté un ZX 81...

Amstrad Amdos et Amstrad CPC 464, 664 et 6128 sont des marques déposées de Amstrad Consumer Electronics.

AMX Mouse est une marque déposée de Advanced Memory Systems Ctd.

# Sommaire

<b>Introduction</b> .....	7
Conseils techniques généraux .....	8
Manettes de jeux et clavier .....	9
Compatibilité entre les différents CPC .....	10
Index de classement par thèmes .....	11
Index de classement par niveaux .....	12
Kim musical .....	13
Autodictée .....	21
Soutien .....	29
Safari .....	39
Lecture rapide .....	47
Lecture débutant .....	55
Pluriel des noms .....	61
Conjugaison .....	69
Addition .....	77
Soustraction .....	85
Multiplication .....	95
Division .....	103
Ravitaillement en vol .....	113
Le train capricieux .....	121
Villes de France .....	129
Construction de phrases .....	137
J'écris tout seul .....	145
Reconstitution de textes .....	153
Cocktail magique .....	161
Masse et volume .....	171
Kim nombres .....	181
<b>Devoirs de vacances</b> .....	191
<b>Conclusion</b> .....	223
<b>Page disquette</b> .....	225
<b>Conseils de lecture</b> .....	227



# 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'assièront 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 programmes 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.

**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 recopiables, 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 tracés 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 votre AMSTRAD CPC), vous ne perdrez pas trop de travail et n'aurez qu'un petit nombre de lignes à taper de nouveau.

## Conseils techniques généraux

La fonction RND (tirage au sort d'un nombre aléatoire entre 0 et 1) est présente dans chaque programme. C'est elle qui permet, à partir des mêmes données, d'obtenir des exercices ou des jeux différents chaque fois.

Sur un AMSTRAD CPC, 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 cha-

que appel sera la même. C'est notamment le cas si on n'initialise pas le générateur au début du programme. A cet effet, le BASIC de l'Amstrad propose une instruction RANDOMIZE. Si cette instruction est utilisée, le générateur est initialisé avec un nombre provenant de l'horloge interne de AMSTRAD CPC (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

La manette de jeux des CPC est aux normes Atari, ce qui signifie, que n'importe quelle manette standard du commerce s'y adapte. Le crayon optique vendu en option n'étant pas une merveille de précision, nous avons décidé de donner des versions de certains de nos programmes pour manette de jeux. Toutefois l'utilisation de la manette entraîne d'autres petits problèmes : l'enfant doit savoir où se trouve le haut, le bas, la droite et la gauche (ceci est un faux problème car nos petites têtes blondes trouvent rapidement l'astuce) ; le second inconvénient est la rapidité de réponse de la manette. Pour palier ce défaut nous introduisons une boucle d'attente systématique.

Récapitulons : les réponses au clavier se feront à l'aide des touches "1" et "0" pour les directions et de la touche "barre espace" pour la validation ; avec manette, le bouton de tir fera office de validation (les programmes comprennent jusqu'à deux boutons de tir), les directions se feront tout naturellement avec la tige de la manette.

## Problèmes de compatibilité entre les différents CPC

Afin de ne connaître aucun problème, les programmes ont été développés avec les instructions du CPC 464. Les utilisateurs d'écran monochromes ne sont pas oubliés, le choix des couleurs ayant été décidé sur un monochrome. Maintenant que tout le monde est satisfait, nous vous signalons que les lignes de REM sont parfaitement inutiles, les programmes ne les utilisant jamais comme lignes de renvoi. Inutiles ne signifie pas superflues, en effet, combien de débutants, après avoir tapé de longues heures et fait des RENUM, nous téléphonent au numéro vert pour nous dire que leur programme refuse de fonctionner. Notre spécialiste pourra répondre d'autant mieux qu'il leur indiquera la ligne exacte qui provoque l'erreur. A vos claviers et bon courage...

## Index de classement par thèmes

Thèmes  Programmes	Jeu	Didact	Etude de situation	Français	Maths	Eveil	Attention	Memoire	Logique	Rapidité	Réponse	
											Clavier	Manette de jeux
Kim musical	X					X	X	X			X	
Autodictée		X		X			X	X			X	
Soutien	X	X		X	X	X		X			X	
Safari	X				X		X			X	X	
Lecture rapide			X	X			X	X			X	X
Lecture débutant			X	X			X	X			X	X
Pluriel des noms		X		X			X	X			X	
Conjugaison		X		X			X	X			X	
Addition		X			X		X				X	
Soustraction		X			X		X				X	
Multiplication		X			X		X				X	
Division		X			X		X				X	
Ravitaillement en vol	X				X		X		X		X	
Le train capricieux	X			X	X	X	X			X	X	
Villes de France			X			X	X	X			X	
Construction de phrases			X	X			X		X		X	
J'écris tout seul			X	X			X		X		X	X
Reconstitution de textes			X	X			X		X		X	X
Cocktail magique			X		X		X	X	X		X	
Masse et volume			X		X		X	X	X		X	X
Kim nombres	X				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 l'AMSTRAD CPC 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 l'AMSTRAD CPC.

Pour continuer il faut appuyer sur une touche.

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. L'AMSTRAD CPC 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 l'AMSTRAD CPC.

## Commentaires

- |                  |   |
|------------------|---|
| <b>100 à 200</b> | Module de présentation du programme, avec tout d'abord une mise en page graphique : ouverture quatre fois de la même fenêtre pour donner une impression de relief.  |
| <b>240 à 280</b> | Initialisation et lecture des notes de la gamme. Les noms et périodes des notes sont sous la forme de DATAs (ligne 270) ; une fois lues, elles sont rangées dans deux tableaux note\$(i) et période(i). Ils n'ont pas besoin d'être déclarés car les indices sont définis de 1 à 7. |
| <b>320</b>       | Appel de la procédure d'affichage de la gamme.  |
| <b>350 à 440</b> | Affichage et jeu de la séquence musicale. Chaque note est tirée au hasard (ligne 380) et affichée (ligne 390), son nom apparaît en clair les trois premiers jeux (ligne 400). Et enfin une fois jouée en ligne 410, elle est mémorisée à la ligne 420.                              |

- 480 à 560** Affichage et jeu de la séquence qui comporte une note de moins. Celle-ci est tirée au sort (ligne 480), la partition est jouée et affichée en ne prenant pas en compte la note choisie (test ligne 560). Les positions des notes dépendent de leur indice dans la séquence ; à la place de la note enlevée on trouve un espace.
- 600 à 680** Module de réponse. On demande la réponse du joueur, on la vérifie, en cas d'erreur on affiche un message et on donne une nouvelle chance (ligne 680).
- 720 à 760** Non, vous n'avez pas trouvé, alors l'Amstrad vous montre ce qu'il fallait répondre.
- 800 à 810** Cas de réponse correcte.
- 820 à 890** Test de fin de programme. Si le troisième jeu n'a pas encore été effectué, on recommence, sinon on affiche un message pour avertir que les séquences suivantes se feront sans le nom des notes. Si jeu=5 alors fin du programme (ligne 840).
- 930 à 1020** Sous-programme de dessin de la gamme, faisant lui-même appel au sous-programme suivant. Les notes sont jouées et affichées sur la portée.
- 1060 à 1100** Sous-programme de dessin de la portée.

## Particularités techniques

On remarquera l'utilisation des fenêtres pour donner une impression de relief grâce aux ombres portées (lignes 110 et 120, ainsi que 140 et 150).

L'Amstrad CPC possède des instructions pour gérer les sons, en particulier SOUND ; cette instruction nécessite plusieurs paramètres dont la période qui nous permet de repérer nos notes.

DO	478
RE	426
MI	379
FA	358
SOL	319
LA	284
SI	253

Les instructions LOCATE présentes lors des boucles permettent l'affichage des noms des notes ainsi que du symbole sur la portée (exemple : boucle 960-1000).

Le hasard, lui, est géré par deux instructions RND et RANDOMIZE. Le paramètre de cette dernière instruction (ligne 240) est TIME / 1000. Pourquoi cette valeur ? Il semblerait que le fait d'aller chercher de profondes décimales influe sur le hasard. En effet la séquence aléatoire a plus de chance de se répéter avec un RANDOMIZE simple. Les voies du hasard sont impénétrables.

## Modifications

Il est possible d'augmenter le nombre de notes de la séquence musicale (lignes 170, 370, 480, 510 et 610). Mais si vous voulez placer plus de neuf notes sur l'écran, il faut aussi diminuer l'écartement entre l'affichage des notes (lignes 390 et 530).

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 :

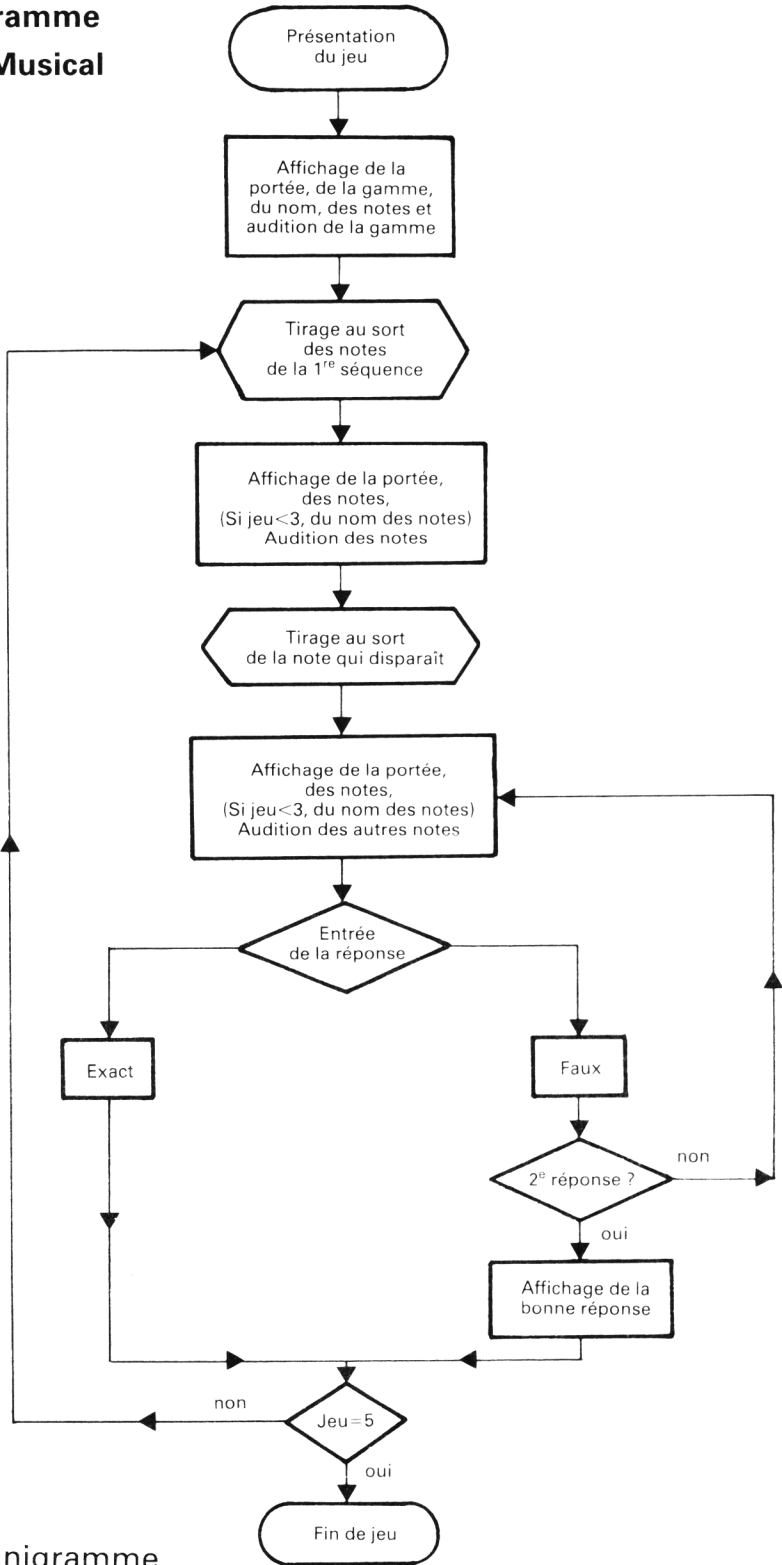
- avant chaque jeu (lignes 830 et 840 : GOSUB 930 avant le GOTO) ;
- en cas de première réponse erronée, avant de rejouer la séquence  
680 C=1 : GOSUB 930  
685 GOTO 490

Par contre, avec des enfants ayant de bonnes connaissances musicales, on peut supprimer les lignes 960 à 1000, les notes apparaîtront ainsi sans leur nom dès le premier jeu.

## Adaptations

Si vous préférez que l'AMSTRAD CPC vous joue de vraies mélodies, au lieu d'une suite de notes déterminée au hasard et dont l'harmonie peut paraître parfois douteuse, vous pouvez remplacer les lignes 370 à 430 la séquence aléatoire par une lecture de notes inscrites en DATA contenant les indices des notes de votre mélodie.

# Programme Kim Musical



Organigramme

```

10 REM -----
20 REM ----- KIM MUSICAL -----
30 REM ---- D.Nielsen nov 83 ----
40 REM --- Augustin jan 86 ---
50 REM -----
60 ON BREAK GOSUB 880
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 13:CLS:PEN 0:PEN#3,0
110 WINDOW#3,15,27,5,7:PAPER #3,0:CLS#3
120 WINDOW#3,14,26,4,6:PAPER#3,10:CLS#3
130 PEN#3,0:LOCATE#3,2,2:PRINT#3,"KIM MUSICAL"
140 WINDOW#3,5,38,11,22:PAPER#3,0:CLS#3
150 WINDOW#3,4,37,10,21:PAPER#3,10:CLS#3
160 LOCATE#3,2,2:PRINT#3,"Tu vas d'abord enten
dre la gamme"
170 PRINT#3,CHR$(10);" Ensuite, tu entendras u
n morceau de 6 notes"
180 PRINT#3,CHR$(10);" Puis le meme morceau av
ec une note en moins"
190 PRINT#3,CHR$(10);" A toi de retrouver cett
e note et de la taper"
200 a$=INKEY$:IF a$="" THEN 200
210 REM -----
220 REM -> initialisation
230 REM -----
240 CLS:RANDOMIZE TIME/1000 +
250 WINDOW#3,5,37,4,23:PAPER#3,0:CLS#3
260 WINDOW#3,4,36,3,22:PAPER#3,10:CLS#3
270 jeu=0:DATA "DO",478,"RE",426,"MI",379,"FA"
,358,"SOL",319,"LA",284,"SI",253
280 FOR i=1 TO 7:READ note$(i), periode(i):NEX
T i
290 REM -----
300 REM -> Module 6 notes
310 REM -----
320 GOSUB 930 +
330 CLS#3:LOCATE#3,2,2:PRINT#3,"Maintenant a t
oi de jouer"
340 PRINT#3," Appuie sur une touche"
350 a$=INKEY$:IF a$="" THEN 350
360 CLS#3:c=0:GOSUB 1060
370 FOR i=1 TO 6
380 y=INT(RND*7)+1
390 LOCATE#3,1+4*i,17-y:PRINT#3,CHR$(236)
400 IF jeu<3 THEN LOCATE#3,1+4*i,19:PRINT#3,no
te$(y)

```

```

410 SOUND 1,periode(y),40,15
420 d(i)=y:nt$(i)=note$(y)
430 NEXT i
440 FOR i=1 TO 6000:NEXT i
450 REM -----
460 REM -> Module 5 notes
470 REM -----
480 f=INT(RND*5)+1
490 CLS#3:PRINT#3," Voici une nouvelle musique
avec 5 notes"
500 GOSUB 1060
510 FOR i=1 TO 6
520 IF i=f THEN 560
530 LOCATE#3,1+4*i,17-d(i):PRINT#3,CHR$(236)
540 IF jeu<3 THEN LOCATE#3,1+4*i,19:PRINT#3,nt +
$(i)
550 SOUND 1,periode(d(i)),40,15
560 NEXT i
570 REM -----
580 REM -> Module reponse
590 REM -----
600 LOCATE#3,2,3:INPUT#3,"Quelle est ta repons
e ";rep$
610 FOR i=1 TO 7
620 IF UPPER$(rep$)=note$(i) THEN SOUND 1,peri
ode(i),40
630 NEXT i
640 IF UPPER$(rep$)=nt$(f) THEN 800
650 IF c=1 THEN 720
660 LOCATE #3,1,3:PRINT#3," Non, ce n'est pas
cela, ecoute une deuxieme fois"
670 FOR i=1 TO 3000:NEXT i
680 c=1:GOTO 490
690 REM -----
700 REM -> Affichage bonne reponse
710 REM -----
720 LOCATE#3,1,3:PRINT#3," Non, c'est encore f
aux "
730 PRINT#3," La bonne reponse etait ";nt$(f)
740 SOUND 1,periode(f),40,15
750 ENT 2,=71,20,=89,20,=71,20,=0,0:FOR i=1 TO
3:SOUND 1,0,100,15,,2:NEXT i
760 GOTO 810
770 REM -----
780 REM -> Gagne
790 REM -----
800 LOCATE#3,2,5:PRINT#3,"C'est exact"
810 FOR i=1 TO 3000:NEXT i

```

```

820 jeu=jeu+1
830 IF jeu<3 THEN 360
840 IF jeu=5 THEN 880
850 CLS#3:LOCATE#3,1,2:PRINT#3," Maintenant tu
   vas jouer sans lire le nom des notes"
860 FOR i=1 TO 4000:NEXT i
870 GOTO 360
880 CLS:LOCATE 10,10:PRINT"Au revoir..."
890 END
900 REM -----
910 REM -> Module dessin de la gamme
920 REM -----
930 CLS#3:LOCATE#3,11,2
940 PRINT#3,"Voici la gamme"
950 GOSUB 1060
960 FOR i=1 TO 7
970 LOCATE#3,1+4*i,17-i:PRINT#3,CHR$(236)
980 LOCATE#3,1+4*i,19:PRINT#3,note$(i)
990 SOUND 1,periode(i),40,15
1000 NEXT i
1010 a$=INKEY$:IF a$="" THEN 1010
1020 RETURN
1030 REM -----
1040 REM -> Module dessin de la portee
1050 REM -----
1060 FOR i=1 TO 5
1070 PLOT 80,300-30*i,0
1080 DRAW 550,300-30*i,0
1090 NEXT i
1100 RETURN

```

# AUTODICTÉE

## 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. Lorsqu'il appuie sur la touche ENTER pour confirmer la fin de son texte, AMSTRAD CPC 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 à l'écran et les fautes éventuelles sont visualisées par un symbole d'ogre souriant.

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 l'AMSTRAD CPC.

## Commentaires

- |                  |   |
|------------------|---|
| <b>60</b>        | Cette instruction permet de transformer la séquence de Break (ESC, puis ESC) en un GOTO effacement de l'écran et fin. |
| <b>100 à 210</b> | Présentation du programme.  |
| <b>280 à 310</b> | Boucle de lecture des 5 phrases. L'instruction INPUT ne permet pas de lire les virgules.                              |
| <b>350 à 400</b> | Préparation des 5 phrases, en vue de l'appel du sous-programme qui travaillera les phrases (GOSUB 530).               |

<b>410 à 520</b>	Affichage du score, des commentaires, et de la fin de jeu. Le jeu peut être relancé avec les mêmes phrases pour un autre enfant.
<b>530 à 570</b>	Affichage de la phrase sur laquelle l'enfant va travailler et attente de la frappe d'une touche.
<b>580 à 640</b>	Entrée du texte de l'enfant.
<b>650 à 680</b>	Test de la réponse. Test de la longueur, puis analyse lettre à lettre (lignes 670 et 680) des deux phrases.
<b>710</b>	Affichage (aucune faute).
<b>720 à 730</b>	Affichage (nombre de fautes).
<b>780 à 790</b>	Affichage du texte exact si l'enfant en était à son troisième essai.
<b>830 à 920</b>	Sous-programme d'analyse de la longueur des phrases.
<b>930 à 960</b>	Sous-programme musical et graphique. Il affiche le petit ogre à la place des fautes et joue une musique.

## Particularités techniques

Il serait dommage de faire des fautes d'orthographe dans un programme de dictée ! C'est pourquoi la ligne 430 ainsi que 720-730, 780-790 et 860-870 tiennent compte de la valeur des variables T, K et X afin de ne pas afficher « 1 FAUTES » !

Dans le test réponse, la ligne 650 compare la longueur des phrases puis, si les deux phrases ont le même nombre de lettres, celles-ci sont analysées une à une à l'intérieur de la boucle des lignes 660 à 680.

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 ESC puis ESC) car on doit alors entrer de nouveau les phrases.

## Modifications

Comme dans « Soutien », on peut supprimer le commentaire du score (lignes 450 à 470).

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 #2, « Relis bien la phrase »**  
**746 LOCATE 2, 2,9 : PRINT #2, v\$**

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 600 et 720-790.

## 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 :

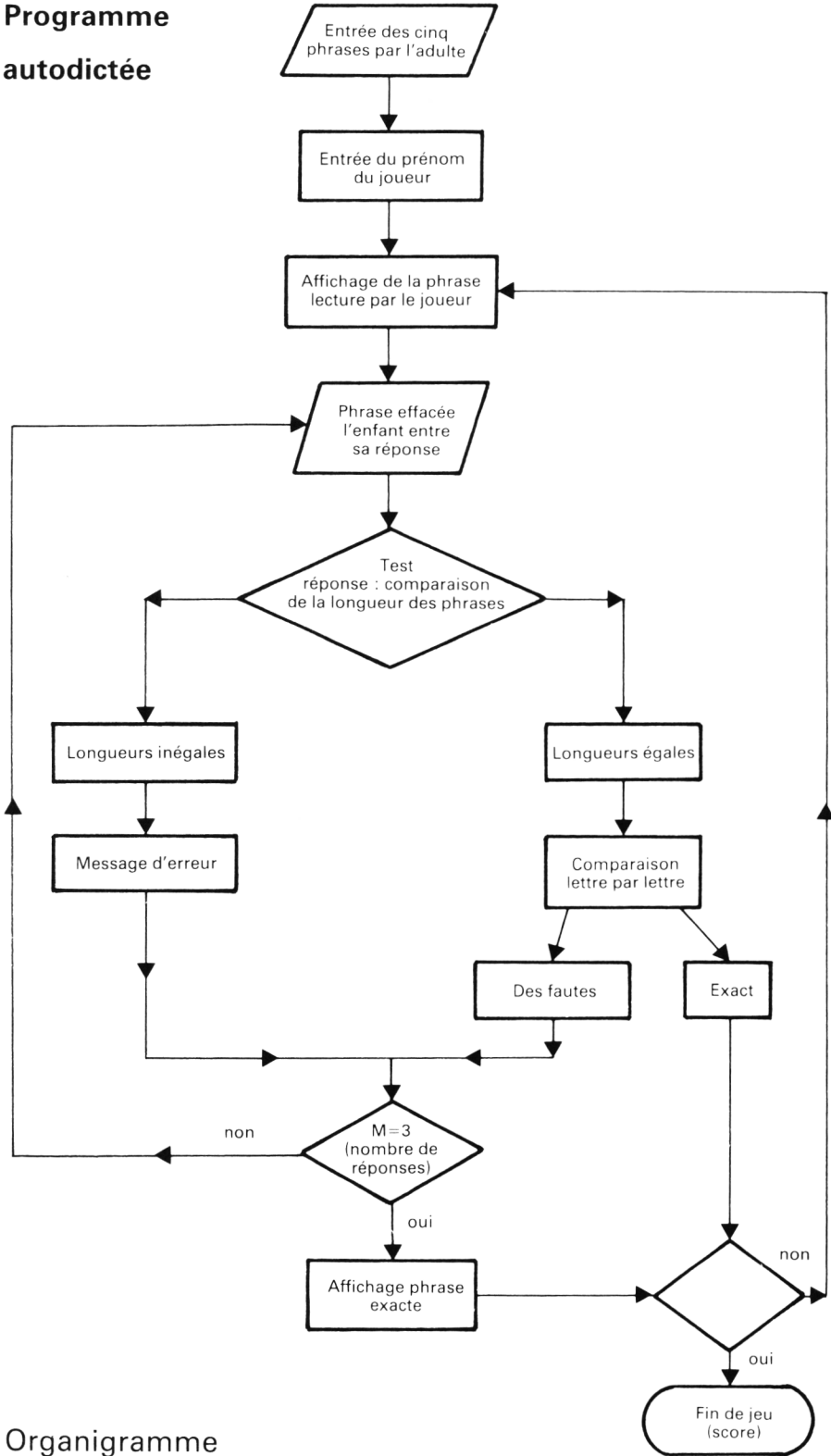
- phrases très courtes, étudiées en lecture, en C.P. ;
- phrases d'une dictée déjà faite, ou à faire ensuite, et qui comportent certaines difficultés particulières, en C.M.

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 PRINT #8, v\$, m\$**

# Programme autodictée



Organigramme

```

10 REM -----
20 REM ----- AUTODICTEE -----
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin fev 86 ---
50 REM -----
60 ON BREAK GOSUB 520
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 13:CLS:PEN 0
110 WINDOW#3,15,27,5,7:PAPER#3,0:CLS#3
120 WINDOW#3,14,26,4,6:PAPER#3,10:CLS#3
130 WINDOW#2,5,38,11,22:PAPER#2,0:CLS#2
140 WINDOW#2,4,37,10,21:PAPER#2,10:CLS#2
150 PEN#3,0:LOCATE#3,2,2:PRINT#3,"AUTODICTEE"
160 PEN#2,0:LOCATE#2,2,2:PRINT#2,"Vous allez e
ntrer votre texte"
170 LOCATE#2,12,4:PRINT#2,"Attention"
180 LOCATE#2,2,6:PRINT#2,"Deux lignes maximum
par phrase"
190 LOCATE#2,2,8:PRINT#2,"Il ne faut pas utili
ser"," de virgule"
200 LOCATE#2,2,11:PRINT#2,"Appuie sur une touc
he"
210 i#=INKEY#:IF i#="" THEN 210 ELSE CLS#2
220 b#(1)="Premiere":b#(2)="Deuxieme"
230 b#(3)="Troisieme":b#(4)="Quatrieme"
240 b#(5)="Cinquieme"
250 REM -----
260 REM -> lecture des phrases
270 REM -----
280 FOR i=1 TO 5
290 PRINT#2,b#(i);" phrase"
300 INPUT#2,a#(i)
310 NEXT i
320 REM -----
330 REM -> Module jeu
340 REM -----
350 CLS#2
360 INPUT#2,"Quel est ton prenom ";nom#:nom#=U
PPER$(nom#)
370 t=0:x=0
380 FOR j=1 TO 5
390 v#=a#(j):GOSUB 530
400 NEXT j
410 CLS#2
420 PRINT#2,"Ton score est de";x;"faute";

```

```

430 IF x>1 THEN PRINT#2,"s" ELSE PRINT#2
440 PRINT#2
450 IF x<5 THEN PRINT#2,"C'est tres bien"
460 IF x>=5 AND x<10 THEN PRINT#2,"C'est un bo
n resultat"
470 IF x>10 THEN PRINT#2,"Il faudra etre plus
attentif","la prochaine fois"
480 FOR i=1 TO 6000:NEXT i
490 PRINT#2:PRINT#2, "Un autre exercice ( o /
n ) ?"
500 i$=INKEY$:IF i$="" THEN 500
510 IF i$="0" OR i$="o" THEN 350
520 CLS:PRINT "Au revoir ";nom$:END
530 CLS#2
540 m=0:t=0
550 PRINT#2,"Tu vas lire cette phrase","lorsqu
e tu la connaistras bien","appuie sur une touch
e"
560 PRINT#2:PRINT#2,v$
570 i$=INKEY$:IF i$="" THEN 570
580 x=x+t
590 t=0
600 IF m=3 THEN 780
610 CLS#2
620 PRINT#2,"Tape le texte sans oublier","la p
onctuation"
630 PRINT#2,"Appuie sur ENTER","lorsque tu aur
as fini"
640 INPUT#2,m$
650 IF LEN(m$)<>LEN(v$) THEN GOSUB 830:GOTO 58
0
660 FOR i=1 TO LEN(v$)
670 IF MID$(m$,i,1)=MID$(v$,i,1) THEN PRINT#2,
MID$(v$,i,1); ELSE GOSUB 930
680 NEXT i
690 PRINT#2
700 m=m+1
710 IF t=0 THEN PRINT#2,"Tu n'as fait aucune f
aute ";nom$;"C'est tres bien":GOTO 800
720 IF t>1 AND m<3 THEN PRINT#2,"Tu as fait";t
;"fautes ";nom$
730 IF t=1 AND m<3 THEN PRINT#2,"Tu as fait";t
;"faute ";nom$
740 IF t>=1 AND m<3 THEN PRINT#2,nom$;", tu pe
ux recommencer"
750 LOCATE#2,2,11:PRINT#2,"Appuie sur une touc
he"
760 i$=INKEY$:IF i$="" THEN 760 ELSE CLS#2

```

```

770 IF t>0 AND m<3 THEN 580
780 IF t>1 AND m=3 THEN PRINT#2,"Voici le text
e exact":PRINT#2,v#:PRINT#2,"tu as fait";t;"fa
utes"
790 IF t=1 AND m=3 THEN PRINT#2,"Voici le text
e exact":PRINT#2,v#:PRINT#2,"tu as fait";t;"fa
ute"
800 x=x+t
810 LOCATE#2,1,11:PRINT#2,"Appuie sur une touc
he"
820 i#=INKEY#:IF i#="" THEN 820 ELSE RETURN
830 CLS#2
840 PRINT#2,"Tu as fait une erreur","Ton texte
n'a pas le meme nombre","de lettre que le pre
mier","Il faut recommencer !"
850 k=LEN(v#)-LEN(m#)
860 IF k>1 THEN PRINT#2,"Il manque";k;"lettres
","dans ton texte":GOTO 900
870 IF k=1 THEN PRINT#2,"Il manque";k;"lettre"
,"dans ton texte": GOTO 900
880 IF k<=-1 THEN PRINT#2,"Ton texte a";ABS(k)
;"lettre";
890 IF k=-1 THEN PRINT#2," de trop" ELSE PRINT
#2,"s de trop"
900 t=t+1:m=m+1
910 LOCATE#2,1,11:PRINT#2,"Appuie sur une touc
he"
920 i#=INKEY#:IF i#="" THEN 920 ELSE RETURN
930 PRINT#2,CHR$(224);
940 SOUND 1,200,10,15
950 t=t+1
960 RETURN

```

# 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. Une question apparaît, le joueur doit déplacer le curseur devant la réponse qu'il pense être juste. Pour monter tapez 1, 0 pour descendre, barre d'espace pour valider. 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 questions 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>100 à 180</b>	Présentation du programme.
<b>190 à 550</b>	Entrée des données.
<b>590 à 700</b>	Présentation joueur.
<b>740 à 1000</b>	Jeu.
<b>1040 à 1120</b>	Score – fin de jeu.
<b>1160 à 1230</b>	Place aléatoire des réponses.
<b>1270 à 1430</b>	Gestion réponse.

## 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 ESC puis ESC) sous peine de devoir retaper les questions et les réponses.

Enfin, ces données peuvent être sauvegardées sur cassette (ou disquette). Il suffit de les écrire sur le support avec les instructions OPENOUT, OPENIN ET PRINT #9.

## Modifications

Les commentaires du score final (lignes 1060 à 1080) 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 190 à 550 et entrer :

190 Q\$(1)=« votre question »  
 200 R\$(1)=« sa réponse »  
 210 etc...

Vous devrez ensuite sauvegarder sur cassette ou disquette 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 contenant différentes séries de questions-réponses.

D'autres séries de quatre questions et réponses peuvent être ajoutées : il suffit de modifier la ligne 930 :

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

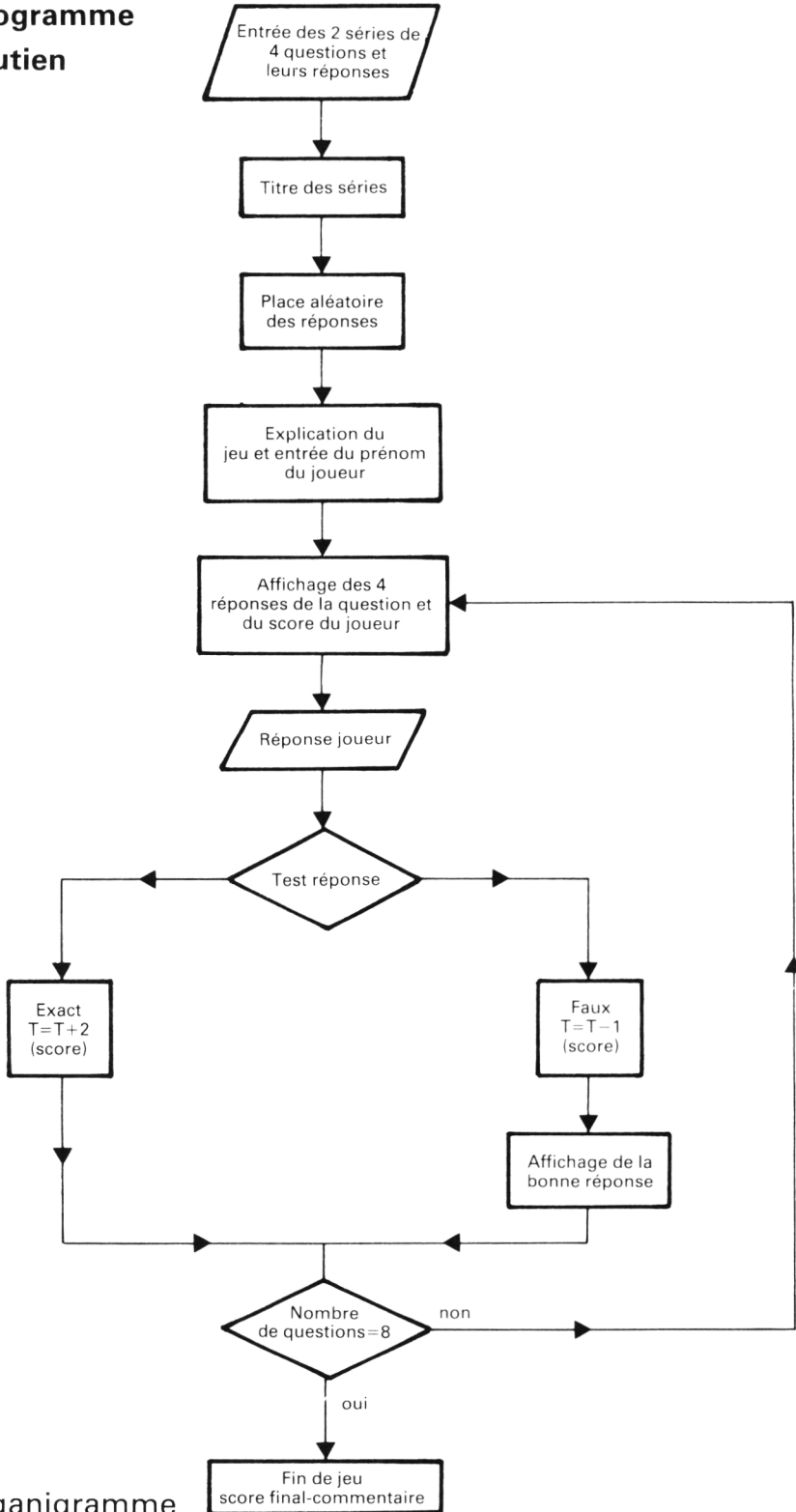
et de dimensionner les tableaux Y,Q\$ et R\$ en début de programme.

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 la ligne 1400.

