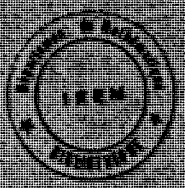
UNIVERSITÉ LOUIS PASTEUR

ulp

Institut
de Recherches
sur l'Enseignement
des Mathématiques
10, rue du Général Zimmer
67084 STRASBOURG CEDEX
tél. (88) 61.48.20

EXPERIENCES SUR LES APPORTS DE L'INFORMATIQUE A L'ENSEIGNEMENT DES MATHEMATIQUES

OPTION ALGORITHMIQUE en TERMINALE T A2



C. DUPUIS
D. GUIN
J.P. IGOT
N. VOGEL

Animateurs à l'I.R.E.M.

1986

AVANT PROPOS

Un groupe de travail a été constitué à l'I.R.E.M. de Strasbourg sur le thème des apports de l'informatique à l'enseignement des mathématiques.
Différentes expériences ont été réalisées avec des adultes enseignant en mathématiques ou d'autres disciplines, ainsi que des jeunes de l'école élémentaire ou du lycée, sur ce thème.
Le présent atelier relate un travail mené dans le cadre du programme de mathématiques de la Terminale A2 et proposant un support d'enseignement pour l'option "activités algorithmiques" prévue sur 15 heures.

Les fiches proposées, centrées sur les problèmes de rangement en mémoire et de classement, ont été expérimentées en 1984/85 dans deux classes de Terminales A2 de Lycées de Strasbourg qui ont réagi très différemment l'une de l'autre.

Le présent fascicule regroupe :

- l'ensemble des fiches distribuées aux élèves
- un jeu de fiches-professeurs comprenant les corrigés sommaires des exercices proposés.

FICHE 1 - A

LE ROBOT

Vous disposez d'un robot qui peut exécuter un certain nombre d'actions élémentaires

AV	avancer d'une unité
DR 45	pivoter à droite de 45°
DR 9Ø	pivoter à droite de 9Ø°
GA 45	pivoter à gauche de 45°
GA 9Ø	pivoter à gauche de 9Ø°

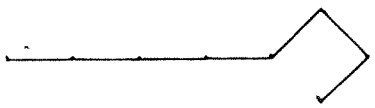
En outre, ce robot sait reconnaître si la condition BLEU : "le voyant bleu est allumé" est vraie ou fausse.

Pour commander le robot, il faut lui indiquer les actions élémentaires à effectuer, dans l'ordre voulu.

Exemple 1

répéter 4 fois
  AV
GA 45
répéter 2 fois
  AV
  DR9Ø
AV

donne :



Exemple 2

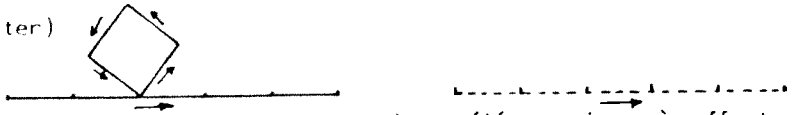
AV
si BLEU alors
  DR 45
  AV
sinon
  GA 45
  AV
AV
AV

donne :

Voici un trajet destiné à être parcouru par ce robot :

. en trait plein, le trajet parcouru si le voyant bleu est allumé

. en pointillé, le trajet parcouru sinon (sens des flèches à respecter)



Ecrivez la succession des actions élémentaires à effectuer par le robot pour parcourir ce trajet.

FICHE 1 - B — QUID ?

Répondre aux questions suivantes, en vous servant d'un "QUID". NOTEZ au fur et à mesure ce que vous faites pour trouver la réponse.

Question 1 Quels sont les meilleurs millésimes de Bourgogne depuis 1970 ?

Question 2 Le grand Rabbin de France est-il nommé ou élu, et par qui ?

Question 3 Le Bonheur, d'Agnès Varda, fut-il un film primé. De quand date-t-il ?

Question 4 On parle souvent de "puces" à propos des ordinateurs. En quoi est faite une "puce" et quelle quantité d'information peut-elle contenir ?

FICHE 1 - C — ALGORITHME DU DICTIONNAIRE

Vous savez chercher un mot dans un dictionnaire.
Déterminez les actions élémentaires nécessaires et
Ecrivez l'algorithme de recherche d'un mot (succession des actions élémentaires nécessaires)

Exercice

Ecrivez, sous forme d'algorithme exécutable par l'un d'entre vous, la confection d'une recette de cuisine que vous connaissez.

FICHE 2 - A

UNE HISTOIRE DE BOLS

(1) Vous disposez d'un bol BLANC contenant du lait
d'un bol ROUGE contenant du café

Il s'agit d'échanger les contenus, à l'aide d'un troisième bol
La seule opération permise est :
"Verser le contenu d'un bol dans un bol vide", notée ←
par exemple : ROUGE ← BLANC signifie verser le contenu du bol BLANC dans le bol ROUGE.
Ecrivez l'algorithme correspondant, qu'on appellera

ECHANGE (BLANC,ROUGE)

Y a-t-il d'autres algorithmes permettant d'obtenir le même résultat ?

(2) Un quatrième bol bleu contient du thé. Ecrivez l'algorithme permettant le passage du contenu 1 au contenu 2 :

	BLEU	BLANC	ROUGE
CONTENU 1	Thé	Lait	Café
CONTENU 2	Lait	Café	Thé

. Illustrez le déroulement de l'algorithme en représentant les contenus successifs des différents bols.

. Appliquez l'algorithme obtenu, en prenant comme contenu initial, le contenu 2. Que contiennent alors les différents bols ?

. Y a-t-il d'autres algorithmes possibles pour obtenir le même résultat ?

. Réécrivez l'un de ces algorithmes en vous servant de l'opération ECHANGE (X,Y), vue au (1), où X,Y désignent deux bols quelconques.

FICHE 2 - B | MEMOIRES

(3) Dans trois cases mémoires notées M1, M2, M3 sont rangés trois nombres.

L'opération notée "M1 ⟵ M2" signifie maintenant "recopier le contenu de la mémoire M2 dans M1", en lieu et place du contenu actuel de cette dernière, à la manière d'une copie de cassette magnétique.

Exécutez l'algorithme suivant, les contenus initiaux de M1, M2 et M3 étant 2, 4 et - 1.

```
. R  ⟵  M2
. M2 ⟵  M1
. M1 ⟵  M3
. M3 ⟵  R
```

(4) Si les contenus initiaux de M1, M2, M3 sont 0, - 16, - 20, écrivez un algorithme permettant de trouver ces trois nombres dans l'ordre croissant en allant de M1 à M3.

Y a - t- il d'autres contenus initiaux que cet algorithme permet de classer dans l'ordre croissant ? Donnez un exemple où l'algorithme convient, un autre où il ne convient pas.

