LOGO K 7
EDIL

EDIL

BELIN

UTILISATION DU PROGRAMME LANGAGE LOGO K7

LOGO est un langage d'initiation à la programmation. Vous trouverez ci-dessous la liste des instructions ainsi que des exemples d'utilisation.

Le MENU PRINCIPAL proposé contient les modes :

P rogrammation	P
L iste	L
E xécution	E
F in	F
C hargement	C
S auvegarde	S
Votre choix ?	

Pour réaliser votre choix,
il suffit d'appuyer sur la touche correspondante.

EXPLICATION DES MODES

LISTE Dans ce mode, l'ordinateur affiche à l'écran la totalité des procédures en mémoire. Il suffit pour cela, à chaque ligne, d'appuyer sur la touche ENTREE. Pour sortir de ce mode, frappez Q.

EXÉCUTION Pour exécuter une procédure, répondre à la question : PROCÉDURE A EXÉCUTER ? par le nom de la procédure. Pour arrêter l'exécution, frappez Q.

FIN Arrête le programme LOGO. Attention ! le dernier programme tapé est perdu. Si vous souhaitez le conserver, utilisez d'abord le mode SAUVEGARDE.

SAUVEGARDE Permet de conserver sur cassette un programme écrit en LOGO. Pour ce faire, frappez le nom du programme, positionnez la cassette, appuyez sur la touche enregistrement du lecteur de cassette puis sur ENTREE. (Attention ! utilisez une cassette personnelle pour sauvegarder vos programmes.)

CHARGEMENT Permet de récupérer un programme écrit en LOGO et préalablement sauvé sur cassette à l'aide de l'instruction SAUVEGARDE. Positionner la cassette, frapper le nom du programme puis ENTREE.

PROGRAMMATION Ce mode permet d'écrire ou de modifier un programme. Un programme consiste à "apprendre" à l'ordinateur à effectuer des actions en lui donnant des ordres.
Ces ordres sont :
– soit des instructions,
– soit des groupes d'instructions appelés procédures.
Le LOGO permet :
– de dessiner,
– de calculer,
– de manipuler des lettres.
Pour ce faire, voici la marche à suivre :
En mode de PROGRAMMATION, sont affichées :
– quelques lignes de texte (nous y reviendrons ultérieurement...),
– et l'expression PROGRAMME A ÉDITER ?.
La règle d'or du LOGO est la suivante :
– chaque programme, ou chaque portion de programme (procédure), porte un nom et n'est connu que par son nom.
– si vous désirez modifier une procédure existante, il suffit de répondre à cette question (PROGRAMME A ÉDITER ?) par le nom de la procédure à éditer,
– dans le cas où elle n'existerait pas, le LOGO vous positionnera au début des programmes déjà écrits,
– dans le cas d'une nouvelle procédure, appuyez sur ENTREE,
– puis l'ordinateur affiche : ? et il attend une ligne de programme.

ATTENTION : les versions TO7 et MO5 diffèrent quelque peu !

(Et nous revenons à présent aux "quelques lignes de texte" évoquées précédemment...)

Sur TO7 :

1) Pour passer d'une ligne à la suivante, frappez ENTREE.
2) Pour insérer une ligne entre deux lignes existantes, se positionner sur la seconde, frapper I puis ENTREE.
3) Pour effacer une ligne, se positionner sur celle-ci, frapper D puis ENTREE.
4) Pour effacer la totalité du programme, frapper EFFACETOUT puis ENTREE.

Sur MO5 :

Dans les cas 2, 3 et 4 traités ci-dessus, avant de frapper ENTREE effacer la fin de la ligne à l'aide de la touche EFF.

COMMENT PROGRAMMER ? QUELQUES CONSEILS...

L'ordinateur, en tant que machine, ne comprend qu'un langage bien défini et limité par des règles d'orthographe et de syntaxe.
Une ligne de programme comprend en général :
1) une instruction (et une seule),
2) un espace (symbolisé ici par [_]),
3) un argument.

INSTRUCTIONS

Chapitre 1 : POUR et FIN

Comme nous l'avons vu précédemment, toute procédure est entièrement définie par son nom. Pour définir une procédure, il suffit de frapper sur une ligne :

POUR _ *nom de la procédure*

Le programme se situera alors entre ce POUR et la ligne :

FIN _ *nom de la procédure*

et sera connu désormais sous le titre *nom de la procédure*.