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

## Programme soutien



## Organigramme

```

10 REM -----
20 REM ----- SOUTIEN -----
30 REM --- D.Nielsen jun 83 ---
40 REM --- Augustin fev 86 ---
50 REM -----
60 ON BREAK GOSUB 1120
70 REM -----
80 REM -> Module lecture des phrases
90 REM -----
100 MODE 2:PAPER 13:PEN 0:CLS
110 LOCATE 2,2:PRINT "SOUTIEN"
120 LOCATE 2,5:PRINT "Ce programme va demander
  de choisir entre quatre reponses", " pour quat
  re questions posees separement"
130 LOCATE 2,7:PRINT "Il y a deux series de qu
  estions"
140 LOCATE 2,9:PRINT "L'ordre des reponses n'e
  st pas toujours le meme"
150 LOCATE 2,11:PRINT "Vous allez d'abord entr
  er les questions et les reponses", " qui vont s
  ervir au jeu"
160 LOCATE 2,13:PRINT "Attention , il ne faut
  pas que, dans chaque serie,", " il y ait deux r
  eponses identiques"
170 LOCATE 2,24:PRINT "APPUYEZ SUR UNE TOUCHE
  POUR CONTINUER"
180 i#=INKEY#: IF i#="" THEN 180 ELSE CLS
190 LOCATE 10,6:PRINT "1ere serie":PRINT
200 LOCATE 2,8:PRINT "Tapez la premiere phrase
  ( ou question )":PRINT
210 INPUT q$(1)
220 PRINT " Maintenant sa reponse + ENTER"
230 INPUT r$(1)
240 PRINT " Deuxieme phrase"
250 INPUT q$(2)
260 PRINT " Sa reponse"
270 INPUT r$(2)
280 PRINT " Troisieme phrase"
290 INPUT q$(3)
300 PRINT " Sa reponse"
310 INPUT r$(3)
320 PRINT " Quatrieme phrase"
330 INPUT q$(4)
340 PRINT " Sa reponse"
350 INPUT r$(4)
360 CLS:LOCATE 10,6:PRINT "2eme serie":PRINT
370 LOCATE 2,8:PRINT "Tapez la cinquieme phras
  e ( ou question )":PRINT

```

```

380 INPUT q$(5)
390 PRINT " Sa reponse"
400 INPUT r$(5)
410 PRINT " Sixieme phrase"
420 INPUT q$(6)
430 PRINT " Sa reponse"
440 INPUT r$(6)
450 PRINT " Septieme phrase"
460 INPUT q$(7)
470 PRINT " Sa reponse"
480 INPUT r$(7)
490 PRINT " Huitieme phrase"
500 INPUT q$(8)
510 PRINT " Sa reponse"
520 INPUT r$(8)
530 CLS
540 LOCATE 2,9:PRINT "Donner un titre a cette
serie"
550 INPUT titre$
560 REM -----
570 REM -> Module jeu
580 REM -----
590 MODE 1:CLS
600 RANDOMIZE TIME/1000
610 LOCATE 2,5:PRINT "Tu vas avoir le choix ",
" entre quatre reponses"," Indique la bonne re
ponse"," avec le curseur"
620 PRINT:PRINT " Les touches de deplacement s
ont : "
630 PRINT " -> 1 pour monter"
640 PRINT " -> 0 pour descendre"
650 PRINT " -> Barre d'espace pour valider"
660 PRINT:INPUT " Quel est ton prenom ";nom$:n
om%=UPPER$(nom%)
670 LOCATE 2,24:PRINT "APPUIE SUR UNE TOUCHE P
OUR CONTINUER"
680 i%=INKEY$:IF i%="" THEN 680 ELSE CLS
690 jeu=1:t=0:h=0
700 GOSUB 1160
710 REM -----
720 REM -> Module 1ere serie
730 REM -----
740 FOR i=jeu TO jeu+3
750 GOSUB 1270
760 LOCATE 2,16:PRINT q$(i)
770 j=6
780 LOCATE 8,j:PRINT CHR$(224)
790 i%=INKEY$:IF i%="" THEN 790

```

```

800 IF i$<>"1" AND i$<>"0" AND i$<>" " THEN SO
UND 1,400,20,15:GOTO 790
810 IF i$=" " THEN 880
820 LOCATE 8,j:PRINT " "
830 IF i$="0" THEN j=j+2 ELSE j=j-2
840 IF j=4 THEN j=12
850 IF j=14 THEN j=6
860 SOUND 1,100,10,15
870 GOTO 780
880 s=-5*(j=6)-6*(j=8)-7*(j=10)-8*(j=12)
890 IF i=y(j-s+h) THEN GOSUB 1350 ELSE GOSUB
1390
900 FOR k=1 TO 6000:NEXT k
910 CLS
920 NEXT i
930 IF jeu=5 THEN 1040
940 REM -----
950 REM -> Module 2eme serie
960 REM -----
970 jeu=jeu+4
980 h=4
990 GOSUB 1130
1000 GOTO 740
1010 REM -----
1020 REM -> Module fin du jeu
1030 REM -----
1040 CLS
1050 LOCATE 2,2:PRINT nom$;" ton score est de"
;t;"point(s)"
1060 IF t>=15 THEN PRINT:PRINT " C'est tres bi
en"
1070 IF t<15 AND t>8 THEN PRINT:PRINT " C'est
bien, mais sois plus"," attentif la prochaine
fois"
1080 IF t<=8 THEN PRINT:PRINT " Il faudra revo
ir le travail"," correspondant a cet exercice"
1090 LOCATE 2,20:PRINT "Le meme exercice ( o /
n ) ?"
1100 i$=INKEY$:IF i$="" THEN 1100
1110 IF i$="0" OR i$="o" THEN 590
1120 MODE 1:PRINT "Au revoir":END
1130 REM -----
1140 REM -> Module place des reponses
1150 REM -----
1160 FOR i=jeu TO jeu+3
1170 y(i)=INT(RND*4)+jeu
1180 IF i=jeu THEN 1220
1190 FOR k=jeu TO i-1
1200 IF y(i)=y(k) THEN 1170

```

```

1210 NEXT k
1220 NEXT i
1230 RETURN
1240 REM -----
1250 REM -> Module affichage reponses
1260 REM -----
1270 CLS:LOCATE 2,2:PRINT titre$
1280 FOR k=1 TO 4
1290 LOCATE 10,k*2+4
1300 af=k+jeu-1
1310 PRINT r$(y(af));" ?"
1320 NEXT k
1330 LOCATE 30,1:PRINT "Score : ";t
1340 RETURN
1350 FOR k=1 TO 300 STEP 10:SOUND 1,k,10,15:NE
XT k
1360 LOCATE 2,20:PRINT "C'est exact ";nom$
1370 t=t+2
1380 RETURN
1390 FOR k=300 TO 1 STEP -10:SOUND 1,k,10,15:N
EXT k
1400 LOCATE 2,20:PRINT "Tu n'a pas trouve ";no
m$
1410 LOCATE 2,22:PRINT "La bonne reponse etait
";r$(i)
1420 t=t-1
1430 RETURN

```



# 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 proposées. 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>100 à 180</b>	Présentation du jeu.
<b>220 à 310</b>	Niveau du jeu.
<b>350 à 630</b>	Création et affichage du lion, du bonhomme et de la voiture.
<b>670 à 800</b>	Test de fin de partie et affichage de l'opération.
<b>840</b>	r est le résultat.
<b>850 à 900</b>	Boucle chronomètre, avec un GOTO 930 en cas de réponse.
<b>910 à 920</b>	Pas de réponse, alors affichage de la réponse et appel de la procédure qui fait avancer le lion.
<b>930 à 1010</b>	Test des réponses (unités et dizaines).
<b>1020 à 1090</b>	Mauvaise réponse, affichage de la bonne réponse, le lion avance et la partie continue.
<b>1100 à 1150</b>	Bonne réponse, le bonhomme avance et on continue la partie (ligne 1150).
<b>1190 à 1260</b>	Fin de partie, dans le cas favorable.
<b>1300 à 1320</b>	Fin de partie au fond de l'estomac du lion.

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 reprogrammés. Il est assez long à taper mais il pimenter cet exercice de calcul mental d'une course contre la montre (l'instruction pour redéfinir un caractère est SYMBOL).

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 en haut de l'écran.

## Modifications

**560** « Temps de réponse de base ». Il peut être allongé en le modifiant.

**700 à 710**

**740 à 750**

**770 à 780**

Modifient les niveaux de jeu.

## Adaptations

En modifiant les lignes de niveau de jeu 700 à 780 et la ligne 840, on peut facilement transposer ce jeu en des exercices de soustraction, de multiplication ou de division.

Exemple pour la multiplication :

**700 A=RND(15)**

**710 B=RND(9)**

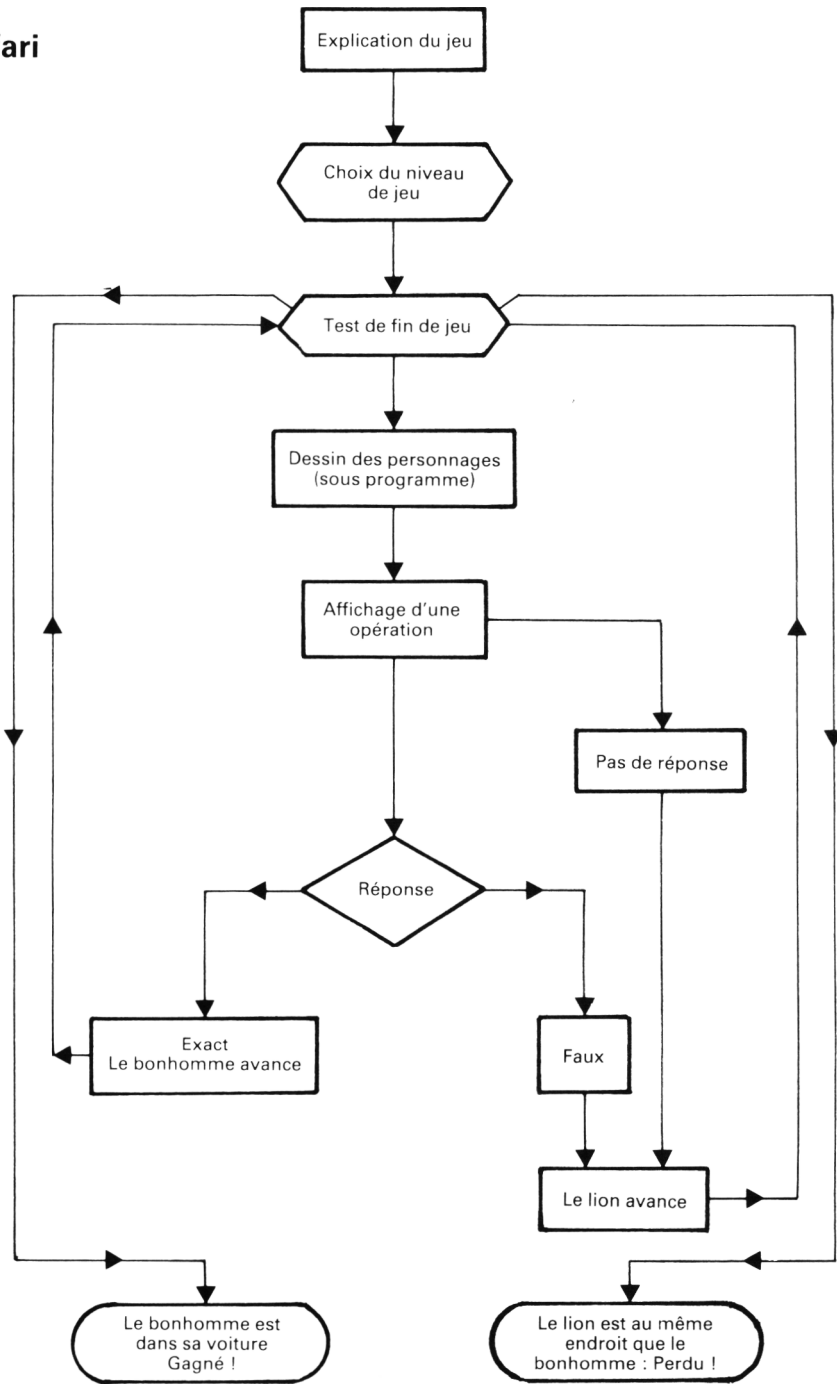
**740 A=RND(20)**

**750 B=RND(12)**

**840 R=A\*B ...**

## Programme

### Safari



## Organigramme

```

10 REM -----
20 REM ----- SAFARI -----
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin fev 86 ---
50 REM -----
60 ON BREAK GOSUB 1260
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 13:PEN 0:CLS
110 RANDOMIZE TIME/1000
120 LOCATE 3,2:PRINT "COURSE DE VITESSE AVEC U
N LION"
130 LOCATE 2,4:PRINT "Seule ta rapidite a trou
ver"," les reponses aux operations"," proposee
s peut te sauver"
140 PRINT:PRINT " Tu es sorti de ta voiture","
 pour photographier un lion endormi"," il vien
t de se reveiller"
150 PRINT:PRINT " Tu vas devoir repondre aux",
" operations proposees"
160 PRINT:PRINT " Si tu reponds bien"," tu av
ances vers ta voiture"
170 PRINT " Si tu te trompes"," le lion vient
 vers toi pour te manger"
180 PRINT:PRINT " A chaque operation le temps"
," de reponse diminue
190 REM -----
200 REM -> Module debut du jeu
210 REM -----
220 LOCATE 2,24:PRINT "APPUIE SUR UNE TOUCHE P
OUR CONTINUER"
230 i$=INKEY$:IF i$="" THEN 230
240 CLS
250 LOCATE 5,5:PRINT "NIVEAU DE JEU"
260 LOCATE 5,7:PRINT "FACILE.....1"
270 LOCATE 5,9:PRINT "MOYEN.....2"
280 LOCATE 5,11:PRINT "DIFFICILE...3"
290 LOCATE 5,13:INPUT "Ton choix ";choix
300 IF choix<1 OR choix>3 THEN 240
310 LOCATE 5,15:INPUT "Quel est ton prenom ";n
om$:nom$=UPPER$(nom$)
320 REM -----
330 REM -> Module affichage
340 REM -----
350 MODE 0:PAPER 5:CLS
360 p=9:m=1
370 SYMBOL AFTER 239
380 SYMBOL 240,0,1,2,4,5,5,37,36

```

```

390 SYMBOL 241,248,4,114,137,85,37,37,137
400 SYMBOL 242,66,92,160,192,131,156,134,126
410 SYMBOL 243,114,4,8,40,208,144,72,108
420 lion$(1)=CHR$(240)+CHR$(241)
430 lion$(2)=CHR$(242)+CHR$(243)
440 SYMBOL 244,49,19,18,19,9,4,3,1
450 SYMBOL 245,192,224,160,224,64,128,224,208
460 SYMBOL 246,1,1,1,3,4,8,80,96
470 SYMBOL 247,202,198,192,224,16,8,5,3
480 enfant$(1)=CHR$(244)+CHR$(245)
490 enfant$(2)=CHR$(246)+CHR$(247)
500 SYMBOL 248,0,0,255,128,128,128,128,255
510 SYMBOL 249,0,0,224,32,32,32,32,254
520 SYMBOL 250,132,132,132,132,255,136,68,56
530 SYMBOL 251,35,227,35,35,254,130,68,56
540 vehi$(1)=CHR$(248)+CHR$(249)
550 vehi$(2)=CHR$(250)+CHR$(251)
560 s=38:v=0
570 PEN 1
580 LOCATE 1,8:PRINT lion$(1)
590 LOCATE 1,9:PRINT lion$(2)
600 LOCATE 10,8:PRINT enfant$(1)
610 LOCATE 10,9:PRINT enfant$(2)
620 LOCATE 19,8:PRINT vehi$(1)
630 LOCATE 19,9:PRINT vehi$(2)
640 REM -----
650 REM -> Module jeu
660 REM -----
670 IF m=p THEN 1300
680 IF p=19 THEN 1190
690 v=v+1
700 ON choix GOTO 710,740,770
710 a=INT(RND*9)
720 b=INT(RND*(9-a))+1
730 GOTO 790
740 a=INT(RND*19)
750 b=INT(RND*(19-a))+1
760 GOTO 790
770 a=INT(RND*39)
780 b=INT(RND*(39-a))+1
790 LOCATE 1,14:PRINT SPACE$(19)
800 LOCATE 5,14:PRINT a;" ";b;"="
810 REM -----
820 REM -> Module reponse
830 REM -----
840 r=a+b
850 FOR i=1 TO s

```

```

860 LOCATE 2,2:PRINT " ":LOCATE 2,2:PRINT i
870 rep#=INKEY#:IF rep#<>" " THEN 930
880 SOUND 1,400-INT(RND*200),5,15
890 FOR j=1 TO 100:NEXT j
900 NEXT i
910 LOCATE 15,14:PRINT r
920 GOTO 1050
930 IF r>9 THEN 960
940 rep=VAL(rep#):LOCATE 13,14:PRINT rep
950 IF r=rep THEN 1100 ELSE 1020
960 dz=VAL(rep#)
970 rp#=INKEY#:IF rp#="" THEN 960
980 unit=VAL(rp#)
990 rep=unit+10*dz
1000 LOCATE 15,14:PRINT rep
1010 IF rep=r THEN 1100
1020 LOCATE 1,24:PRINT SPACE$(39)
1030 LOCATE 1,24:PRINT "FAUX ! C'etait";r
1040 FOR i=1 TO 1000:NEXT i
1050 LOCATE 1,24:PRINT SPACE$(39)
1060 LOCATE 2,24:PRINT "ATTENTION AU LION"
1070 m=m+2:LOCATE m-2,8:PRINT " ";lion$(1):LO
CATE m-2,9:PRINT " ";lion$(2)
1080 FOR i=1 TO 2000:NEXT i
1090 GOTO 670
1100 LOCATE 1,24:PRINT SPACE$(39)
1110 LOCATE 10,24:PRINT "OUF !"
1120 p=p+2:LOCATE p-2,8:PRINT " ";enfant$(1):
LOCATE p-2,9:PRINT " ";enfant$(2)
1130 s=s-2
1140 FOR i=1 TO 2000:NEXT i
1150 GOTO 670
1160 REM -----
1170 REM -> Module gagne
1180 REM -----
1190 CLS
1200 LOCATE 2,5:PRINT nom#;" C'est gagne"
1210 PRINT:PRINT " En";v;"operations"
1220 FOR i=100 TO 1000 STEP 10
1230 SOUND 1,i,5,15
1240 NEXT i
1250 FOR i=1 TO 2000:NEXT i
1260 MODE 1:PEN 0:CLS:END
1270 REM -----
1280 REM -> Module perdu
1290 REM -----
1300 CLS
1310 LOCATE 2,5:PRINT nom#;" au revoir"
1320 GOTO 1220

```



# 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, pendant une seconde.

Puis une liste de 12 mots (les 10 précédents, plus 2 « intrus ») s'affiche (dans des couleurs différentes). Le joueur déplace un curseur à l'aide des touches "0" et "1" 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>100 à 200</b> | Initialisation et présentation du jeu.  |
| <b>210 à 280</b> | Affichage des 10 mots (les mots sont lus dans les lignes de DATAs par une boucle du type FOR NEXT). |
| <b>320 à 380</b> | Affichage des 12 mots. (la ligne 350 tire au sort la couleur du mot).                               |

- 420 à 500** Repérage des mots par l'indice, ainsi l'AMSTRAD CPC ne travaillera pas sur les chaînes de caractères.
- 510 à 690** Lecture du clavier grâce à l'instruction INKEY\$. Déplacement du curseur dans la direction désirée et enfin validation si la touche enfoncée est "barre espace" (ligne 600).
- 700 à 730** Test des réponses. Envoi du message d'erreur ou de réussite et si les deux bonnes réponses sont trouvées : nouvelle séquence.
- 740 à 770** Affichage des réponses exactes.
- 780 à 810** Test le nombre de jeux (f). Jeu suivant ou fin du programme.
- 1000 à 1220** Liste des mots en DATAs. Pour chaque jeu, il a deux listes :  
 – la liste des 10 mots,  
 – la liste des 12 mots dans un ordre différent, dont les deux intrus.

## Particularités techniques

Nous vous proposons une version pour manette de jeu, il suffit de modifier ainsi les lignes 580 à 630.

```

580 rep=JOY(0)
585 FOR k=1 TO 100:NEXT k
590 IF rep<>1 AND rep<>2 AND rep<>16 AND rep<>
32 THEN 580
600 IF rep=16 OR rep=32 THEN 700
610 IF j>6 THEN LOCATE 20,3*j-15 ELSE LOCATE 1
,3+3*j
620 PRINT " "
630 IF rep=1 THEN j=j+1 ELSE j=j-1

```

## Modifications

La temporisation d'affichage des 10 mots lus séparément est fixée à la ligne 260 (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 !

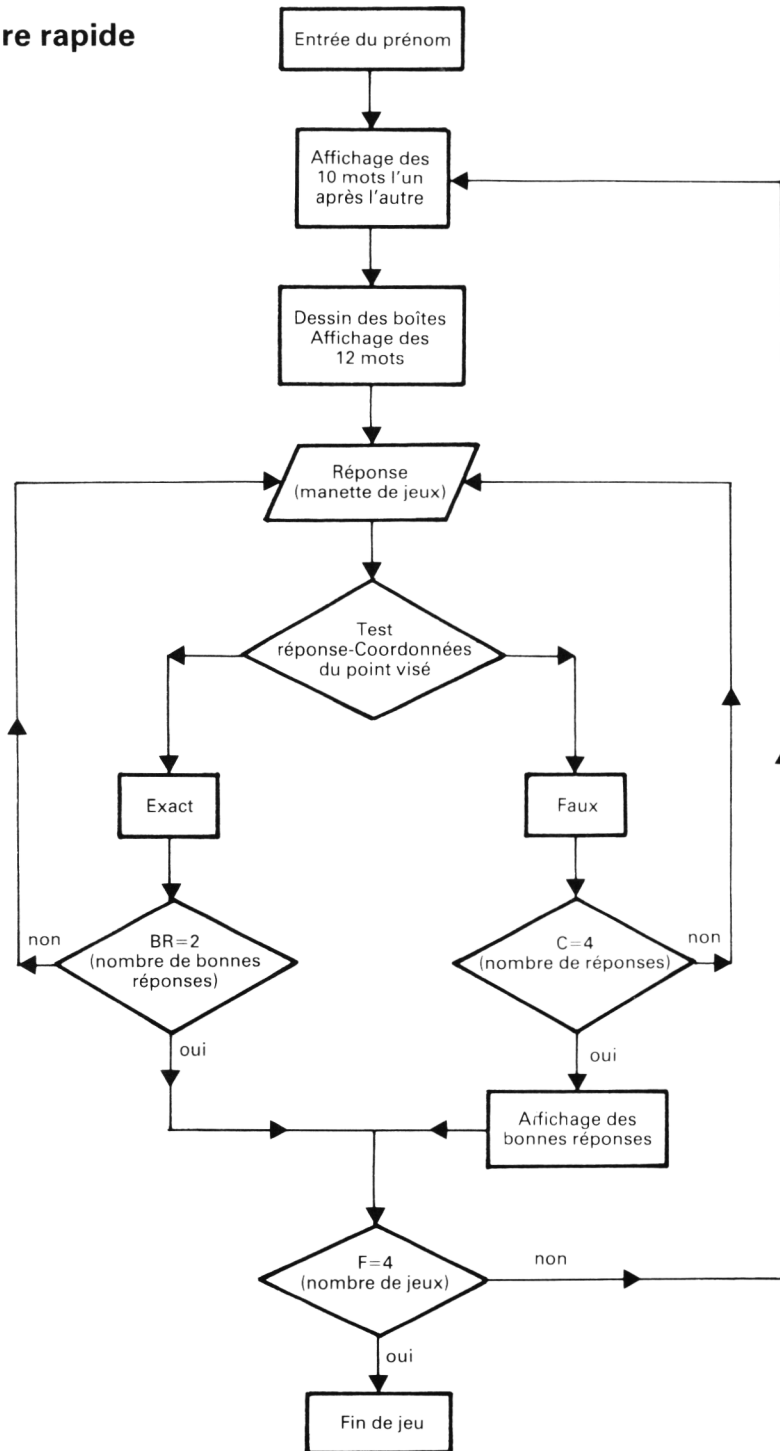
Afin de simplifier le programme, les listes de mots lues en DATA sont doubles, pour un jeu. On pourrait imaginer de lire en une seule fois les 12 mots, de tirer les 2 intrus au sort, et d'afficher les autres. Le jeu serait ainsi différent à chaque fois.

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 :

**790 PRINT #8, nom\$, tr ; « BONNES REPONSES » ; rf;  
« REPONSES FAUSSES »**

## Programme Lecture rapide



Organigramme

```

10 REM -----
20 REM ---- LECTURE RAPIDE ----
30 REM --- D.Nielsen oct 83 ---
40 REM --- Augustin fev 86 ---
50 REM -----
60 ON BREAK GOSUB 1220
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 13:CLS:PEN 0
110 LOCATE 2,2:PRINT "TEST LECTURE RAPIDE"
120 ENV 1,20,1,1,35,-1,1
130 FOR i=1 TO 10:SOUND 1,3780,0,0,1,1,10:NEXT
i
140 LOCATE 2,4:INPUT "Quel est ton prenom ";no
m#:nom#=UPPER$(nom#)
150 FOR i=0 TO 200 STEP 10:SOUND 1,i,10,15:NEX
T i
160 RANDOMIZE TIME/1000
170 RESTORE
180 MODE 0:PAPER 5:PEN 1
190 DIM a$(12),mot$(10)
200 f=0:tr=0:br=0
210 CLS
220 FOR i=1 TO 10
230 READ mot$(i)
240 LOCATE 5,10:PRINT mot$(i)
250 SOUND 1,200,10,15:
260 FOR j=1 TO 2000:NEXT j
270 CLS
280 NEXT i
290 REM -----
300 REM -> Module affichage
310 REM -----
320 PAPER 1:CLS
330 FOR k=1 TO 12
340 READ a$(k)
350 j=INT(RND*13):IF j=1 OR j=2 OR j=4 THEN 35
0
360 IF k>6 THEN LOCATE 11,3*k-15 ELSE LOCATE 2
,3+3*k
370 PEN j:PRINT a$(k)
380 NEXT k
390 REM -----
400 REM -> Module debut du jeu
410 REM -----
420 f=f+1:PEN 5
430 c=0:br=0:t=1:s=0
440 FOR j=1 TO 12

```

```

450 FOR i=1 TO 10
460 IF mot$(i)=a$(j) THEN a$(j)=" "
470 NEXT i,j
480 FOR i=1 TO 12
490 IF a$(i)<>" " THEN num(t)=i:t=t+1
500 NEXT i
510 j=1
520 IF c=4 OR br=2 THEN 730
530 FOR i=1 TO 1000:NEXT i
540 LOCATE 4,24:PRINT SPACE$(6)
550 c=c+1
560 IF j>6 THEN LOCATE 20,3*j-15 ELSE LOCATE 1
,3+3*j
570 PRINT CHR$(224)
580 i$=INKEY$:IF i$="" THEN 580
590 IF i$<>"1" AND i$<>"0" AND i$<>" " THEN SO
UND 1,400,20,15:GOTO 580
600 IF i$="" THEN 700
610 IF j>6 THEN LOCATE 20,3*j-15 ELSE LOCATE 1
,3+3*j
620 PRINT " "
630 IF i$="1" THEN j=j+1 ELSE j=j-1
640 IF j=13 THEN j=1
650 IF j=0 THEN j=12
660 SOUND 1,100,10,15
670 IF j>6 THEN LOCATE 20,3*j-15 ELSE LOCATE 1
,3+3*j
680 PRINT CHR$(224)
690 GOTO 580
700 IF (j=num(1) OR j=num(2)) AND s=0 THEN 720

710 IF (j=num(1) OR j=num(2)) AND j<>s THEN 72
0 ELSE LOCATE 4,24:PRINT "FAUX ":GOTO 520
720 LOCATE 4,24:PRINT "BRAVO":br=br+1:s=j:GOTO
520
730 tr=tr+br
740 IF br=2 THEN 770
750 LOCATE 1,1:PRINT "les mots nouveaux","etai
ent"
760 PRINT a$(num(1));" ";a$(num(2))
770 FOR i=1 TO 2500:NEXT i
780 IF f<4 THEN 210
790 MODE 1:CLS:PEN 0:LOCATE 2,2:PRINT nom$;",
tu as trouve";tr;"mots nouveaux"
800 FOR i=0 TO 200 STEP 10:SOUND 1,i,10,15:NEX
T i
810 LOCATE 2,5:PRINT "C'est termine, au revoir
":END

```

```
1000 DATA LAPIN,CHEVAL,PIGEON,VACHE,COQ
1010 DATA POULE,CANARD,ANE,MOUTON,DINDE
1020 DATA DINDE,CANARD,SINGE,CHEVAL
1030 DATA VACHE,MOUTON,POULE,COQ,LAPIN
1040 DATA PIGEON,LION,ANE
1050 DATA PLEURER,RIRE,SOUPIRER,CHANTER
1060 DATA GRIMACER,SOURIRE,GROGNER
1070 DATA CRIER,SIFFLER,RICANER
1080 DATA COURIR,CRIER,GRIMACER,SOUPIRER
1090 DATA PLEURER,GROGNER,SAUTER,CHANTER
1100 DATA RICANER,SIFFLER,RIRE,SOURIRE
1110 DATA DEUX,SIX,UN,HUIT,QUATRE,CINQ
1120 DATA NEUF,DOUZE,SEPT,ONZE
1130 DATA UN,DEUX,TROIS,QUATRE,CINQ,SIX
1140 DATA SEPT,HUIT,NEUF,DIX,ONZE,DOUZE
1150 DATA JACQUES,PIERRE,ELISABETH
1160 DATA ANNIE,CLAUDE,PHILIPPE,NATHALIE
1170 DATA MARTINE,PASCAL,CHRISTINE
1180 DATA PIERRE,NATHALIE,CLAUDE,ANNIE
1190 DATA JACQUES,DANIEL,CHRISTINE
1200 DATA AUGUSTIN,MARTINE,ELISABETH
1210 DATA PASCAL,PHILIPPE
1220 MODE 1:PEN 0:CLS:END
```

# 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 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>210</b>       | Cinq mots au lieu de dix.        |
| <b>320 à 360</b> | Affichage sur une seule colonne. |

## Modifications

Elles sont les mêmes que pour « lecture rapide » :

- durée de la temporisation de l'affichage des mots de la première liste, ligne 250 ;

- nombre de réponses possibles : on peut considérer, surtout si ce programme sert à l'apprentissage de mots nouveaux, que l'enfant doit trouver de lui-même la bonne réponse, quel que soit le nombre de réponses. Il faut alors supprimer les lignes :

**500**

**530**

**680 à 690**

Pour jouer avec une manette de jeux, il faut remplacer les lignes 550 à 590 par :

```
550 rep=JOY(0)
555 FOR k=1 TO 100:NEXT k
560 IF rep<>1 AND rep<>2 AND rep<>16 AND rep<>
32 THEN 550
570 IF rep=16 OR rep=32 THEN 650
580 LOCATE 8,3+3*j:PRINT " "
590 IF rep=1 THEN j=j+1 ELSE j=j-1
```

## 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).

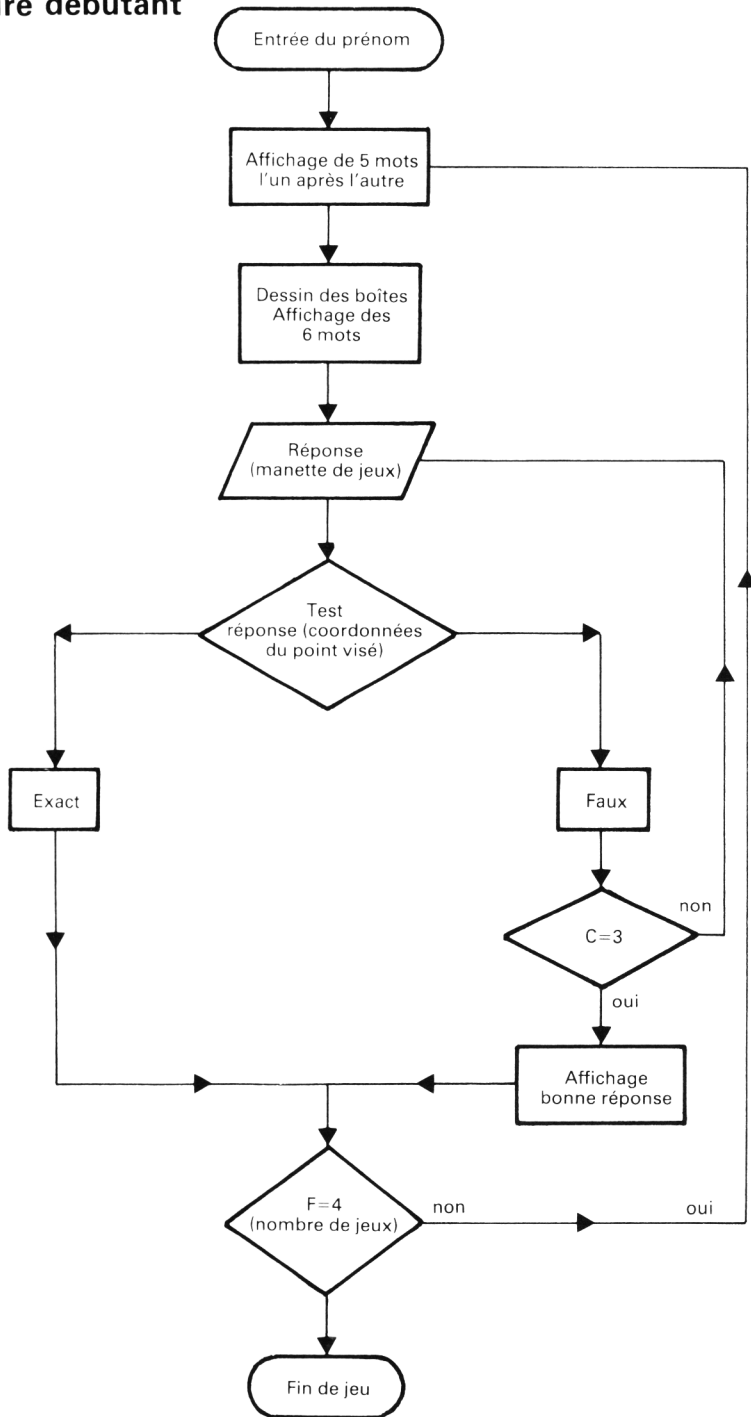
On peut encore entrer en DATA non pas des mots entiers mais des graphèmes (le, la, lo, lu, au, o, dre, bre, etc.).

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.

## Programme

### Lecture débutant



Organigramme

```

10 REM -----
20 REM ---- LECTURE RAPIDE ----
30 REM ---- D.Nielsen oct 83 ----
40 REM ---- Augustin fev 86 ----
50 REM -----
60 ON BREAK GOSUB 1080
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 13:CLS:PEN 0
110 LOCATE 2,2:PRINT "TEST LECTURE RAPIDE DEBU
TANT"
120 ENV 1,20,1,1,35,-1,1
130 FOR i=1 TO 10:SOUND 1,3780,0,0,1,1,10:NEXT
i
140 LOCATE 2,4:INPUT "Quel est ton prenom ";no
m#:nom#=UPPER$(nom#)
150 FOR i=0 TO 200 STEP 10:SOUND 1,i,10,15:NEX
T i
160 RANDOMIZE TIME/1000
170 RESTORE
180 MODE 0:PAPER 5:PEN 1
190 f=0:tr=0:br=0
200 CLS
210 FOR i=1 TO 5
220 READ mot$(i)
230 LOCATE 5,10:PRINT mot$(i)
240 SOUND 1,200,10,15:
250 FOR j=1 TO 2000:NEXT j
260 CLS
270 NEXT i
280 REM -----
290 REM -> Module affichage
300 REM -----
310 PAPER 1:CLS
320 FOR k=1 TO 6
330 READ a$(k)
340 j=INT(RND*13):IF j=1 OR j=2 OR j=4 THEN 34
0
350 LOCATE 10,3+3*k:PEN j:PRINT a$(k)
360 NEXT k
370 REM -----
380 REM -> Module debut du jeu
390 REM -----
400 f=f+1:PEN 5
410 c=1:br=0
420 FOR j=1 TO 6
430 FOR i=1 TO 5
440 IF mot$(i)=a$(j) THEN a$(j)="

```

```

450 NEXT i,j
460 FOR i=1 TO 6
470 IF a$(i)<>"" THEN num=i
480 NEXT i
490 j=1
500 IF c=4 THEN 670
510 FOR i=1 TO 1000:NEXT i
520 LOCATE 4,24:PRINT SPACE$(6)
530 c=c+1
540 LOCATE 8,3+3*j:PRINT CHR$(224)
550 i$=INKEY$:IF i$="" THEN 550
560 IF i$<>"1" AND i$<>"0" AND i$<>" " THEN SO
UND 1,400,20,15:GOTO 550
570 IF i$="" THEN 650
580 LOCATE 8,3+3*j:PRINT " "
590 IF i$="1" THEN j=j+1 ELSE j=j-1
600 IF j=7 THEN j=1
610 IF j=0 THEN j=6
620 SOUND 1,100,10,15
630 LOCATE 8,3+3*j:PRINT CHR$(224)
640 GOTO 550
650 LOCATE 4,24
660 IF j=num THEN PRINT "BRAVO":br=br+1 ELSE P
RINT "FAUX ":GOTO 500
670 tr=tr+br
680 IF br=1 THEN 700
690 LOCATE 1,1:PRINT "le mot nouveau","etait :
";a$(num)
700 FOR i=1 TO 2500:NEXT i
710 IF f<4 THEN 200
720 MODE 1:CLS:PEN 0:LOCATE 2,2:PRINT nom$;",
tu as trouve";tr;"mots nouveaux"
730 FOR i=0 TO 200 STEP 10:SOUND 1,i,10,15:NEX
T i
740 LOCATE 2,5:PRINT "C'est termine, au revoir
":END
1000 DATA un,une,le,de,il
1010 DATA un,le,deux,il,de,une
1020 DATA maman,papa,petit,chat,fait
1030 DATA papa,fait,chat,maman,pepe,petit
1040 DATA dans,est,il,avec,elle
1050 DATA elue,avec,elle,il,dans,est
1060 DATA fait,elle,de,est,et
1070 DATA et,fait,de,four,est,elle
1080 MODE 1:PEN 0:CLS:END

```

# 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 fautive, 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

60	Procédure anti-“BREAK”.
100 à 160	Définition des fenêtres.
200 à 240	Initialisation et lecture des mots en DATAs. La ligne 240 tire au hasard un mot.
280 à 350	Lecture de la réponse de l'enfant, avec une vérification immédiate de l'article.
390 à 510	Contrôle de la réponse, et le cas échéant aiguillage vers les exceptions.
550 à 560	La règle est simple.
600 à 620	Bonne réponse, et si nb < > 10 nouvelles questions.
1000 à 1320	DATAs des mots.
2000 à 2030	Contrôle des noms en S, X, Z.
3000 à 3030	Contrôle des noms en AU et EU.
4000 à 4130	Contrôle des noms en AL.
5000 à 5080	Contrôle des noms en OU.
6000 à 6080	Contrôle des noms en ALL.

## Particularités techniques

Lorsque la réponse est donnée, le nom proposé est comparé aux différentes exceptions à la règle générale, afin de déterminer la bonne réponse (lignes 390 à 510).

Les exceptions sont inscrites en DATA, elles peuvent ainsi servir à la comparaison avec le nom donné (lignes 390 à 510) et aussi à l'affichage de la règle en cas de réponse erronée (instruction RESTORE pour pouvoir relire le bloc DATA).

## Modifications

On peut augmenter le nombre de mots à tirer au sort en DATA (lignes 1000 à 1100) en changeant seulement la valeur du multiplicateur de la fonction RND de la ligne 240.

*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

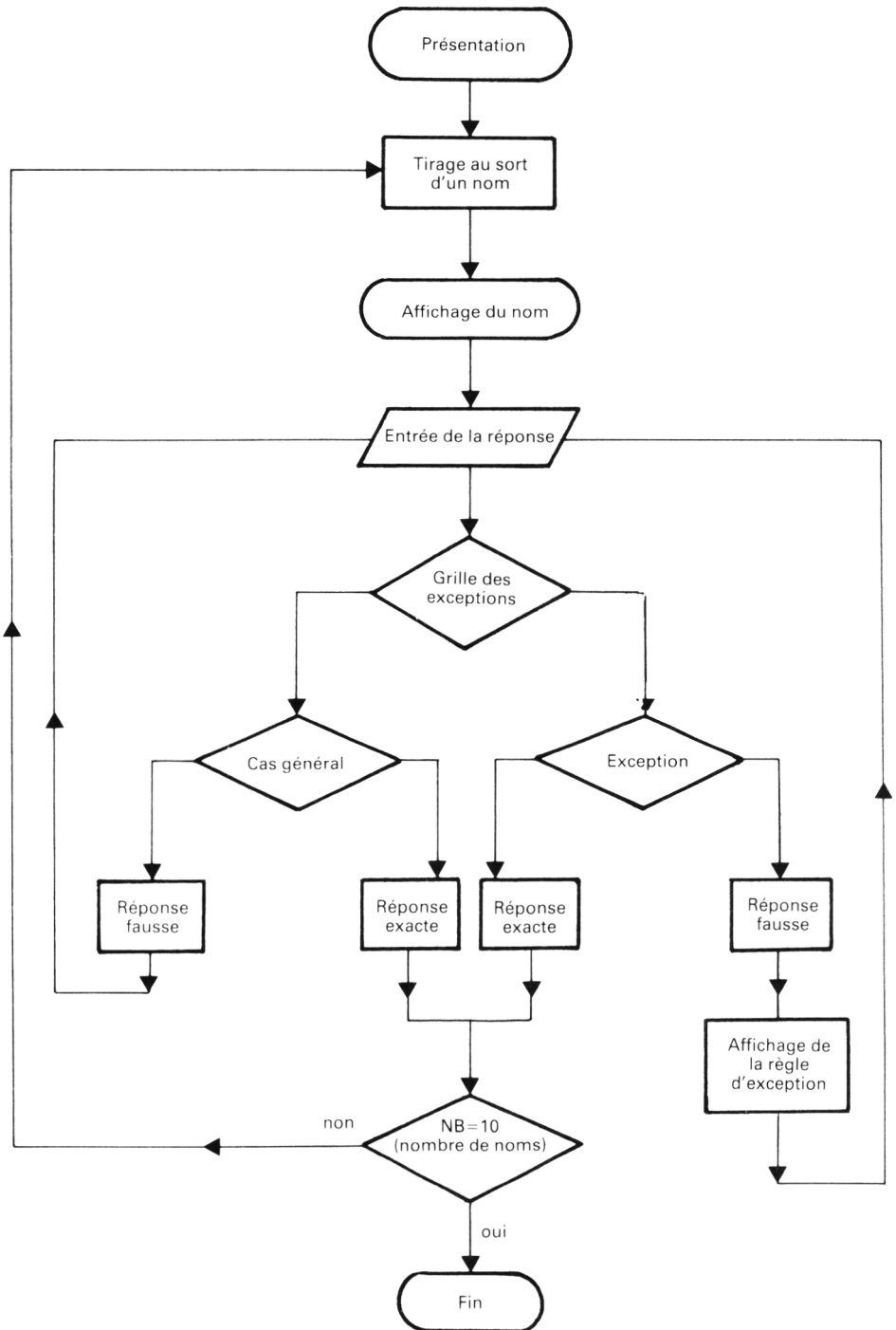
Si l'on possède une imprimante, il est possible comme pour le programme de conjugaison de garder une trace des erreurs rencontrées, afin de revenir ensuite sur ces règles.