(5) Quelles valeurs peut-on mettre au départ dans M1, M2, M3 pour que l'algorithme suivant réalise le classement par ordre croissant ?

```
. R  ⟵  M1
. M1 ⟵  M2
. M2 ⟵  M3
. M3 ⟵  R
```

Même question pour l'algorithme :

```
. R  ⟵  M1
. M1 ⟵  M2
. M2 ⟵  R
```

FICHE 3 - A

UNE DICTEE

1 On va dicter dix nombres.

. Pendant cette dictée vous ne notez aucun nombre.
. Vous ne pouvez garder en mémoire que deux nombres

a) A la fin de la dictée, vous devez donner la moyenne des dix nombres. Comment procédez-vous ?

b) A la fin de la dictée, vous devez donner le plus petit des dix nombres. Comment faites-vous ?

c) Même question pour dix mots en utilisant l'ordre alphabétique.

FICHE 3 - B

MEMOIRES (suite) - CLASSEMENTS

(2) a) Exécuter l'algorithme ci-contre, sur les mémoires M1, M2, M3 contenant respectivement - 5, 0, - 3.
Que contiennent-elles après cette exécution ?

. R ← M1
. si M2 < M1 alors R ← M2
. M2 ← M3
. M3 ← R

b) Même question si les contenus initiaux sont - 3, - 5, 0.

(3) On désigne par x, y les contenus inconnus de deux mémoires M1, M2. Ecrivez un algorithme permettant de classer ces deux nombres, dans l'ordre croissant, dans les mémoires M1, M2. (On utilisera ← et si ... alors ...)

(4) Ecrivez l'algorithme correspondant pour trois mémoires M1, M2, M3 contenant trois nombres inconnus x, y, z.

(5) Déroulez l'algorithme du (4) si les mémoires M1, M2, M3 contiennent au départ 2, 1, 0 en indiquant les contenus intermédiaires de toutes les mémoires utilisées, à chaque étape du déroulement. Combien d'essais analogues faut-il faire pour vérifier que l'algorithme fonctionne bien dans tous les cas ?

FICHE 4 - A — CLASSEMENT DE 4 NOMBRES

(1) Ecrivez l'algorithme permettant de classer 4 nombres, dans l'ordre croissant, dans les mémoires M1, M2, M3 et M4, en utilisant la même méthode qu'au (4) de la fiche 3 - B.

(2) Exécutez l'algorithme sur les contenus suivants :

	M1	M2	M3	M4
a)	-2	4	-5	-3
b)	2	1	0	-2

(3) Variante de 1

Ecrivez l'algorithme de classement des 4 nombres en utilisant la méthode suivante :
Déterminer le numéro p de la case mémoire M_p qui contient le plus petit des 4 nombres, échanger alors son contenu avec celui de M_1.
Recommencer avec les 3 cases mémoires restantes M_2, M_3 et M_4 et enfin avec M_3 et M_4.

FICHE 4 - B — CONTROLE N° 1

Exercice 1

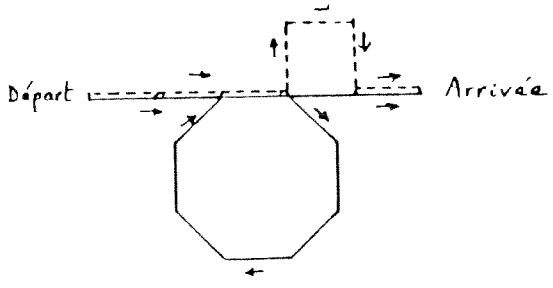
On dispose d'un robot qui sait exécuter les actions élémentaires :

AV	avancer d'une unité
GA 45	tourner à gauche de 45°
GA 90	tourner à gauche de 90°
DR 45	tourner à droite de 45°
DR 90	tourner à droite de 90°

et qui sait reconnaître si la condition :

(1) : le feu est bleu est vraie ou fausse.

Ecrivez la succession des ordres élémentaires à donner pour que le robot parcourre le trajet en continu si le feu est bleu et le trajet en pointillé si le feu n'est pas bleu.



Exercice 2 :

On dispose de 4 mémoires T_1 T_2 T_3 T_4 contenant chacune un nombre entier

T_1	T_2	T_3	T_4
x	y	z	n

Les nombres x, y et z vérifient la relation $x < y < z$.

ALGORITHME

SI $T_4 \leq T_1$ ALORS [$T_3 \leftarrow T_2$; $T_2 \leftarrow T_1$; $T_1 \leftarrow T_4$]

SINON [SI $T_4 \leq T_2$ ALORS [$T_3 \leftarrow T_2$; $T_2 \leftarrow T_4$]

SINON [SI $T_4 \leq T_3$ ALORS [$T_3 \leftarrow T_4$

Question 1 On donne les contenus suivants :

T_1	T_2	T_3	T_4
4	7	8	2

et on exécute l'algorithme ci-dessus.

Indiquez, dans un tableau, les contenus de T_1 T_2 T_3 T_4 à chaque étape de l'exécution.

Question 2 Même question dans le cas où au départ les contenus sont

T_1	T_2	T_3	T_4
-2	1	6	3

Question 3 Plus généralement les contenus au départ sont :

T_1	T_2	T_3	T_4
x	y	z	n

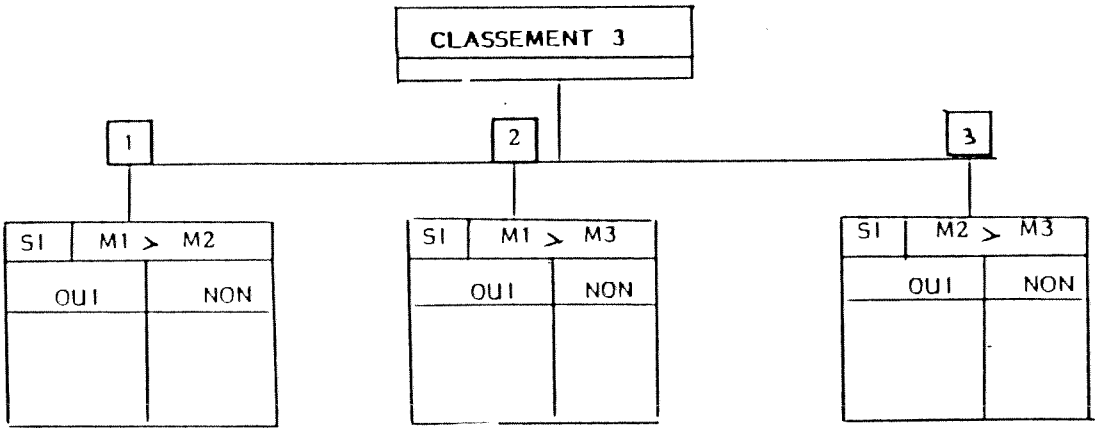
avec $x < y < z$

a) si $n \leq x$, quels sont les contenus de T_1 T_2 T_3 à la fin de l'exécution de l'algorithme ?

b) si vous ignorez la position de n par rapport à x, y et z, que pouvez-vous dire des contenus de T_1 T_2 T_3 à la fin de l'exécution de l'algorithme ?