Exemple :

```
POUR _ TRAIT
AVANCE _ 3Ø
FIN _ TRAIT
```

Chapitre 2 : Dessinons !

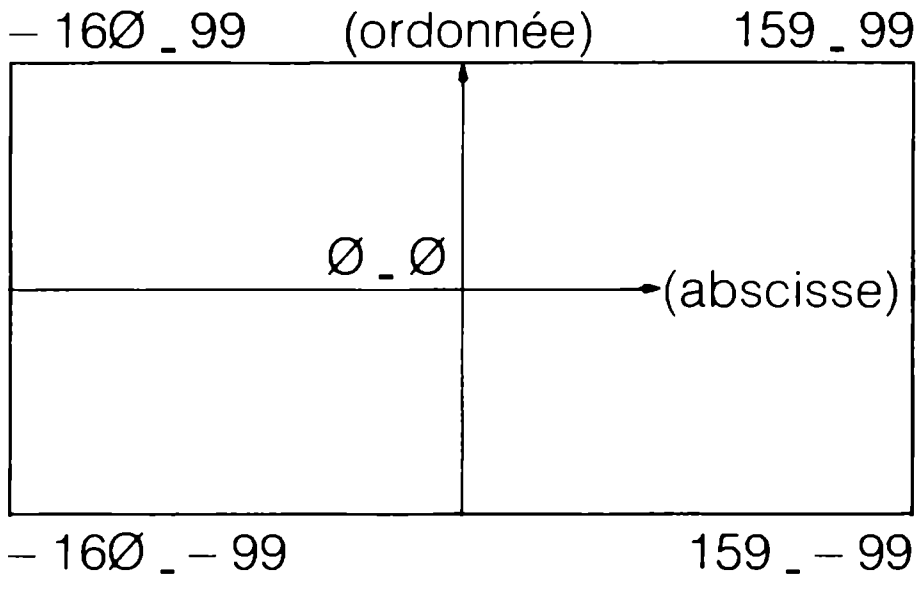
COULEURCR _ *couleur* où *couleur* est un nombre compris entre Ø et 7 sur TO7, entre Ø et 15 sur MO5, permet de choisir la couleur du tracé.

FIXEFOND _ *couleur* définit la couleur du fond.

FIXECADRE _ *couleur* définit la couleur du cadre.

FIXEPOS _ *abscisse* _ *ordonnée* positionne le point de départ du dessin.

Exemple :

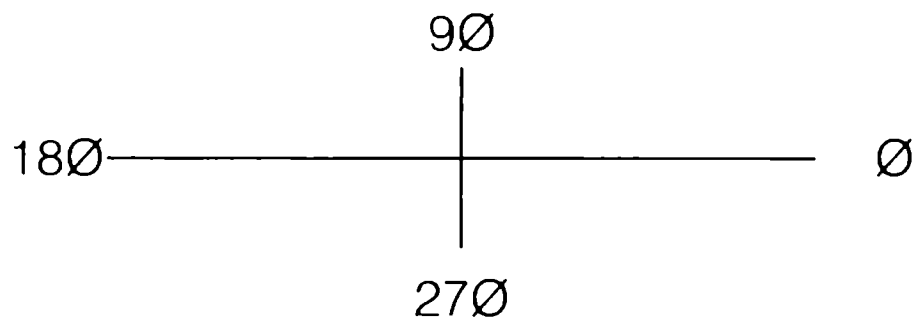


Exemple : FIXEPOS _ 4Ø _ 2Ø

LEVECR (Cette instruction n'utilise ni espace, ni argument ; de même pour toutes celles qui ne seront pas suivies du signe _).
Rend invisibles les dessins.

BAISSECR rend visibles les dessins.

FIXECAP _ *angle* définit, par rapport à l'horizontale, l'angle avec lequel les dessins seront effectués ; la valeur de l'angle est en degrés.



Exemple :

```
POUR _ TRAIT
FIXECAP _ 45
AVANCE _ 3Ø
FIN _ TRAIT
```

AVANCE _ *longueur* dessine sur l'écran un segment de telle *longueur* en tenant compte de FIXECAP, LEVECR, BAISSECR.

Exemple :

```
POUR _ EXEMPLE
AVANCE _ 3Ø
LEVECR
AVANCE _ 3Ø
FIXECAP _ 9Ø
BAISSECR
AVANCE _ 5Ø
FIN _ EXEMPLE
```

RECULE _ *longueur* Devinez !