Il faut alors ajouter ces lignes :

**2025, 3025, 4115, 5075, 6075**

**Ex. : 2025 PRINT#8, nom\$, n\$(t), rep\$**

## Programme Pluriel des noms



Organigramme

```

10 REM -----
20 REM --- PLURIEL DES NOMS ---
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin jan 86 ---
50 REM -----
60 ON BREAK GOSUB 690
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 13:CLS:PEN 0
110 WINDOW#3,11,31,5,7:PAPER#3,0:CLS#3
120 WINDOW#3,10,30,4,6:PAPER#3,10:CLS#3
130 WINDOW#2,5,38,11,22:PAPER#2,0:CLS#2
140 WINDOW#2,4,37,10,21:PAPER#2,10:CLS#2
150 PEN#3,0:LOCATE#3,3,2:PRINT#3,"PLURIEL DES
NOMS"
160 PEN#2,0:LOCATE#2,2,2:INPUT#2,"Quel est ton
pre nom ";nom$:nom%=UPPER$(nom%)
170 REM -----
180 REM -> initialisation et lecture
190 REM -----
200 DIM n$(31),a$(15)
210 nb=0:v=0
220 RESTORE:RANDOMIZE TIME/1000
230 READ n$(v+1):IF n$(v+1)<>"ZZZZ" THEN v=v+1
:GOTO 230
240 t=INT(RND*30)+1
250 REM -----
260 REM -> Module pluriel
270 REM -----
280 FOR i=1 TO 6000:NEXT i
290 CLS#2:PRINT#2,"Ecris ce nom au pluriel"
300 CLS#3:LOCATE#3,2,2:PRINT#3,n$(t)
310 PRINT#2,"Attention, n'oublie pas l'article
"
320 LOCATE#2,2,6:INPUT#2,rep$
330 PRINT#2:rep%=UPPER$(rep%)
340 l=LEN(n$(t)):p=LEN(rep%)
350 IF LEFT$(rep$,3)<>"DES" THEN PRINT#2:PRINT
#2,"Tu as oublie l'article":GOTO 280
360 REM -----
370 REM -> Module grille des exceptions
380 REM -----
390 IF RIGHT$(n$(t),1)="Z" OR RIGHT$(n$(t),1)=
"S" OR RIGHT$(n$(t),1)="X" THEN 2000
400 IF RIGHT$(n$(t),2)="EU" OR RIGHT$(n$(t),2)
="AU" THEN 3000
410 IF RIGHT$(n$(t),2)="AL" THEN 4000
420 RESTORE 1200

```

```

430 FOR i=1 TO 7
440 READ exc$(i)
450 IF exc$(i)=n$(t) THEN 5000
460 NEXT i
470 RESTORE 1230
480 FOR i=1 TO 8
490 READ exc$(i)
500 IF exc$(i)=n$(t) THEN 6000
510 NEXT i
520 REM -----
530 REM -> Module reponse
540 REM -----
550 IF RIGHT$(rep$,3)=RIGHT$(n$(t),2)+"S" THEN
600
560 PRINT#2," Tu te trompes, ";nom$," La regle
est simple, recommence":GOTO 280
570 REM -----
580 REM -> Module reponse exacte
590 REM -----
600 PRINT#2," C'est exact, ";nom$
610 nb=nb+1
620 IF nb<>10 THEN PRINT#2," Voici un autre no
m":GOTO 240
630 REM -----
640 REM -> Module fin
650 REM -----
660 CLS#3:LOCATE#3,4,2:PRINT#3,nom$
670 PRINT#2," C'est termine, au revoir"
680 FOR i=1 TO 3500:NEXT i
690 CLS:END
1000 DATA UN CHAT,UN CHEVAL,UN CANARD
1010 DATA UN CHEVEU,UN BAL,UN CHIEN
1020 DATA UNE JOURNEE,UN PRIX,UN RAT
1030 DATA UNE FOURRURE,UN TROU,UN BIJOU
1040 DATA UN FOUR,UNE SOURIS,UN VITRAIL
1050 DATA UN NEZ,UN CHACAL,UN OBJECTIF
1060 DATA UNE NOIX,UN PINGOUIN,UN GAZ
1070 DATA UN TRAVAIL,UN RELAIS,UN CHOU
1080 DATA UNE SERRURE,UN PUIT,UN RECITAL
1090 DATA UN JOUJOU,UNE TISANE,UN VERRE
1100 DATA ZZZZ
1200 DATA UN BIJOU,UN JOUJOU,UN CHOU
1210 DATA UN GENOU,UN HIBOU,UN CAILLOU
1220 DATA UN POU
1230 DATA UN VITRAIL,UN TRAVAIL,UN EMAIL
1240 DATA UN BAIL,UN SOUPIRAIL
1250 DATA UN CORAIL,UN EVANTAIL

```

```

1260 DATA UN FERMAIL
1300 DATA UN BAL,UN FESTIVAL,UN CARNAVAL
1310 DATA UN REGAL,UN CEREMONIAL
1320 DATA UN RECITAL,UN CHACAL
1330 REM -----
1340 REM -> Module les noms en S,Z,X
1350 REM -----
2000 IF RIGHT$(n$(t),3)=RIGHT$(rep$,3) THEN 600
2010 PRINT#2," Les noms termines au singulier"
," par S ou Z ou X gardent la meme"," forme au
pluriel"
2020 PRINT#2," Exemple : UN BOIS -> DES BOIS"
2030 GOTO 280
2040 REM -----
2050 REM -> Module noms en AU et EU
2060 REM -----
3000 IF RIGHT$(rep$,3)=RIGHT$(n$(t),2)+"X" THE
N 600
3010 PRINT#2," Les noms termines au singulier"
," par AU ou EAU ou EU ou OEU"," prennent un X
au pluriel"
3020 PRINT#2," Exemple : UN ADIEU -> DES ADIEU
X"
3030 GOTO 280
3040 REM -----
3050 REM -> Module noms en AL
3060 REM -----
4000 RESTORE 1300
4010 FOR i=1 TO 7
4020 READ exc$(i)
4030 IF n$(t)=exc$(i) THEN 4130
4040 NEXT i
4050 IF RIGHT$(rep$,3)=MID$(n$(t),1-1,1)+"UX"
THEN 600
4060 PRINT#2,"Les noms termines au singulier p
ar AL ont un pluriel en AUX, sauf :"
4070 RESTORE 1300
4080 FOR i=1 TO 7
4090 READ exc$(i):PRINT#2,"-> ";exc$(i)
4100 NEXT i
4110 PRINT#2:PRINT#2," qui ont un pluriel en A
LS"
4120 GOTO 280
4130 IF RIGHT$(rep$,3)=RIGHT$(n$(t),2)+"S" THE
N 600 ELSE 4060
4140 REM -----

```

```
4150 REM -> Module noms en OU
4160 REM -----
5000 IF RIGHT$(rep$,3)=RIGHT$(n$(t),2)+"X" THE
N 600
5010 CLS#2:PRINT#2," Les noms suivants":PRINT#
2
5020 RESTORE 1200
5030 FOR i=1 TO 7
5040 READ exc$(i)
5050 PRINT#2,"-> ";exc$(i)
5060 NEXT i
5070 PRINT#2:PRINT#2," prennent un X au plurié
1"
5080 GOTO 280
5090 REM -----
5100 REM -> Module noms en AIL
5110 REM -----
6000 IF RIGHT$(rep$,3)=MID$(n$(t),1-2,1)+"UX"
THEN 600
6010 CLS#2:PRINT#2," Les noms suivants"
6020 RESTORE 1230
6030 FOR i=1 TO 8
6040 READ exc$(i)
6050 PRINT#2,"-> ";exc$(i)
6060 NEXT i
6070 PRINT#2:PRINT#2," ont un pluriel en AUX"
6080 GOTO 280
```

# 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 fautive, 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>100 à 200</b>	Présentation du jeu et saisie du prénom du joueur.
<b>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>340 à 360</b>	Tirage au sort d'un verbe, d'une personne, d'un temps.
<b>400 à 430</b>	Entrée de la réponse de l'enfant.
<b>470 à 660</b>	Test de la réponse.
<b>700 à 800</b>	Réponse fautive. Affichage de la conjugaison-type.
<b>810 à 850</b>	Nouvelle entrée de réponse.
<b>890 à 940</b>	Bonne réponse, et test de suite du jeu.
<b>980 à 1020</b>	Fin du jeu.
<b>1030 à 1130</b>	DATA verbes.
<b>1140 à 1190</b>	DATA personnes.
<b>1200 à 1230</b>	DATA temps.
<b>1240 à 1310</b>	DATA terminaisons.

## 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 et du temps du verbe demandés.

La conjugaison-type, en cas d'erreur, est affichée par une boucle FOR NEXT aux lignes 740 à 770.

## 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.

Ligne 240 : la lecture des verbes dans les DATA peut se faire 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 dimensions de tableaux de la ligne 170 si le nombre de verbes devient trop important). Il suffira de modifier le paramètre multiplicateur de la fonction RND de la ligne 340 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 verbe 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 1200 à 1230.

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 470 à 570 ;
- conjugaison de l'auxiliaire et lecture des participes passés en DATA.

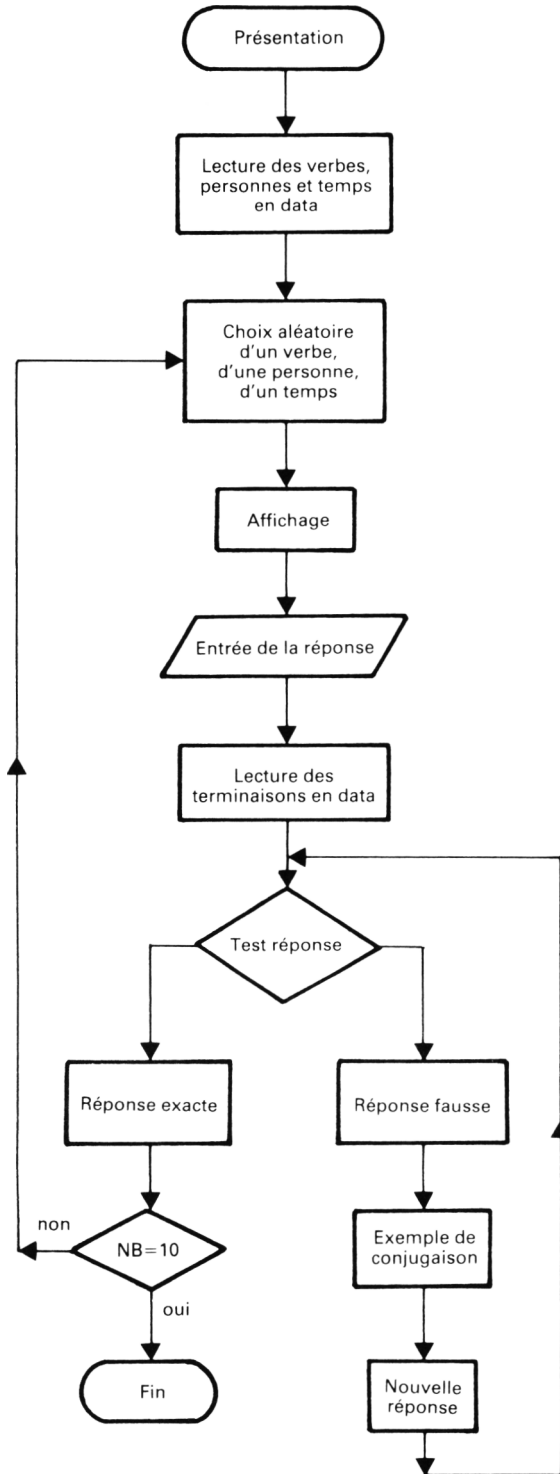
L'enfant ne pouvant sortir du programme qu'en trouvant la bonne réponse, il est inutile de comptabiliser le score. Par contre, et surtout dans une école, il est très utile de garder une trace écrite des erreurs.

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

```
701 PRINT#8, nom$, "erreur au verbe" ; v$(v) ; per$ (p) ;  
temp$(t)
```

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

**Programme  
Conjugaison**



Organigramme

```

10 REM -----
20 REM ----- CONJUGAISON -----
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin jan 86 ---
50 REM -----
60 ON BREAK GOSUB 1020
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 13:CLS:PEN 0
110 WINDOW#3,15,27,5,7:PAPER#3,0:CLS#3
120 WINDOW#3,14,26,4,6:PAPER#3,10:CLS#3
130 WINDOW#2,5,38,11,22:PAPER#2,0:CLS#2
140 WINDOW#2,4,37,10,21:PAPER#2,10:CLS#2
150 PEN#3,0:LOCATE#3,2,2:PRINT#3,"CONJUGAISON"
160 PEN#2,0:LOCATE#2,2,2:INPUT#2,"Quel est ton
   prenom ";nom$:nom$=UPPER$(nom$)
170 DIM v$(40),pr$(6)
180 nb=0:l=0
190 RANDOMIZE TIME/1000
200 RESTORE
210 REM -----
220 REM -> Module lecture des DATAs
230 REM -----
240 READ v$(1+1): IF v$(1+1)<>"ZZZZ" THEN l=l+
   1:GOTO 240
250 FOR i=1 TO 6
260 READ pr$(i),pers$(i)
270 NEXT i
280 FOR i=1 TO 4
290 READ temps$(i)
300 NEXT i
310 REM -----
320 REM -> tirage au sort d'un verbe
330 REM -----
340 v=INT(RND*30)+1
350 p=INT(RND*6)+1
360 t=INT(RND*4)+1
370 REM -----
380 REM -> Module conjugaison
390 REM -----
400 CLS#2:CLS#3:LOCATE#3,2,2:PRINT#3,v$(v)
410 LOCATE#2,1,2:PRINT#2,"Conjugué ce verbe au
   ";temps$(t)
420 PRINT#2,"a la ";pers$(p)
430 INPUT#2,rep$:rep$=UPPER$(rep$)
440 REM -----
450 REM -> Module reponse
460 REM -----

```

```

470 IF RIGHT$(v$(v),2)="IR" THEN 530
480 ON t GOTO 490,500,510,520
490 RESTORE 1240:GOTO 610
500 RESTORE 1250:GOTO 610
510 RESTORE 1260:GOTO 610
520 RESTORE 1270:GOTO 610
530 ON t GOTO 540,550,560,570
540 RESTORE 1280:GOTO 610
550 RESTORE 1290:GOTO 610
560 RESTORE 1300:GOTO 610
570 RESTORE 1310
580 REM -----
590 REM -> Module test reponse
600 REM -----
610 FOR i=1 TO 6
620 READ term$(i)
630 NEXT i
640 l=LEN(v$(v))
650 radi$=LEFT$(v$(v),l-2)
660 IF rep$=pr$(p)+" "+radi$+term$(p) THEN 890
670 REM -----
680 REM -> Module reponse fausse
690 REM -----
700 PRINT#2,"Tu te trompes, ";nom$
710 FOR i=1 TO 2000:NEXT i
720 CLS#2:PRINT#2,"Voici un exemple de conjuga
ison","d'un verbe semblable au ";temps$(t)
730 IF RIGHT$(v$(v),2)="IR" THEN ex$="FIN" EL
SE ex$="CHANT"
740 FOR i=1 TO 6
750 LOCATE#2,10,4+i
760 PRINT#2, pr$(i);" ";ex$;term$(i)
770 NEXT i
780 PRINT#2," As-tu trouve ?"
790 PRINT#2," Appuie sur une touche"
800 i$=INKEY$:IF i$="" THEN 800
810 CLS#2:PRINT#2:PRINT#2,pers$(p)
820 PRINT#2,"du ";temps$(t)
830 PRINT#2:PRINT#2,"Quelle est ta nouvelle re
ponse"
840 INPUT#2,rep$:rep$=UPPER$(rep$)
850 GOTO 660
860 REM -----
870 REM -> Module reponse exacte
880 REM -----
890 PRINT#2,"C'est exact, ";nom$
900 nb=nb+1
910 IF nb=9 THEN 980

```

```

920 PRINT#2,"Voici un autre verbe"
930 FOR i=1 TO 2000:NEXT i
940 GOTO 340
950 REM -----
960 REM -> Module fin du jeu
970 REM -----
980 FOR i=1 TO 3500:NEXT i
990 CLS#2:CLS#3:LOCATE#3,2,2:PRINT#3,nom#
1000 LOCATE#2,2,2:PRINT#2,"C'est termine, au r
evoir"
1010 FOR i=1 TO 2000:NEXT i
1020 CLS:END
1030 DATA CHAHUTER,PLEURER,JOUER
1040 DATA COPIER,CREER,PUNIR
1050 DATA FLEURIR,SOLDER,FREMIR
1060 DATA FRISER,FLECHIR,FROTTER
1070 DATA LOUER,NOIRCIR,PALIR
1080 DATA PARFUMER,PEDALER,MESURER
1090 DATA RACCOURCIR,NOURRIR,REFUSER
1100 DATA RONFLER,REUNIR,REUSSIR
1110 DATA SERRER,PUNIR,BONDIR
1120 DATA VIEILLIR,TRAHIR,CONVERTIR
1130 DATA ZZZZ
1140 DATA JE,"1ere PERSONNE DU SINGULIER"
1150 DATA TU,"2eme PERSONNE DU SINGULIER"
1160 DATA IL,"3eme PERSONNE DU SINGULIER"
1170 DATA NOUS,"1ere PERSONNE DU PLURIEL"
1180 DATA VOUS,"2eme PERSONNE DU PLURIEL"
1190 DATA ILS,"3eme PERSONNE DU PLURIEL"
1200 DATA "PRESENT DE L'INDICATIF"
1210 DATA "IMPARFAIT DE L'INDICATIF"
1220 DATA "FUTUR SIMPLE"
1230 DATA "PASSE SIMPLE"
1240 DATA E,ES,E,ONS,EZ,ENT
1250 DATA AIS,AIS,AIT,IONS,IEZ,AIENT
1260 DATA ERAI,ERAS,ERA,ERONS,EREZ,ERONT
1270 DATA AI,AS,A,AMES,ATES,ERENT
1280 DATA IS,IS,IT,ISSONS,ISSEZ,ISSENT
1290 DATA ISSAIS,ISSAIS,ISSAIT,ISSIONS,ISSIEZ,
ISSAIENT
1300 DATA IRAI,IRAS,IRA,IRONS,IREZ,IRONTE
1310 DATA IS,IS,IT,IMES,ITES,IRENT

```

# 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 dans une fenêtre. 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.

AMSTRAD CPC 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>60</b>	Procédure anti-"BREAK".
<b>100 à 190</b>	Initialisation des fenêtres et présentation du programme. Choix du nombre de chiffres.
<b>230 à 310</b>	Les chiffres de chaque nombre sont tirés au sort séparément.
<b>350 à 420</b>	Affichage de l'opération.
<b>430</b>	Mise à zéro des retenues.
<b>440 à 520</b>	Par où commencer l'opération ?
<b>560 à 670</b>	Addition des unités.

*Lignes 560 à 580* : Réponse et test de celle-ci. En cas de réponse fausse, envoi de la table d'addition, puis nouvelle réponse.

*Ligne 590* : Affichage de la réponse si elle ne comporte qu'un seul chiffre.

*Lignes 600 à 640* : Décomposition du résultat en unités et dizaines.

*Lignes 650 à 670* : Affichage de la retenue et des unités.

<b>710 à 840</b>	Addition des dizaines. (même déroulement que pour les unités).
<b>880 à 930</b>	Addition des centaines.
<b>940 à 980</b>	Fin de l'opération et passage à la suivante.
<b>1020 à 1110</b>	Sous-programme : Table d'addition. Elle est affichée en cas de réponse fausse pour une des additions simples. La variable "t" prend la valeur du chiffre (unités, dizaines, centaines) du nombre supérieur "a".
<b>1120 à 1130</b>	Fin du jeu.

## Particularités techniques

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

L'affichage dans une fenêtre permet de placer les chiffres les uns juste à côté des autres. Le signe opératoire est placé à l'extrême gauche de l'opération.

L'instruction LOCATE est très utilisée, notamment pour placer les retenues ou les tables d'addition.

Les lignes 250, 440 et 480, 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 600 et 770 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.

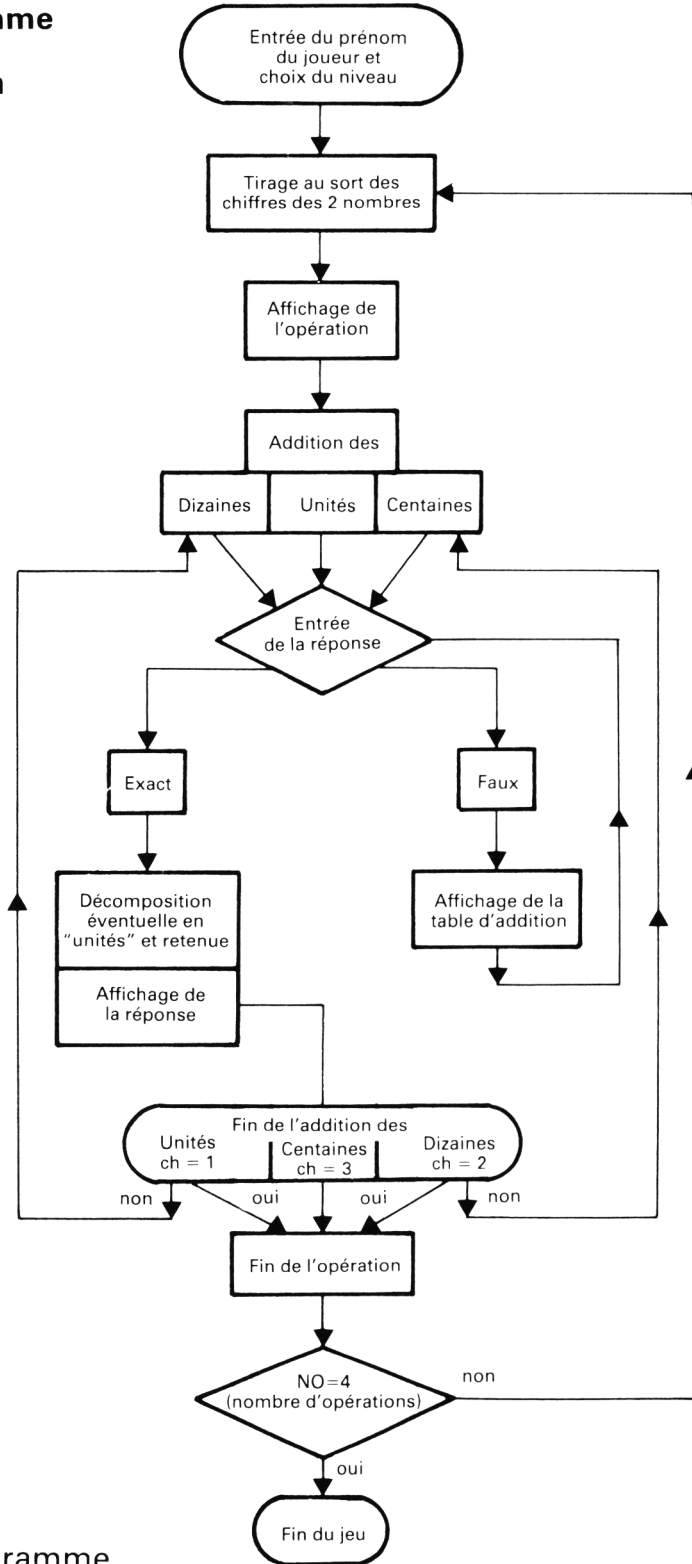
## Adaptations

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

Le mécanisme opératoire (lignes 560 à 980) 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.

**Programme**  
**Addition**



Organigramme

```

10 REM -----
20 REM ----- ADDITION -----
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin jan 86 ---
50 REM -----
60 ON BREAK GOSUB 1130
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1
110 WINDOW#1,2,39,13,25:PAPER#1,13:PEN#1,0:CLS
#1
120 WINDOW#2,14,26,4,10:PAPER#2,13:PEN#2,0
130 WINDOW#3,14,26,2,2:PAPER#3,13:PEN#3,0
140 n=0
150 LOCATE#1,16,2:PRINT#1,"ADDITION"
160 PRINT#1:INPUT#1,"Quel est ton prenom ";nom
#:PRINT#1
170 PRINT#1,"Additions avec des nombres a un,
deux ou trois chiffres ?"
180 PRINT#1:PRINT#1,"Indique ton choix ( 1, 2
ou 3) ?"
190 rep#=INKEY#:IF (rep#<>"1") AND (rep#<>"2")
AND (rep#<>"3") THEN 190
200 REM -----
210 REM -> Module choix des chiffres
220 REM -----
230 RANDOMIZE TIME/1000
240 ch=VAL(rep#)
250 IF n=4 THEN 1120
260 ON ch GOTO 290,280,270
270 ac=INT(RND*10):bc=INT(RND*10)
280 ad=INT(RND*10):bd=INT(RND*10)
290 au=INT(RND*10):bu=INT(RND*10)
300 a=ac*100+ad*10+au
310 b=bc*100+bd*10+bu
320 REM -----
330 REM -> Module affichage operation
340 REM -----
350 CLS#2:LOCATE#2,4,2:PRINT#2,ac;ad;au
360 IF rep#="1" THEN LOCATE#2,4,2:PRINT#2,SPAC
E#(5)
370 IF rep#="2" THEN LOCATE#2,4,2:PRINT#2,SPAC
E#(3)
380 LOCATE#2,3,3:PRINT#2,"+"
390 LOCATE#2,4,4:PRINT#2,bc;bd;bu
400 IF rep#="1" THEN LOCATE#2,4,4:PRINT#2,SPAC
E#(5)

```

```

410 IF rep#="2" THEN LOCATE#2,4,4:PRINT#2,SPAC
E$(3)
420 LOCATE#2,2,5:PRINT#2,"-----"
430 CLS#3:red=0:rec=0
440 IF ch=1 THEN 560
450 CLS#1:PRINT#1:PRINT#1,"Par quels chiffres
commences-tu      ton addition ?"
460 PRINT#1:PRINT#1,"Les unites.....1"
470 PRINT#1,"Les dizaines...2"
480 IF ch=3 THEN PRINT#1,"Les centaines..3"
490 PRINT#1:INPUT#1,"Ton choix ";deb
500 IF deb=1 THEN 560
510 PRINT#1,"Tu te trompes, ";nom#;" reflechis
encore"
520 FOR i=1 TO 1000:NEXT i:GOTO 450
530 REM -----
540 REM -> Module addition des unites
550 REM -----
560 CLS#1:PRINT#1:PRINT#1,"Que font";au;"unite
(s) +";bu;"unite(s) ?"
570 INPUT#1,ru
580 IF ru<>au+bu THEN t=au:GOSUB 1020:GOTO 560

590 IF ru<10 THEN LOCATE#2,10,6:PRINT#2,ru:GOT
O 710
600 IF ru>9 AND ch=1 THEN LOCATE#2,8,6:PRINT#2
,"1 ";ru-10:GOTO 710
610 PRINT#1,"C'est exact, ";nom#:PRINT#1
620 PRINT#1,"Cela fait 1 dizaine, plus";ru-10;
"unite(s)"
630 PRINT#1,"Il faut donc ecrire";ru-10;"dans
la colonne des unites et","1 dizaine au dessus
de la colonne","des dizaines"
640 FOR i=1 TO 2000:NEXT i
650 LOCATE#2,10,6:PRINT#2,ru-10
660 LOCATE#3,7,1:PRINT#3,"+1"
670 FOR i=1 TO 6000:NEXT i
680 REM -----
690 REM -> Module addition des dizaines
700 REM -----
710 IF ch=1 THEN 940
720 CLS#1:PRINT#1:PRINT#1,"Que font";ad;"dizai
ne(s) +";bd;"dizaine(s) ?";
730 IF ru>9 THEN PRINT#1,"plus une retenue":re
d=1
740 INPUT#1,rd
750 IF rd<>ad+bd+red THEN t=ad:GOSUB 1020:GOTO
720
760 IF rd<10 THEN LOCATE#2,7,6:PRINT#2,rd:GOTO
880

```

```

770 IF rd>9 AND ch=2 THEN LOCATE#2,5,6:PRINT#2
,"1 ";rd-10:GOTO 880
780 PRINT#1,"C'est exact, ";nom#:PRINT#1
790 PRINT#1,"Cela fait 1 centaine, plus";rd-10
;"dizaine(s)"
800 PRINT#1,"Il faut donc ecrire";rd-10;"dans
la colonne des dizaines et","1 centaine au des
sus de la colonne","des centaines"
810 FOR i=1 TO 2000:NEXT i
820 LOCATE#2,7,6:PRINT#2,rd-10
830 LOCATE#3,4,1:PRINT#3,"+1"
840 FOR i=1 TO 6000:NEXT i
850 REM -----
860 REM -> Module addition des centaines
870 REM -----
880 IF ch=2 THEN 940
890 CLS#1:PRINT#1:PRINT#1,"Que font";ac;"centa
ine(s) +";bc;"centaine(s) ?"
900 IF rd>9 THEN PRINT#1,"plus une retenue":re
c=1
910 INPUT#1,rc
920 IF rc<>ac+bc+rec THEN t=ac:GOSUB 1020:GOTO
890
930 IF rc<10 THEN LOCATE#2,4,6:PRINT#2,rc ELSE
LOCATE#2,2,6:PRINT#2,"1 ";rc-10
940 CLS#1:PRINT#1:PRINT#1,"Bravo ";nom#:PRINT#
1
950 PRINT#1,"Voici maintenant une autre operat
ion"
960 n=n+1
970 FOR i=1 TO 3000:NEXT i
980 GOTO 240
990 REM -----
1000 REM -> Module table d'addition
1010 REM -----
1020 CLS#1:PRINT#1,"Tu te trompes, ";nom#;" re
lis la table d'addition"
1030 FOR i=1 TO 1000:NEXT i
1040 CLS#1:FOR i=0 TO 10
1050 PRINT#1,t;"+";i;"=";t+i
1060 NEXT i
1070 FOR i=1 TO 2000:NEXT i
1080 PRINT#1,"As-tu trouve la bonne reponse ";
nom#;" ?"
1090 PRINT#1,"Appuie sur une touche"
1100 i#=INKEY#:IF i#="" THEN 1100
1110 RETURN
1120 CLS:PRINT "C'est termine":PRINT "Au revoi
r ";nom#:END
1130 CLS:END

```

# 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 opération s'inscrit dans une fenêtre. Chaque chiffre étant tiré au sort entre 0 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>100 à 180</b>	Initialisation et présentation du programme.
<b>220 à 300</b>	Tirage au sort des chiffres des deux nombres.
<b>340 à 410</b>	Affichage de l'opération.
<b>420 à 520</b>	Sens de l'opération.

*Ligne 420* : Si  $A < B$  (soustraction impossible), le programme va en 1280.

*Lignes 430 à 520* : Demandent au joueur par où il veut commencer.

- 530 à 550** 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 730).
- 560 à 590** Si CH (choix du niveau) est 2 ou 3, affectent aux variables les valeurs correspondant à la soustraction des dizaines.
- 600 à 630** Si CH est 3, affectent aux variables les valeurs correspondant à la soustraction des centaines.
- 640 à 660** Fin d'opération.
- 670** Envoie à une autre opération.
- 680 à 690** Fin du programme.
- 730 à 860** Opération.  
*Ligne 740* : S'il y a une retenue de l'opération précédente, envoie au sous-programme 1040. La réponse est entrée ligne 770 : si elle est « N » (pour « NON »), le programme va en ligne 900 (retenue).  
*Ligne 790* : Convertit la chaîne REP\$ en sa valeur numérique si l'opération est possible.  
*Ligne 800* : Si la réponse est bonne, envoie en 830.  
*Lignes 810 à 820* : Réponse fausse. Envoie au sous-programme d'affichage de la table d'addition puis attend une nouvelle réponse (GOSUB 1150).
- 900 à 1000** Retenue. Même déroulement, mais en prenant une unité supérieure : décomposition de la retenue.
- 1040 à 1100** Affichage et explications de la nouvelle opération en cas de retenue dans la précédente soustraction.
- 1140 à 1240** Sous-programme : Affichage de la table d'addition en cas d'erreur de calcul mental.
- 1280 à 1370** Interroge l'enfant si l'opération est impossible ( $A < B$ ).

## 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 (730 à 860) 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 420 à 520 peuvent alors être supprimées.

Si l'on veut éviter l'affichage d'opérations impossibles, il suffit de supprimer les lignes 1250 à 1370 et de modifier la ligne :

```
420 IF A<B THEN 250
```

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 en ligne 240 (IF N<4).

## Adaptations

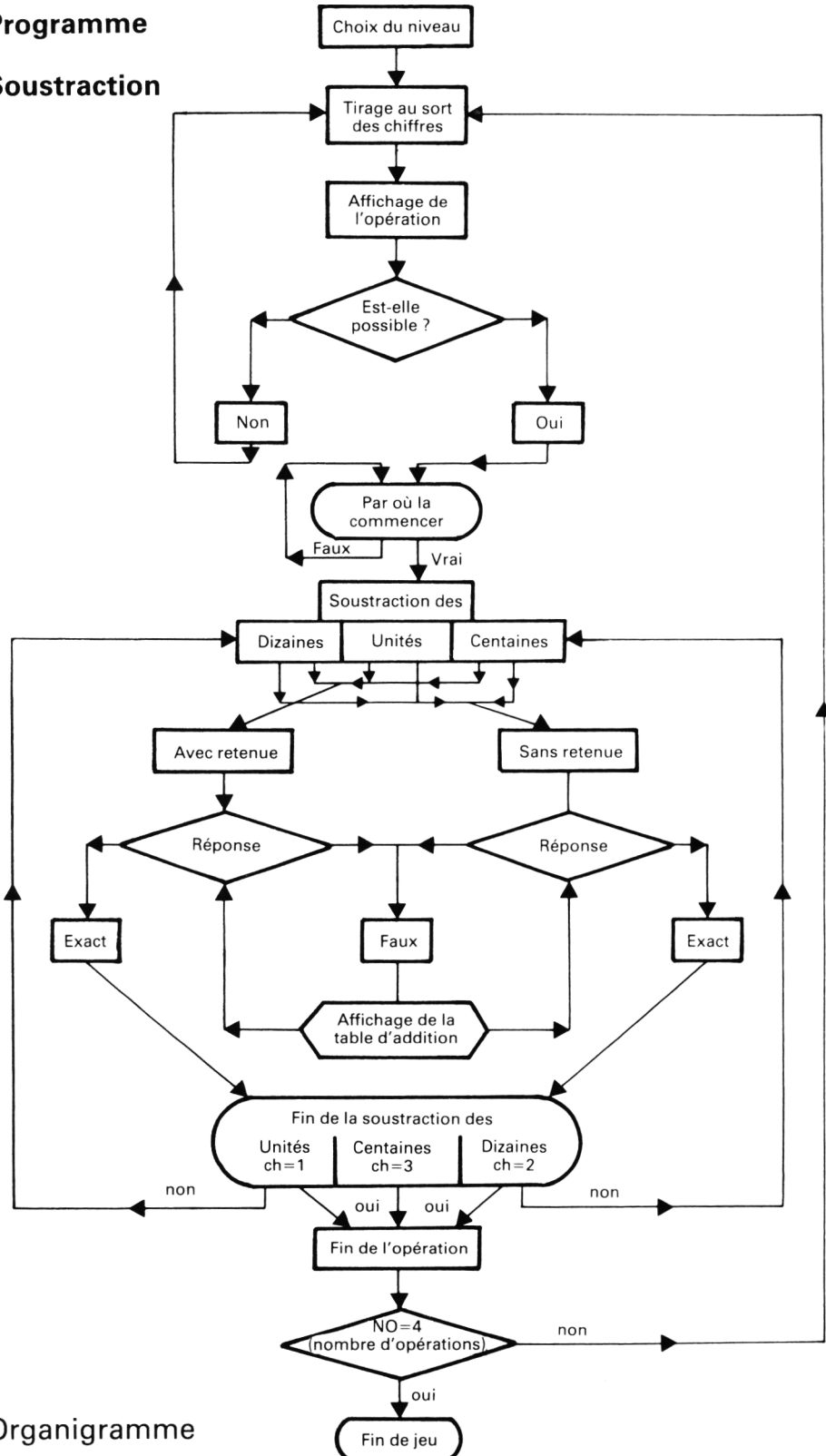
Ce programme peut, lui aussi, être utilisé à l'intérieur d'autres programmes de raisonnement (en supprimant les lignes 530 à 630 et en affectant aux variables les valeurs du programme).

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.