FICHE 5 | FONCTIONNEMENT D'UN ALGORITHME DE CLASSEMENT

Le schéma suivant représente l'algorithme de classement de 3 nombres de la fiche (3)



(1) Complétez les cases OUI NON respectives avec les instructions de l'algorithme de classement.

(2) M1, M2, M3 contiennent, au départ - 2, 4, - 5.
Indiquez pour chacun des tests [1] [2] [3] , laquelle des cases OUI ou NON est parcourue pour effectuer le classement de ces trois nombres.

(3) Même question pour 2, 1, 0

(4) Quelles étaient les valeurs initiales possibles de M1, M2 et M3 pour que l'algorithme se déroule de la façon suivante :

[1]	[2]	[3]
OUI	NON	OUI

(5) Quels sont tous les autres déroulements possibles ? Donnez un exemples de contenus pour chaque cas.

(6) Construire un schéma similaire pour le classement de 4 nombres par l'une ou l'autre des méthodes vues dans la fiche 4-A.

FICHE 6

UN ALGORITHME GENERAL DE CLASSEMENT

(1) Que fait cet algorithme pour N > 2. Donnez-lui un nom.

```
répéter, pour J de 2 à N
        | si M1 > MJ alors ECHANGE (M1,MJ)
```

Faites un essai pour N = 4 et les contenus

M1	M2	M3	M4
4	2	0	1

(2) Utilisez l'algorithme du (1) pour réécrire l'algorithme de classement de 4 nombres vu dans la fiche 4.

(3) Donnez l'algorithme de classement de N nombres dans les N cases mémoires M1, M2, ... MN.

ECRITURE EN LANGAGE L.S.E. DE

L'ALGORITHME DE LA FICHE 6

Note : Le L.S.E. est un langage du type BASIC en français, disposant de procédures à variables locales (possibilité non utilisée ci-dessous).

```
0010 *CLASSEM'   -N
002'  .IRE[/,'DONNER N '] N
0030 TABLEAU CHAINE M[N];CHAINE R
0040 FAIRE 50 POUR J←1 JUSQUA N
0050   LIRE[/,'DONNER UN ELEMENT: '] M[J]
0060 FAIRE 70 POUR J←1 JUSQUA N-1
0070   &MINIM(J,N)
0080 FAIRE 80 POUR J←1 JUSQUA N ; AFFICHER M[
0090 TERMINER
0110 PROCEDUF  &MINIM(A,B)
 :20   FAIRE 130 POUR I←A+1 JUSQUA B
0130     SI M[A]>M[I] ALORS &ECHAN(M[A],M[I])
0140 RETOUR
0160 PROCEDURE &ECHAN(C,D)
0170   R←M[C]
0180   M[C]←M[D]
0190   M[D]←R
0200 RETOUR
```

ECRITURE EN BASIC
du même algorithme

```
0010 REM CLASSEMENT N
0020 INPUT "DONNER N ";N
0030 DIM M(N
0040   R J=1 TO N : INPUT "DONNER UN ELEMENT ";M(J) :NEXT J
0050 FOR J=1 TO N-1 : GOSUB 110 : NEXT J
0060 FOR J=1 TO N : PRINT M(J), : NEXT J
0070 PRINT
0080 END
0110 REM MINIM(J,N)
0120   FOR I=J+1 TO N
0130     IF M(J)>M(I) THEN GOSUB 150
0140 RETURN
0150 REM ECHAN M(J),M(I))
 160   R=M(J)
0170   M(J)=M(I)
0180   M(I)=R
0190 RETURN
```

CONTROLE N° 2

(1) Dérouler l'algorithme suivant :

si M1 < M3 alors échange M1, M2
si M1 < M3 alors échange M1, M3
si M1 < M4 alors échange M1, M4
si M2 < M3 alors échange M2, M3
si M2 < M4 alors échange M2, M4
si M3 < M4 alors échange M3, M4

Que fait cet algorithme ?

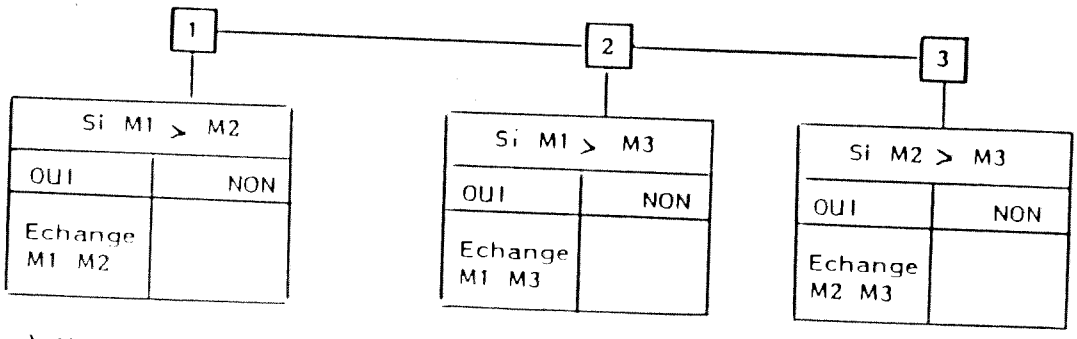
M1	M2	M3	M4
2	4	3	5

(2) Ecrire un algorithme faisant le même travail sur 3 mémoires M1, M2, M3

vérifier sur

M1	M2	M3
2	0	4

(3) Soit l'arbre suivant



a) M1 M2 M3 contiennent respectivement - 2, 4, - 5. Indiquez pour chaque test si l'on parcourt la branche OUI ou NON

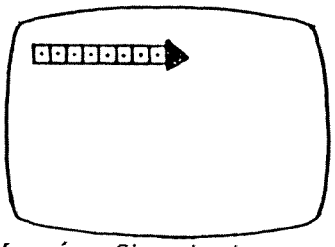
1	2	3

b) Donnez un exemple de contenu de M1 M2 M3 tel que l'on ait :

1	2	3
OUI	NON	OUI

FICHE 7A

LE JEU DU COLIMACON (Préliminaires)



Vous disposez d'un écran et d'un clavier ainsi que des ordres suivants :

INIT : effacer l'écran

CURSEUR (X,Y) : allumer la case d'abscisse et d'ordonnée Y

CLAVIER : vérifier si une touche du clavier est enfoncée. Si oui, la case mémoire REPONSE contient le caractère fourni par cette touche
sinon, la case mémoire REPONSE n'est pas modifiée

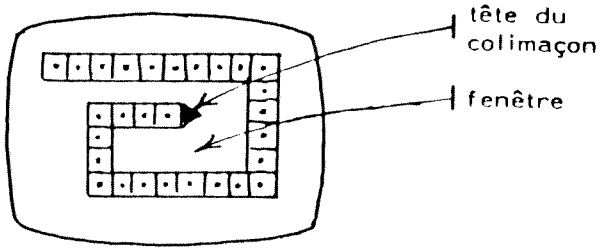
1. Ecrivez un algorithme permettant l'allumage automatique d'un segment horizontal, à partir de la case supérieure gauche de l'écran et ce, jusqu'à ce que le joueur frappe une touche.