CARRÉ _ *côté* Essayez !

ARCD _ *rayon* _ *angle*

ARCG _ *rayon* _ *angle*

Exemple :

```
POUR _ EXEMPLE
FIXECAP _ 45
ARCD _ 2Ø _ 36Ø
ARCG _ 5Ø _ 4Ø
FIN _ EXEMPLE
```

DROITE _ *angle* augmente

ou

GAUCHE _ *angle* diminue le CAP de *angle*.

VIDECRAN efface la totalité de l'écran.

INVERSE passe l'écran en négatif.

Exemple :

```
POUR _ NEG
CARRÉ _ 5Ø
INVERSE
FIN _ NEG
```

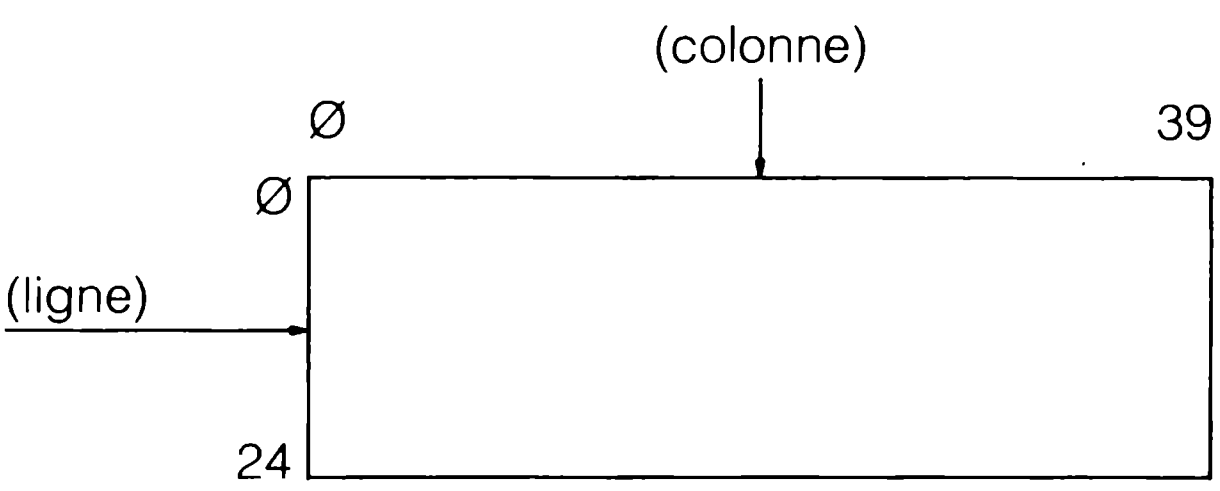
FIXECHEHOR _ *échelle* modifie l'échelle horizontale.

FIXECHEVER _ *échelle* modifie l'échelle verticale.

Exemple :

```
POUR _ ELLIPSE
FIXECHEHOR _ Ø.5
ARCD _ 5Ø _ 36Ø
FIN _ ELLIPSE
```

Chapitre 3 : Pour afficher quelque chose...



FIXECURSEUR _ *colonne* _ *ligne*

Exemple :

```
POUR _ EX
FIXECURSEUR _ 2Ø _ 1Ø
ECRIS _ "BONJOUR
FIN _ EX
```

ECRIS _ *expression* affiche à l'écran un nombre ou un texte. Dans le cas d'un texte, celui-ci doit être précédé de guillemets.

Exemples :

```
POUR _ EX1
ECRIS _ 12
FIN _ EX1
```

```
POUR _ EX2
ECRIS _ "BONJOUR
FIN _ EX2
```

Remarque : après avoir affiché votre message, l'instruction ECRIS ne passe pas à la ligne suivante. Il faut pour cela utiliser :

ALALIGNE

Exemples :

```
POUR _ E1          POUR _ E2
ECRIS _ 12         ECRIS _ 12
ALALIGNE           ECRIS _ "NON,13!
ECRIS _ "NON,13!   FIN _ E2
FIN _ E1
```

FIXETAILLE _ *largeur* _ *hauteur* définit la taille des caractères :
si le nombre est Ø : taille normale,
si le nombre est 1 : taille double.

Chapitre 4 : Instructions utilisant des variables.