Programme

Soustraction



Organigramme

```

10 REM -----
20 REM ----- SOUSTRACTION -----
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin fev 86 ---
50 REM -----
60 ON BREAK GOSUB 680
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1
110 WINDOW#1,2,39,13,25:PAPER#1,13:PEN#1,0:CLS
#1
120 WINDOW#2,12,28,4,10:PAPER#2,13:PEN#2,0
130 a$(1)="unite":a$(2)="dizaine":a$(3)="centa
ine"
140 LOCATE#1,12,2:PRINT#1,"SOUSTRACTION"
150 PRINT#1:INPUT#1,"Quel est ton prenom ";nom
$:PRINT#1
160 PRINT#1,"Soustractions avec des nombres","
a un, deux ou trois chiffres ?"
170 PRINT#1:PRINT#1,"Indique ton choix ( 1, 2
ou 3) ?"
180 ch$=INKEY$:IF (ch$<>"1") AND (ch$<>"2") AN
D (ch$<>"3") THEN 180
190 REM -----
200 REM -> Module choix des chiffres
210 REM -----
220 RANDOMIZE TIME/1000
230 ch=VAL(ch$)
240 IF n=4 THEN 680
250 ON ch GOTO 280,270,260
260 ac=INT(RND*10):bc=INT(RND*10)
270 ad=INT(RND*10):bd=INT(RND*10)
280 au=INT(RND*10):bu=INT(RND*10)
290 a=ac*100+ad*10+au
300 b=bc*100+bd*10+bu
310 REM -----
320 REM -> Module affichage operation
330 REM -----
340 CLS#2:LOCATE#2,4,2:PRINT#2,ac;" ";ad;" "
;au
350 IF ch$="1" THEN LOCATE#2,4,2:PRINT#2,SPACE
$(7)
360 IF ch$="2" THEN LOCATE#2,4,2:PRINT#2,SPACE
$(5)
370 LOCATE#2,1,3:PRINT#2,"-"
380 LOCATE#2,4,4:PRINT#2,bc;" ";bd;" ";bu
390 IF ch$="1" THEN LOCATE#2,4,4:PRINT#2,SPACE
$(7)

```

```

400 IF ch#="2" THEN LOCATE#2,4,4:PRINT#2,SPACE
#(5)
410 LOCATE#2,2,5:PRINT#2,"-----"
420 IF a<b THEN 1280
430 ret=0
440 IF ch=1 THEN 530
450 CLS#1:PRINT#1:PRINT#1,"Par quels chiffres
commences-tu","ta soustraction ?"
460 PRINT#1:PRINT#1,"Les unites.....1"
470 PRINT#1,"Les dizaines...2"
480 IF ch=3 THEN PRINT#1,"Les centaines..3"
490 PRINT#1:INPUT#1,"Ton choix ";deb
500 IF deb=1 THEN 530
510 PRINT#1,"Tu te trompes, ";nom#;" reflechis
encore"
520 FOR i=1 TO 1000:NEXT i:GOTO 450
530 n1=au:n2=bu:x=1
540 t#=a#(1):v#=a#(2)
550 GOSUB 730
560 IF ch=1 THEN 640
570 n1=ad:n2=bd:x=2
580 t#=a#(2):v#=a#(3)
590 GOSUB 730
600 IF ch=2 THEN 640
610 n1=ac:n2=bc:x=3
620 t#=a#(3)
630 GOSUB 730
640 PRINT#1:PRINT#1,"Bravo, ";nom#
650 n=n+1
660 FOR i=1 TO 2500:NEXT i
670 GOTO 240
680 CLS:PRINT "C'est termine, ";nom#:PRINT"Au
revoir"
690 END
700 REM -----
710 REM -> Module soustraction
720 REM -----
730 CLS#1:PRINT#1
740 IF ret=1 THEN GOSUB 1040
750 PRINT#1,"Que font";n1;t#;"(s) -";n2;t#;"(s
)"
760 PRINT#1:PRINT#1,"Si tu ne peux pas faire c
ette","soustraction, tapes -N-"
770 PRINT#1:INPUT#1,rep#:rep#=UPPER$(rep#)
780 IF rep#="N" AND n1<n2 THEN 900
790 r=VAL(rep#)
800 IF r=n1-n2 THEN 830
810 GOSUB 1150

```

```

820 CLS#1:GOTO 750
830 PRINT#1:PRINT#1,"C'est exact, ";nom#
840 LOCATE#2,19-5*x,6:PRINT#2,r
850 FOR i=1 TO 2500:NEXT i
860 RETURN
870 REM -----
880 REM -> Module retenue
890 REM -----
900 CLS#1:PRINT#1
910 PRINT#1,"Il faut donc prendre une ";v#,"et
  l'ajouter aux";n1;t#;"(s)"
920 PRINT#1,"Cela fait";n1+10;t#;"(s)"
930 ret=1
940 LOCATE#2,19-5*x,2:PRINT#2,"1"
950 FOR i=1 TO 1500:NEXT i
960 PRINT#1:PRINT#1,"Que font";n1+10;"-";n2;"="
  ";
970 INPUT#1,r
980 IF r=n1+10-n2 THEN 830
990 GOSUB 1150
1000 GOTO 960
1010 REM -----
1020 REM -> Module retenue precedente
1030 REM -----
1040 PRINT#1,"Nous avons utilise une ";t#,"pou
  r effectuer la soustraction","precedente"
1050 PRINT#1,"L'operation a faire maintenant",
  "est donc";n1;"-";n2+1
1060 n2=n2+1:ret=0
1070 LOCATE#2,18-5*x,4:PRINT#2,"1+"
1080 FOR i=1 TO 6000:NEXT i
1090 CLS#1:PRINT#1
1100 RETURN
1110 REM -----
1120 REM -> Module table d'addition
1130 REM -----
1140 CLS#1
1150 PRINT#1,"Tu te trompes, ";nom#,"relis ta
  table d'addition"
1160 FOR i=1 TO 1000:NEXT i
1170 CLS#1
1180 FOR i=0 TO 10
1190 PRINT#1,n2;"+";i;"=";n2+i
1200 NEXT i
1210 PRINT#1,"As-tu trouve la bonne reponse ";
  nom#;" ?"
1220 PRINT#1,"Appuie sur une touche"
1230 i#=INKEY#:IF i#="" THEN 1230
1240 RETURN

```

```
1250 REM -----
1260 REM -> soustraction impossible
1270 REM -----
1280 CLS#1:PRINT#1
1290 PRINT#1,"Est-ce que tu peux faire","cette
  operation?":PRINT#1
1300 INPUT#1,"Oui -> 1 / Non -> 2 ";t
1310 IF t=2 THEN 1350
1320 PRINT#1:PRINT#1,"Attention ";nom#;" regard
de bien","les deux nombres","et reflechis enco
re"
1330 FOR i=1 TO 3500:NEXT i
1340 GOTO 1280
1350 PRINT#1:PRINT#1,"Tu as raison, ";nom#;" n
ous allons","donc faire une autre operation"
1360 FOR i=1 TO 3500:NEXT i
1370 GOTO 250
```



# 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 deux ou trois chiffres, mais il peut être multiplié par un nombre B formé de un, deux, trois chiffres.

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

AMSTRAD CPC propose d'abord de multiplier A par le chiffre des unités de B. 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, AMSTRAD CPC 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 pour obtenir la réponse finale.

A chaque opération, l'enfant peut arrêter ou poursuivre le programme avec une autre opération.

## Commentaires

<b>100 à 140</b>	Initialisation, présentation et choix du niveau.
<b>180 à 240</b>	Tirage au sort des nombres. La valeur du nombre de chiffres de chacun des deux nombres est fonction du niveau choisi.
<b>280 à 320</b>	Affichage de l'opération.
<b>360 à 540</b>	Multiplications successives. Les deux boucles (i=1 to nb2) et (k=1 to nb1) assurent toutes les

possibilités de la décomposition en petites multiplications. En cas d'erreur la table de multiplication est appelée.

- 580 à 800** Une fois les multiplications faites, il s'agit d'additionner pour obtenir le résultat final. Les résultats successifs ont été conservés dans un tableau à deux dimensions :  $r(i, k)$ .
- 840 à 950** Fin du programme. L'Amstrad demande si le joueur veut poursuivre le programme, ou s'il désire s'arrêter.
- 990 à 1070** Sous-programme. Utilisé en cas d'erreur, il affiche la table de multiplication pour permettre à l'enfant de voir son erreur. Une nouvelle réponse sera demandée.
- 1110 à 1190** Sous-programme. Utilisé lors d'une retenue précédente, il explique qu'il faut ajouter cette retenue au nombre suivant.

## Particularités techniques

L'affichage est entièrement simulé par des LOCATE, en effet il est préférable de toujours conserver la multiplication à l'écran. Les lignes de l'écran 20 à 25 servent uniquement pour les commentaires.

Les multiplications simples sont réalisées dans deux boucles imbriquées. De cette manière, il suffit de jouer sur les indices du tableau à deux dimensions pour conserver les résultats.

## 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 côte à côte, comme la table de multiplication, pour des raisons de place sur l'écran).

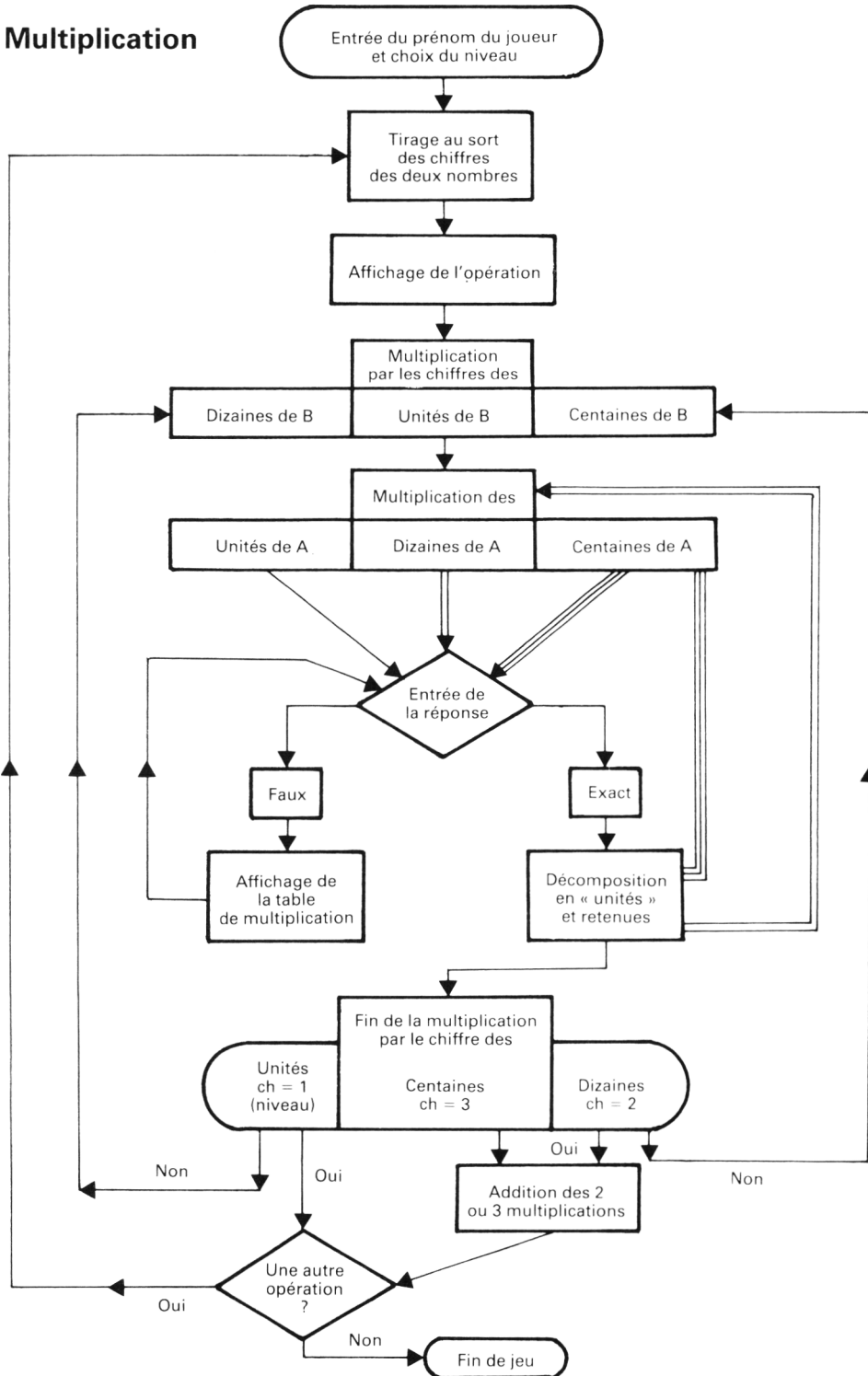
## Adaptations

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.

**Programme**  
**Multiplication**



Organigramme

```

10 REM -----
20 REM ---- MULTIPLICATION ----
30 REM --- D.Nielsen aou 85 ---
40 REM --- Augustin mar 86 ---
50 REM -----
60 ON BREAK GOSUB 950
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:CLS:PRINT "MULTIPLICATION"
110 LOCATE 1,10:PRINT "Multiplication de nombre
de 2 ou 3","chiffres par un nombre","de 1 a
3 chiffres"
120 PRINT:PRINT "Quel niveau choisis-tu ?","(1
...facile a 5...difficile)"
130 i#=INKEY#:IF i#="" THEN 130
140 ch=VAL(i#):IF ch<1 OR ch>5 THEN 130
150 REM -----
160 REM -> Tirage au sort des nombres
170 REM -----
180 IF ch=1 THEN nb1=2:nb2=1
190 IF ch=2 THEN nb1=3:nb2=1
200 IF ch=3 THEN nb1=2:nb2=2
210 IF ch=4 THEN nb1=3:nb2=2
220 IF ch=5 THEN nb1=3:nb2=3
230 FOR i=1 TO nb1:a(i)=INT(RND*9)+1:NEXT i
240 FOR i=1 TO nb2:b(i)=INT(RND*9)+1:NEXT i
250 REM -----
260 REM -> Affichage de l'operation
270 REM -----
280 CLS
290 FOR i=1 TO nb1:LOCATE 37-(i*6),2:PRINT a(i)
):NEXT i
300 FOR i=1 TO nb2:LOCATE 37-(i*6),5:PRINT b(i)
):NEXT i
310 LOCATE 14,5:PRINT "x"
320 FOR i=1 TO 22:LOCATE 35-i,6:PRINT "-":NEXT
i
330 REM -----
340 REM -> Multiplications
350 REM -----
360 FOR i=1 TO nb2
370 FOR k=1 TO nb1
380 FOR j=20 TO 25:LOCATE 1,j:PRINT SPACE$(39)
:NEXT j
390 LOCATE 1,20:PRINT "Tu vas multiplier";a(k)
;"par";b(i)
400 PRINT "Que font";a(k);"x";b(i);
410 INPUT rep

```

```

420 IF rep=a(k)*b(i) THEN 440
430 GOSUB 990:GOTO 380
440 IF rt>0 THEN GOSUB 1110
450 r(i,k+i)=rep MOD 10:rt=rep\10
460 LOCATE 43-(i*6)-(k*6),7+(i*2):PRINT r(i,k+
i)
470 IF rt>0 AND nb1>k THEN LOCATE 1,23:PRINT "
nous ecrivons";r(i,k+i);"dans l'operation et g
ardons";rt,"en retenue":LOCATE 32-(k*6),1:PRIN
T rt
480 IF rt>0 AND k=nb1 THEN LOCATE 37-(k*6)-(i*
6),7+(i*2):PRINT rt:r(i,k+i+1)=rt:rt=0
490 FOR j=1 TO 3500:NEXT j
500 NEXT k
510 LOCATE 1,1:PRINT SPACE$(39)
520 NEXT i
530 FOR t=20 TO 25:LOCATE 1,t:PRINT SPACE$(39)
:NEXT t
540 IF ch<3 THEN 840 ELSE LOCATE 1,20:PRINT "I
l faut maintenant additionner","ces multiplica
tions"
550 REM -----
560 REM -> Additions
570 REM -----
580 LOCATE 31,16:PRINT r(1,2)
590 LOCATE 1,22:PRINT "Nous allons d'abord pos
er";r(1,2)
600 FOR t=1 TO 3500:NEXT t
610 FOR k=3 TO nb1+nb2+1
620 FOR t=20 TO 25:LOCATE 1,t:PRINT SPACE$(39)
:NEXT t:LOCATE 1,20
630 rtad=0:PRINT "que font";
640 FOR i=1 TO nb2
650 IF i>1 THEN PRINT "+";
660 rtad=rtad+r(i,k):PRINT r(i,k);
670 NEXT i
680 IF rtd>0 THEN rtad=rtad+rtd:PRINT "+";rtd;
"de retenue ";
690 IF rtad=0 THEN 800
700 INPUT rep2
710 IF rep2=rtad THEN 750
720 PRINT "Tu te trompes, reflechis"
730 FOR j=1 TO 3500:NEXT j
740 GOTO 620
750 IF rtad<10 THEN rtd=0:GOTO 780
760 rtd=rtad\10:rtad=rtad MOD 10
770 PRINT "nous ecrivons";rtad;"et nous reteno
ns";rtd;"pour l'addition suivante"
780 LOCATE 43-(k*6),16:PRINT rtad

```

```

790 FOR j=1 TO 3500:NEXT j
800 NEXT k
810 REM -----
820 REM -> Suite ou fin de programme
830 REM -----
840 CLS:PRINT "Cette operation est terminee"
850 PRINT "Une autre operation (O/N) ?"
860 i#=INKEY#:IF i#="" THEN 860
870 i#=UPPER$(i#)
880 IF i#="N" THEN 950
890 IF i#<>"O" THEN 860
900 FOR i=1 TO nb2
910 FOR k=2 TO nb1+nb2+1
920 r(i,k)=0
930 NEXT k,i
940 CLS:RUN
950 PRINT "Au revoir":END
960 REM -----
970 REM -> Table de multiplication
980 REM -----
990 FOR j=18 TO 24:LOCATE 1,j:PRINT SPACE$(39)
:NEXT j
1000 LOCATE 1,18:PRINT "tu te trompes, relis c
ette table"
1010 FOR j=1 TO 10
1020 IF j>5 THEN LOCATE 20,13+j ELSE LOCATE 1,
18+j
1030 PRINT j;"x";b(i);"=";j*b(i)
1040 NEXT j
1050 FOR j=1 TO 5000:NEXT j
1060 FOR j=18 TO 24:LOCATE 1,j:PRINT SPACE$(39)
):NEXT j
1070 RETURN
1080 REM -----
1090 REM -> Retenue precedente
1100 REM -----
1110 FOR t=20 TO 24:LOCATE 1,t:PRINT SPACE$(39)
):NEXT
1120 LOCATE 1,20:PRINT "Il faut maintenant ajo
uter la retenue","de la multiplication precede
nte"
1130 PRINT "Que font";rep;"+";rt;
1140 INPUT rep1
1150 IF rep1=rep+rt THEN 1190
1160 PRINT "Tu te trompes, reflechis"
1170 FOR j=1 TO 5000:NEXT j
1180 GOTO 1110
1190 rep=rep1:rt=0:RETURN

```

# 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 du programme

Le joueur va d'abord choisir son niveau de jeu :

- 1 : division d'un nombre à 1 chiffre par un nombre à 1 chiffre
- 2 : division d'un nombre à 2 chiffres par un nombre à 1 chiffre
- 3 : division d'un nombre à 3 chiffres par un nombre à 2 chiffres
- 4 : division d'un nombre à 4 chiffres par un nombre à 2 chiffres
- 5 : division d'un nombre à 5 chiffres par un nombre à 3 chiffres

Une opération va ensuite s'inscrire en haut de l'écran, la partie basse étant réservée pour les questions et les commentaires.

Si le niveau de jeu est supérieur à 1, le joueur va d'abord indiquer quel nombre doit être divisé pour commencer (en fonction du diviseur). En cas d'erreur, le programme détectera le type d'erreur faite et guidera le joueur pour formuler une nouvelle réponse. Ce nombre va s'afficher en bleu, à l'intérieur du dividende.

Les différentes divisions (selon le nombre de chiffres du quotient) vont ensuite s'effectuer selon le même principe. Le joueur doit trouver combien de fois B (le diviseur) est contenu dans le nombre à diviser. Il va proposer un chiffre : l'ordinateur inscrit alors, en bas de l'écran, le produit correspondant et demande au joueur si cela lui convient. Il doit donc comparer ce produit et le nombre à diviser, puis choisir éventuellement de recommencer avec un autre chiffre (produit trop grand, par exemple). Toute erreur est détectée par le programme. Ensuite, l'ordinateur calcule et affiche le reste de la division : de nouveau, il appartient au joueur de comparer ce reste au diviseur et de juger de sa réponse. Si cette division est exacte, les valeurs correspondantes (quotient, produit et reste) s'affichent à leur place dans l'opération en haut de l'écran et si la division n'est pas terminée, la nouvelle opération s'y inscrit (reste de l'opération précédente + chiffre suivant du dividende).

L'ordinateur comptabilise chacun des essais pour trouver le quotient juste et affiche ce « score » à la fin de l'opération.

A la fin de chaque opération, on peut choisir de continuer avec une autre opération (de même niveau) ou d'arrêter.

## Commentaires

- 90 à 130** Initialisation et présentation du programme.  
*Ligne 90* : initialisation de RND, nombre aléatoire, pendant que le joueur lit la présentation.
- 170 à 290** Tirage au sort des nombres.  
*Lignes 180 à 200* : A est le dividende, ch (niveau de jeu) représente le nombre de chiffres de A. Chacun de ces chiffres A(l) est tiré au sort. La ligne 210 tire au sort le même nombre de chiffres pour B (diviseur).  
*Lignes 230 à 250* : la valeur de B est définie, selon ch, niveau de jeu choisi.  
*Ligne 260* : si le dividende est plus petit que le diviseur (quotient entier = 0), on recommence le tirage au sort.  
*Ligne 270* : détermine AF\$, chaîne de caractères égale au nombre de chiffres du diviseur, pour réaliser l'affichage des nombres.  
*Ligne 280* : détermine de la même façon une chaîne de caractères EF\$, formée d'espaces, pour effacer des réponses fausses.  
*Ligne 290* : calcule le quotient de la division et le nombre de chiffres de ce quotient (NCQ).
- 330 à 350** Affichage de l'opération.  
*Ligne 360* : affiche le diviseur B.  
*Ligne 370* : affiche le dividende A, en fonction du nombre de chiffres.
- 410 à 510** Début de l'opération.  
*Ligne 450* : entrée de la réponse.  
*Ligne 460* : commentaire correspondant à une réponse plus petite que le diviseur.  
*Ligne 470* : commentaire correspondant à une réponse trop grande (diviseur \* 10).  
*Ligne 480* : calcul de DV(1) qui représente les l chiffres de gauche du dividende (l = nombre de chiffres du diviseur).  
*Lignes 490 à 500* : selon les valeurs respectives de DV(1) et de B (diviseur), commentaires correspondant à une réponse fausse.  
*Ligne 510* : C1 = nombre de chiffres de la réponse exacte (pour l'affichage).
- 550 à 770** Divisions successives.  
*Ligne 560* : question.

*Ligne 570* : réponse du joueur : seuls les chiffres (0 à 9) sont acceptés comme réponse.

*Ligne 590* : affichage du produit correspondant à la réponse.

*Lignes 600 à 640* : le joueur juge lui-même sa réponse (+ commentaires éventuels).

*Ligne 650* : LC détermine la position d'affichage du produit dans l'opération en haut de l'écran.

*Lignes 660 à 690* : affichage du quotient et de la soustraction.

*Lignes 700 à 750* : affichage du reste de la soustraction : le joueur doit confirmer sa réponse, commentaires selon son choix (reste trop grand, ...).

### 780 à 830

Suite des divisions successives.

*Ligne 780* : si J (nombre de divisions successives) est égal à NCQ (nombre de chiffres du quotient), la division est terminée.

*Lignes 790 à 830* : modification des variables. D(J), nombre à diviser, est formé à partir de R(J-1), reste de la division précédente, et A(ch-T), chiffre suivant du dividende.

### 870 à 950

Fin du programme.

*Ligne 900* : affichage du nombre de réponses du joueur et du nombre minimum de réponses.

*Ligne 910* : propose une autre opération.

### 960

Sous-programme de pause (pour lire un commentaire).

### 970

Sous-programme de commentaire (début d'opération).

### 1010 à 1040

Sous-programme d'effacement des nombres dans l'opération, en cas de réponse fautive ou de changement de réponse.

## Particularités techniques

Pour former le dividende A et le diviseur B, on tire au sort autant de chiffres que le niveau choisi (ch). Le dividende est formé de la suite de ces chiffres (ligne 200), par exemple, les chiffres 2,6,8,3 donneront le nombre 2683. Ainsi, au niveau 4, ce sera un nombre de 4 chiffres. Pour le diviseur, seuls les 3

premiers chiffres tirés au sort seront utilisés, selon le niveau choisi (lignes 230 à 250).

Les chaînes AF\$ et EF\$ seront utilisées pour une bonne présentation de l'affichage des nombres à l'intérieur de l'opération : AF\$ servira de valeur de base pour une instruction PRINT USING (ligne 690), EF\$, formé d'un certain nombre de cases « espace », sera utilisé pour effacer les nombres dans l'opération en cas d'erreur ou de nouveau choix du joueur.

La ligne 290 calcule le nombre de chiffres du quotient qui est calculé dans la variable QUOTIENT. Une variable numérique étant précédée d'un espace vide (s'il s'agit d'un nombre positif), on transforme QUOTIENT en une chaîne de caractères STR\$(QUOTIENT) dont on élimine le premier caractère dans l'instruction MID\$. Le nombre de chiffres du quotient est alors égal au nombre de caractères (LEN) de la chaîne Q\$ ainsi formée.

Chaque opération commence par la recherche du premier nombre à diviser, qui est formé des 2, 3, 4... chiffres de gauche du dividende, selon la valeur du diviseur.

Le test de la réponse du joueur est effectué en fonction des valeurs respectives du diviseur et du nombre à diviser DV(1) : si le diviseur est, par exemple, 345 et le dividende 4578, DV(1) sera 457 car les 3 premiers chiffres du dividende forment un nombre supérieur au diviseur ; si, par contre, le dividende est 27596, DV(1) sera alors 2759. Cette partie du programme est un peu longue, mais elle permet de travailler avec des nombres quelconques.

Les divisions successives sont réalisées à l'intérieur d'un même programme. L'affichage des nombres dans l'opération est réalisé à partir de J (nombre d'opérations réalisées). Le programme vérifie lui-même si l'opération est terminée : ce nombre d'opérations (J) est alors égal au nombre de chiffres du quotient NCQ. NCQ représente aussi, évidemment, le nombre minimum de « coups » pour faire cette opération.

## Modifications

Les valeurs des nombres, obtenues à partir du choix du niveau (voir Particularités techniques) peuvent être modifiées de manière très simple à partir des lignes 170 à 220. On pourrait, en particulier, tirer le diviseur et le dividende, dans leur entier :  $(\text{INT}(\text{RND} * 99) + 1)$  donnera, par exemple, un nombre entre 1 et 99. Mais il faudra ensuite rechercher chacun des chiffres du

dividende, car ils sont nécessaires pour la suite du programme :  $A \text{ MOD } (10^I)$  donnera ces chiffres, dans une boucle où  $I$  représente le nombre de chiffres de ce dividende.

La structure du programme permet en effet de travailler avec des nombres quelconques. Seule, la place sur l'écran, pour afficher l'opération, limitera vos choix.

Ce programme pourrait être transformé en un programme plus complet sur la technique de la division en ne fournissant pas les résultats des opérations intermédiaires (multiplication et soustraction). Mais il faudrait alors que l'élève travaille avec un cahier sous la main, ou bien qu'un sous-programme de « calculatrice » soit ajouté dans le programme, car tout ne peut être effectué en calcul mental.

On pourrait envisager une suite automatique des opérations en fonction du rapport score/nombre minimum d'opérations

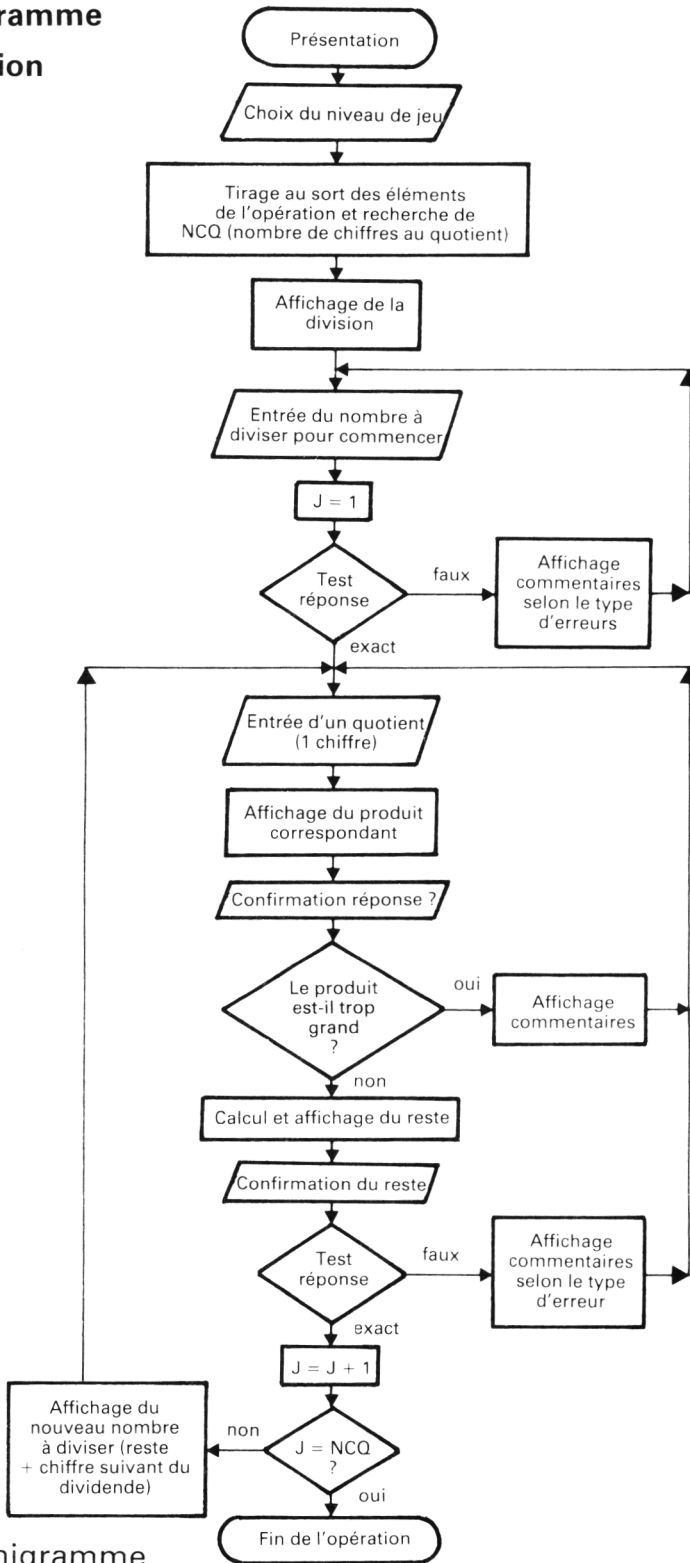
```
950 IF REP=NCQ+1 THEN N=N+1 : IF N >5 THEN N=5
952 IF REP>NCQ+4 THEN N=N-1 : IF N<1 THEN N=1
954 GOTO 10
```

## Adaptations

Le programme de divisions intermédiaires peut travailler avec n'importe quel nombre. On peut donc rechercher la division exacte, ou du moins avec deux ou trois chiffres après la virgule. Il suffit de déplacer l'affichage du dividende à l'extrême gauche de l'écran, pour laisser la place aux décimaux.

En modifiant les valeurs des tirages au sort des chiffres du diviseur et du dividende, des divisions de nombres décimaux peuvent aussi être effectuées à l'aide de ce programme.

**Programme**  
**Division**



Organigramme

```

10 REM -----
20 REM ----- DIVISION -----
30 REM --- D.Nielsen avr 85 ---
40 REM --- Augustin mar 86 ---
50 REM -----
60 REM -----
70 REM -> Module presentation
80 REM -----
90 MODE 1:CLS:RANDOMIZE TIME/1000:PRINT "DIVIS
ION"
100 LOCATE 1,10:PRINT "Division de nombres par
un diviseur","de 1 a 3 chiffres"
110 PRINT:PRINT "Quel niveau choisis-tu ?","(1
...facile a 5...difficile)"
120 i#=INKEY#:IF i#="" THEN 120
130 ch=VAL(i#):IF ch<1 OR ch>5 THEN 120
140 REM -----
150 REM -> Tirage au sort des nombres
160 REM -----
170 rep=0:a=0:b=0
180 FOR i=1 TO ch
190 a(i)=INT(RND*9)+1
200 IF i=1 THEN a=a(i) ELSE a=a+a(i)*10^(i-1)
210 b(i)=INT(RND*9)+1
220 NEXT i
230 IF ch<3 THEN b=b(1):div=1
240 IF ch>2 THEN b=b(1)+b(2)*10:div=2
250 IF ch=5 THEN b=b(1)+b(2)*10+b(3)*100:div=3
260 IF a<b THEN 180
270 af#="#":FOR i=1 TO div:af#=af#+"#":NEXT i
280 ef#=CHR$(32):FOR i=1 TO div:ef#=ef#+CHR$(3
2):NEXT i
290 quotient=INT(a/b):q#=MID$(STR$(quotient),2
):ncq=LEN(q#)
300 REM -----
310 REM -> Affichage de l'operation
320 REM -----
330 CLS:j=1
340 FOR y=1 TO 14:LOCATE 17,y:PRINT CHR$(143):
NEXT y
350 FOR x=18 TO 26:LOCATE x,4:PRINT CHR$(143):
NEXT x
360 LOCATE 21,2:PRINT MID$(STR$(b),2)
370 l=17-(LEN(STR$(a))*2):LOCATE l,2:PRINT a
380 REM -----
390 REM -> Debut de la division
400 REM -----
410 IF ch=1 THEN d(j)=a:GOTO 550
420 IF ch=2 AND a(i)<b THEN d(j)=a:GOTO 550

```

```

430 FOR k=20 TO 25:LOCATE 1,k:PRINT SPACE$(39)
:NEXT k
440 LOCATE 1,20:PRINT "Quel nombre vas-tu divi
ser par";b;"pour commencer ";
450 INPUT d(j):rep=rep+1
460 IF d(j)<b THEN PRINT "Non, ce nombre est t
rop petit","pour etre divise par";b:GOSUB 960:
GOTO 430
470 IF d(j)>b*10 THEN PRINT "Non, ton quotient
sera superieur a 10":GOSUB 960:GOTO 430
480 FOR i=1 TO div:dv(1)=a(ch+1-i)+dv(1)*10:NE
XT i
490 IF dv(1)>b AND d(j)<>dv(1) THEN GOSUB 970:
GOTO 430
500 IF dv(1)<b AND d(j)<>dv(1)*10+a(ch-div) TH
EN GOSUB 970:GOTO 430
510 c1=LEN(MID$(STR$(d(j)),2))
520 REM -----
530 REM -> Division successives
540 REM -----
550 FOR k=20 TO 25:LOCATE 1,k:PRINT SPACE$(39)
:NEXT k
560 LOCATE 1,20:PRINT "Combien de fois";b;"peu
x-tu enlever de";d(j);" ?"
570 i$=INKEY$:IF i$="" THEN 570
580 rep=rep+1:q(j)=VAL(i$):IF ASC(i$)<48 OR AS
C(i$)>57 THEN 570
590 PRINT b;"x";q(j);"=";q(j)*b
600 PRINT "Ce chiffre te convient-il (O/N)"
610 i$=INKEY$:IF i$="" THEN 610
620 i$=UPPER$(i$):IF i$="N" THEN PRINT "Alors
on recommence":GOSUB 960:GOTO 550
630 IF i$<>"O" THEN 610
640 r(j)=d(j)-q(j)*b:IF r(j)<0 THEN PRINT "Att
ention ce nombre est trop grand":GOSUB 960:GOT
O 550
650 IF q(j)*b>10^div AND j=1 THEN lc=1+j*2 ELS
E lc=1-1+j*2
660 LOCATE lc-2,j*4:PRINT "-";:PRINT USING af$
;q(j)*b
670 LOCATE 19+j*2,6:PRINT MID$(STR$(q(j)),2)
680 LOCATE lc-1,4*j+2:PRINT USING af$;r(j)
690 FOR k=20 TO 25:LOCATE 1,k:PRINT SPACE$(39)
:NEXT k
700 LOCATE 1,20:PRINT d(j);"-";q(j)*b;"=";r(j)
710 PRINT "Regarde le reste"
720 PRINT "Ton quotient est-il exact (O/N) ?"
730 i$=UPPER$(INKEY$):IF i$="" THEN 730
740 IF i$<>"O" AND i$<>"N" THEN 730

```

```

750 IF i#="0" AND r(j)>b THEN PRINT "Tu te trompes, ce reste", " est plus grand que le diviseur", "Il faut choisir un autre quotient":GOSUB 960:GOSUB 1010:GOTO 550
760 IF i#="N" AND r(j)<b THEN PRINT "Et pourtant c'est le bon quotient !":GOSUB 960
770 IF i#="N" AND r(j)>b THEN PRINT "Tu as raison, alors on recommence":GOSUB 960:GOSUB 1010:GOTO 550
780 IF j=ncq THEN 870
790 t=c1+j-1
800 j=j+1
810 d(j)=r(j-1)*10+a(ch-t)
820 LOCATE 1c,j*4-2:PRINT USING af#;d(j)
830 GOTO 550
840 REM -----
850 REM -> Fin de programme
860 REM -----
870 CLS:PRINT "Operation terminee"
880 IF ch=1 THEN 910
890 IF ch=2 AND a(1)<b THEN 910
900 IF rep=ncq+1 THEN PRINT "Bravo, tu as ete le plus rapide possible" ELSE PRINT "Tu as donne";rep;"reponses":PRINT "C'etait faisable en";ncq+1;"coups"
910 PRINT:PRINT "Une autre operation (O/N) ?"
920 i#=UPPER$(INKEY#):IF i#="" THEN 920
930 IF i#<>"O" AND i#<>"N" THEN 920
940 IF i#="N" THEN PRINT "Au revoir":END
950 RUN
960 FOR pause=1 TO 5000:NEXT pause:RETURN
970 PRINT "Ta reponse ne correspond pas", "au nombre a diviser":GOSUB 960:RETURN
980 REM -----
990 REM -> Effacement des nombres
1000 REM -----
1010 LOCATE 1c-2,j*4:PRINT ef#;" "
1020 LOCATE 1c-1,j*4+2:PRINT ef#
1030 LOCATE 19+j*2,6:PRINT " "
1040 RETURN

```

# 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 AMSTRAD CPC 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>100 à 170</b>	Définition du dessin de l'avion.
<b>180 à 360</b>	Présentation du jeu sur l'écran, avec une animation.
<b>400</b>	Choix de la bonne altitude par l'AMSTRAD CPC.
<b>410 à 430</b>	Affichage de l'avion de chasse (en jaune).
<b>440 à 500</b>	Test de l'altitude proposée par le joueur et affichage de l'avion ravitailleur.
<b>510 à 580</b>	Message d'erreur. Compteur de coups restant à jouer (si plus de 10 réponses).
<b>620 à 710</b>	Affichage réponse exacte.
<b>750 à 830</b>	Affichage « perdu ».
<b>860 à 890</b>	Proposent de rejouer ou d'arrêter.