2. Prévoir, en outre, que l'allumage s'arrête si le bord droit de l'écran est atteint, même si le joueur n'intervient pas
(X = XMAX)

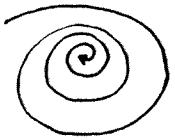
3. Faire en sorte que l'allumage se poursuive en descendant, à partir du moment où le joueur a frappé sa touche. L'allumage devant toujours s'arrêter, par contre, si le bord de l'écran est atteint sans que le joueur ne soit intervenu.

FICHE 7B

LE JEU DU COLIMAÇON (Jeu complet)



escalier
en "colimaçon"



(vu en verticale !!)

Il s'agit d'écrire un algorithme permettant l'allumage automatique des cases consécutives dans une direction donnée, jusqu'à ce que le joueur frappe une touche quelconque du clavier, ce qui a pour effet de modifier la direction du colimaçon, selon l'ordre suivant :

(Cf. schéma ci-dessus)

Le jeu s'arrête si la "tête" du colimaçon entre en collision avec une case déjà allumée.

Le but du jeu consistera donc à dévier "in extremis" la trajectoire en frappant une touche avant la collision.

. Ecrivez l'algorithme complet du jeu

(On pourra désigner par XMIN, XMAX, YMIN, YMAX les limites de la "fenêtre" dans laquelle le colimaçon peut encore se développer à un instant donné.)

FICHE PROFESSEUR

FICHE 1 | PROFESSEUR

A. Le but de la séance est d'acquérir la notion d'enchainement d'actions élémentaires. Un tel enchainement est un algorithme. On précise clairement les actions élémentaires permises. Le vocabulaire et le formalisme sont donc bien limités.

```
. répéter 2 fois
      |AV
.si BLEU alors |GA 45
               |AV
               |répéter 3 fois
               |   |GA 90
               |   |AV
               |GA 45   (*)
         sinon |rien
. répéter 3 fois
      |AV
```

```
GA 45
AV
GA 90
AV
GA 90
AV
GA 90
AV
GA 45
```

Cette action est exécutée que la condition BLEU soit vraie ou fausse

Les décalages sont importants pour préciser dans quelles conditions, l'action est à exécuter (ainsi l'action GA 45 (*) n'est exécutée qu'une fois).

Observer qu'il existe plusieurs algorithmes possibles pour un problème donné. Celui-ci est en quelque sorte "minimal".

B. Le QUID offre en outre l'intérêt d'être problématique pour les élèves (les réponses ne sont pas connues et la démarche peut, éventuellement, ne pas donner une réponse à la question posée).

Les élèves explicitent les actions élémentaires :
- choisir un mot-clé
-chercher ce mot dans la liste
- se reporter à la page indiquée

Les différentes situations qui vont se présenter impliquent des actions conditionnelles. P. ex. :
Si le mot clé est dans la liste alors...
sinon...

Là encore apparaît la possibilité d'algorithmes différents pour aboutir au même résultat

Noter que l'action "choisir un mot-clé" est relativement "intelligente" et donc difficilement automatisable.

Remarques sur les réponses aux questions

- Elles sont dans l'edition 1983 du QUID
- La réponse à la question 4 nécessite une édition récente du QUID, le mot "puce" étant utilisé dans un sens nouveau. Les élèves ont donc à décider quand ils arrêtent leurs recherches.

C. L'algorithme de recherche dans un dictionnaire peut, lui, être entièrement formalisé en actions élémentaires. Il faudra les expliciter en n'allant pas trop loin dans le détail (ex : analyse lettre à lettre de chaque mot pour effectuer une comparaison de deux mots).

La solution ci-dessous suppose de partir de la première page du dictionnaire. Ce n'est peut être pas la plus performante. La recherche dichotomique l'est plus mais est-elle naturelle pour les élèves ?

```
A
B
Si (non(2)) alors | tant que(1) faire
                  |     | C
                  |     | B
                  | E
                  | B
                  | tant que(1) faire
                  |     | D

Si(2) alors F
      sinon G
```

A : ouvrir le dictionnaire à la page 1
B : lire le 1er mot de la page
C : passer à la page suivante
D : lire le mot suivant dans la page
E : passer à la page précédente
F : lire la définition
G : conclure "le mot n'est pas dans le dictionnaire"

(1) le mot lu est avant le mot cherché (ordre alphabétique)
(2) le mot lu est le mot cherché

FICHE 2 | **PROFESSEUR**

(1) Donner les deux algorithmes minimaux possibles

Ⓐ
```
VERT ← BLANC
BLANC ← ROUGE
ROUGE ← BLANC
```

Ⓑ
```
VERT ← ROUGE
ROUGE ← BLANC
BLANC ← VERT
```

. On pourra suggérer de donner un nom à l'algorithme, et l'utiliser par la suite. Par exemple :

ECHANGE(BLANC,ROUGE) pour Ⓐ
ECHANGE(ROUGE,BLANC) pour Ⓑ

(2)

Ⓐ
```
VERT ← BLANC
BLANC ← ROUGE
ROUGE ← BLEU
BLEU ← VERT
```

Ⓑ
```
VERT ← BLEU
BLEU ← BLANC
BLANC ← ROUGE
ROUGE ← VERT
```

Ⓒ
```
VERT ← ROUGE
ROUGE ← BLEU
BLEU ← BLANC
BLANC ← VERT
```

. La première opération fixe les suivantes.

. Dans un premier temps "dérouler" l'algorithme sur les contenus des bols (varier les contenus initiaux pour faire acquérir la distinction contenu-contenant). Montrer l'indépendance de l'algorithme et du contenu des bols. Par la suite, faire trouver les algorithmes sans le déroulement sur un exemple.