Pour écrire des programmes très intéressants, il est nécessaire d'utiliser des variables. On peut se représenter une variable comme un tiroir. Son contenu peut changer. Par exemple, dans une variable numérique, on peut "ranger" un nombre.

Dans le LOGO K7 vous pouvez utiliser 26 variables dont les noms sont les lettres de l'alphabet, de A à Z.

Après avoir défini une variable, le nom de celle-ci se comporte exactement comme le nombre qu'elle représente.

Exemple : POUR _ AV peut s'écrire POUR _ AV

```
AVANCE _ 5Ø          A= _ 5Ø
FIN _ AV             AVANCE _ A
                     FIN _ AV
```

Cela ne paraît pas plus simple, mais attendez...

On peut donc se servir de variables dans toutes les instructions des chapitres 2 et 3 utilisant un argument.

Remarque :

On peut écrire A = _ 5Ø

c'est-à-dire variable = _ nombre

ou A = _ P

c'est-à-dire variable = _ variable

ENTRÉE _ *variable* cette instruction oblige l'ordinateur (à l'exécution de la procédure) à demander à l'utilisateur un nombre pour le ranger dans la variable dont le nom suit ENTREE dans la ligne.

Exemple :

```
POUR _ EX
ENTREE _ N
ECRIS _ N
FIN _ EX
```

TESTE _ *expression1 opération expression2*

Les expressions 1 et 2 sont des nombres ou des variables numériques.

Opération est = ou < ou >.

Cette instruction regarde la véracité de la condition écrite après TESTE.

Exemples :

TESTE _ A=12 ou TESTE _ W<V ou TESTE _ 14>2

SIVAPPELLE _ *nom de procédure* (si vrai)

SIFAPPELLE _ *nom de procédure* (si faux)

Si le résultat des tests est vrai, SIVAPPELLE arrête la procédure en cours d'exécution pour exécuter la nouvelle procédure indiquée, puis revient à la ligne suivante.

Même parcours pour SIFAPPELLE.

Exemple :

```
POUR _ CHANCE     POUR _ GAGNE     POUR _ PERDU
ENTREE _ N        ECRIS _ "GAGNE   ECRIS _ "PERDU
TEXTE _ N=14      FIN _ GAGNE      FIN _ PERDU
SIVAPPELLE _ GAGNE
SIFAPPELLE _ PERDU
FIN _ CHANCE
```

SIVALLER _ *nom de procédure*

SIFALLER _ *nom de procédure*

Si le résultat est vrai, arrête la procédure en cours d'exécution et passe à la procédure indiquée.

Idem pour un résultat faux.

Voir APPELLE et ALLER.

ET _ *expression1 opération expression2*

Syntaxe identique à celle de TESTE ; s'utilise après un TESTE. Le résultat de cette instruction est vrai si et seulement si le résultat précédent et la condition écrite après ET sont vrais.

OU _ *expression1 opération expression2*

Syntaxe identique à TESTE.

Le résultat est vrai si le résultat précédent OU la condition est vrai(e).

Exemple : essayez ces deux programmes, successivement, en répondant 1 puis 6 puis 12. Vous comprendrez alors la différence entre ET et OU.

```
POUR _ ESSAI1          POUR _ ESSAI2
ENTREE _ A             ENTREE _ A
TESTE _ A<1Ø           TESTE _ A<1Ø
ET _ A>5               OU _ A>5
SIVALLER _ GAGNE       SIVALLER _ GAGNE
ECRIS _ "PERDU         ECRIS _ "PERDU
FIN _ ESSAI1           FIN _ ESSAI2
POUR _ GAGNE           POUR _ GAGNE
ECRIS _ "GAGNE         ECRIS _ "GAGNE
FIN _ GAGNE            FIN _ GAGNE
```

Chapitre 5 : Manipulons des lettres.

Il existe un deuxième type de variables dans lesquelles on peut ranger des lettres, des mots, des phrases ou n'importe quelle suite de caractères que l'on peut frapper au clavier.

Elles sont, comme les précédentes, au nombre de 26 et ont pour nom les lettres de l'alphabet suivies de la lettre L : de AL à ZL.

On dit qu'il s'agit de variables alphanumériques.

LISLISTE _ *nom de variable* cette instruction demande à l'utilisateur le contenu de la variable.

Exemple :

```
POUR _ LISL
LISLISTE _ NL
ECRIS _ NL
FIN _ LISL
```

TESTEL _ *expression1 = expression2*

expression1 est une variable alphanumérique,

expression2 est soit une variable, soit un texte précédé de guillemets.