## Particularités techniques

Les deux avions sont dessinés à partir de caractères personnalisés (lignes 110 à 170).

La ligne 690 : il faut bien lire un caractère “\_” (souligné : SHIFT + “0”).

## Modifications

L’avion de chasse peut être placé à n’importe quelle altitude entre 1 et 200, mais la séquence de nombres aléatoires ne sera jamais la même, elle n’est pas initialisée lors d’une instruction RANDOMIZE TIME/1000.

Lignes 540 à 560 : le nombre de réponses maximal peut être ajusté à l’âge du joueur.

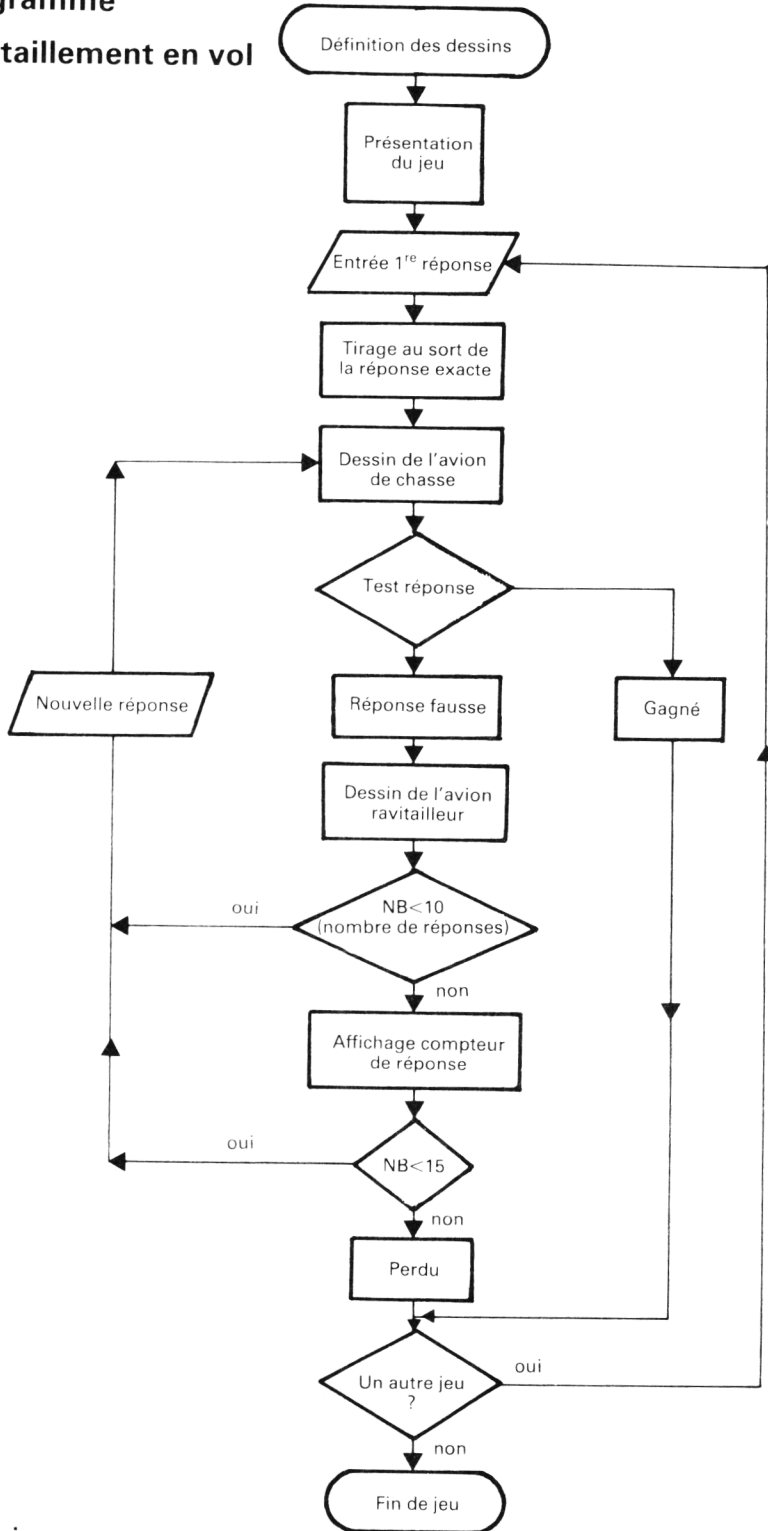
## Adaptations

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

La réponse peut alors être saisie directement par une instruction A\$=INKEY\$ à la ligne 440.

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

**Programme**  
**Ravitaillement en vol**



Organigramme

```

10 REM -----
20 REM -- RAVITAILLEMENT EN VOL --
30 REM ---- D.Nielsen   nov 83 ----
40 REM ---- Augustin   jan 86 ----
50 REM -----
60 RANDOMIZE TIME/1000
70 REM -----
80 REM -> Module initialisation
90 REM -----
100 MODE 1:CLS:PAPER 0:PEN 1
110 SYMBOL AFTER 239
120 SYMBOL 240,0,0,0,1,3,63,85,255
130 SYMBOL 241,0,96,192,193,195,255,255,248
140 SYMBOL 242,63,3,1,0,0,0,0,0
150 SYMBOL 243,192,192,192,192,96,0,0,0
160 avion$(1)=CHR$(240)+CHR$(241)
170 avion$(2)=CHR$(242)+CHR$(243)
180 SOUND 1,0,2300,15,,31
190 TAG
200 FOR i=300 TO 50 STEP -2
210 MOVE 2*i,390:PRINT avion$(1);" RAVITAILLEM
ENT EN VOL ";
220 MOVE 2*i,375:PRINT avion$(2);" ";
230 NEXT i
240 TAGOFF
250 LOCATE 1,10:PRINT " Ravitailler un avion e
n vol"," c'est lui donner du carburant"," san
s qu'il soit obligé d'atterrir"
260 PRINT:PRINT " Pour cela, il faut que l'avi
on"," ravitailleur soit a la meme"," altitude
que l'avion de chasse"
270 LOCATE 3,25:PRINT "APPUIE SUR UNE TOUCHE P
OUR CONTINUER"
280 i$=INKEY$:IF i$="" THEN 280
290 CLS:LOCATE 1,5
300 PRINT " L'altitude est comprise entre"," 0
et 200 metres"
310 PRINT:PRINT " Tu pilotes l'avion ravitaill
eur"," et tu dois donner ton altitude"
320 PRINT:PRINT " Si ton altitude est trop ele
ve"," ton avion monte au dessus de"," l'avion
de chasse"
330 PRINT:PRINT " Si ton altitude est trop bas
se"," ton avion descend au dessous de "," l'av
ion de chasse"
340 PRINT:PRINT " Tu dois trouver l'altitude e
xacte"
350 LOCATE 3,25:PRINT "APPUIE SUR UNE TOUCHE P
OUR CONTINUER"

```

```

360 i#=INKEY#:IF i#="" THEN 360
370 REM -----
380 REM -> Module jeu
390 REM -----
400 c=0:b=INT(RND*200)+1
410 CLS
420 LOCATE 12,8:PRINT avion$(1)
430 LOCATE 12,9:PRINT avion$(2)
440 LOCATE 2,20:INPUT "Quelle altitude propose
s-tu ";a
450 IF a<1 OR a>200 THEN 410
460 c=c+1
470 IF b<a THEN x=5 ELSE x=16
480 IF a=b THEN 620
490 PEN 2:LOCATE 20,x:PRINT avion$(1)
500 LOCATE 20,x+1:PRINT avion$(2):PEN 1
510 LOCATE 1,22
520 PRINT " Ce n'est pas la bonne altitude"
530 SOUND 1,200,20,15
540 IF c>10 AND c<>14 THEN PRINT " Depeches to
i, tu n'as plus que";15-c," altitudes a propos
er"
550 IF c=14 THEN PRINT " C'est ta derniere cha
nce"
560 IF c=15 THEN 750
570 FOR i=1 TO 3500:NEXT i
580 GOTO 410
590 REM -----
600 REM -> Module gagne
610 REM -----
620 FOR i=20 TO 24
630 LOCATE 1,i:PRINT SPACE$(39)
640 NEXT i
650 LOCATE 1,20:PRINT " Bravo, c'est la bonne
altitude", " Tu peut ravitailler l'avion"
660 PRINT:PRINT " Tu as gagne en";c;"reponses"
670 PEN 2:LOCATE 20,8:PRINT avion$(1)
680 LOCATE 20,9:PRINT avion$(2):PEN 1
690 FOR i=14 TO 19:LOCATE i,8:PRINT "_":NEXT i
700 FOR i=1 TO 7000:NEXT i
710 GOTO 870
720 REM -----
730 REM -> Module perdu
740 REM -----
750 CLS:PRINT " Trop tard", " L'avion n'a plus
de carburant"
760 FOR i=1 TO 2000:NEXT i
770 FOR i=1 TO 50
780 SOUND 1,i,1,15

```

```
790 NEXT i
800 SOUND 1,0,300,15,1,,16
810 FOR i=1 TO 10
820 PAPER 0:CLS:PAPER 10:CLS
830 NEXT i
840 REM -----
850 REM -> Module fin de partie
860 REM -----
870 MODE 1:PAPER 0:PEN 1:CLS:PRINT " Une autre
partie ( o / n )"
880 i$=INKEY$:IF i$="" THEN 880
890 IF UPPER$(i$)="O" THEN 400 ELSE END
```



# 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 (les wagons sont ici représentés par des fenêtres).

## Niveau

En fonction des questions utilisées dans ces exercices.

Le système de réponse n'utilisant que les deux touches "1" et "0" 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 tout domaine (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 (AMSTRAD CPC 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 "0" et "1", 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, AMSTRAD CPC 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>100 à 220</b>	Initialisation des fenêtres et présentation du programme.
<b>260 à 560</b>	Entrée des données.
<b>570 à 660</b>	Présentation du programme pour l'enfant.
<b>700 à 890</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 (GOSUB 940).
<b>900</b>	Fin de programme.
<b>940</b>	Initialisation des positions relatives des wagons et locomotive.
<b>950 à 1040</b>	Création des fenêtres et affichage des mots dans celles-ci.
<b>1050</b>	$t=0$ correspond à l'initialisation de la minuterie.
<b>1060 à 1100</b>	Lecture d'un déplacement, et chargement des paramètres des fenêtres.
<b>1110 à 1190</b>	Tirage aléatoire d'un déplacement de la locomotive.
<b>1230 à 1270</b>	Regarde si un wagon est accroché, et si c'est le cas analyse la réponse.

## Modifications

Une temporisation du déplacement des wagons (ligne 1060) peut être ajoutée afin qu'ils avancent moins vite (variable  $t$ ).

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 ?).

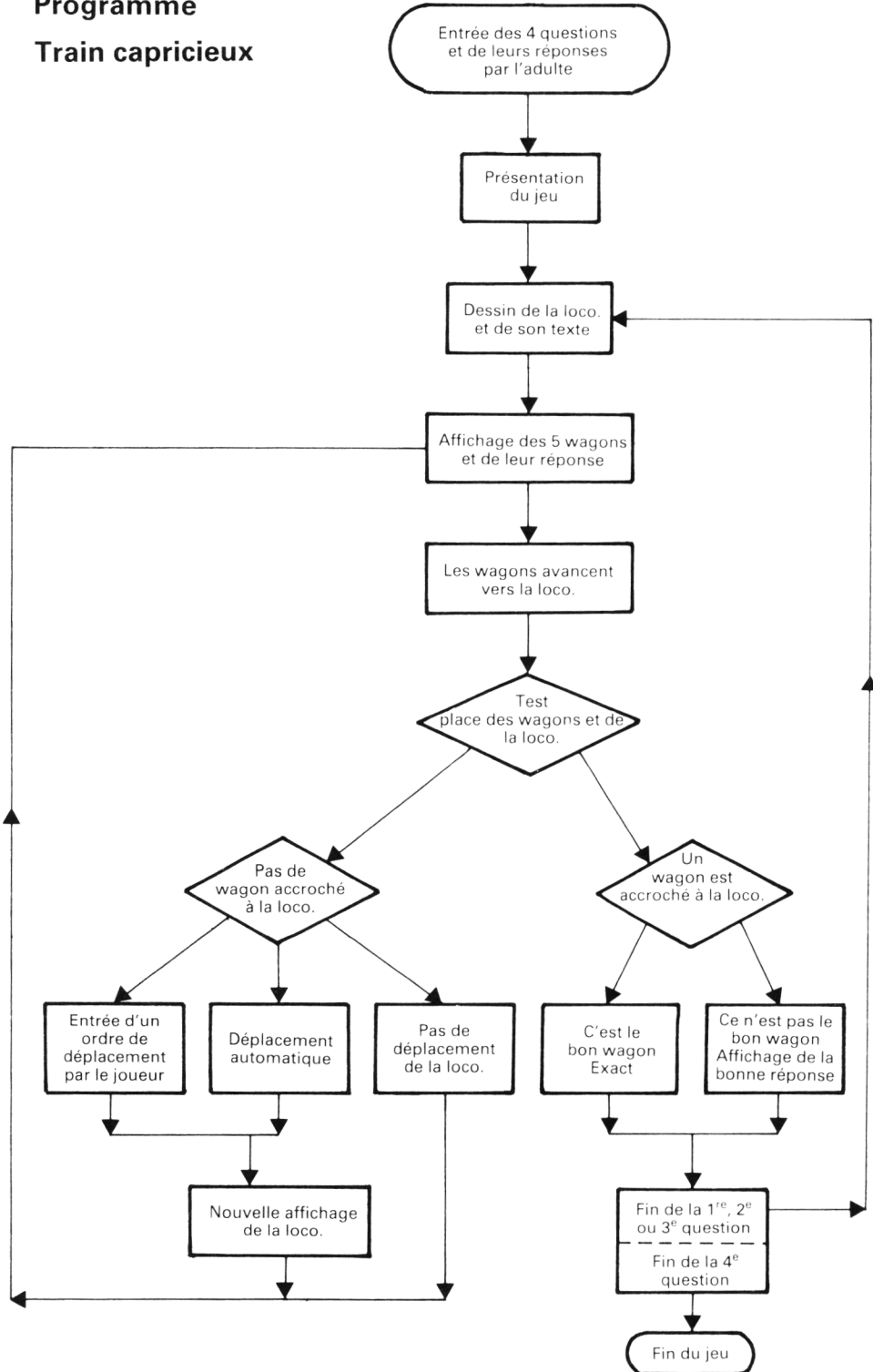
*Géographie*

- 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.

**Programme**  
**Train capricieux**



Organigramme

```

10 REM -----
20 REM ---- TRAIN CAPRICIEUX ----
30 REM --- D.Nielsen nov 83 ----
40 REM --- Augustin fev 86 ----
50 REM -----
60 ON BREAK GOSUB 900
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 13:PEN 0:CLS
110 RANDOMIZE TIME/1000
120 FOR i=1 TO 5
130 j=INT(RND*3)+2
140 WINDOW#i,25,38,27-5*i,29-5*i:PAPER#i,j:CLS#i:PEN#i,j+1
150 NEXT i
160 WINDOW#6,3,9,12,14:PAPER#6,2:PEN#6,3:CLS#6
170 FOR i=1 TO 5:LOCATE#i,2,2:PRINT#i,"CAPRICIEUX":NEXT i
180 LOCATE#6,2,2:PRINT#6,"TRAIN"
190 FOR i=1 TO 300 STEP 10
200 SOUND 1,i,10,15
210 NEXT i
220 FOR i=1 TO 5000 :NEXT i
230 REM -----
240 REM -> Module lecture des phrases
250 REM -----
260 MODE 2:CLS
270 LOCATE 2,2:PRINT "LE TRAIN CAPRICIEUX"
280 LOCATE 2,5:PRINT "Vous allez d'abord entrer les questions et les reponses"," qui vont servir au jeu"
290 LOCATE 10,8:PRINT "1er jeu":PRINT
300 INPUT "Que voulez-vous ecrire dans la locomotive ";a#
310 INPUT "1ere reponse (fausse) ";b#(1)
320 INPUT "2eme reponse (exacte) ";b#(2)
330 INPUT "3eme reponse (fausse) ";b#(3)
340 INPUT "4eme reponse (fausse) ";b#(4)
350 INPUT "5eme reponse (fausse) ";b#(5)
360 CLS:LOCATE 10,8:PRINT "2eme jeu":PRINT
370 INPUT "Que voulez-vous ecrire dans la locomotive ";c#
380 INPUT "1ere reponse (fausse) ";d#(1)
390 INPUT "2eme reponse (fausse) ";d#(2)
400 INPUT "3eme reponse (fausse) ";d#(3)
410 INPUT "4eme reponse (fausse) ";d#(4)
420 INPUT "5eme reponse (exacte) ";d#(5)
430 CLS:LOCATE 10,8:PRINT "3eme jeu":PRINT

```

```

440 INPUT "Que voulez-vous ecrire dans la loco
motive ";e$
450 INPUT "1ere reponse (exacte) ";f$(1)
460 INPUT "2eme reponse (fausse) ";f$(2)
470 INPUT "3eme reponse (fausse) ";f$(3)
480 INPUT "4eme reponse (fausse) ";f$(4)
490 INPUT "5eme reponse (fausse) ";f$(5)
500 CLS:LOCATE 10,8:PRINT "4eme jeu":PRINT
510 INPUT "Que voulez-vous ecrire dans la loco
motive ";g$
520 INPUT "1ere reponse (fausse) ";h$(1)
530 INPUT "2eme reponse (fausse) ";h$(2)
540 INPUT "3eme reponse (fausse) ";h$(3)
550 INPUT "4eme reponse (exacte) ";h$(4)
560 INPUT "5eme reponse (fausse) ";h$(5)
570 MODE 1:CLS
580 LOCATE 2,4:PRINT "TRAIN"
590 LOCATE 2,6:PRINT "Tu vas voir apparaitre",
" une locomotive a gauche de l'ecran"
600 PRINT " et a droite 5 wagons"
610 PRINT:PRINT " Les wagons vont avancer", " v
ers la locomotive"
620 PRINT:PRINT " Tu dois faire monter ou desc
endre", " la locomotive pour accrocher", " le wa
gon qui contient la bonne reponse"
630 PRINT:PRINT " Tape 1 pour monter", " ou 2 p
our descendre"
640 PRINT:PRINT " Attention, la locomotive cha
nge ", " parfois de voie toute seule"
650 LOCATE 2,24:PRINT "APPUIE SUR UNE TOUCHE F
OUR CONTINUER"
660 i$=INKEY$:IF i$="" THEN 660 ELSE CLS
670 REM -----
680 REM -> Module jeu
690 REM -----
700 v$=a$:br=2
710 FOR i=1 TO 5
720 t$(i)=b$(i)
730 NEXT i
740 GOSUB 940
750 v$=c$:br=5
760 FOR i=1 TO 5
770 t$(i)=d$(i)
780 NEXT i
790 GOSUB 940
800 v$=e$:br=1
810 FOR i=1 TO 5
820 t$(i)=f$(i)
830 NEXT i
840 GOSUB 940

```

```

850 v#=g#:br=4
860 FOR i=1 TO 5
870 t#(i)=h#(i)
880 NEXT i
890 GOSUB 940
900 MODE 1:CLS:PRINT "Au revoir":END
910 REM -----
920 REM -> Module affichage
930 REM -----
940 d1=0:d=0
950 CLS
960 FOR i=1 TO 5
970 j=INT(RND*3)+2
980 WINDOW#i,25-d,38-d,27-5*i,29-5*i:PAPER#i,j
::CLS#i:PEN#i,j+1
990 NEXT i
1000 WINDOW#6,3,9,12+d1,14+d1:PAPER#6,2:PEN#6,
3:CLS#6
1010 LOCATE#6,2,2:PRINT#6,v#
1020 FOR i=1 TO 5
1030 LOCATE#i,2,2:PRINT#i,t#(i)
1040 NEXT i
1050 t=0
1060 i#=INKEY#:IF i#="" THEN t=t+1:IF t>500 TH
EN 1120 ELSE 1060
1070 IF i#<>"1" AND i#<>"0" THEN SOUND 1,400,2
0,15:GOTO 1060
1080 IF i#"1" THEN d1=d1-5 ELSE d1=d1+5
1090 IF d1<-10 THEN d1=-10
1100 IF d1>10 THEN d1=10
1110 t=0
1120 IF d=15 THEN 1230
1130 d=d+1
1140 SOUND 1,100,10,15
1150 IF INT(RND*10)=5 THEN d1=d1+5
1160 IF INT(RND*10)=4 THEN d1=d1-5
1170 IF d1<-10 THEN d1=-10
1180 IF d1>10 THEN d1=10
1190 GOTO 950
1200 REM -----
1210 REM -> Module wagon accroche
1220 REM -----
1230 FOR i=1 TO 2000:NEXT i
1240 CLS:LOCATE 2,10
1250 IF d1=15-5*br THEN PRINT "C'est exact " E
LSE PRINT "Faux, c'était : ";t#(br)
1260 LOCATE 2,24:PRINT "APPUIE UNE TOUCHE POUR
CONTINUER"
1270 i#=INKEY#:IF i#="" THEN 1270 ELSE RETURN

```

# 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 AMSTRAD CPC.

Après entrée du prénom du joueur, une carte de France se dessine sur l'écran. Dans cette carte, un carré 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 visualisée à l'écran.

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

A la fin du jeu, AMSTRAD CPC 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>60 à 80</b>	Initialisation et présentation.
<b>120 à 350</b>	Coordonnées des différents points de la carte de France, et dessin de la carte de France.
<b>390 à 580</b>	Lecture des villes en DATA et affichage de la liste des villes.
<b>620 à 630</b>	Initialisation de RND (nombre aléatoire).
<b>640 à 700</b>	Choix aléatoire d'une ville et clignotement du carré sur la carte de France.
<b>710</b>	Affichage de la question.
<b>720 à 730</b>	Test de la réponse.
<b>740 à 790</b>	On efface le carré de la carte.
<b>800</b>	Fin de jeu. Affichage du score.

## 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 160), ces derniers étant marqués sur l'écran (ligne 170) puis reliés entre eux en ordre successif par un segment de droite 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 un carré autour de ce point et le faire clignoter pour que l'enfant le voit (lignes 660 à 700).

Les coordonnées des villes et leur nom sont lus (lignes 400 à 450) par une boucle WHILE/WEND 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 permettant plusieurs réponses, en ajoutant :

```
ligne 631 : REP = 0
ligne 711 : REP = REP + 1
ligne 741 : IF REP > 3 THEN 640
```

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 630) 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 « Construction de phrases »).

## Adaptations

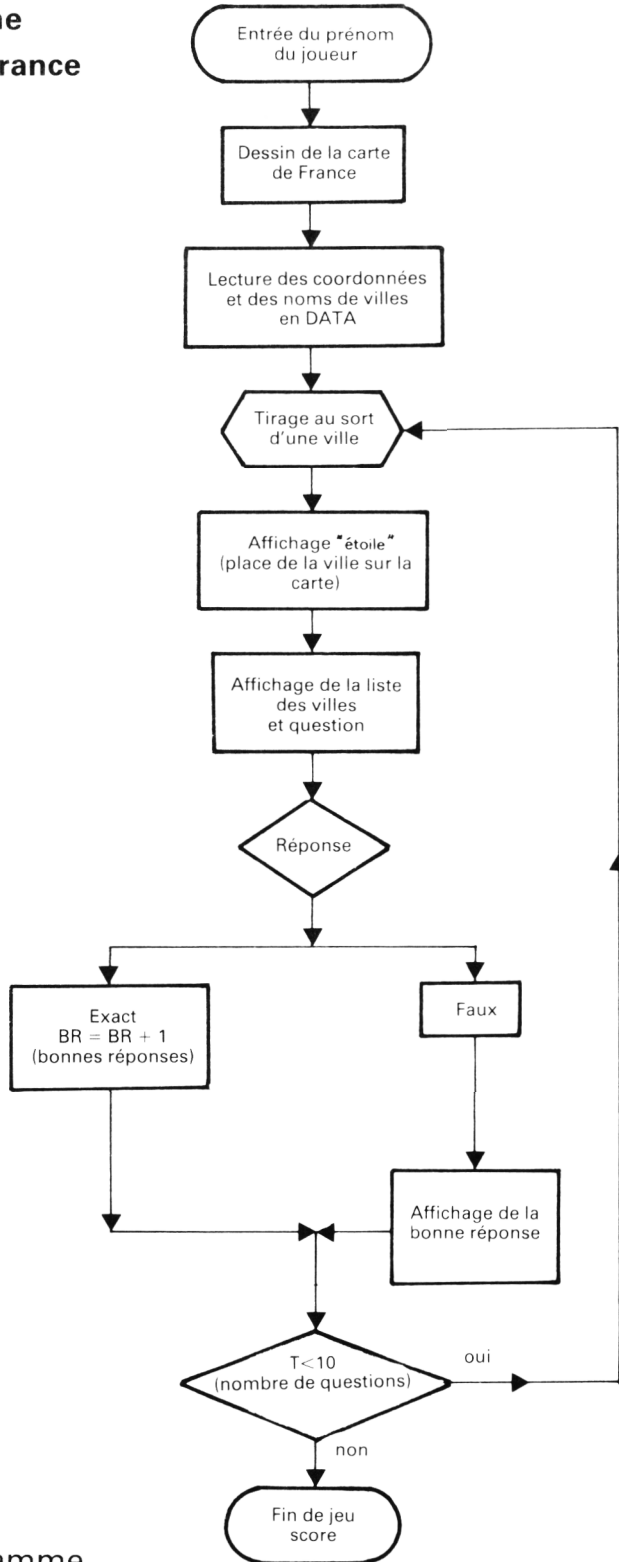
Vous pouvez ajouter d'autres villes au programme, vous pouvez aussi interroger vos enfants et vos amis sur les fleuves, ou les montagnes, ou les ports, etc.

AMSTRAD CPC 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 !

**Programme  
Villes de France**



Organigramme

```

10 REM -----
20 REM ---- VILLES DE FRANCE ----
30 REM --- D.Nielsen oct 83 ---
40 REM --- Augustin jan 86 ---
50 REM -----
60 MODE 1:INK 0,1:INK 1,24
70 LOCATE 10,10:PRINT "GEOGRAPHIE"
80 LOCATE 2,16:INPUT "Quel est ton prenom ";no
m#:nom#=UPPER$(nom#)
90 REM -----
100 REM -> Module Carte de France
110 REM -----
120 CLS
130 RESTORE:c=0
140 PLOT 306,370
150 WHILE c<>48
160 READ x,y
170 DRAWR x,y,1
180 c=c+1
190 WEND
200 DATA 52,-28,70,-24,-18,-38
210 DATA 0,-20,-16,-20,0,-6
220 DATA 8,0,4,6,10,0
230 DATA -8,-14,6,-22,0,-8
240 DATA 12,-8,10,-2,0,-12
250 DATA -22,-8,-18,-10,-14,10
260 DATA -20,4,-14,-6,-10,-14
270 DATA -2,-14,-6,-4,-64,16
280 DATA -40,8,12,32,4,30
290 DATA -6,18,-18,14,-14,22
300 DATA -10,10,-18,8,-14,0
310 DATA 0,16,36,4,14,-6
320 DATA 16,-2,-12,12,-2,20
330 DATA 16,0,4,-6,18,-6
340 DATA 12,0,4,8,4,8
350 DATA 14,8,6,16,14,8
360 REM -----
370 REM -> Module Villes de France
380 REM -----
390 DIM v$(13),x(13),y(13):c=1
400 WHILE c<>14
410 READ x,y,v$(c)
420 x(c)=x:y(c)=y
430 LOCATE 1,c:PRINT v$(c)
440 c=c+1
450 WEND
460 DATA 306,302,PARIS
470 DATA 316,348,LILLE

```

```

480 DATA 344,314,REIMS
490 DATA 272,324,LE HAVRE
500 DATA 226,328,CHERBOURG
510 DATA 172,302,BREST
520 DATA 216,288,RENNES
530 DATA 216,264,NANTES
540 DATA 260,254,POITIERS
550 DATA 244,206,BORDEAUX
560 DATA 366,214,LYON
570 DATA 364,180,AVIGNON
580 DATA 388,168,MARSEILLE
590 REM -----
600 REM -> Module de jeu
610 REM -----
620 RANDOMIZE TIME/1000:t=0:br=0
630 INK 1,24:v=INT(RND*13)+1
640 FOR i=20 TO 22:LOCATE 2,i:PRINT SPACE$(39)
:NEXT i
650 IF t=10 THEN 800
660 FOR j=1 TO 101
670 FOR i=-1 TO 1
680 PLOT x(v)+i,y(v)-i,(j MOD 2)
690 PLOT x(v)+i,y(v)+i,(j MOD 2)
700 NEXT i,j
710 LOCATE 2,20:INPUT "Quelle est cette ville
";rep#:rep#=UPPER$(rep#)
720 t=t+1
730 IF rep#<>v$(v) THEN PRINT " Non, ";nom#;"
,c'etait ";v$(v) ELSE br=br+1:PRINT " C'est e
xact ";nom#
740 FOR i=-1 TO 1
750 PLOT x(v)+i,y(v)-i,0
760 PLOT x(v)+i,y(v)+i,0
770 NEXT i
780 FOR i=1 TO 3500:NEXT i
790 GOTO 630
800 PRINT:PRINT " Score de ";nom#;" :";br;"/ 1
0"

```



# 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>90 à 160</b>	Initialisation et présentation du jeu.
<b>170</b>	Effacement de la mémoire des phrases précédentes.
<b>180 à 230</b>	Lecture de la phrase et décomposition en éléments.
<b>270 à 300</b>	Initialisation du jeu.
<b>340 à 400</b>	Tirage au sort d'un élément : le numéro d'un élément est tiré au sort (ligne 360). Il est comparé aux numéros déjà sortis (lignes 380 à 400). S'il n'a pas déjà été utilisé, il est validé, sinon, on recommence le tirage au sort.
<b>410 à 450</b>	Affichage de chaque élément en une des trois couleurs définies (rouge, bleu clair, bleu marine). Cet affichage se fait sur une ligne et à une place aléatoire déterminée par X (ligne 410).
<b>490 à 500</b>	Entrée de la réponse.
<b>510</b>	Test réponse.
<b>520 à 550</b>	Réponse fausse « Recommence ».
<b>560 à 590</b>	Affichage de la bonne phrase (trois réponses erronées).
<b>600</b>	Affichage réponse exacte.
<b>610 à 620</b>	Test nombre de jeux. Fin de jeu.

**1000 à 1030**      Entrée des quatre phrases en DATA.

## Particularités techniques

La phrase est décomposée en éléments (quel que soit leur nombre) dans les lignes 200 à 230. AMSTRAD CPC 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 220).

L'affichage des éléments est déterminé par  $a(i)$ , 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).

La ligne 170 sert, 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). Cette ligne est superflue, mais elle est là pour que vous ne l'oubliez pas si un jour un de vos programmes utilise cette méthode.

AMSTRAD CPC 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 550. On peut aussi augmenter ou diminuer le nombre de phrases du jeu (variable F, ligne 620) en aménageant les lignes de DATA.

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

**161 : TR = 0**

**601 : TR = TR + 1**

**et en ligne 620 : PRINT « TU AS DONNE »;tr;« REPONSES FAUSSES » cela avant le END**

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

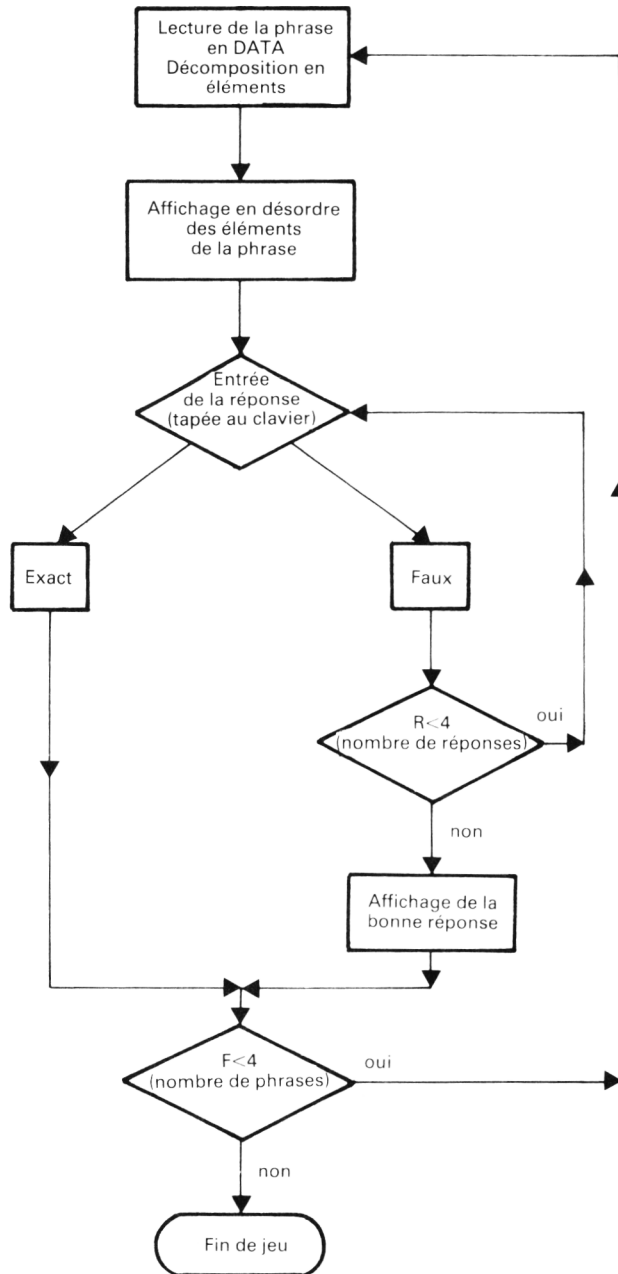
Un autre programme « J'écris tout seul » vous propose une adaptation pour les enfants de 6 et 7 ans, utilisant les manettes de jeu et n'obligeant pas l'enfant à taper la phrase.

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), AMSTRAD CPC 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 à les taper à chaque nouveau chargement (pensez aussi à placer en ligne 162 GOTO 190).

## Programme Construction de phrases



Organigramme

```

10 REM -----
20 REM --- CONSTRUCTION DE PHRASES ---
30 REM ----- D.Nielsen nov 83 -----
40 REM ----- Augustin jan 86 -----
50 REM -----
60 REM -----
70 REM -> Module presentation
80 REM -----
90 MODE 1:PAPER 13:PEN 0:CLS
100 f=0:RANDOMIZE TIME/1000
110 LOCATE 2,10:PRINT "Tu vas voir apparaitre"
," des mots en desordre"
120 PRINT:PRINT " Tu dois les replacer dans le
bon"," ordre pour former une phrase"," correc
te"
130 PRINT:PRINT " Et ensuite, taper cette phra
se"
140 LOCATE 3,24:PRINT "APPUIE SUR UNE TOUCHE P
OUR CONTINUER"
150 i#=INKEY#:IF i#="" THEN 150 ELSE CLS
160 f=f+1
170 IF f<>1 THEN FOR i=1 TO p:y#(i)="":NEXT i
180 READ ph#
190 p=1:l=LEN(ph#)
200 FOR i=1 TO l
210 x#=MID$(ph#,i,1)
220 IF x#<>" " THEN y#=y#+x# ELSE y#(p)=y#:p=p
+1:y#=""
230 NEXT i
240 REM -----
250 REM -> Module debut du jeu
260 REM -----
270 FOR i=1 TO 3500:NEXT i
280 r=0:CLS
290 LOCATE 3,24:PRINT "APPUIE SUR UNE TOUCHE P
OUR CONTINUER"
300 i#=INKEY#:IF i#="" THEN 300
310 REM -----
320 REM -> Tirage au sort et affichage
330 REM -----
340 CLS
350 FOR i=1 TO p
360 a(i)=INT(RND*p)+1
370 IF i=1 THEN 410
380 FOR k=1 TO i-1
390 IF a(i)=a(k) THEN 360
400 NEXT k
410 x=INT(RND*28)+1
420 j=INT(RND*3)+2

```

```

430 PEN j:LOCATE x,i*2-1:PRINT y$(a(i))
440 SOUND 1,35,1,15,,1
450 NEXT i
460 REM -----
470 REM -> Module reponse
480 REM -----
490 PEN 0:LOCATE 2,21:PRINT "A toi de taper le
s mots dans le bon"," ordre, pour former une p
hrase correcte"
500 LOCATE 2,23:INPUT rep$:rep$=UPPER$(rep$)
510 IF rep$=LEFT$(ph$,1-1) THEN 600
520 SOUND 1,402,20,15,,1:PRINT " FAUX"
530 FOR i=1 TO 3000:NEXT i
540 FOR i=21 TO 25:LOCATE 1,i:PRINT SPACE$(39)
:NEXT i
550 r=r+1:IF r<4 THEN LOCATE 2,21:PRINT "Recom
mence":GOTO 500
560 FOR i=20 TO 24:LOCATE 1,i:PRINT SPACE$(39)
:NEXT i
570 LOCATE 1,20:PRINT "La phrase correcte etai
t : "
580 PRINT:PRINT ph$
590 GOTO 610
600 SOUND 1,100,20,15,,1:PRINT " BRAVO"
610 FOR i=1 TO 3000:NEXT i
620 IF f=4 THEN CLS:PRINT " C'est termine, au
revoir":END ELSE 160
1000 DATA "LE PETIT CHAT JOUE AVEC LA PELOTE D
E LAINE "
1010 DATA "LES GARCONS JOUENT SUR LE TERRAIN D
E FOOTBALL "
1020 DATA "MES SOEURS ONT UNE GRANDE POUPEE BL
ONDE "
1030 DATA "LES PETITES HIRONDELLES SONT PERCHE
ES SUR LES FILS ELECTRIQUES "

```



# 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 jeux.

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>100 à 250</b>	Initialisation et présentation du jeu.
<b>270</b>	Effacement de la mémoire des phrases précédentes.
<b>280 à 330</b>	Lecture de la phrase et décomposition en éléments.
<b>370 à 400</b>	Initialisation du jeu.
<b>440 à 530</b>	Tirage au sort d'un élément : le numéro d'un élément est tiré au sort (ligne 450). Il est comparé aux numéros déjà sortis (lignes 470 à 490). S'il n'a pas déjà été utilisé, il est validé, sinon on recommence le tirage au sort.