. Utiliser les procédures ECHANGE. Par exemple, pour Ⓐ

Ⓐ'
```
ECHANGE(BLANC,ROUGE)
ECHANGE(ROUGE,BLEU)
```

(3) La notion de case mémoire correspond à la réalité des ordinateurs et peut être utilement illustrée par les manipulations de cassettes ou de bandes magnétiques en sonorisation. Mettre en évidence les deux faits suivants :

si "M1 ← M2" 1) l'ancien contenu de M1 est remplacé par celui de M2
2) celui de M2 subsiste dans M2

On peut illustrer par :

```
R  ← M2
M2 ← M1
M1 ← M3
M3 ← 2
```

M1	M2	M3	R
2	4	-1	?
2	4	-1	4
2	2	-1	4
-1	2	-1	4
-1	2	4	4

Certains élèves observeront que, globalement, l'algorithme réalise une permutation circulaire des contenus de M1, M2, M3.

④ R ⟵ M3
M3 ⟵ M1
M1 ⟵ R
ou échange(M3,M1), convient dès que M1 > M2 > M3

⑤ a)

M1	M2	M3
0	-20	-16

ou toute disposition

M1	M2	M3
3	1	2

b)

M1	M2	M3
-16	-20	0

"

M1	M2	M3
2	1	3

FICHE 3 | **PROFESSEUR**

(1) Aboutir à

a)

```
M ← 0
répéter 10 fois
  | M ← M+nombre dicté
Ecrire M/10
```

On ne retient que :
. le contenu de M
. le nombre dicté

b)

```
MIN ← nombre dicté en premier
répéter 9 fois
  | D ← nombre dicté
  | Si D < MIN alors |MIN ← D
écrire MIN
```

Mettre en évidence les graphismes (les structures) :

```
répéter . . fois
  | ordres élémentaires
```

et

```
si . . . alors
  |
  | ordres élément
  .
sinon
  |
  | ordres élément.
  .
```

. On décale vers la droite le (ou les) ordre(s) qui doi(vent) être répété(s) ou exécuté(s) conditionnellement ("Sinon" peut être omis, s'il n'y a rien à faire). P.ex observer que écrire MIN n'est exécuté qu'une seule foi
. La notation D < MIN concerne les contenus alors que D ← MIN concerne la case D et le contenu de MIN.
. Observer dans b) le traitement particulier du 1er nombre dicté, la répétition ne portant que sur les 9 autres, ce qui permet de comparer deux nombres.
. Ces procédés économisent la mémoire mais on ne dispose plus, pour la suite, des nombres dictés ce qui ne permet pas, par exemple, de les classer tous.

(3) et (4) Les exercices qui suivent ont pour but de préparer l'écriture d'un algorithme de classement des contenus de n cases mémoires.

```
n = 2     Si M2 < M1 alors   ECHANGE(M1,M2)
                     sinon   RIEN
```

n = 3 plusieurs solutions seront sans doute proposées par les élèves, une fois vaincue la difficulté des contenus inconnus. Ces algorithmes seront le plus difficiles à généraliser à 4 ou plus cases (cf. fiche 4). Proposer alors la méthode suivante :

1° rechercher le plus petit des trois contenus et le mettre dans M1

2° recommencer avec les deux cases restantes M2 et M3 d'où l'algorithme :

```
si M2 < M1   alors    ECHANGE (M1,M2)
si M3 < M1   alors    ECHANGE (M1,M3)
                (* M1 contient alors le plus petit nombre)
si M2 > M3   alors    ECHANGE (M2,M3)
                (* M2 contient le plus petit des deux restants)
```

ou en détaillant :

```
si M1 > M2   alors | R  ← M1
                   | M1 ← M2
                   | M2 ← R
si M1 > M3   alors | R  ← M1
                   | M1 ← M3
                   | M3 ← R
si M2 > M3   alors | R  ← M2
                   | M2 ← M3
                   | M3 ← R
```

. Entamer ici une discussion sur la généralisation à plus de 3 cases.

1) comparer M1 à tous les suivants, échanger dès que M1 est plus grand
2) comparer M2 à tous les suivants, " " " M2 " " "
   etc...

. Variante possible :

1) au lieu d'échanger chaque fois les contenus, rechercher la case contenant le plus petit, en parcourant toutes les cases concernées. Echanger ensuite le contenu de cette case avec celui de la case M1.
2) Procéder de même avec les cases M2, M3 ...

On aura alors :

```
P ← 1
Si M2 < MP   alors   P ← 2
Si M3 < MP   alors   P ← 3
exécuter ECHANGE(M1,MP)
P ← 2
Si M3 < MP   alors   P ← 3
exécuter ECHANGE (M2,MP)
```

(5) Mettre en évidence ici la nécessité de parcourir les deux branches de chaque alternative (Soit 6 essais pour le cas n = 3)

FICHE 4 | **PROFESSEUR**

①

```
Si M1 > M2   alors   ECHANGE (M1,M2)
Si M1 > M3   alors   ECHANGE (M1,M3)
Si M1 > M4   alors   ECHANGE (M1,M4)
          (*M1 contient le plus petit des 4 *)
Si M2 > M3   alors ECHANGE (M2,M3)
Si M2 > M4   alors   ECHANGE (M2,M4)
          (*M2 contient le plus petit des 3 derniers *)
Si M3 > M4   alors   ECHANGE (M3,M4)
```

MINIMUM(1,4)

MINIMUM(2,4)

MINIMUM(3,4)

. On pourra à l'issue de ce travail, préparer la généralisation à N en faisant observer que l'on met successivement dans les cases-mémoires Mi le plus petit des contenus de Mi, Mi+1 ... Mn, d'où la notation MINIMUM (I,4)

. Les déroulements proposés en ② contribuent à dissocier l'algorithme des données initiales.

M1	M2	M3	M4	ECH
-2	4	-5	-3	?
-2	4	-5	-3	?
-5	4	-2	-3	-2
(-5)	4	-2	-3	-2
⋮	-2	4	-3	4
⋮	(-3)	4	-2	-2
⋮	⋮	(-2)	4	4

③ Variante :