<b>510 à 530</b>	Boucle pour afficher les différents éléments de la phrase, dans leur fenêtre.
<b>570 à 590</b>	Question.
<b>600 à 710</b>	Réponse par déplacement du curseur et validation par "barre espace" ou bouton de tir.
<b>720-730</b>	Bonne réponse : affichage de « EXACT » : une séquence musicale est jouée : les éléments de la phrase déjà trouvés sont affichés. Réponse fausse : affichage de "FAUX" et même séquence que précédemment.
<b>740-760</b>	Phrase complète - Test nombre de jeux.
<b>770 à 790</b>	Phrases du jeu en DATA.
<b>800</b>	Fin de jeu.

## Particularités techniques

Jusqu'à la ligne 390, le programme est très proche de « Construction de phrases ». Seules différences :

- la phrase, plus courte, est lue en un seul DATA ;
- le principe de découpage de la phrase en éléments (lignes 300 à 330) 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 fenêtre (I) (une à sept au maximum) est affectée du numéro d'ordre de l'élément dans la phrase a(i).

C'est ce qui permet de tester ensuite la valeur de la réponse ligne 720 : a(j) = S

S : numéro de l'élément demandé.

## Modifications

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 760 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 » ligne 720).

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.

Pour jouer avec une manette de jeux il faut remplacer les lignes 620 à 660 par

```
620 rep=JOY(0)
625 FOR k=1 TO 100:NEXT k
630 IF rep<>1 AND rep<>2 AND rep<>16 AND rep<>
32 THEN 620
640 IF rep=16 OR rep=32 THEN 720
650 LOCATE#j,13,2:PRINT#j,"."
660 IF rep=1 THEN j=j+1 ELSE j=j-1
```

## 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 :

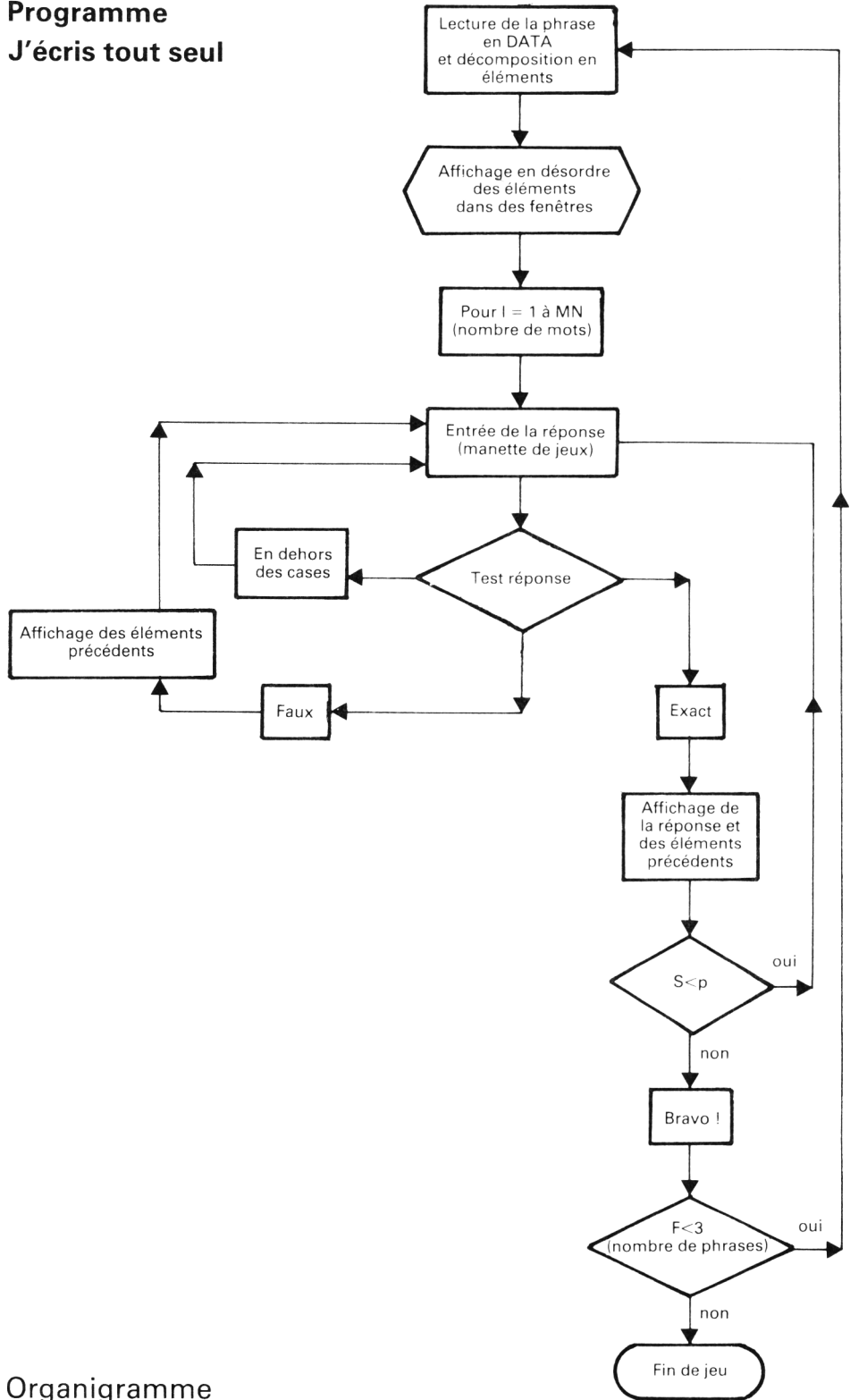
- pas plus de sept éléments dans la phrase (place sur l'écran) ;
- vous devez terminer la phrase par un espace ;
- si vous utilisez des mots de plus de dix lettres, il faut agrandir les fenêtres.

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 7<sup>e</sup> au maximum !) dans une fenêtre 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.

**Programme  
J'écris tout seul**



Organigramme

```

10 REM -----
20 REM -- J'ECRIS TOUT SEUL --
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin jan 86 ---
50 REM -----
60 ON BREAK GOSUB 800
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 13:PEN 0:CLS
110 f=0:RANDOMIZE TIME/1000
120 FOR i=1 TO 5
130 j=INT(RND*3)+2
140 WINDOW#i,3,16,22-4*i,24-4*i:PAPER#i,j::CLS
#i:PEN#i,j+1
150 NEXT i
160 WINDOW#6,24,38,2,4:PAPER#6,0:PEN#6,1:CLS#6
170 WINDOW#7,24,38,6,8:PAPER#7,2:PEN#7,3:CLS#7
180 WINDOW 2,38,22,24:PAPER 10:PEN 0:CLS
190 FOR i=1 TO 5:LOCATE#i,2,2:PRINT#i,"J'ECRIS
":NEXT i
200 LOCATE#6,2,2:PRINT#6,"TOUT"
210 LOCATE#7,2,2:PRINT#7,"SEUL"
220 PRINT " Tu vas voir apparaitre"," des mots
en desordre"
230 FOR i=1 TO 5000:NEXT i
240 PRINT " Tu dois les replacer dans le bon",
" ordre pour former une phrase"," correcte"
250 FOR i=1 TO 5000:NEXT i
260 f=f+1
270 IF f<>1 THEN FOR i=1 TO p:y$(i)="":NEXT i
280 READ ph$
290 p=1:l=LEN(ph$)
300 FOR i=1 TO l
310 x#=MID$(ph$,i,1)
320 IF x#<>" " THEN y#=y#+x$ ELSE y$(p)=y$:p=p
+1:y#=""
330 NEXT i
340 REM -----
350 REM -> Module debut du jeu
360 REM -----
370 s=1:m#=""::CLS
380 LOCATE 1,2:PRINT "APPUIE SUR UNE TOUCHE PO
UR CONTINUER"
390 i#=INKEY#:IF i#="" THEN 390
400 nm=p-1
410 REM -----
420 REM -> Tirage au sort

```

```

430 REM -----
440 FOR i=1 TO nm
450 a(i)=INT(RND*nm)+1
460 IF i=1 THEN 500
470 FOR k=1 TO i-1
480 IF a(i)=a(k) THEN 450
490 NEXT k
500 NEXT i
510 FOR i=1 TO 7
520 CLS#i:LOCATE#i,2,2:PRINT#i,y$(a(i))
530 NEXT i
540 REM -----
550 REM -> Module reponse
560 REM -----
570 CLS
580 FOR i=1 TO nm
590 LOCATE 2,3:PRINT "Quel est le mot No";i;"d
e la phrase"
600 LOCATE#i,13,2:PRINT#i,CHR$(224)
610 j=1
620 i$=INKEY$:IF i$="" THEN 620
630 IF i$<>"1" AND i$<>"0" AND i$<>" " THEN SO
UND 1,400,20,15:GOTO 620
640 IF i$="" THEN 720
650 LOCATE#j,13,2:PRINT#j,"."
660 IF i$="1" THEN j=j+1 ELSE j=j-1
670 IF j=8 THEN j=1
680 IF j=0 THEN j=7
690 SOUND 1,100,10,15
700 LOCATE#j,13,2:PRINT#j,CHR$(224)
710 GOTO 620
720 IF a(j)=s THEN CLS:PRINT " C'est exact":m$
=m$+" "+y$(i):PRINT m$ ELSE SOUND 1,400,20:CLS
:PRINT " Faux, recommence":PRINT m$:GOTO 620
730 LOCATE#j,13,2:PRINT#j,"."
740 s=s+1:IF s<>p THEN 760
750 IF f=3 THEN 800 ELSE 250
760 NEXT i
770 DATA "ELLE EST DANS LA RUE "
780 DATA "LE BEBE EST DANS SON LIT "
790 DATA "LE PETIT CHAT JOUE AVEC LA BALLE "
800 CLS:PRINT "AU REVOIR":END

```



# 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 ou huit parties d'un texte s'affichent en désordre, dans des cases sur l'écran. Le joueur doit ensuite sélectionner avec la manette (ou le clavier) 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 bonne, son numéro d'ordre dans la phrase s'affiche à côté de l'élément.

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 :

<b>1030-1100</b>	Les 7 ou 8 parties des textes sont entrées séparément, mais dans l'ordre exact, en DATA.
<b>1140-1210</b>	
<b>1250-1330</b>	
<b>190 à 200</b>	Lecture des différentes parties d'un texte en DATA.
<b>500 à 610</b>	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 ou huit éléments maximum. Pour une question de mise en place sur l'écran : c'est le nombre limite de lignes pouvant être affichées à la fois dans le désordre, en haut de l'écran afin d'y laisser la place des messages.

Toutefois, le principe de lecture des parties de texte en DATA (lignes 190 à 200) 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 » ligne 610.

Le nombre de textes est fixé ligne 660 : des textes supplémentaires pourront être ajoutés en modifiant cette ligne et... en les entrant en DATA !

Pour jouer avec la manette de jeux, il faut remplacer les lignes 500 à 550 par

```
500 rep=JOY(0)
510 FOR k=1 TO 100:NEXT k
520 IF rep<>1 AND rep<>2 AND rep<>16 AND rep<>
32 THEN 500
530 IF rep=16 OR rep=32 THEN 620
540 LOCATE 39,2*j-1:PRINT "."
550 IF rep=1 THEN j=j+1 ELSE j=j-1
```

## Adaptations

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

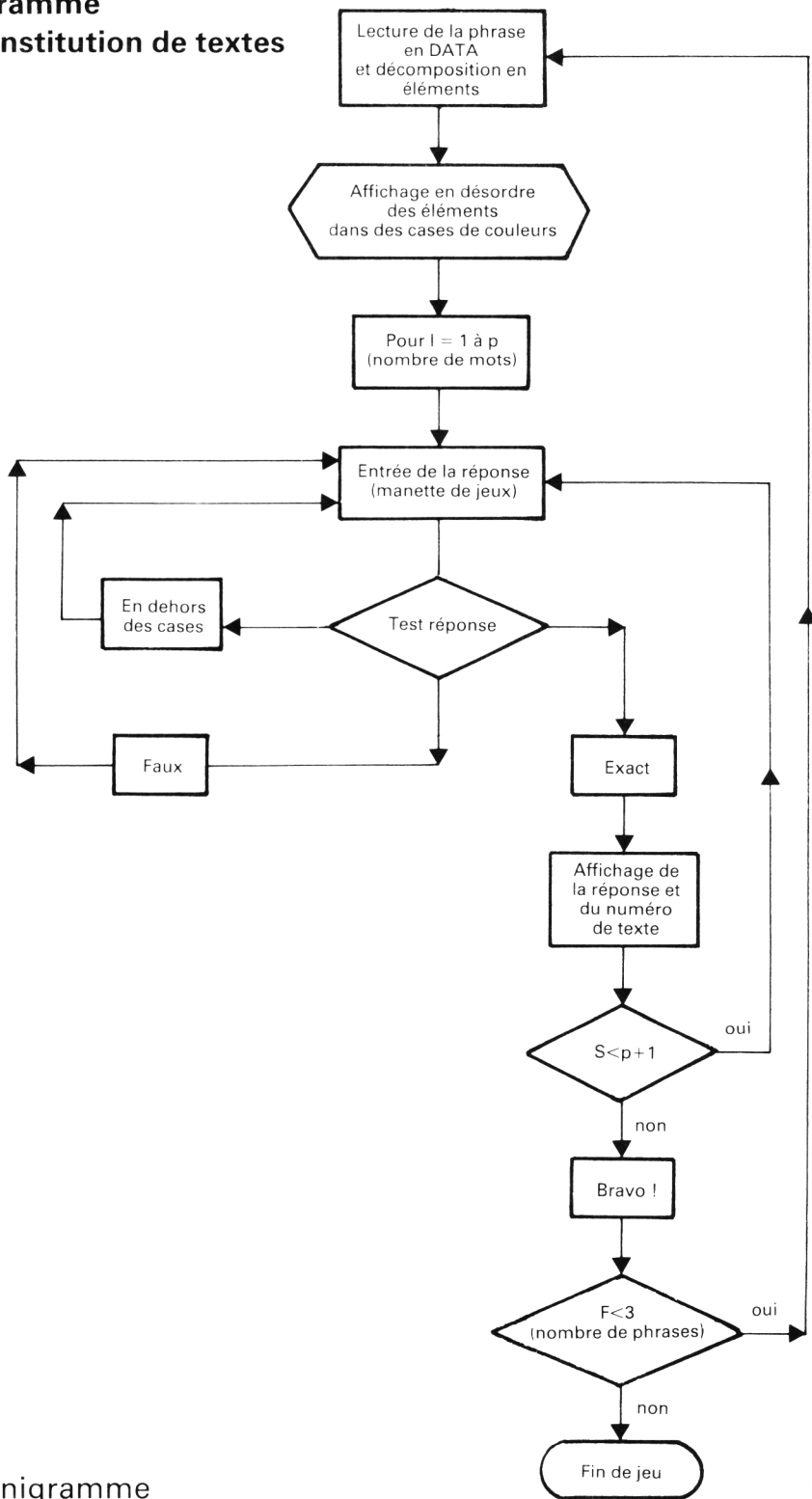
Pour donner toute sa valeur à ce travail, il faut choisir des textes d'auteurs, suffisamment courts pour être partagés en sept ou huit éléments (chaque élément ne pouvant contenir que 34 caractères ou espaces pour être affiché sur une seule ligne).

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 ».

# Programme Reconstitution de textes



Organigramme

```

10 REM -----
20 REM -- RECONSTITUTION DE TEXTE --
30 REM ----- D.Nielsen nov 83 -----
40 REM ----- Augustin jan 86 -----
50 REM -----
60 REM -----
70 REM -> Module presentation
80 REM -----
90 MODE 1:PAPER 13:PEN 0:CLS
100 f=0:RANDOMIZE TIME/1000
110 LOCATE 8,5:PRINT "RECONSTITUTION DE TEXTE"
120 LOCATE 2,10:PRINT "Tu vas voir apparaitre"
," des morceaux de texte en desordre"
130 PRINT:PRINT " Tu dois les replacer dans le
bon"," ordre pour former une phrase"," correc
te"
140 PRINT:PRINT " Deplace le curseur sur les e
lements"," choisis les uns apres les autres"
150 LOCATE 3,24:PRINT "APPUIE SUR UNE TOUCHE P
OUR CONTINUER"
160 i#=INKEY#:IF i#="" THEN 160:ELSE CLS
170 f=f+1
180 p=0
190 READ ph#
200 IF ph#<>"ZZZZ" THEN p=p+1:y#(p)=ph#:GOTO 1
90
210 REM -----
220 REM -> Module debut du jeu
230 REM -----
240 FOR i=1 TO 3500:NEXT i
250 s=1:CLS
260 LOCATE 3,24:PRINT "APPUIE SUR UNE TOUCHE P
OUR CONTINUER"
270 i#=INKEY#:IF i#="" THEN 270
280 REM -----
290 REM -> Tirage au sort et affichage
300 REM -----
310 CLS
320 FOR i=1 TO p
330 a(i)=INT(RND*p)+1
340 IF i=1 THEN 380
350 FOR k=1 TO i-1
360 IF a(i)=a(k) THEN 330
370 NEXT k
380 j=INT(RND*3)+2
390 PEN j:LOCATE 1,i*2-1:PRINT y#(a(i))
400 SOUND 1,35,1,15,,1

```

```

410 NEXT i
420 REM -----
430 REM -> Module réponse
440 REM -----
450 PEN 0:LOCATE 2,21:PRINT "A toi de trouver
les morceaux"," dans le bon ordre"
460 FOR i=1 TO p
470 LOCATE 2,24:PRINT "Quel est le morceau No"
;i;"de la phrase"
480 LOCATE 39,1:PRINT CHR$(224)
490 j=1
500 i$=INKEY$:IF i$="" THEN 500
510 IF i$<>"1" AND i$<>"0" AND i$<>" " THEN SO
UND 1,400,20,15:GOTO 500
520 IF i$="" THEN 610
530 LOCATE 39,2*j-1:PRINT "."
540 IF i$="1" THEN j=j+1 ELSE j=j-1
550 IF j=p+1 THEN j=1
560 IF j=0 THEN j=p
570 SOUND 1,111,10,15
580 LOCATE 39,2*j-1:PRINT CHR$(224)
590 FOR k=20 TO 23:LOCATE 1,k:PRINT SPACE$(39)
:NEXT k
600 GOTO 500
610 IF a(j)=s THEN LOCATE 2,20:PRINT "C'est ex
act":LOCATE 37,2*j-1:PRINT i ELSE SOUND 1,400,
20,15:LOCATE 2,20:PRINT "Faux, recommence":GOT
O 500
620 LOCATE 39,2*j-1:PRINT "."
630 s=s+1:IF s<>p+1 THEN 670
640 CLS:FOR k=1 TO p:PRINT y$(k):NEXT k
650 FOR k=1 TO 3000:NEXT k
660 IF f=3 THEN 680 ELSE 170
670 NEXT i
680 FOR i=1 TO 3500:NEXT i
690 CLS:PRINT "AU REVOIR":END
1000 REM -----
1010 REM -> LE PETIT PRINCE DE SAINT EXUPERY
1020 REM -----
1030 DATA "MA VIE EST MONTONE."
1040 DATA "JE CHASSE LES POULES,"
1050 DATA "LES HOMMES ME CHASSENT."
1060 DATA "TOUTES LES POULES SE RESSEMBLENT"
1070 DATA "ET TOUS LES HOMMES SE RESSEMBLENT."
1080 DATA "MAIS SI TU M'APPRIVOISES,"
1090 DATA "MA VIE SERA COMME ENSOLEILLE."
1100 DATA "ZZZZ"

```

```
1110 REM -----
1120 REM -> D'apres MAURICE GENEVOIS
1130 REM -----
1140 DATA "C'EST UN PETIT PASSEREAU"
1150 DATA "A ROBE GRISE"
1160 DATA "D'UN GRIS VELOUTE ET PROFOND."
1170 DATA "LORSQUE LE VENT REMUE LA RAMILLE"
1180 DATA "QUI LE PORTE,"
1190 DATA "IL OSCILLE DOUCEMENT,"
1200 DATA "SUR SES FATTES."
1210 DATA "ZZZZ"
1220 REM -----
1230 REM -> d'apres M. VAN DER MEERSCH
1240 REM -----
1250 DATA "DEHORS,"
1260 DATA "LA TEMPETE S'ETAIT LEVEE;"
1270 DATA "LE VENT D'OUEST GEMISSAIT"
1280 DATA "DANS LA CHEMINEE."
1290 DATA "ON L'ENTENDAIT RODER"
1300 DATA "DANS LES COULOIRS"
1310 DATA "COMME UN GRAND CHIEN INQUIET"
1320 DATA "QUI PLEURE."
1330 DATA "ZZZZ"
```



# 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 à l'écran.

## Commentaires

<b>100 à 160</b>	Présentation du programme.
<b>170 à 260</b>	Choix du cocktail.
<b>300</b>	Choix aléatoire du volume de la carafe (10 - 20 - 30 - 40 - 50 - 60 ... 100 cl).
<b>320 à 400</b>	Dessin de la carafe et indication des proportions.
<b>600 à 850</b>	Calcul du volume de chaque composant (1 à 3).
<b>890 à 1000</b>	Visualisation dans la carafe de chaque bonne réponse.
<b>1040 à 1250</b>	Calcul du complément d'eau et visualisation de la réponse donnée, même si elle est fautive (la carafe « déborde » si la réponse est trop grande).
<b>1260 à 1310</b>	Fin de programme. Retour possible à un autre calcul avec un cocktail différent et un nouveau volume de carafe.
<b>1320 à 1490</b>	Sous-programme « la carafe déborde » en cas de complément d'eau trop important, puis effacement de cette visualisation.
<b>1500 à 1520</b>	
<b>1530 à 1550</b>	
<b>1560 à 1580</b>	
<b>1590 à 1610</b>	
<b>1620 à 1640</b>	Liste des proportions des cocktails et couleurs des composants.
<b>1650 à 1670</b>	

## Particularités techniques

Les proportions des cocktails sont inscrites en DATA. Le choix du cocktail renvoie à une ligne (450 à 500) 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 600 à 1000). 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 680 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 la fonction RND.

## 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 !

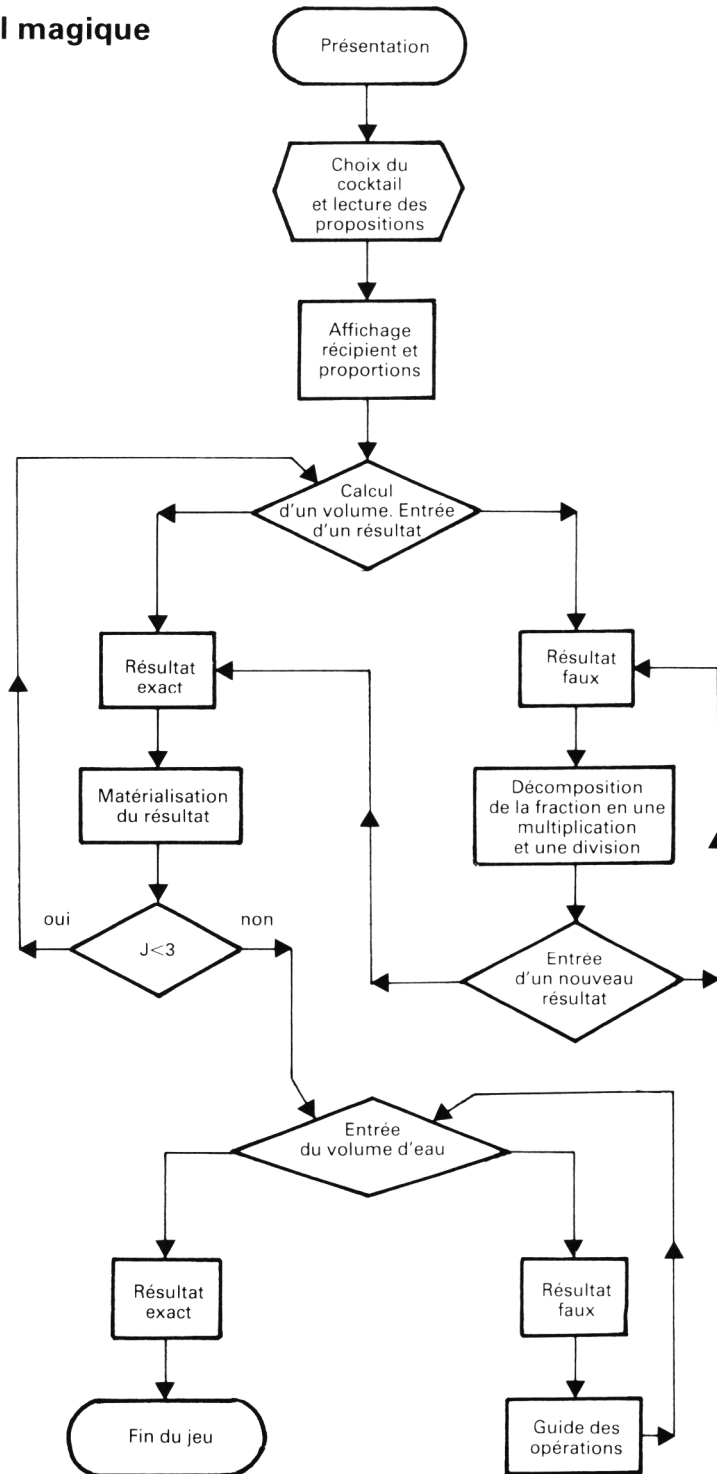
Vous pouvez modifier les lignes 1500 à 1670 en y composant vos propres cocktails.

*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 !

**Programme  
Cocktail magique**



Organigramme

```

10 REM -----
20 REM --- COCKTAIL MAGIQUE ---
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin fev 86 ---
50 REM -----
60 ON BREAK GOSUB 1270
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 13:PEN 0:CLS
110 RANDOMIZE TIME/1000
120 LOCATE 10,5:PRINT "COCKTAIL MAGIQUE"
130 LOCATE 2,8:PRINT "Un cocktail est un melange",
" de plusieurs boissons dans un verre",
" ou une carafe"
140 PRINT:PRINT " Tu vas devoir preparer",
" un cocktail de ton choix"
150 LOCATE 2,24:PRINT "APPUIE SUR UNE TOUCHE F
OUR CONTINUER"
160 i#=INKEY#:IF i#="" THEN 160
170 CLS
180 LOCATE 5,5:PRINT "HAWAI DRINK.....1"
190 LOCATE 5,7:PRINT "COCKTAIL EXOTIQUE...2"
200 LOCATE 5,9:PRINT "COCKTAIL ESQUIMAU...3"
210 LOCATE 5,11:PRINT "BOISSON ROSE.....4"
220 LOCATE 5,13:PRINT "BONNE MINE.....5"
230 LOCATE 5,15:PRINT "VIN DE FRAISES.....6"
240 LOCATE 5,17:INPUT "Ton choix ";choix
250 IF choix<1 OR choix>6 THEN 170
260 LOCATE 5,20:INPUT "Quel est ton prenom ";n
om#:nom#=UPPER$(nom#)
270 REM -----
280 REM -> Module dessin recipient
290 REM -----
300 CLS
310 vol=INT(RND*9+1)*10
320 FOR i=1 TO 12
330 LOCATE 5,3+i:PRINT CHR$(143)
340 LOCATE 11,3+i:PRINT CHR$(143)
350 NEXT i
360 FOR i=6 TO 10:LOCATE i,15:PRINT CHR$(143):
NEXT i
370 LOCATE 20,4:PRINT "Le volume de la"
380 LOCATE 20,5:PRINT "carafe est de"
390 LOCATE 19,6:PRINT vol;"centilitres"
400 f=14:t=2
410 REM -----
420 REM -> Module proportions

```

```

430 REM -----
440 ON choix GOTO 450,460,470,480,490,500
450 RESTORE 1500:GOTO 510
460 RESTORE 1530:GOTO 510
470 RESTORE 1560:GOTO 510
480 RESTORE 1590:GOTO 510
490 RESTORE 1620:GOTO 510
500 RESTORE 1650
510 FOR i=1 TO 3
520 READ nu(i),de(i),a$(i)
530 LOCATE 20,6+2*i:PRINT nu(i);"/";de(i);"de"

540 LOCATE 20,7+2*i:PRINT a$(i)
550 NEXT i
560 LOCATE 20,15:PRINT "et de l'eau gazeuse"

570 REM -----
580 REM -> Module calcul des volumes
590 REM -----
600 FOR j=1 TO 3
610 FOR i=17 TO 25:LOCATE 1,i:PRINT SPACE$(39)
:NEXT i
620 LOCATE 1,17:PRINT "Combien dois-tu verser
de ",a$(j)
630 PRINT "Il faut que tu calcules ce que ","r
epresente";nu(j);"/";de(j);"de";vol;"centilitr
es"
640 PRINT "Attention, il faut que tu arrondiss
es","ton resultat au nombre entier inferieur",
"( pas de chiffres decimaux)"
650 INPUT "Quelle est ta reponse ";rep(j)
660 IF rep(j)<>INT(rep(j)) THEN 610
670 IF rep(j)=INT(vol*nu(j)/de(j)) THEN 890
680 FOR i=17 TO 25:LOCATE 1,i:PRINT SPACE$(39)
:NEXT i
690 ENT 2,=71,20,=89,20,=71,20,=0,0:FOR i=1 TO
3:SOUND 1,0,100,15,,2:NEXT i
700 LOCATE 1,17:PRINT "Tu te trompes ";nom$;"V
oici comment tu dois calculer"
710 PRINT "Pour prendre les";nu(j);"/";de(j);"
de";vol;"cl","il faut d'abord multiplier";vol;
"par";nu(j)
720 INPUT "Cela fait combien ";rn(j)
730 IF rn(j)=vol*nu(j) THEN 770
740 PRINT "Ce n'est pas cela, ";nom$;" recomme
nce"
750 FOR i=1 TO 5000:NEXT i
760 GOTO 680

```

```

770 FOR i=17 TO 25:LOCATE 1,i:PRINT SPACE$(39)
:NEXT i
780 LOCATE 1,17:PRINT "Il faut maintenant divi
ser";rn(j);"par";de(j)
790 PRINT "Attention division entiere seulemen
t","(pas de chiffres decimaux"
800 INPUT "Cela fait combien ";rd(j)
810 IF rd(j)=INT(rn(j)/de(j)) THEN 890
820 PRINT "Ce n'est pas cela, ";nom#;" recomme
nce"
830 FOR i=1 TO 5000:NEXT i
840 ENT 2,=71,20,=89,20,=71,20,=0,0:FOR i=1 TO
3:SOUND 1,0,100,15,,2:NEXT i
850 GOTO 770
860 REM -----
870 REM -> Module resultat exact
880 REM -----
890 rep(j)=INT(vol*nu(j)/de(j))
900 PRINT "C'est exact ";nom#
910 h=INT(6*nu(j)/de(j))
920 PEN t:t=t+1
930 FOR i=f-h TO f
940 FOR k=6 TO 10
950 LOCATE k,i:PRINT CHR$(143)
960 NEXT k
970 NEXT i
980 PEN 0:f=f-h-1
990 FOR i=1 TO 5000:NEXT i
1000 NEXT j
1010 REM -----
1020 REM -> Module complement d'eau
1030 REM -----
1040 FOR i=17 TO 25:LOCATE 1,i:PRINT SPACE$(39)
):NEXT i
1050 PEN 2
1060 FOR i=4 TO f
1070 FOR k=6 TO 10
1080 LOCATE k,i:PRINT CHR$(143)
1090 NEXT k
1100 NEXT i
1110 PEN 0
1120 LOCATE 1,17:PRINT "Combien d'eau gazeuse
dois-tu rajouter"
1130 INPUT r
1140 v=vol-(rep(1)+rep(2)+rep(3))
1150 IF r<v THEN PRINT "Ce n'est pas assez ";n
om#:GOTO 1180
1160 IF r>v THEN GOSUB 1320

```

```

1170 IF r=v THEN 1250
1180 PRINT "Il faut d'abord additionner le vol
umes","des trois jus ou sirops verses"
1190 ENT-2,=71,20,=89,20,=71,20,=0,0:FOR i=1 T
O 3:SOUND 1,0,100,15,,2:NEXT i
1200 PRINT rep(1);"+";rep(2);"+";rep(3)
1210 INPUT "Cela fait combien";tr
1220 IF tr<>vol-v THEN PRINT "Non, ";nom#;" re
commence" ELSE PRINT "Exact, ";nom#
1230 FOR i=i TO 5000:NEXT i
1240 GOTO 1040
1250 FOR i=17 TO 25:LOCATE 1,i:PRINT SPACE$(39
):NEXT i
1260 LOCATE 1,17:PRINT "C'est exact ";nom#;"tu
peux devenir un vrai barman"
1270 PRINT:INPUT "Un autre cocktail ( o / n )
";rep#
1280 rep#=UPPER$(rep#)
1290 IF rep#<>"O" THEN CLS:PRINT "Au revoir":E
ND
1300 FOR i=1 TO 5000:NEXT i
1310 GOTO 170
1320 FOR i=17 TO 25:LOCATE 1,i:PRINT SPACE$(39
):NEXT i
1330 FOR i=9 TO 15
1340 FOR j=1 TO 4
1350 LOCATE j,i:PRINT CHR$(143)
1360 LOCATE 11+j,i:PRINT CHR$(143)
1370 NEXT j
1380 NEXT i
1390 FOR i=17 TO 25:LOCATE 1,i:PRINT SPACE$(39
):NEXT i
1400 LOCATE 5,17:PRINT "CA DEBORDE ....."
1410 FOR i=1 TO 5000:NEXT i
1420 FOR i=9 TO 15
1430 FOR j=1 TO 4
1440 LOCATE j,i:PRINT " "
1450 LOCATE 11+j,i:PRINT " "
1460 NEXT j
1470 NEXT i
1480 LOCATE 1,17
1490 RETURN
1500 DATA 1,3,JUS D ORANGE
1510 DATA 1,5,LAIT DE COCO
1520 DATA 1,4,SIROP DE SUCRE
1530 DATA 1,5,JUS DE MANGUE
1540 DATA 2,8,JUS DE PAPAYE

```

1550 DATA 1,10,JUS DE GOYAVE  
1560 DATA 2,5,CITRON PRESSE  
1570 DATA 1,10,SIROP DE MENTHE  
1580 DATA 1,3,GLACE PILEE  
1590 DATA 1,3,JUS DE FRAMBOISE  
1600 DATA 1,4,SIROP DE SUCRE  
1610 DATA 1,10,CREME FRAICHE  
1620 DATA 3,5,JUS DE TOMATE  
1630 DATA 1,4,JUS DE CAROTTE  
1640 DATA 1,10,JUS DE CELERI  
1650 DATA 2,5,JUS DE FRAISE  
1660 DATA 2,10,JUS DE CITRON  
1670 DATA 1,4,SIROP DE SUCRE

# 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 marquées 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>100 à 170</b> | Initialisation et présentation du jeu.  |
| <b>210 à 310</b> | Dessin du cube (en coordonnées graphiques).   |
| <b>350</b>       | Le volume du cube est tiré au sort.   |
| <b>360 à 570</b> | Quel est son volume ? Si la réponse donnée n'est pas exacte : on décompose le problème en deux questions (calcul de l'aire, puis du volume à partir de l'aire). |

<b>610 à 730</b>	Recherche de la masse du cube. Dessin de la balance, des poids de mesure. h et t déterminent les positions des deux plateaux.
<b>740 à 880</b>	Déplacements du curseur sous les poids, lecture de la réponse validée par "barre espace" ou le bouton de tir.
<b>890</b>	Détermination du poids désigné.
<b>900 à 960</b>	Modification des paramètres h et t des balances en fonction du nouveau poids.
<b>970</b>	Pas assez !
<b>980 à 1060</b>	Gagné, le bon poids est trouvé. Fin de jeu ou possibilité de continuer.
<b>1070 à 1100</b>	Trop de poids !
<b>1140 à 1280</b>	Sous-programme de dessin de la balance et de l'afficheur digital.

## Particularités techniques

Le choix des poids se fait avec le curseur sous 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 !).

Ensuite, la balance est redessinée : les lignes 1140 à 1280 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 890 un compteur.