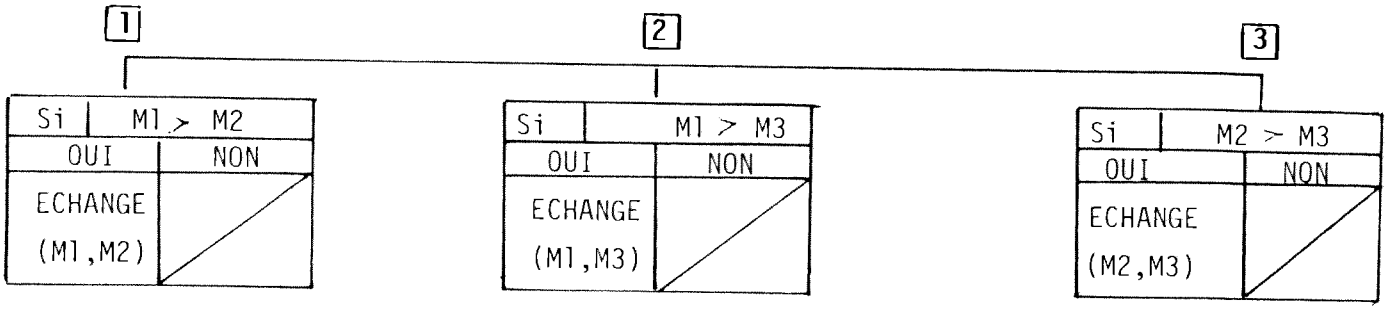
```
p ← 1
si Mp > M2   alors  p ← 2
si Mp > M3   alors  p ← 3
si Mp > M4   alors  p ← 4
exécuter ECHANGE (M1,Mp)

p ← 2
si Mp > M3   alors  p ← 3
si Mp > M4   alors  p ← 4
exécuter ECHANGE (M2,Mp)

p ← 3
si Mp > M4   alors  p ← 4
exécuter ECHANGE (M3,Mp)
```

FICHE 5

PROFESSEUR



N.B On pourra barrer la case vide ou bien y mettre RIEN

(2)

- 2	4	-5
- 2	4	-5
- 5	4	-2
	-2	4

1	2	3
NON	OUI	OUI

(Faire remarquer que le contenu de M1, M3 évolue éventuellement d'un test à l'autre.)

(3)

2	1	0
1	2	0
0	2	1
	1	2

1	2	3
OUI	OUI	OUI

(4)

1	2	3
OUI	NON	OUI

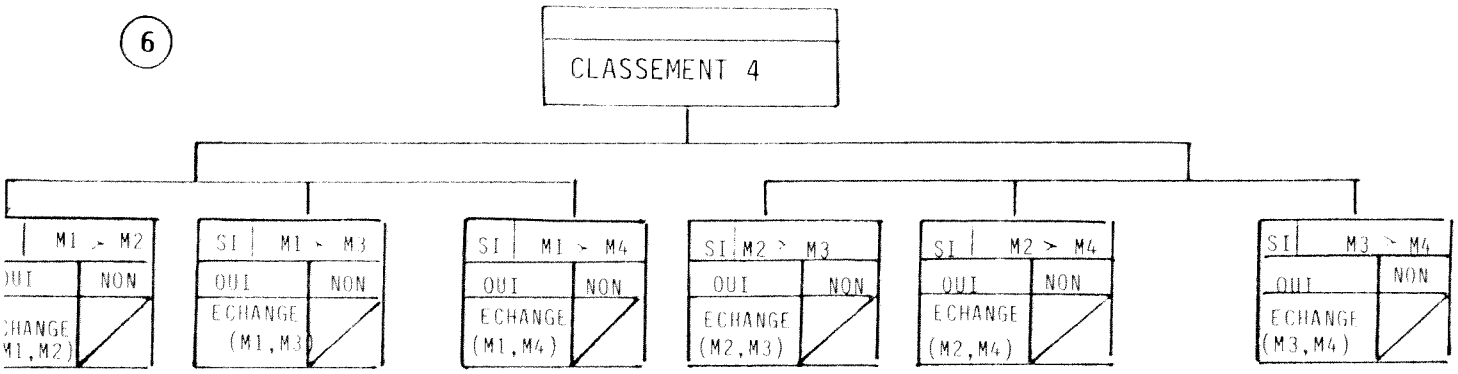
M1	M2	M3
2	0	1
0	2	1
	1	2

(5)

1	2	3
OUI	OUI	OUI
OUI	OUI	NON
OUI	NON	OUI
OUI	NON	NON
NON	OUI	OUI
NON	OUI	NON
NON	NON	OUI
NON	NON	NON

M1	M2	M3

(6)



FICHE 6- A

PROFESSEUR

(1) C'est l'algorithme du minimum de N nombres. Ce minimum est rangé dans M1.
On peut le nommer MINIMUM (1,N)

4	2	0	1
2	4	0	1
0	4	2	1
(0)	4	2	1

(2)
MINIMUM (1,4)
MINIMUM (2,4)
MINIMUM (3,4)

(3)
```
Répéter, pour I de 1 à (N-1)
    | MINIMUM (I,N)
```

ou encore

```
Répéter, pour I de 1 à (N-1)
    |répéter pour J de (I+1) à N
    ||Si MI > MJ alors ECHANGE (MI,MJ)
```

Attention aux bornes

(N-1)

(I+1)

```
0010 *CLASSEMENT-N
0020 LIRE[/,'DONNER N '] N
0030 TABLEAU CHAINE    ]:CHAINE R
00   FAIRE 50 POUR J←1 JUSQUA N
0050   LIRE[/,'DONNER UN ELEMENT: '] M[J]
0060 FAIRE 70 POUR J←1 JUSQUA N-1
0070   &MINIM(J,N)
0080 FAIRE 80 POUR J←1 JUSQUA N : AFFICHER M[J]
0090 TERMINER
0110 PROCEDURE &MINIM(A,B)
0120   FAIRE 130 POUR I←A+1 J   UA B
0130     S.  [A]>M[I] ALORS &ECHAN(M[A],M[I])
0140 RETOUR
0160 PROCEDURE &ECHAN(C,D)
0170   R←M[C]
0180   M[C]←M[D]
0190   M[D]←R
0200 RETOUR
```

```
0010 REM CLASSEMENT N
0020 INPUT "DONNER N ";N
0030 DIM M(N)
0040 FOR J=1 TO N : INPUT "DONNER UN ELEMENT ";M(J) :NEXT J
0050 FOR J=1 TO N-1 : GOSUB 110 : NEXT J
0060 FOR J=1 TO N : PRI`  M(J), : NEXT J
00   PRINT
0080 END
0110 REM MINIM(J,N)
0120   FOR I=J+1 TO N
0130     IF M(J)>M(I) THEN GOSUB 150
0140 RETURN
0150 REM ECHAN(M(J),M(I))
0160   R=M(J)
0170   M(J)=M(I)
0180   M(I)=R
0190 RETURN
```

FICHE 6-A | PROFESSEUR | **Réalisation possible sur ordinateur (suite)**

Variante utilisant la position P de la case mémoire Mp

```
0010 REM CLASSEMENT N
0020 INPUT "DONNER N ";N
0030 DIM M(N)
0040 FOR J=1 TO N : INPUT "DONNER UN ELEMENT ";M(J) : NEXT J
0050 FOR J=1 TO N-1 : GOSUB 100 :GOSUB 140: NEXT J
0060 FOR J=1 TO N : PRINT M(J), : NEXT J
0070 PRINT
0080 END
0100 REM RECHE(J,N)
0105   P=J
0110   FOR I=J+1 TO N
0120     IF M(P)>M(I) THEN P=I
0130 RETURN
0140 REM ECHAN(M(J),M(P))
0150   R=M(J)
0160   M(J)=M(P)
0170   M(P)=R
0180 RETURN
```

```
0010 *CLASSEMENT N (VARIANTE POSITION P)
0020 LIRE[/,'NOMBRE D''ELEMENTS? '] N
0030 TABLEAU CHAINE M[N] ; CHAINE R
0040 FAIRE 50 POUR J=1 JUSQUA N
0050   LIRE[/,'DONNER UN ELEMENT : '] M[J]
0060 FAIRE 80 POUR J=1 JUSQUA N-1
0070   &RECHE(J,N)
0080   &ECHAN(J,P)
0090 FAIRE 90 POUR J=1 JUSQUA N ; AFFICHER M[J]
0100 TERMINER
0110 PROCEDURE &RECHE(A,B)
0120 P=A
0130 FAIRE 140 POUR I=A+1 JUSQUA B
0140   SI M[P]>M[I] ALORS P=I
0150 RETOUR
0160 PROCEDURE &ECHAN(C,D)
0170   R=M[C]
0180   M[C]=M[D]
0190   M[D]=R
0200 RETOUR
```

Mise en route :

Basculer le bouton à gauche de l'appareil; dès que l'écran est chaud apparaît le message MICRAL ... Introduire la disquette "système" BASIC" dans le lecteur de gauche étiquette vers vous puis appuyer sur la touche [VALIDE] Après divers bruits et des messages indiquant que le système d'exploitation prologue et le GBASIC sont chargés apparaît le message OK. Ceci indique que l'ordinateur est prêt et que vous pouvez taper votre programme.

Taper le programme

Taper la première ligne du programme puis appuyer sur [VALIDE] ce qui remet le curseur au début de la ligne suivante et enregistre la ligne dans la mémoire de l'ordinateur. Recommencer l'opération pour chaque ligne.

Vérifier la frappe du programme : LIST [VALIDE]

Cette commande permet de réécrire sur l'écran les lignes de programme déjà enregistrées.

LIST [VALIDE] liste tous le programme
LIST 100 - 150 [VALIDE] liste à partir de la ligne 100 jusqu'à 150
LIST - 70 [VALIDE] liste du début jusqu'à la ligne 70
LIST 30 - [VALIDE] liste à partir de la ligne 30 jusqu'à la fin du programme.

Exécution du programme : RUN [VALIDE]

Le programme s'exécute depuis le début. En cas d'erreur dans le programme l'exécution est arrêtée et un message apparaît à l'écran : par exemple SN ERROR IN 30. Les deux premières lettres indiquent le type de l'erreur rencontrée.

SN ERROR IN 30

signifie que l'ordinateur a rencontré une erreur de syntaxe en ligne 30 (ou autre). Appuyer sur L ce que listera la ligne à l'écran puis sur [VALIDE] . Réécrire ensuite la ligne 30 corrigée puis appuyer sur [VALIDE.]

FICHE 7-A ET 7-B | **PROFESSEUR**

Les préparatifs de la fiche 7-A permettent de se familiariser avec la gestion d'un curseur lumineux sur l'écran (base de beaucoup de jeux).

1 On peut proposer

```
INIT
REPONSE ← VIDE
X ← Ø
Y ← Ø
tant que REPONSE = VIDE
    X ← X + 1
    CURSEUR(X,Y)
    CLAVIER
```

2 puis la condition tant que (REPONSE = VIDE) et (X < XMAX) en ayant initialisé XMAX à 8Ø par exemple

3

```
INIT
REPONSE ← vide
XMAX ← 8Ø   X ← Ø   YMAX ← 25   Y ← Ø
tant que (REPONSE = vide) et X   XMAX)
    X ← X+1
    CURSEUR(X,Y)
    CLAVIER
REPONSE ← vide
tant que (REPONSE = vide) et (Y < YMAX)
    Y ← Y+1
    CURSEUR(X,Y)
    CLAVIER
```

Observer que cette solution n'arrête pas le jeu après collision avec le bord droit de l'écran X=XMAX, on pourra proposer l'une des deux solutions suivantes :

3 Variante 1

```
INIT
REPONSE ← vide
XMAX ← 8Ø   X ← Ø   YMAX ← 25   Y ← Ø
tant que (REPONSE=vide) et (XMAX < X)
```

```
    X<-- X+1
    CURSEUR(X,Y)
    CLAVIER
REPONSE <-- vide
Si X=XMAX alors REPONSE <-- non vide

tant que (REPONSE=vide) et (Y< YMAX)
    Y <-- Y+1
    CURSEUR(X,Y)
    CLAVIER

REPONSE <-- vide
Si X=XMAX ou YMAX=Y alors REPONSE <-- non vide
```

3 Variante 2

```
INIT
REPONSE <--vide
XMAX <-- 80   X <--0   YMAX <-- 25   Y <--0
COLLISION <-- 0
tant que (REPONSE=vide) et (COLLISION=0)
    CURSEUR(X,Y)
    X <-- X+1
    Si X=XMAX alors COLLISION <-- 1
    CLAVIER

REPONSE <--vide
XMAX <-- X+1
tant que (REPONSE=vide) et (COLLISION=0)
    CURSEUR(X,Y)
    Y <-- Y+1
    Si Y=YMAX alors COLLISION <--1
    CLAVIER
```

La délimitation de la fenêtre peut s'obtenir en modifiant XMAX,YMAX,XMIN, YMIN après chaque changement de direction. (P.ex. XMAX <-- X+1 nouvelle marge droite) d'où la solution complète :

```
4  INIT
   XMAX ← 80 YMAX ← 25   XMIN ← -1   YMIN ← -1
   X ← 0   Y ← 0   REPONSE ← VIDE
   tant que (REPONSE = vide)
   |  tant que (REPONSE=vide) et (X < XMAX)
   |  |  X ← X+1                                  }
   |  |  CURSEUR (X,Y)                            }  (1)
   |  |  CLAVIER                                  }
   |  REPONSE ← vide                              }
   |  si (X=XMAX) alors REPONSE ← non vide        }
   |  XM ← X
   |  tant que (REPONSE=vide) et (Y < YMAX)       }
   |  |  Y ← Y+1                                  }
   |  |  CURSEUR  (X,Y)                           }  (2)
   |  |  CLAVIER                                  }
   |  REPONSE ← vide                              }
   |  si (YMAX= Y) ou (XMAX=X) alors REPONSE ← non vide
   |  YM ← Y
   |  tant que (REPONSE=vide) et (X > XMIN)       }
   |  |  X ← X-1                                  }
   |  |  CURSEUR(X,Y)                             }  (3)
   |  |  CLAVIER                                  }
   |  REPONSE ← VIDE
   |  Si (XMIN=X) ou (YMAX=Y) ou (XMAX=X) alors (REPONSE ← non vide)
   |  tant que (REPONSE=vide) et (Y > YMIN)
   |  |  Y ← Y-1                                  }
   |  |  CURSEUR (X,Y)                            }  (4)
   |  |  CLAVIER                                  }
   |  REPONSE ← vide
   |  Si (XMIN=X) ou (YMAX=Y) ou (XMAX=X) ou (YMIN=Y) alors REPONSE ← non vide
   |  YMIN ← Y   XMIN ← X   YMAX ← YM   XMAX ← XM
   |
```