*Exemple :*

```

886 A=1 THEN 1300
887 A=A+1
1300 PRINT« IL N'Y A PLUS CE POIDS DANS LA BOITE »
1310 GOTO 790

```

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

Pour jouer avec la manette de jeux, il faut remplacer les lignes 790 à 830 par :

```

790 rep=JOY(0)
800 IF rep<>4 AND rep<>8 AND rep<>16 AND rep<>
32 THEN 790
810 LOCATE j,24:PRINT " "
815 FOR i=1 TO 100:NEXT i
820 IF rep=16 OR rep=32 THEN 890
830 IF rep=8 THEN j=j+4 ELSE j=j-4

```

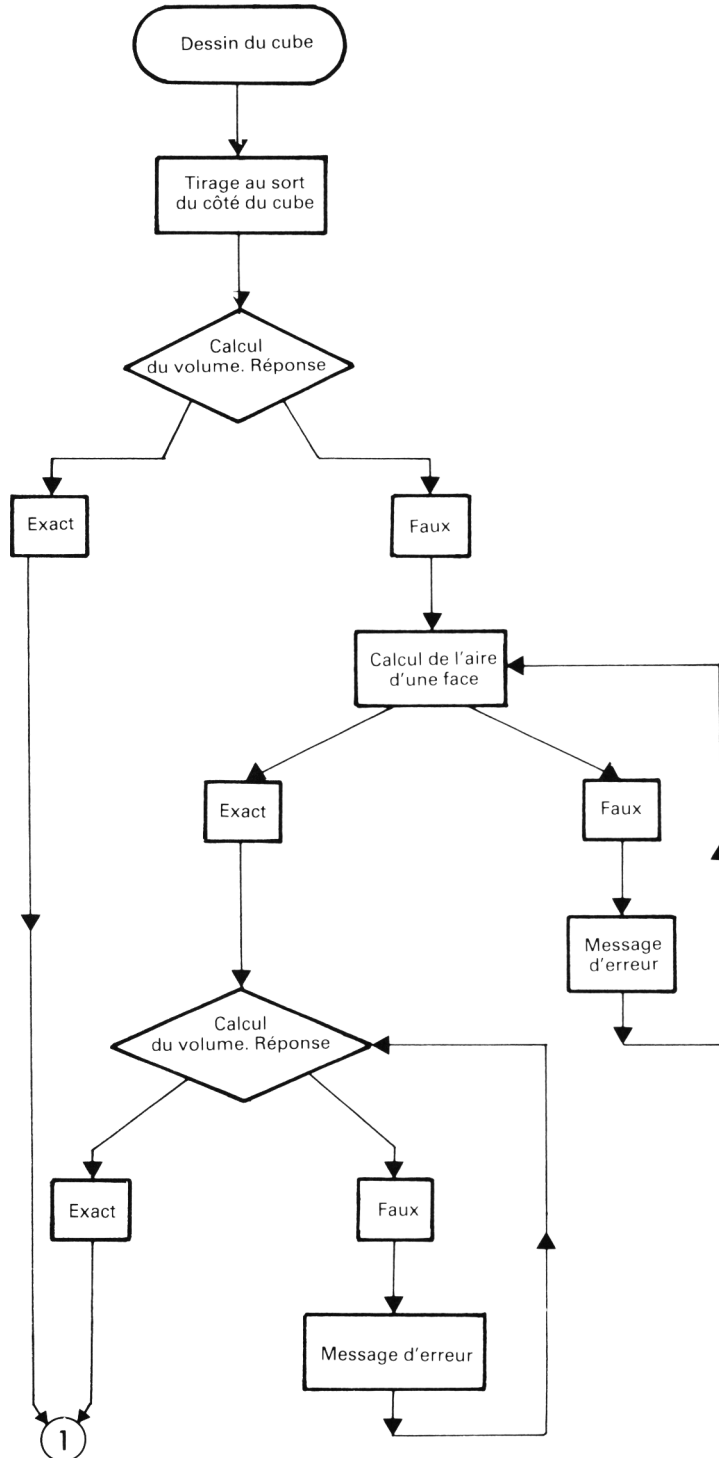
## Adaptations

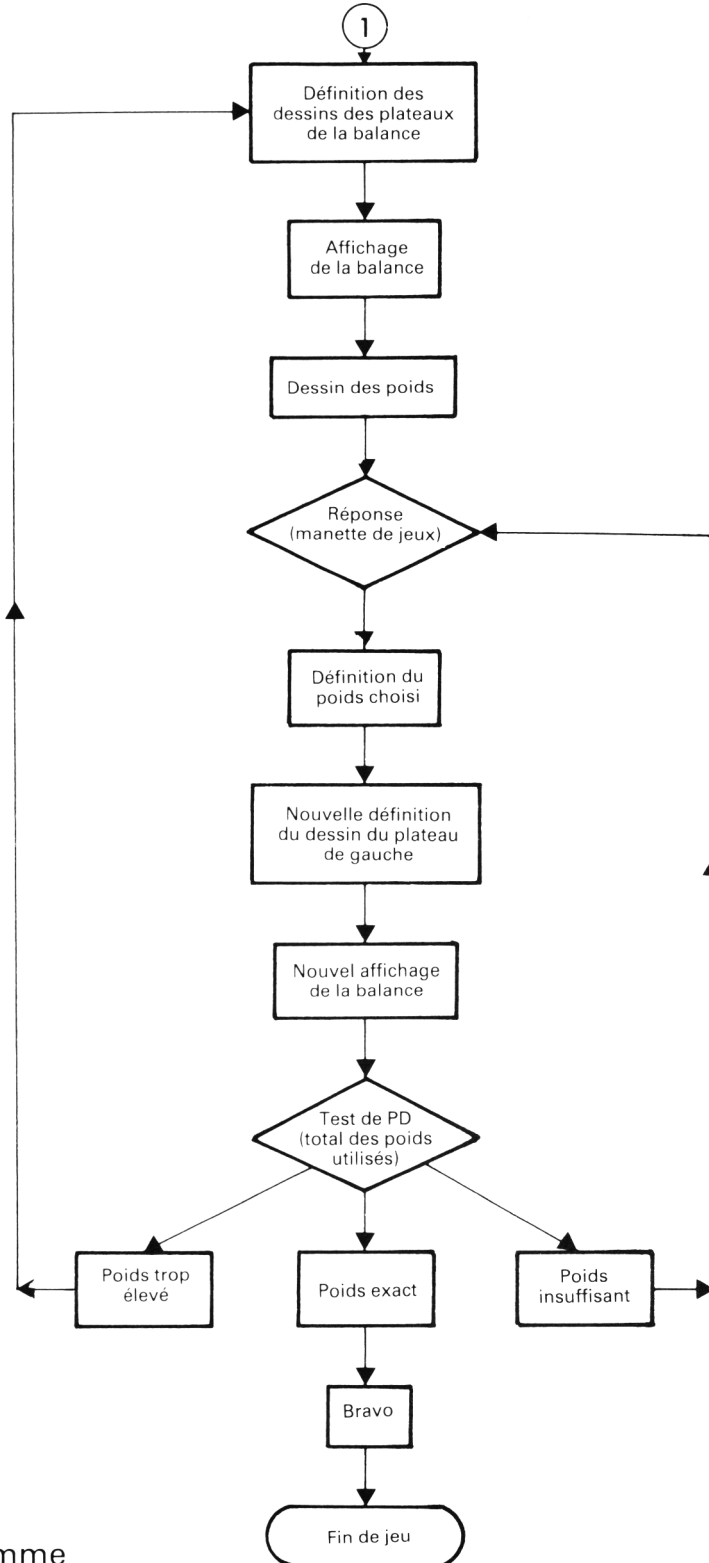
Ce programme permet une approche de la notion de masse volumique avec un liquide simple : l'eau.

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.

Programme Masse et volume





Organigramme

```

10 REM -----
20 REM ---- MASSE ET VOLUME ----
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin fev 86 ---
50 REM -----
60 ON BREAK GOSUB 1060
70 REM -----
80 REM -> Module presentation
90 REM -----
100 MODE 1:PAPER 0:PEN 1:CLS
110 RANDOMIZE TIME/1000
120 masse#=CHR$(143)+CHR$(143)+CHR$(143)
130 LOCATE 10,5:PRINT "MASSE ET VOLUME"
140 LOCATE 2,8:PRINT "Tu vas devoir calculer",
" le volume d'un cube"," sa masse rempli d'eau",
" puis verifier cette masse"," sur une balance"
150 LOCATE 2,24:PRINT "APPUIE SUR UNE TOUCHE P
OUR CONTINUER"
160 i#=INKEY#:IF i#="" THEN 160
170 CLS
180 REM -----
190 REM -> Module dessin du cube
200 REM -----
210 PLOT 200,300,1
220 DRAW 300,300,1
230 DRAW 300,200,1
240 DRAW 200,200,1
250 DRAW 200,300,1
260 DRAW 250,350,1
270 DRAW 350,350,1
280 DRAW 300,300,1
290 PLOT 350,350,1
300 DRAW 350,250,1
310 DRAW 300,200,1
320 REM -----
330 REM -> Module volume du cube
340 REM -----
350 c=INT(RND*10)+1
360 LOCATE 1,16:PRINT "Le cote de ce cube mesure";c;"cm"
370 INPUT "Quel est son volume, en cm3 ";vol
380 IF vol=c*c*c THEN 550
390 PRINT "Non, ce n'est pas cela"
400 ENT 2,=71,20,=89,20,=71,20,=0,0:FOR i=1 TO
3:SOUND 1,0,100,15,,2:NEXT i
410 FOR i=1 TO 3000:NEXT i

```

```

420 ENT 2,=71,20,=89,20,=71,20,=0,0:FOR i=1 TO
  3:SOUND 1,0,100,15,,2:NEXT i
430 FOR i=16 TO 25:LOCATE 1,i:PRINT SPACE$(39)
:NEXT i
440 LOCATE 1,16:PRINT "Pour calculer le volume
  d'un cube,", "il faut chercher l'aire", "d'une
  de ces faces"
450 PRINT "Combien fait :";c;"x";c;"= ";
460 INPUT sur
470 IF sur<>c*c THEN PRINT "Faux, recommence":
GOTO 410
480 FOR i=1 TO 3500:NEXT i
490 ENT 2,=71,20,=89,20,=71,20,=0,0:FOR i=1 TO
  3:SOUND 1,0,100,15,,2:NEXT i
500 FOR i=16 TO 25:LOCATE 1,i:PRINT SPACE$(39)
:NEXT i
510 LOCATE 1,16:PRINT "Bien ! maintenant, pour
  trouver", "son volume, il faut multiplier", "ce
  tte aire par la hauteur du cube"
520 PRINT "c'est a dire";sur;"x";c
530 INPUT "Cela fait combien ";volume
540 IF volume=sur*c THEN vol=volume ELSE PRINT
  "Faux, recommence":GOTO 480
550 PRINT "Bravo"
560 LOCATE 2,24:PRINT "Appuie sur une touche p
  our continuer"
570 i#=INKEY#:IF i#="" THEN 570 ELSE CLS
580 REM -----
590 REM -> Module recherche du poids
600 REM -----
610 t=16:h=-16:poids=0
620 GOSUB 1140
630 LOCATE 1,16:PRINT "Le cube est rempli d'ea
  u", "Tu dois trouver sa masse"
640 PRINT "Pour la trouver, tu vas peser ce cu
  be"
650 PRINT "Voici les poids dont tu disposes"

660 PEN 3
670 FOR i=1 TO 8
680 LOCATE 4*i,22:PRINT masse#
690 LOCATE 4*i,23:PRINT masse#
700 NEXT i
710 PEN 1
720 LOCATE 4,21:PRINT "500 200 100  50  20  10
  5   1"
730 FOR i=1 TO 4500:NEXT i

```

```

740 FOR i=16 TO 20:LOCATE 1,i:PRINT SPACE$(39)
:NEXT i
750 LOCATE 1,16:PRINT "Designe avec le curseur
le poids"
760 PRINT "Attention, tu ne pourras pas enleve
r","de poids de la balance"
770 j=5
780 LOCATE 5,24:PRINT CHR$(224)
790 i$=INKEY$:IF i$="" THEN 790
800 IF i$<>"1" AND i$<>"0" AND i$<>" " THEN SO
UND 1,400,20,15:GOTO 790
810 LOCATE j,24:PRINT " "
820 IF i$="" THEN 890
830 IF i$="1" THEN j=j+4 ELSE j=j-4
840 IF j=1 THEN j=33
850 IF j=37 THEN j=5
860 SOUND 1,100,10,15
870 LOCATE j,24:PRINT CHR$(224)
880 GOTO 790
890 poids=poids-500*(j=5)-200*(j=9)-100*(j=13)
-50*(j=17)-20*(j=21)-10*(j=25)-5*(j=29)-(j=33)
900 SOUND 1,100,20,15
910 FOR i=1 TO 19:LOCATE 5,i:PRINT SPACE$(39):
NEXT i
920 IF poids=vol THEN t=0:h=0
930 IF poids>vol THEN t=-16:h=16
940 GOSUB 1140
950 IF poids>vol THEN 1070
960 IF poids=vol THEN 990
970 LOCATE 1,16:PRINT "Il faut encore ajouter
des poids"
980 GOTO 730
990 LOCATE 1,16:PRINT "Bravo"
1000 PRINT "1 cm3 d'eau pese 1 gramme donc",vo
l;"cm3 pese";:IF vol>1 THEN PRINT "nt";
1010 PRINT vol;"gramme";:IF vol>1 THEN PRINT "
s"
1020 FOR i=1 TO 6000:NEXT i
1030 CLS
1040 INPUT "Un autre jeu ( o / n ) ";choix$:ch
oix$=UPPER$(choix$)
1050 IF choix$="O" THEN 170
1060 CLS:PRINT "Au revoir":END
1070 LOCATE 1,16:PRINT "C'est trop !"
1080 PRINT "Recommence"
1090 FOR i=1 TO 3000:NEXT i
1100 CLS:GOTO 610

```

```
1110 REM -----
1120 REM -> Module dessin de la balance
1130 REM -----
1140 PLOT 150,255-t,1:DRAWR +100,0,1
1150 PLOT 350,255-h,1:DRAWR +130,0,1
1160 PLOT 200,255-t,1:DRAWR 0,-25,1
1170 PLOT 415,255-h,1:DRAWR 0,-25,1
1180 DRAW 200,228-t,1
1190 PLOT 300,230,1:DRAWR 0,-25,1
1200 PEN 3
1210 LOCATE 12,8+(t=-16)-(t=16):PRINT masse$
1220 LOCATE 24,7+(h=-16)-(h=16):PRINT masse$+m
asse$
1230 LOCATE 24,8+(h=-16)-(h=16):PRINT masse$+m
asse$
1240 LOCATE 12,9+(t=-16)-(t=16):PRINT masse$
1250 LOCATE 24,9+(h=-16)-(h=16):PRINT masse$+m
asse$
1260 PEN 1
1270 LOCATE 25,8+(h=-16)-(h=16):PRINT USING "#
###";poids
1280 RETURN
```

# 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, AMSTRAD CPC 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

**90 à 180** Initialisation et présentation du jeu.

**220 à 310** Première séquence.

*Ligne 240* : Un nombre est tiré au sort dans un encadrement différent selon le niveau.

*Ligne 250* : La variable aléatoire X inscrit le nombre sur l'écran à une place chaque fois différente.

*Lignes 260 à 270* : Pour chaque nombre, une fenêtre est déterminée sur 20 colonnes de large et 2 (ou 4 version 3) de haut. Le nombre s'inscrit 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.

*Lignes 290 à 310* : C'est le joueur qui décide du moment où l'on passe à la suite.

### 350 à 410

Affichage de la liste moins un nombre.

*Ligne 360* : 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.

### 450 à 470

Réponse - Test de la réponse.

Affichage bonne réponse, en cas d'erreur.

Affichage « gagné ».

### 480 à 490

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

### 500 à 510

Affichage du score et fin de jeu.

## Particularités techniques

La séquence aléatoire n'est pas initialisée, elle est laissée au bon plaisir de l'AMSTRAD CPC.

Dans la première séquence, chaque nombre tiré au sort est affecté d'un numéro d'ordre A(I), ce qui permet de l'afficher de nouveau dans la deuxième séquence (le nombre retiré est remplacé par « ? », ligne 400).

## 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.

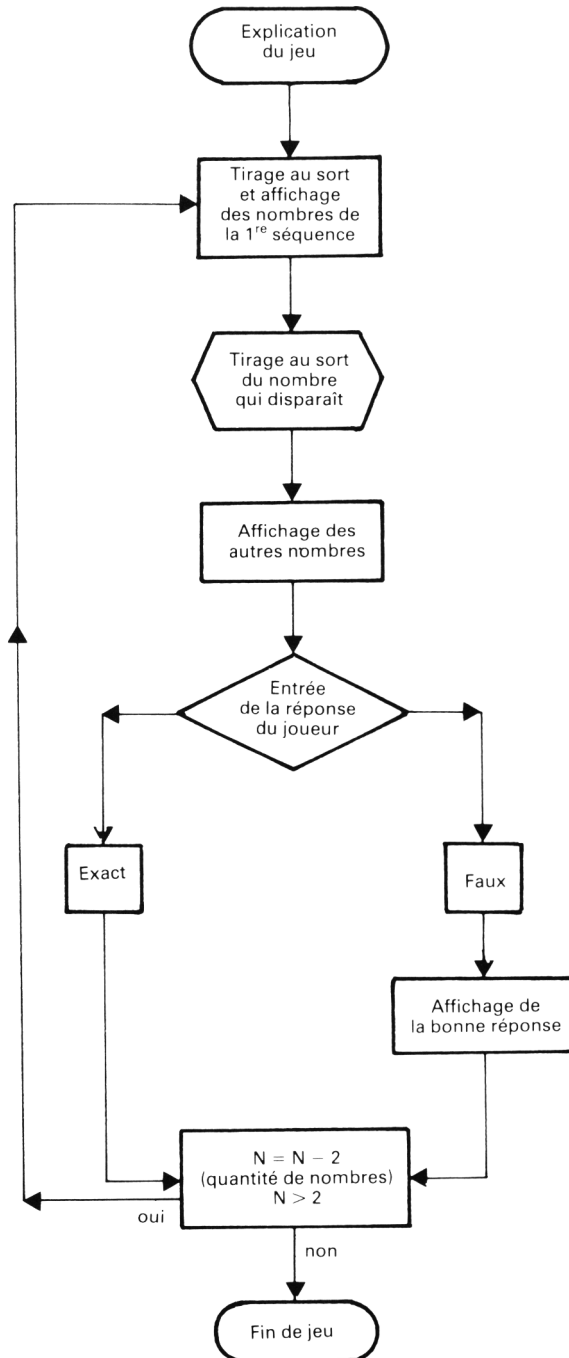
Des jeunes enfants (version facile) peuvent être gênés par l'affichage du même nombre plusieurs fois (ligne 240 :  $A(l)$  est déterminé par RND). Si vous voulez éviter cette répétition, vous pouvez ajouter entre les lignes 240 et 250 un petit programme qui va comparer le nombre  $A(l)$  à ceux déjà sortis (voir le programme « Construction de phrases »).

## 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.

Programme Kim facile  
 Programme Kim moyen  
 Programme Kim difficile



Organigramme

```

10 REM -----
20 REM -- KIM NOMBRES FACILE --
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin jan 86 ---
50 REM -----
60 REM -----
70 REM -> Module presentation
80 REM -----
90 DIM a(14):n=10:br=0
100 MODE 1:PAPER 13:CLS:PEN 0
110 LOCATE 12,5:PRINT "JEU DE KIM"
120 LOCATE 1,8:PRINT "Tu vas voir des nombres
s'inscrire sur"
130 PRINT " l'écran, regarde les bien":PRINT
140 PRINT "Ensuite les nombres vont changer de
":PRINT" place et un d'entre eux va disparaître"
150 PRINT:PRINT "A toi de le reconnaître"
160 LOCATE 1,22:PRINT "Quand tu seras prêt à jouer"
170 PRINT "Appuie sur une touche"
180 i$=INKEY$:IF i$="" THEN 180
190 REM -----
200 REM -> Première sequence
210 REM -----
220 CLS
230 FOR i=1 TO n
240 a(i)=INT(RND*9)+1
250 x=INT(RND*15)+1
260 IF i<=5 THEN LOCATE x,i*4 ELSE LOCATE x+18
,(i-5)*4
270 PRINT a(i)
280 NEXT i
290 LOCATE 4,23:PRINT "QUAND TU AURAS BIEN LU
CES NOMBRES"
300 LOCATE 10,24:PRINT "APPUIE SUR UNE TOUCHE"
310 i$=INKEY$:IF i$="" THEN 310
320 REM -----
330 REM -> Disparition d'un nombre
340 REM -----
350 CLS
360 f=INT(RND*(n-1))+1
370 FOR i=1 TO n
380 x=INT(RND*14)+3
390 IF i<=5 THEN LOCATE x,i*4 ELSE LOCATE x+18
,(i-5)*4
400 IF i=f THEN PRINT "?" ELSE PRINT a(i)
410 NEXT i
420 REM -----

```

```

430 REM -> module reponse
440 REM -----
450 LOCATE 1,23:INPUT "QUEL NOMBRE A DISPARU "
;rep
460 IF rep=a(f) THEN PRINT "BRAVO TU AS GAGNE"
;br=br+1 ELSE PRINT "NON, C'ETAIT";a(f)
470 FOR i=1 TO 2000:NEXT i
480 n=n-2
490 IF n<>2 THEN 220
500 CLS:LOCATE 12,5:PRINT "C'est termine":LOCA
TE 5,7:PRINT "Tu as trouve";br;"nombre(s) cach
es"
510 LOCATE 1,24:END

```

```

10 REM -----
20 REM -- KIM NOMBRES MOYEN --
30 REM --- D.Nielsen nov 83 ---
40 REM --- Augustin jan 86 ---
50 REM -----
60 REM -----
70 REM -> Module presentation
80 REM -----
90 DIM a(14):n=14:br=0
100 MODE 1:PAPER 13:CLS:PEN 0
110 LOCATE 12,5:PRINT "JEU DE KIM"
120 LOCATE 1,8:PRINT "Tu vas voir des nombres
s'inscrire sur"
130 PRINT " l'ecran, regarde les bien":PRINT
140 PRINT "Ensuite les nombres vont changer de
":PRINT" place et un d'entre eux va disparaitr
e"
150 PRINT:PRINT "A toi de le reconnaitre"
160 LOCATE 1,22:PRINT "Quand tu seras pret a j
ouer"
170 PRINT "Appuie sur une touche"
180 i$=INKEY$:IF i$="" THEN 180
190 REM -----
200 REM -> Premiere sequence
210 REM -----
220 CLS
230 FOR i=1 TO n
240 a(i)=INT(RND*20)+1
250 x=INT(RND*15)+2
260 IF i<=7 THEN LOCATE x,i*2 ELSE LOCATE x+18
,(i-7)*2
270 PRINT a(i)
280 NEXT i

```

```

290 LOCATE 4,23:PRINT "QUAND TU AURAS BIEN LU
CES NOMBRES"
300 LOCATE 10,24:PRINT "APPUIE SUR UNE TOUCHE"
310 i#=INKEY#:IF i#="" THEN 310
320 REM -----
330 REM -> Disparition d'un nombre
340 REM -----
350 CLS
360 f=INT(RND*(n-1))+1
370 FOR i=1 TO n
380 x=INT(RND*14)+3
390 IF i<=7 THEN LOCATE x+18,16-i*2 ELSE LOCAT
E x,16-(i-7)*2
400 IF i=f THEN PRINT "?" ELSE PRINT a(i)
410 NEXT i
420 REM -----
430 REM -> module reponse
440 REM -----
450 LOCATE 1,23:INPUT "QUEL NOMBRE A DISPARU "
;rep
460 IF rep=a(f) THEN PRINT "BRAVO TU AS GAGNE"
;br=br+1 ELSE PRINT "NON, C'ETAIT";a(f)
470 FOR i=1 TO 2000:NEXT i
480 n=n-2
490 IF n<>2 THEN 220
500 CLS:LOCATE 12,5:PRINT "C'est termine":LOCA
TE 5,7:PRINT "Tu as trouve";br;"nombre(s) cach
es"
510 LOCATE 1,24:END

```

```

10 REM -----
20 REM -- KIM NOMBRES DIFFICILE --
30 REM ---- D.Nielsen nov 83 ----
40 REM ---- Augustin jan 86 ----
50 REM -----
60 REM -----
70 REM -> Module presentation
80 REM -----
90 DIM a(20):n=20;br=0
100 MODE 1:PAPER 13:CLS:PEN 0
110 LOCATE 12,5:PRINT "JEU DE KIM"
120 LOCATE 1,8:PRINT "Tu vas voir des nombres
s'inscrire sur"
130 PRINT " l'écran, regarde les bien":PRINT
140 PRINT "Ensuite les nombres vont changer de
":PRINT" place et un d'entre eux va disparaîtr
e"

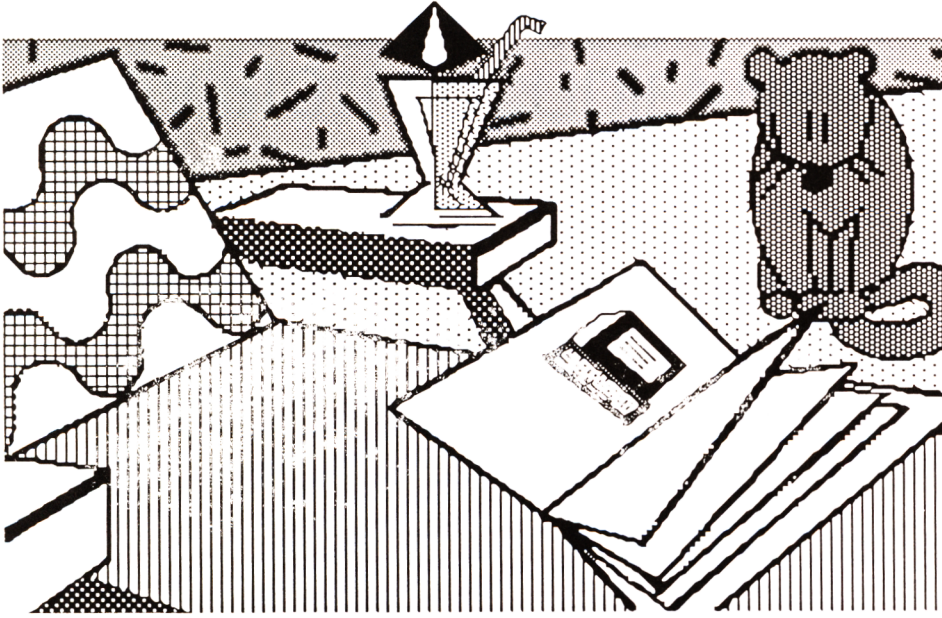
```

```

150 PRINT:PRINT "A toi de le reconnaitre"
160 LOCATE 1,22:PRINT "Quand tu seras pret a j
ouer"
170 PRINT "Appuie sur une touche"
180 i#=INKEY#:IF i#="" THEN 180
190 REM -----
200 REM -> Premiere sequence
210 REM -----
220 CLS
230 FOR i=1 TO n
240 a(i)=INT(RND*50)+1
250 x=INT(RND*14)+3
260 IF i<=10 THEN LOCATE x,i*2 ELSE LOCATE x+1
8,(i-10)*2
270 PRINT a(i)
280 NEXT i
290 LOCATE 4,23:PRINT "QUAND TU AURAS BIEN LU
CES NOMBRES"
300 LOCATE 10,24:PRINT "APPUIE SUR UNE TOUCHE"
310 i#=INKEY#:IF i#="" THEN 310
320 REM -----
330 REM -> Disparition d'un nombre
340 REM -----
350 CLS
360 f=INT(RND*(n-1))+1
370 FOR i=1 TO n
380 x=INT(RND*14)+3
390 IF i<=5 THEN LOCATE x+18,10+i*2
400 IF i>5 AND i<=10 THEN LOCATE x,22-i*2
410 IF i>10 AND i<=15 THEN LOCATE x,42-i*2
420 IF i>15 THEN LOCATE x+18,22-(i-10)*2
430 IF i=f THEN PRINT "?" ELSE PRINT a(i)
440 NEXT i
450 REM -----
460 REM -> module reponse
470 REM -----
480 LOCATE 1,23:INPUT "QUEL NOMBRE A DISPARU "
;rep
490 IF rep=a(f) THEN PRINT "BRAVO TU AS GAGNE"
;br=br+1 ELSE PRINT "NON, C'ETAIT";a(f)
500 FOR i=1 TO 2000:NEXT i
510 n=n-2
520 IF n<>2 THEN 220
530 CLS:LOCATE 12,5:PRINT "C'est termine":LOCA
TE 5,7:PRINT "Tu as trouve";br;"nombre(s) cach
es"
540 LOCATE 1,24:END

```



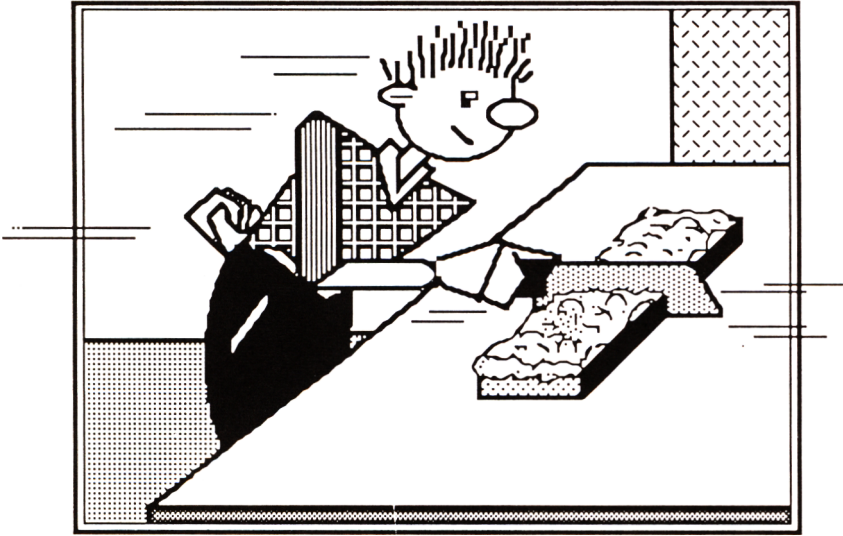


## PETIT CAHIER DE VACANCES SUR AMSTRAD

Tu as envie de jouer avec ton micro et de voir si tu n'as pas complètement oublié l'école ? Ouvre vite ton petit cahier de vacances pour y découvrir 10 jeux. Ces dix sujets de calcul et de français t'aideront à réviser ces deux matières en t'amusant. Tu verras, les programmes sont courts et les résultats ne se font pas attendre...



# VOUS AVEZ DIT DIVISION EUCLIDIENNE ?



**T**u vas rechercher le quotient entier et le reste d'une division.

## Comment vas-tu t'y prendre ?

L'ordinateur fait toutes les opérations que tu lui demandes. Mais sa précision de calcul peut parfois être un défaut : si tu poses une division, il recherchera jusqu'à 6 chiffres décimaux...

« Combien de livres à 8 F puis-je acheter avec 257 francs ? Combien me restera-t-il ? » Ce problème, il ne sait pas le résoudre tout seul... Il sait seulement rechercher le quotient entier et calculer le reste.

Voici la solution que je te propose.

Tu entres le dividende (nombre à diviser) sous le nom de variable N, et le diviseur sous le nom de D.

L'opérateur \ donne, ligne 40, le quotient entier. MOD renvoie, à la ligne 50, le reste de la division de N par D.

Tu aurais pu calculer ces deux valeurs autrement :

$Q = \text{INT}(N/D)$  (partie entière du quotient)

$R = N - (Q * D)$  (c'est ce que l'on fait, dans l'opération, pour calculer le reste).

```

5 REM ----- QUOTIENT ENTIER -----
10 MODE 1:CLS
20 INPUT "ENTREZ LE NOMBRE A DIVISER ";N
30 INPUT "ENTREZ LE DIVISEUR ";D
40 Q=N\D
50 R=N MOD D
60 PRINT "LE QUOTIENT ENTIER DE";N;"DIVISE PAR
";D;"EST";Q
70 PRINT "LE RESTE EST";R

```

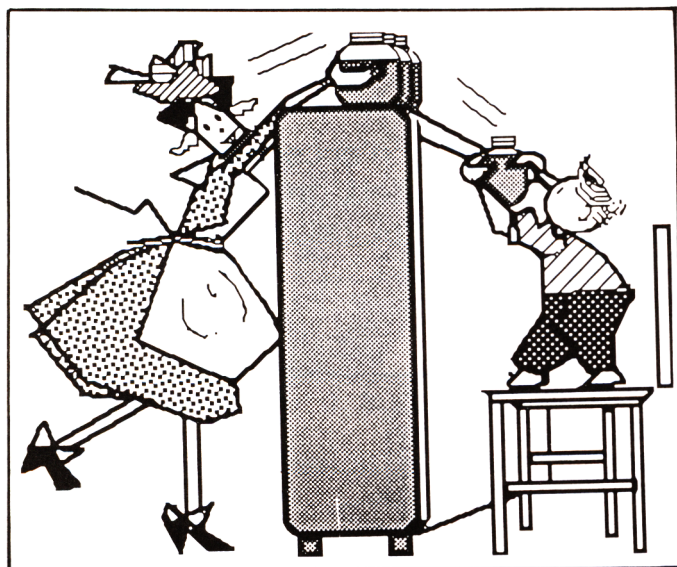
Que vois-tu à l'écran ?

```

ENTREZ LE NOMBRE A DIVISER ? 36
ENTREZ LE DIVISEUR ? 5
LE QUOTIENT ENTIER DE 36 DIVISE PAR 5
EST 7
LE RESTE EST 1

```

# HISTOIRES DE MULTIPLES ET DE DIVISEURS



**T**u vas rechercher tous les diviseurs d'un nombre entré au clavier.

Comment vas-tu t'y prendre ?

Un nombre  $l$  est diviseur d'un nombre  $A$  (ou  $A$  est multiple de  $l$ ) si leur quotient est un nombre entier, donc si le reste de la division euclidienne de  $A$  par  $l$  est 0.

## Voici la solution que je te propose :

La variable I va prendre toutes les valeurs de 1 à A (boucle de la ligne 30).

Dans la ligne 40, l'opérateur MOD donne le reste de la division euclidienne de A par I.

Si ce reste est nul ( $R=0$ ), on peut afficher que A est multiple de I (ou que I est un diviseur de A).

Attention : ton Amstrad n'acceptera pas de nombre supérieur à 32767, et au-delà de 1000, même s'il calcule beaucoup plus vite que toi, cela devient long, long...

```

5 REM ---- DIVISEURS D'UN NOMBRE ----
10 MODE 1:CLS
20 INPUT "DONNE UN NOMBRE ";A
30 FOR I=1 TO A
40 R=A MOD I
50 IF R=0 THEN PRINT A;"EST MULTIPLE DE";I
60 NEXT I

```

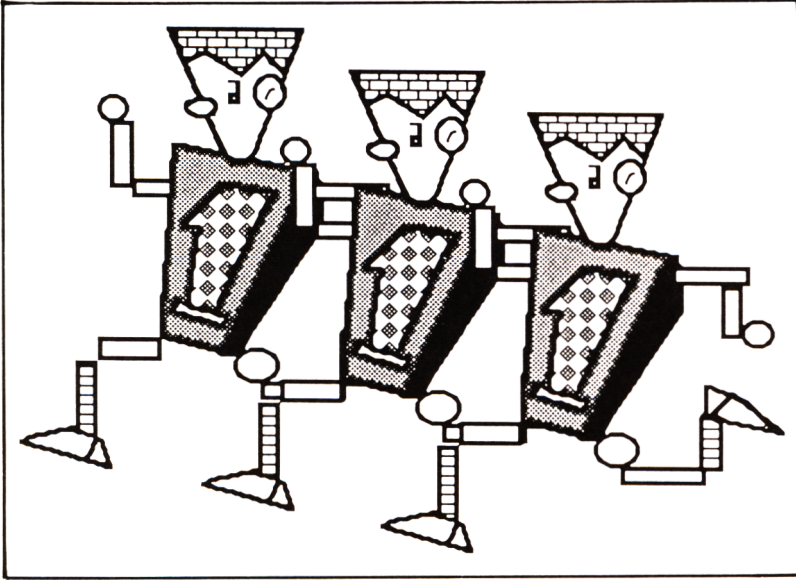
## Que vois-tu à l'écran ?

```

DONNE UN NOMBRE ? 75
75 EST MULTIPLE DE 1
75 EST MULTIPLE DE 3
75 EST MULTIPLE DE 5
75 EST MULTIPLE DE 15
75 EST MULTIPLE DE 25
75 EST MULTIPLE DE 75

```

# AU COMMENCEMENT ÉTAIENT LES NOMBRES PREMIERS



**T**u vas rechercher tous les nombres premiers entre 1 et 100.

Comment vas-tu t'y prendre ?

Un nombre  $l$  est un nombre premier s'il n'est divisible que par 1 et lui-même. Il faut donc rechercher tous les nombres qui n'ont pas d'autres diviseurs (aide-toi du programme précédent).

## Voici la solution que je te propose :

Le programme se réalise entre deux boucles :

1° pour  $I = 1$  à 100 (le programme va tester chaque nombre)

2° pour chaque valeur de  $I$ , tu cherches ses diviseurs (regarde le programme précédent) et tu comptabilises le nombre de diviseurs dans la variable DIVIS.

Si  $DIVIS=2$ , c'est que ce nombre n'a que deux diviseurs (1 et lui-même). Il s'agit donc d'un nombre premier.

Tu pourrais augmenter la valeur maximum dans la boucle de la ligne 20 (de 1 à 1000, par exemple), mais cela devient très long... Alors, pour gagner du temps, tu peux aussi éliminer un nombre dès qu'il a 3 diviseurs, sans les rechercher tous, en ajoutant :

65 IF DIVIS>2 THEN 90

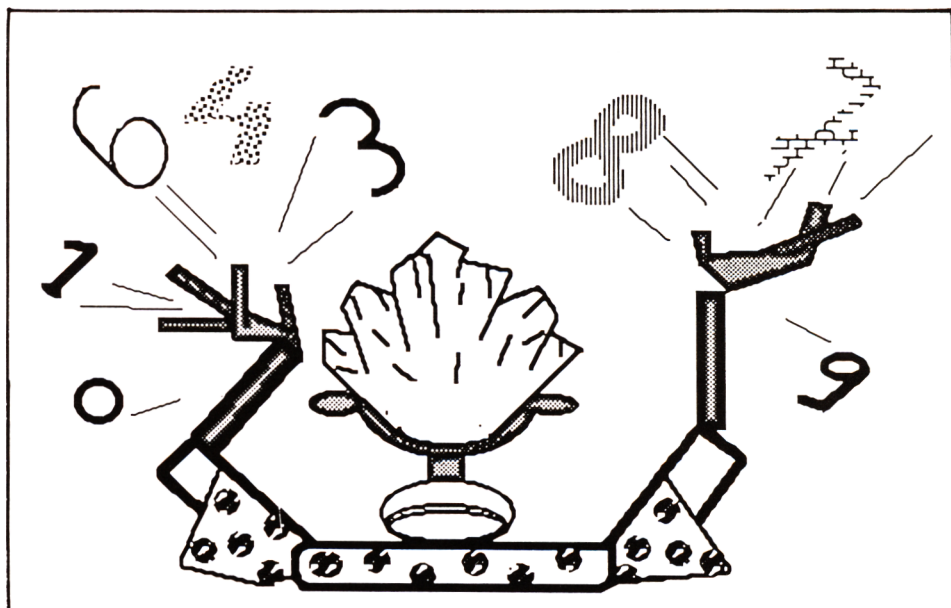
```

5 REM ----- NOMBRES PREMIERS -----
10 MODE 1:CLS
20 FOR I=1 TO 100
30 DIVIS=0
40 FOR K=1 TO I
50 R=I MOD K
60 IF R=0 THEN DIVIS=DIVIS+1
70 NEXT K
80 IF DIVIS=2 THEN PRINT I;"EST UN NOMBRE PREM
IER"
90 NEXT I
```

## Que vois-tu à l'écran ?

2	EST	UN	NOMBRE	PREMIER
3	EST	UN	NOMBRE	PREMIER
5	EST	UN	NOMBRE	PREMIER
7	EST	UN	NOMBRE	PREMIER
11	EST	UN	NOMBRE	PREMIER
13	EST	UN	NOMBRE	PREMIER
17	EST	UN	NOMBRE	PREMIER
19	EST	UN	NOMBRE	PREMIER
23	EST	UN	NOMBRE	PREMIER
29	EST	UN	NOMBRE	PREMIER
31	EST	UN	NOMBRE	PREMIER
37	EST	UN	NOMBRE	PREMIER
41	EST	UN	NOMBRE	PREMIER
43	EST	UN	NOMBRE	PREMIER
47	EST	UN	NOMBRE	PREMIER
53	EST	UN	NOMBRE	PREMIER
59	EST	UN	NOMBRE	PREMIER
61	EST	UN	NOMBRE	PREMIER
67	EST	UN	NOMBRE	PREMIER
71	EST	UN	NOMBRE	PREMIER
73	EST	UN	NOMBRE	PREMIER
79	EST	UN	NOMBRE	PREMIER
83	EST	UN	NOMBRE	PREMIER
89	EST	UN	NOMBRE	PREMIER
97	EST	UN	NOMBRE	PREMIER

## A PROPOS DE TRI DE NOMBRES



**T**u vas classer des nombres du plus grand au plus petit.

### Comment vas-tu t'y prendre ?

Pour classer des nombres, plusieurs techniques sont possibles. Je t'en propose deux, les plus simples (mais elles ne seraient pas les plus rapides s'il y avait beaucoup de nombres à trier).

1° – **par élimination** : tu regardes tous les nombres. Tu prends le plus grand pour le mettre à part. Tu regardes ceux qui restent et tu recommences autant de fois qu'il y a de nombres à tirer.

2° – **par permutation** : tu regardes deux nombres qui se suivent dans la liste. S'il ne sont pas dans l'ordre correct, tu les changes de place entre eux. Tu recommences avec les autres nombres, jusqu'à ce que tous les nombres soient correctement classés.

**Cette fois-ci, je te propose deux solutions :**

– Partie commune (10 à 60) :

Tu vas d'abord demander le nombre d'éléments à classer. Au fur et à mesure de leur entrée au clavier, les nombres sont mis en mémoire dans une variable indicée A(I), I représentant le numéro d'entrée du nombre dans la liste.

**Attention :** si tu veux classer des listes de plus de 10 nombres, il te faudra auparavant dimensionner un tableau A en ajoutant :  
15 DIM A(30) par exemple, pour un maximum de 30 nombres.

### 1<sup>re</sup> solution

– tri par élimination : tu recherches le plus grand des nombres de la liste et tu le gardes dans MAXI (lignes 90 à 110), tu affiches ce nombre, puis tu l'élimines de la liste :  
 $A(T) = 0$ .

Cette recherche est effectuée autant de fois qu'il y a de nombres : boucle FOR I = 1 TO NB de la ligne 70 à la ligne 130.

```

5 REM ----- TRI DE NOMBRES -----
10 MODE 1:CLS
20 INPUT "COMBIEN DE NOMBRES A CLASSER ";NB
30 FOR I=1 TO NB
40 PRINT "NOMBRE No";I;:INPUT A(I)
50 NEXT I
60 PRINT "VOICI LES NOMBRES DU PLUS GRAND", "AU
  PLUS PETIT"
70 FOR I=1 TO NB
80 MAXI=0:T=0
90 FOR K=1 TO NB
100 IF A(K)>MAXI THEN MAXI=A(K):T=K
110 NEXT K
120 PRINT MAXI:A(T)=0
130 NEXT I

```

Que vois-tu à l'écran ?

```

COMBIEN DE NOMBRES A CLASSER ? 8
NOMBRE No 1 2 3 4 5 6 7 8
NOMBRE No 2 3 4 5 6 7 8 1
NOMBRE No 3 4 5 6 7 8 1 2
NOMBRE No 4 5 6 7 8 1 2 3
NOMBRE No 5 6 7 8 1 2 3 4
NOMBRE No 6 7 8 1 2 3 4 5
NOMBRE No 7 8 1 2 3 4 5 6
NOMBRE No 8 1 2 3 4 5 6 7
VOICI LES NOMBRES DU PLUS GRAND
AU PLUS PETIT
8
7
6
5
4
3
2
1

```

2<sup>e</sup> solution

– tri par permutation : PERM est une variable de contrôle qui indique si l'ordinateur a fait une ou plusieurs permutations (PERM = 1).

Dans les lignes 80 à 100, tu compares chaque nombre à celui qui le précède : s'il est plus grand, tu les changes tous les deux de place (ligne 90).

S'il y a eu des permutations, tu recommences pour contrôler l'ordre (ligne 110). Si PERM = 0, c'est que tous les nombres sont dans le bon ordre, tu peux alors les afficher.

Cette deuxième solution est plus rapide et permet ensuite de réutiliser les nombres triés dans un programme car ils sont, non seulement rangés dans le bon ordre sur l'écran, mais aussi dans la mémoire de l'ordinateur.

```

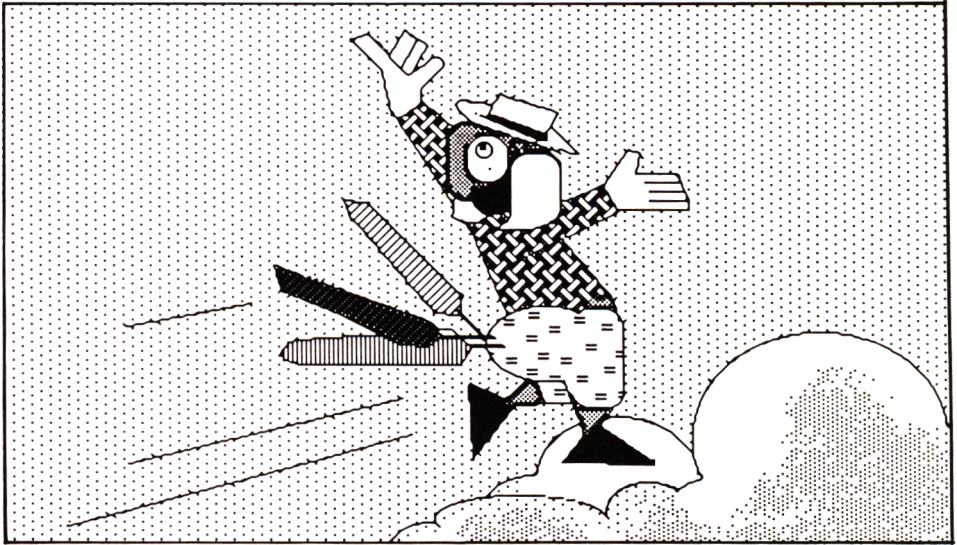
5 REM ----- TRI DE NOMBRES VERSION 2 -----
10 MODE 1:CLS
20 INPUT "COMBIEN DE NOMBRES A CLASSER ";NB
30 FOR I=1 TO NB
40 PRINT "NOMBRE No";I;:INPUT A(I)
50 NEXT I
60 PRINT "VOICI LES NOMBRES DU PLUS GRAND","AU
  PLUS PETIT"
70 PERM=0
80 FOR I=2 TO NB
90 IF A(I)>A(I-1) THEN R=A(I):A(I)=A(I-1):A(I-
  1)=R:PERM=1
100 NEXT I
110 IF PERM=1 THEN 70
120 FOR I=1 TO NB
130 PRINT A(I)
140 NEXT I
    
```

Que vois-tu à l'écran ?

```

COMBIEN DE NOMBRES A CLASSER ? 8
NOMBRE No 1 ? 11
NOMBRE No 2 ? 0
NOMBRE No 3 ? 88
NOMBRE No 4 ? 110
NOMBRE No 5 ? 45
NOMBRE No 6 ? 3
NOMBRE No 7 ? 4
NOMBRE No 8 ? 76
VOICI LES NOMBRES DU PLUS GRAND
AU PLUS PETIT
110
88
76
45
11
4
3
0
    
```

# BUENOS AIRES



**T**u vas calculer automatiquement l'aire d'une surface géométrique simple (carré, rectangle, triangle, parallélogramme)

Comment vas-tu t'y prendre ?

Comme toi, Amstrad peut calculer l'aire d'une surface géométrique s'il sait de quelle surface il s'agit, s'il connaît la ou les dimensions utiles ainsi que la formule de calcul.

## Voici la solution que je te propose :

Ce programme est un peu plus compliqué que les précédents car j'ai voulu tenir compte des unités de mesure pour éviter de calculer une aire à partir de dimensions données dans des unités différentes.

```

5 REM ---- AIRES ----
10 MODE 1:CLS
20 FOR I=1 TO 4
30 READ N$(I),D1$(I),D2$(I)
40 PRINT I;".....";N$(I)
50 NEXT I
60 PRINT:INPUT "VOTRE CHOIX ";CH:IF CH<1 OR CH
>4 THEN 60
70 PRINT "QUELLE EST LA MESURE DE ";D1$(CH);" (
valeur + unite)";:INPUT M1$:M1=VAL(M1$)
80 FOR I=1 TO LEN(M1$):N$=MID$(M1$,I,1):IF N$=
CHR$(32) THEN UN1$=RIGHT$(M1$,LEN(M1$)-1)
90 NEXT I
100 IF CH=1 THEN 150
110 PRINT "QUELLE EST LA MESURE DE ";D2$(CH);"
(valeur + unite)";:INPUT M2$:M2=VAL(M2$)
120 FOR I=1 TO LEN(M2$):N$=MID$(M2$,I,1):IF N$
=CHR$(32) THEN UN2$=RIGHT$(M2$,LEN(M2$)-1)
130 NEXT I
140 IF UN1$<>UN2$ THEN PRINT "ATTENTION, UNITE
S DIFFERENTES":GOTO 70
150 ON CH GOSUB 200,210,220,230
160 PRINT "LA SURFACE DE CE ";N$(CH);" EST";S;
UN1$;"2"
170 END
200 S=M1*M1:RETURN
210 S=M1*M2:RETURN
220 S=M1*M2/2:RETURN
230 S=M1*M2:RETURN
300 DATA CARRE,SON COTE,SON COTE
310 DATA RECTANGLE,SA LONGUEUR,SA LARGEUR
320 DATA TRIANGLE,SA BASE,SA HAUTEUR CORRESPON
DANTE
330 DATA PARALLELOGRAMME,L'UN DES COTES,LA HAU
TEUR CORRESPONDANTE

```

Les lignes 20 à 50 lisent en DATA les noms des figures, et le nom des deux dimensions utiles pour calculer leur aire (attention : pour le carré, on a écrit deux fois COTE pour éviter un décalage dans la lecture des DATA).

La ligne 60 permet de choisir la figure.

Les lignes 70 à 110 (70 seulement pour le carré) permettent d'entrer chaque dimension avec son nom d'unité : par exemple 35 CM . M1 et M2 prendront les valeurs numériques de ces dimensions.

Les lignes 80, 90, 120, 130 et 140 contrôlent qu'il s'agit bien de la même unité : tu recherches le nom des unités à droite de M1\$ et M2\$ en te basant sur l'espace entre le nombre et cette unité. S'ils ne correspondent pas, il faut recommencer.

La ligne 150 envoie, selon la figure à l'un des sous-programmes 200 à 230 pour calculer l'aire.

Il ne reste plus qu'à afficher ce résultat par la ligne 160.

Tu peux adapter ce programme à d'autres calculs :

- en ajoutant d'autres figures (et leurs mesures utiles) en DATA, ainsi que la formule de calcul.
- pour calculer des périmètres, ou des volumes.

## Que vois-tu à l'écran ?

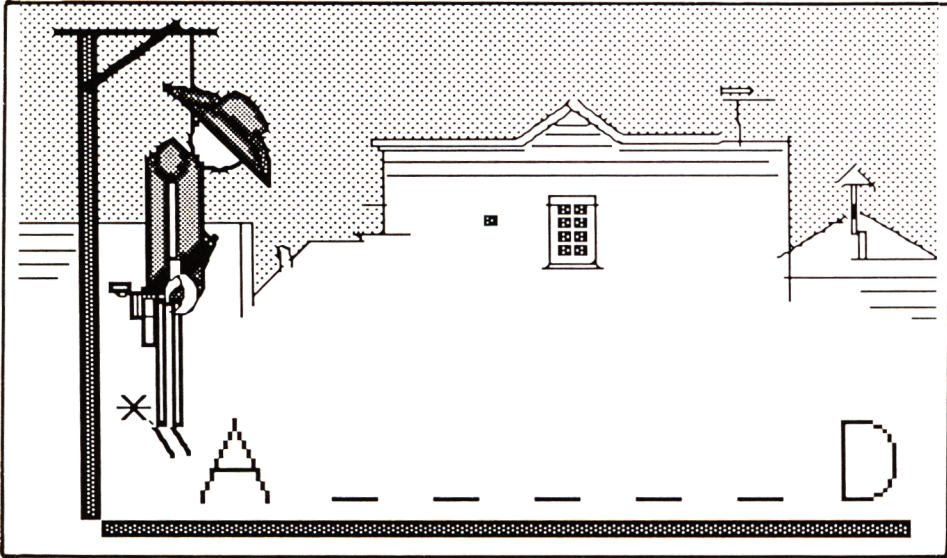
```

1 .....CARRE
2 .....RECTANGLE
3 .....TRIANGLE
4 .....PARALLELOGRAMME

VOTRE CHOIX ? 3
QUELLE EST LA MESURE DE SA BASE
(valeur + unite)? 2 cm
QUELLE EST LA MESURE DE
SA HAUTEUR CORRESPONDANTE
(valeur + unite)? 3 cm
LA SURFACE DE CE TRIANGLE EST 3 cm2

```

# LE PAUVRE PENDU



**T**u va deviner et reconstituer un mot lettre par lettre

Comment vas-tu t'y prendre ?

C'est un des jeux les plus classiques qui soit ! De nombreuses versions existent sur ordinateur avec des dessins de plus ou moins bon goût.

Tu te contenteras de mettre au point le cœur du programme, sans graphisme.

Au cours du jeu, il faut comparer la lettre proposée par le joueur à toutes les lettres du mot : une fonction sera alors très utile, c'est MID\$.

Mais, attention, pour que le jeu tourne correctement, il faudra tenir compte du nombre de lettres fausses et de lettres exactes pour arrêter le jeu lorsque le mot est trouvé ou qu'il y a trop de mauvaises réponses.

### Voici la solution que je te propose :

Le mot à découvrir est enregistré dans les lignes 30 et 50. Il est pris, lettre à lettre, par l'instruction INKEY\$, ce qui permet de ne pas l'afficher sur l'écran. Le caractère " " (blanc ou espace), indique que le mot est terminé.

Les lignes 60 à 80 affichent autant d'étoiles qu'il y a de lettres dans le mot.

Le jeu se déroule entre les lignes 90 et 180 : la lettre entrée par le joueur est comparée à chaque lettre du mot (ligne 130). Si elle correspond à une de ses lettres elle est affichée à sa place dans le mot, la variable A(I) prend la valeur 1 pour éviter que la même réponse ne soit prise en compte deux fois, le compteur BR enregistre le nombre de lettres trouvées et la variable REP prend la valeur 1 (réponse exacte). T correspond au nombre de lettres identiques.

La ligne 150 affiche un message lorsque la lettre n'est pas dans le mot et compte le nombre de réponses fausses (RF).

Les lignes 160 et 170 servent de test de fin de jeu : 160  
 $BR=L-T$  : le mot est trouvé.

170  $RF = 10$  : Dix réponses fausses, vous êtes pendu !

Si tu veux conserver le réalisme du jeu, il ne te reste plus qu'à dessiner l'échafaud, le bourreau, et ... le pendu !

```

5 REM ---- LE PENDU ----
10 MODE 1:CLS
20 PRINT "ENTREZ LE MOT A DECOUVRIR","PUIS UN
ESPACE POUR VALIDER LE MOT"
30 I$=INKEY$:I$=UPPER$(I$):IF I$="" THEN 30
40 IF I$=" " THEN 60
50 MOT$=MOT$+I$:L=L+1:GOTO 30
60 FOR I=1 TO L
70 LOCATE I*2,5:PRINT "*"
80 NEXT I
90 LOCATE 1,8:PRINT "TAPEZ UNE LETTRE":REP=0
100 I$=INKEY$:I$=UPPER$(I$):IF I$="" THEN 100
110 LOCATE 1,8:PRINT SPACE$(39)
120 FOR I=1 TO L
130 IF I$=MID$(MOT$,I,1) AND A(I)=0 THEN LOCAT
E I*2,5:PRINT I$:A(I)=1:BR=BR+1:IF REP=1 THEN
BR=BR-1:REP=1:T=T+1 ELSE REP=1
140 NEXT I
150 IF REP=0 THEN LOCATE 1,10+RF:PRINT I$;" n
est pas dans ce mot":RF=RF+1
160 IF BR=L-T THEN LOCATE 1,22:PRINT "BRAVO !
MOT TROUVE EN";BR+RF;"COUPS":END
170 IF RF=10 THEN LOCATE 1,22:PRINT "TROP TAR
D ! LE MOT ETAIT ";MOT$:END
180 GOTO 90

```

Que vois-tu à l'écran ?

**ENTREZ LE MOT A DECOUVRIR  
 PUIS UN ESPACE POUR VALIDER LE MOT**

**A U G U S T I N**

**E n'est pas dans ce mot  
 O n'est pas dans ce mot  
 Y n'est pas dans ce mot  
 B n'est pas dans ce mot  
 M n'est pas dans ce mot**

**BRAVO ! MOT TROUVE EN 12 COUPS**

## LA PHRASE ACCORDÉON



**J**eu pour 7 joueurs : chacun entre un morceau de phrase, sans savoir ce que les autres ont écrit. La phrase complète est dévoilée fin de jeu.

### Comment vas-tu t'y prendre ?

Ce programme, sur un thème cher aux surréalistes (le cadavre exquis de la grammaire), ne doit pas poser de problème de développement. Il est basé sur la concaténation d'éléments de phrases (ce mot barbare signifie simplement « coller côte à côte »).

Mais, afin d'obtenir une phrase grammaticalement correcte, tu dois indiquer à chaque joueur le type d'élément qu'il devra ajouter (sujet, verbe, complément d'objet, etc.).

Voici la solution que je te propose :

```

5 REM ---- LA PHRASE ACCORDEON ----
10 MODE 1:CLS
20 FOR I=1 TO 7
30 READ F$(I)
40 CLS:PRINT "TU DOIS ECRIRE LE ";F$(I);" DE L
A PHRASE"
50 LINE INPUT "ENTRE TON TEXTE ";ELEM$(I)
60 PH$=PH$+CHR$(32)+ELEM$(I)
70 NEXT I
80 CLS:LOCATE 1,5:PRINT "VOICI LA PHRASE QUE V
OUS AVEZ CONSTRUITE"
90 PRINT PH$
100 END
200 DATA COMPLEMENT CIRCONSTANCIEL DE LIEU
210 DATA COMPLEMENT CIRCONSTANCIEL DE TEMPS
220 DATA SUJET (a la 3eme personne du singulier)
230 DATA VERBE (a la 3eme personne du singulier)
240 DATA COMPLEMENT D'OBJET DIRECT
250 DATA COMPLEMENT D'OBJET INDIRECT
260 DATA COMPLEMENT CIRCONSTANCIEL DE BUT

```

Les lignes 20 à 70 lisent et affichent la définition du type d'élément à ajouter par le joueur. L'instruction CLS de la ligne 40 permet d'éviter que chaque joueur ne voie le texte de ses prédécesseurs. L'élément I est entré dans la ligne 50 par LINEINPUT qui permet d'utiliser des virgules dans un élément.

La ligne 60 effectue cette fameuse concaténation, en ajoutant un espace entre chaque élément.

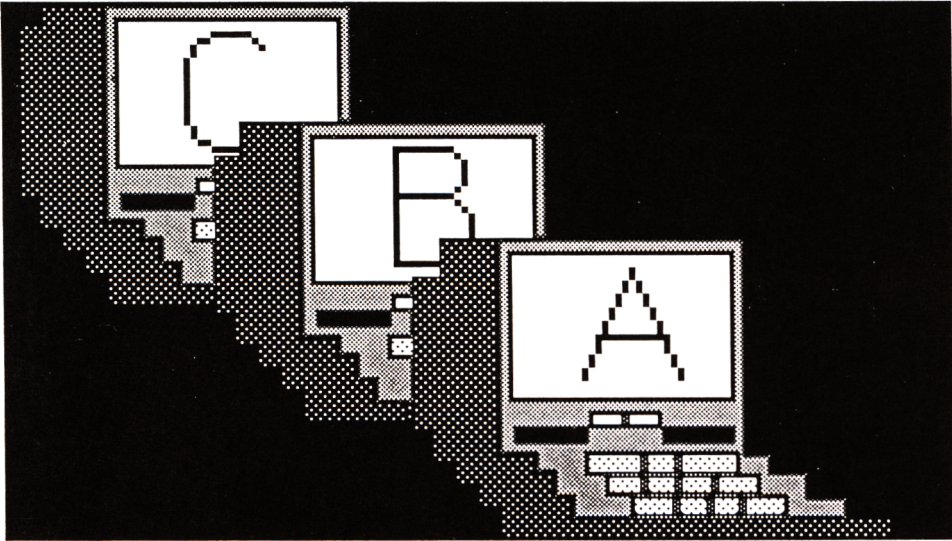
La phrase complète est affichée par la ligne 90.

Les types d'éléments sont inscrits en DATA. J'ai choisi une structure de phrase que tu peux évidemment modifier selon ton choix.

Que vois-tu à l'écran ?

```
TU DOIS ECRIRE LE  
COMPLEMENT CIRCONSTANCIEL DE LIEU  
DE LA PHRASE  
ENTRE TON TEXTE ? dans une salle  
sombre
```

# PAR ORDRE ALPHABÉTIQUE



**T**u vas classer une liste de mots dans l'ordre alphabétique.

Comment vas-tu t'y prendre ?

Pour ton Amstrad, un mot est en fait formé de chiffres ! Ce sont les codes "ASCII" de chaque lettre qu'il conserve en mémoire. Alors pour lui, classer des mots par ordre alphabétique ressemble beaucoup à trier des nombres, du plus petit à plus grand. Révise le programme de tri de nombres et tu trouveras vite la solution !

## Voici la solution que je te propose :

Le programme de classement par ordre alphabétique ressemble en tout point à la deuxième version du programme de tri de nombres. Simplement, au lieu de comparer des variables "nombre", tu compares des variables "chaîne alphabétique" et dans le sens inverse (afin d'obtenir un classement du plus petit au plus grand).

Tu pourras réutiliser ce petit programme dans des carnets d'adresse, par exemple, car l'ordinateur va garder les mots en mémoire dans le bon ordre.

```

5 REM ----- ORDRE ALPHABETIQUE -----
10 MODE 1:CLS
20 INPUT "COMBIEN DE MOT A CLASSER ";NB
30 FOR I=1 TO NB
40 PRINT "MOT No";I;:INPUT A$(I)
50 NEXT I
60 PRINT "VOICI LA LISTE DE MOTS","DANS L'ORDR
E ALPHABETIQUE"
70 PERM=0
80 FOR I=2 TO NB
90 IF A$(I)<A$(I-1) THEN R$=A$(I):A$(I)=A$(I-1
):A$(I-1)=R$:PERM=1
100 NEXT I
110 IF PERM=1 THEN 70
120 FOR I=1 TO NB
130 PRINT A$(I)
140 NEXT I

```

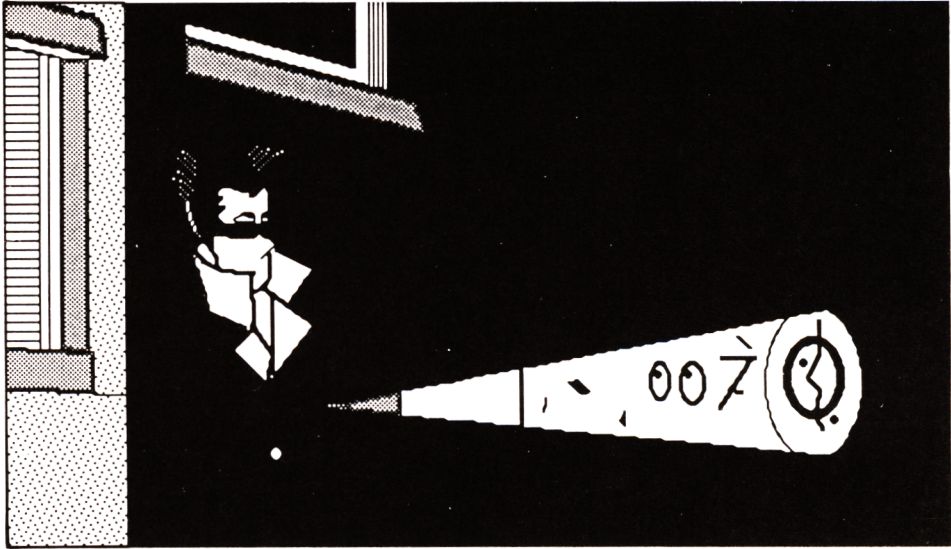
## Que vois-tu à l'écran ?

```

COMBIEN DE MOT A CLASSER ? 8
MOT No 1 ? AUGUSTIN
MOT No 2 ? TERESA
MOT No 3 ? DANIEL
MOT No 4 ? JACQUES
MOT No 5 ? FRANCOIS
MOT No 6 ? PAUL
MOT No 7 ? SIMONE
MOT No 8 ? AGNES
VOICI LA LISTE DE MOTS
DANS L'ORDRE ALPHABETIQUE
AGNES
AUGUSTIN
DANIEL
FRANCOIS
JACQUES
PAUL
SIMONE
TERESA

```

# CODE SECRET



**T**u vas transformer l'ordinateur en machine à coder et décoder des messages dans un langage secret.

## Comment vas-tu t'y prendre ?

De nombreux programmes de codage existent. Certains sont très compliqués et destinés aux services secrets !... Nous limiterons notre ambition à un principe de codage simple : A chaque lettre de l'alphabet, correspond une autre lettre, en respectant toujours l'ordre alphabétique.

Par exemple, si tu remplaces A par K, tu remplaceras B par L, C par M, etc. Il suffit donc de connaître la lettre qui correspond à A pour chiffrer ou déchiffrer un message.

C'est simple, mais parfois fastidieux et l'ordinateur peut faire cela très vite !

### Voici la solution que je te propose :

La ligne 20 enregistre la lettre qui correspond à A en clair.

Dans la ligne 30, la variable CODE prend en compte la différence de place entre ces deux lettres. C'est ce qui va permettre le codage.

Dans les lignes 40 à 60, le joueur choisit de coder, ou au contraire de décoder un message.

Le message est pris en compte lettre par lettre, dans A\$, ligne 80. B\$ sera son correspondant codé, ou décodé. S'il ne s'agit pas d'une lettre (espace, ponctuation...) B\$ = A\$ (ligne 110). (Attention il faut bien lire GOTO 150).

Les lignes 120 et 130 vont définir B\$ en fonction de CODE. B\$ est ajouté à la phrase résultat PH\$.

Lorsque le joueur tape un point, c'est que le message est terminé. Tu affiches PH\$ (ligne 170).

Ainsi, tu pourras communiquer seulement avec ceux que tu choisiras en leur donnant ton message codé, sa clé (la lettre de référence qui correspond à A) et ... le programme pour le décoder !

```

5 REM ---- CODE SECRET ----
10 MODE 1:CLS:PRINT "Quelle lettre remplace la
  lettre A"
20 I#=INKEY#:IF I#="" THEN 20
30 I#=UPPER$(I#):CODE=ASC(I#)-64
40 PRINT:PRINT "CODAGE.....1"
50 PRINT "DECODAGE....2"
60 INPUT "VOTRE CHOIX ";CH
70 PRINT "Ecrivez votre message","et tapez un
  . A la fin"
80 A#=INKEY#:IF A#="" THEN 80
90 A#=UPPER$(A#):B#="":PRINT A#;
100 IF A#="." THEN 170
110 R=ASC(A#):IF R<65 OR R>90 THEN B#=A#:GOTO
  150
120 IF CH=1 THEN R=R+CODE:IF R>90 THEN R=R-26
130 IF CH=2 THEN R=R-CODE:IF R<65 THEN R=R+26
140 B#=CHR$(R)
150 PH#=PH#+B#
160 GOTO 80
170 PRINT:PRINT PH#:END

```

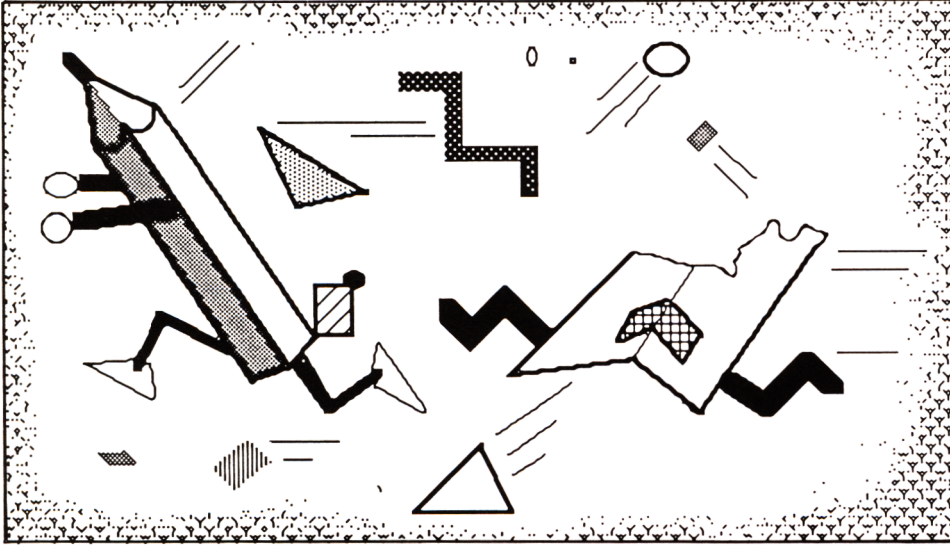
Que vois-tu à l'écran ?

```

Quelle lettre remplace la lettre A
CODAGE.....1
DECODAGE....2
VOTRE CHOIX ? 1
Ecrivez votre message
et tapez un . A la fin
Voici un exemple de codage.
MFZTZ LE UOUDGCU UU TFURXU

```

# MELI-MELO



**T**u vas retrouver un mot dont l'ordinateur aura mélangé les lettres.

Comment vas-tu t'y prendre ?

Tu l'as vu dans les programmes précédents, ton Amstrad est capable de décomposer un mot en lettres et de "jouer" avec ces lettres, c'est-à-dire de les changer de place entre elles. Mais toi, saurais-tu faire l'inverse ?

Comme il faut entrer un mot au début de jeu, je te propose de réaliser un jeu à deux, l'un contre l'autre, avec un score pour chacun.

## Voici la solution que je te propose :

Le début du jeu ressemble à celui du PENDU puisqu'il faut entrer un mot. J'ai choisi le même principe, qui permet de ne pas afficher ce mot, mais chaque lettre est gardée en mémoire sous un nom de variable indicée I\$(L), ce qui évite de décomposer ensuite le mot en lettres.

Attention: si tu veux utiliser des mots de plus de 10 lettres, il faudra dimensionner un tableau I\$, en ajoutant :

7 DIM I\$(20) par exemple

La boucle de la ligne 10 à la ligne 270 permet de recommencer deux fois le jeu : chaque joueur change de rôle.

Pour mélanger les lettres de MOT\$, j'ai ici repris une routine souvent utilisée dans les programmes de ce livre : tu tires au sort une des lettres (ligne 60), tu vérifies qu'elle n'est pas déjà sortie (lignes 80 à 100), puis tu l'affiches.

La réponse du joueur est entrée dans la ligne 200. Si la réponse est exacte (ligne 220), le score du joueur tient compte du nombre de réponses fausses auparavant données.

Si sa réponse est fausse, Amstrad l'affiche (même principe que dans le pendu). Après 8 réponses fausses, l'ordinateur donne le bon mot.

Tu peux prolonger le jeu en gardant les deux scores en mémoire :

265 SC(JEU) = SC(JEU) + 10 - RF

et en ajoutant :

280 GOTO 10

```

5 REM ----- MELIMELO -----
10 FOR JEU=1 TO 2:MODE 1:CLS:L=1:MOT$=""
20 PRINT "ENTREZ LE MOT A DECOUVRIR","PUIS UN
ESPACE POUR VALIDER LE MOT"
30 I$(L)=UPPER$(INKEY$):IF I$(L)="" THEN 30 EL
SE IF I$(L)=" " THEN L=L-1:GOTO 50
40 MOT$=MOT$+I$(L):L=L+1:GOTO 30
50 CLS:FOR I=1 TO L
60 R(I)=INT(RND*L)+1
70 IF I=1 THEN 110
80 FOR J=1 TO I-1
90 IF R(I)=R(J) THEN 60
100 NEXT J
110 LOCATE I*2,5:PRINT I$(R(I))
120 NEXT I
130 RF=0
200 LOCATE 1,10:INPUT "TA REPONSE ";REP$:REP$=
UPPER$(REP$)
210 LOCATE 1,10:PRINT SPACE$(39)
220 LOCATE 1,13+RF:IF REP$=MOT$ THEN PRINT "BR
AVO ! TU AS GAGNE";10-RF;"POINTS":GOTO 260
230 PRINT "NON ! ";REP$;" N'EST PAS LE BON MOT
";RF=RF+1
240 IF RF<8 THEN 200
250 PRINT "C'ETAIT ";MOT$
260 IF JEU=1 THEN LOCATE 1,23:PRINT "APPUIE SU
R UNE TOUCHE","POUR CHANGER DE JOUEUR ":A$=INK
EY$:IF A$="" THEN 260
270 NEXT JEU

```

Que vois-tu à l'écran ?

**O I S A N M**

**NON ! MOISAN N'EST PAS LE BON MOT  
NON ! NASIMO N'EST PAS LE BON MOT  
BRAVO ! TU AS GAGNE 8 POINTS**

**APPUIE SUR UNE TOUCHE  
POUR CHANGER DE JOUEUR**



# 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) et 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'Éducation 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'Éducation nationale et le plan « Informatique pour tous ».

# ANNEXE

## DISQUETTE D'ACCOMPAGNEMENT

L'ouvrage "Amstrad à l'école" vous propose un ensemble de programme Basic permettant une utilisation pédagogique de votre Amstrad CPC. Sa disquette d'accompagnement s'adresse notamment à tous ceux qui n'ont pas le temps de recopier le programme sur leur ordinateur.

Vente par correspondance

### BON DE COMMANDE

Je commande la disquette d'accompagnement du livre : « Amstrad à l'école » au prix de 150,00 FF\*.

Renvoyez-nous ce bon rempli (découpé ou copié), avec votre règlement, plus 10,00 FF\*\* de frais de port et d'emballage, par chèque bancaire ou postal établi à l'ordre de LA CONSOLE.

Adresse : La Console, 5, place du Colonel Fabien, 75010 Paris

Nom ..... Prénom .....

Adresse .....

..... Ville .....

Code postal ..... Pays .....

\* Prix valable jusqu'au 31 décembre 1986.

\*\* Hors CEE, prévoyez 15,00 FF pour supplément de frais d'expédition.

```

dir
A: D1          BAS : D2          BAS : D3          BAS : D4          BAS
A: D5          BAS : D6          BAS : D7          BAS : D8          BAS
A: D9          BAS : D10         BAS : D11         BAS : MENU        BAS
A: KIM         BAS : VILLES       BAS : DICTEE      BAS : SOUTIEN     BAS
A: SAFARI     BAS : LECRM        BAS : LECR        BAS : LECRD       BAS
A: LECRDM     BAS : FLURIEL     BAS : CONJU      BAS : ADDITION    BAS
A: SOUS       BAS : RAVIT       BAS : TRAIN      BAS : PHRASE      BAS
A: KIMNO1     BAS : KIMNO2     BAS : KIMNO3     BAS : MASSE       BAS
A: MASSEM     BAS : COCKTAIL   BAS : TEXTE      BAS : TEXTEM      BAS
A: ECRIS      BAS : ECRISAM    BAS : MULTI      BAS : DIVI        BAS
A: JEU        BAS : GOTH       DAT
A>

```

Elle contient :

- 39 programmes en Basic Amstrad, reflet des chapitres de ce livre. Ils se distinguent par l'extension ".BAS"

Le programme "MENU.BAS", écrit en Basic permet d'enchaîner les différents programmes du livre. Le END de chaque programme est remplacé par RUN "MENU" ce qui permet un retour au menu. Pour modifier les programmes il vous suffit de remplacer RUN "MENU" par END.

- Le programme JEU.BAS permet de charger un jeu de caractères que nous vous offrons. Il utilise le fichier "GOTH.DAT".

```

! " # $ % & ' ( ) * + , - . / 0 1 2 3 4 5 6 7 8 9 : ; < = > ? @ A B C D E F G H
I J K L M N O P Q R S T U V W X Y Z [ \ ] ^ _ ` a b c d e f g h i j k l m n o p
q r s t u v w x y z
Ready

```

# CONSEILS DE LECTURE

Pour approfondir vos connaissances en BASIC Amstrad et mieux connaître le système des CPC 464, 664 et 6128, P.S.I. vous propose une palette d'ouvrages utiles.

## **Pour mieux maîtriser le BASIC Amstrad**

- **BASIC plus 80 routines sur Amstrad** – Michel Martin (Editions du P.S.I.)

Pour pousser votre Amstrad au maximum de ses capacités ; 80 routines de simulation d'instructions qui n'existent pas en BASIC Amstrad.

- **BASIC Amstrad, CPC 464, 664 et 6128 : 1 – Méthodes pratiques ; 2 – Programmes** – Jacques Boisgontier (Editions du P.S.I.)

Pour ceux qui ont déjà pratiqué un BASIC, voici deux ouvrages de perfectionnement au BASIC Amstrad, illustrés par de nombreux programmes-exemples.

## **Pour être informé régulièrement de l'actualité des micros Amstrad**

- **MICROSTAD, revue mensuelle du Groupe Test**

Pour exploiter au mieux les capacités de votre micro, vous trouverez au sommaire de chaque numéro, un rendez-vous avec les rubriques clés :

- Découvrez la face cachée de votre CPC ou PCW : astuces, idées, conseils, tout pour comprendre votre micro, son anatomie, son fonctionnement, sa programmation et exploiter ses capacités graphiques et sonores.
- Domptez votre CPC 464, 664, 6128, PCW 8256 ou 8512 : passionnés, petits ou grands, spécialistes ou débutants, une information pratique et la compétence d'experts au service de votre micro.
- Programmez votre micro Amstrad : dans chaque numéro de MICROSTAD, un cocktail de programmes (dessins, jeux, utilitaires, gestion, etc.) et des trucs de programmation.

Achévé d'imprimer en mai 1986  
sur les presses de l'imprimerie Laballery  
58500 Clamecy  
Dépôt légal : mai 1986  
N° d'impression : 605024  
N° d'édition : 86595-343-1  
ISBN : 2-86595-343-2



## **Votre avis nous intéresse**

Pour nous permettre de faire de meilleurs livres, adressez-nous vos critiques sur le présent ouvrage.

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

.....  
.....  
.....  
.....

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

.....  
.....  
.....  
.....

Si vous souhaitez des éclaircissements techniques, écrivez-nous, nous ne manquerons pas de vous répondre directement.

**Où avez-vous acheté ce livre ?**

- cadeau                       librairie                       autres  
 exposition                       boutique micro

**Comment en avez-vous eu connaissance ?**

- publicité                       catalogue                       autres  
 exposition                       conseils d'un ami

**Avez-vous déjà acquis des livres P.S.I. ?**

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**



# AMSTRAD A L'ÉCOLE

**"A**mstrad 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 sur Amstrad CPC 464, 664 et 6128.

**U**n petit cahier de vacances, en fin d'ouvrage, leur permettra de concevoir eux-mêmes des programmes informatiques simples grâce auxquels ils pourront réviser calcul et français.



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

**ISBN 2-86595-343-2**

**120 FF**

# AMSTRAD A L'ÉCOLE



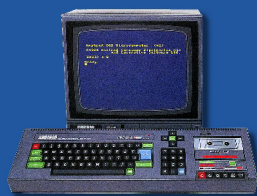


Document numérisé avec amour par

# AMSTRAD

CPC 

# MÉMOIRE ÉCRITE



<https://acpc.me/>