ou bien la variante utilisant COLLISION :

```
4  Variante 2
   INIT
   XMAX ← 80   YMAX ← 25   XMIN ← 0   YMIN ← 0
   X ← 0   Y ← 0   COLLISION ← 0   REPONSE ← vide
```

```
tant que COLLISION = Ø
    tant que (REPONSE=vide) et (COLLISION=Ø)
        CURSEUR(X,Y)
        X ← X+1
        Si X=XMAX alors COLLISION ← 1                    ①
        CLAVIER
    XMAX ← X+1    REPONSE ← vide
    Tant que (REPONSE=vide) et (COLLISION=Ø)
        CURSEUR(X,Y)
        Y ← Y+1
        Si Y=YMAX alors COLLISION ← 1                    ②
        CLAVIER
    YMAX ← Y+1    REPONSE ← vide
    tant que (REPONSE=vide) et (COLLISION=Ø)
        CURSEUR(X,Y)
        X ← X-1
        Si X=XMIN alors COLLISION ← 1                    ③
        CLAVIER
    XMIN ← X-1    REPONSE ← vide
    tant que (REPONSE=vide) et (COLLISION=Ø)
        CURSEUR(X,Y)
        Y ← Y-1
        Si Y=YMIN alors COLLISION ← 1                    ④
        CLAVIER
    YMIN ← Y-1    REPONSE ← vide
```

. Observer que les 4 modules se ressemblent. Introduire le module :

```
AVANCE(DX,DY)
    REPONSE ← vide
    tant que (REPONSE=vide) et COLLISION=Ø)
        CURSEUR(X,Y)
        X ← X+DX    Y ← Y+DY
        Si (X=XMAX) ou (Y=YMAX) ou (X=XMIN) ou (Y=YMIN)
        alors COLLISION ← 1
        CLAVIER
```

La partie ① ② ③ ④ pourra se simplifier par :

```
tant que COLLISION = Ø
    exécuter AVANCE(1,Ø) : XMAX ← X+1
    exécuter AVANCE(Ø,1) : YMAX ← Y+1
    exécuter AVANCE(-1,Ø) : XMIN ← X-1
    exécuter AVANCE(Ø,-1) : YMIN ← Y-1
```

FICHE 7-B

PROFESSEUR (suite 3)

La réalisation sur machine peut être envisagée dans le cadre de clubs informatiques

- sur des machines disposant de fonctions graphiques, on trouvera les analogues directs de INIT, CURSEUR(X,Y) et CLAVIER
- sur les autres, il conviendra de simuler la fonction CURSEUR(X,Y) à l'aide d'ordres d'écriture d'un caractère. (P. ex. "*").

L'exemple ci-dessous réalise l'algorithme complet du jeu en BASIC structuré . avec fonctions graphiques séquence 260-280
. sans fonctions graphiques (séquence 210-250)

```
10 REM Principal          (Variante 2)
20 PRINT "■":X=0:Y=0:REP$=""
30 XMAX=80:YMAX=25:XMIN=-1:YMIN=-1
40 COLL=0
50 WHILE COLL=0 DO
60   DX=1:DY=0:GOSUB130:X1=X
70   DX=0:DY=1:GOSUB130:Y1=Y
80   DX=-1:DY=0:GOSUB130
90   DX=0:DY=-1:GOSUB130:YMIN=Y
95   XMIN=X:YMAX=Y1:XMAX=X1
100 WEND
120 END
130 REM Avance DX,DY
140   REP$=""
150   WHILE (REP$="")AND(COLL=0) DO
160     X=X+DX:Y=Y+DY
170     GOSUB210:GET REP$
180     IF (X=XMAX)OR(Y=YMAX)OR(X=XMIN)OR(Y=YMIN)THEN COLL=1
190   WEND
200 RETURN
210 REM CURSEUR(X,Y)   (Sans fonctions graphiques)
220   PRINT"■"
230   FOR I=0 TO Y:PRINT:NEXT I
240   PRINT TAB(X);"*"
250 RETURN
260 REM CURSEUR (Avec fonctions graphiques)
270   SET X,Y:RETURN
280 RETURN
```

FICHE 7-B

PROFESSEUR (suite 4)

(Variante 1)

```
10 REM Principal
20 PRINT "■":X=0:Y=0:REP$=""
30 XMAX=80:YMAX=25:XMIN=-1:YMIN=-1
50 WHILE REP$="" DO
60    WHILE (REP$="")AND(X<XMAX)DO
70     X=X+1:GOSUB 210:GET REP$
90    WEND
100   REP$="":X1=X:IF X=XMAX THEN REP$="+"
120   WHILE (REP$="")AND(Y<YMAX)DO
130     Y=Y+1:GOSUB 210:GET REP$
140   WEND
150   REP$="":Y1=Y:IF (Y=YMAX)OR(X=XMAX)THEN REP$="+"
160   WHILE (REP$="")AND(X>XMIN) DO
170     X=X-1:GOSUB 210:GET REP$
180   WEND
185   REP$="":IF (X=XMIN)OR(X=XMAX)OR(Y=YMIN)OR(Y=YMAX)THEN REP$="+"
190   WHILE (REP$="")AND(Y>YMIN) DO
195     Y=Y-1:GOSUB 210:GET REP$
200   WEND
205   REP$="":IF (X=XMAX)OR(Y=YMAX)OR(X=XMIN)OR(Y=YMIN)THEN REP$="+"
207   XMAX=X1:YMAX=Y1:XMIN=X:YMIN=Y
208 WEND
209 END
210 REM CURSEUR(X,Y)   (Sans fonctions graphiques)
220   PRINT"■"
230   FOR I=0 TO Y:PRINT:NEXT I
240   PRINT TAB(X);"*"
250 RETURN
260 REM CURSEUR (Avec fonctions graphiques)
270   SET X,Y:RETURN
280 RETURN
```