Cette instruction teste l'égalité des expressions 1 et 2.

Exemple : TESTEL _ BL="BONJOUR TESTEL _ BL=NL

ETL _ *expression1 = expression2*

OUL _ *expression1 = expression2*

Même fonctionnement que ET et OU mais avec la syntaxe de TESTEL.

Remarque : la véracité des tests se vérifie par SIVRAI et SIFAUX.

METD _ *variable1* _ *variable2*

METF _ *variable1* _ *variable2*

Ces instructions n'utilisent que des variables, permettent de placer la *variable1* au début (METD) ou à la fin (METF) de la *variable2*.

Exemple :

```
POUR _ MET1
AL= _ "BON
BL= _ "JOUR
METD _ AL _ BL
ECRIS _ BL
FIN _ MET1
```

```
POUR _ MET2
AL= _ "BON
BL= _ "JOUR
METF _ AL _ BL
ECRIS _ BL
FIN _ MET2
```

Chapitre 6 : Boucles et appels.

REPETE _ *fois* Comme son nom l'indique, ceci exécute *fois* consécutivement la totalité des lignes comprises entre REPETE et BOUCLE.

BOUCLE

Exemple :

```
POUR _ RE
REPETE _ 1Ø
ECRIS _ "BONJOUR
ALALIGNE
BOUCLE
FIN _ RE
```

APPELLE _ *nom de procédure*

Interrompt la procédure en cours pour aller exécuter la procédure appelée. A la fin de cette dernière, l'exécution reprend immédiatement après la ligne de l'interruption.

Exemple :

```
POUR _ BO
REPETE _ 1Ø
APPELLE _ S
BOUCLE
FIN _ BO
POUR _ S
ECRIS _ "SALUT
ALALIGNE
FIN _ S
```

ALLER _ *nom de procédure*

Idem que pour APPELLE sans effectuer de retour après l'interruption lors de la FIN de la procédure appelée.

Exemple :

```
POUR _ S
ECRIS _ "BONJOUR
ALALIGNE
ALLER _ CONTINUE
ECRIS _ "CA VA ?
FIN _ S
POUR _ CONTINUE
ECRIS _ "SALUT !
FIN _ CONTINUE
```

Chapitre 7 : Instructions spéciales et fonctions.

LISCR lis les données fournies par le crayon optique.

POSCR positionne l'origine des graphismes au point lu par LISCR.

JUSQUECR trace un segment du point précédent au point lu par LISCR en conservant l'origine.

Exemple :

```
POUR _ RAYONS
LISCR
JUSQUECR
ALLER _ RAYONS
FIN _ RAYONS
```

Remarque : les 3 instructions ci-dessus ne peuvent être utilisées que si vous disposez du crayon optique.

REMARQUE _ *texte* à l'exécution, le LOGO K7 ne tient pas compte de cette ligne; elle vous sert seulement pour écrire, à votre intention, des remarques concernant votre programme.

SON _ *mélodie*

Exemple :

```
POUR _ CLAIR
SON _ DODODOREL48MIREL24DOMIREREL48DO
FIN _ CLAIR
```

Chapitre 8 : Fonctions

Certains mots sont réservés dans la syntaxe LOGO.

ECHELLE donne le rapport échelle horizontale / échelle verticale.

HASARD donne un nombre entier tiré au hasard entre Ø et 9.

XCOOR est égal à l'abscisse graphique.

YCOOR est égal à l'ordonnée graphique.

CAP est égal à l'angle de tracé graphique.

SOMME _ *expression1* _ *expression2*
expression1 + expression2.

QUOTIENT _ *expression1* _ *expression2*
expression1/expression2.

DIFFERENCE _ *expression1* _ *expression2*
expression1 − expression2.

PRODUIT _ *expression1* _ *expression2*
*expression1 * expression2.*

Remarque : *expression1* et *expression2* sont soit des nombres, soit des variables numériques.

COMPTE _ (*expression*) *expression* est une variable alphanumérique; cette instruction donne la longueur de la variable alphanumérique.

DÉFINITION DES VARIABLES ALPHANUMÉRIQUES :
exp = "texte
exp1 = _ *exp2*
exp1 = _ PREMIER _ (*exp2*)
exp1 = _ DERNIER _ (*exp2*)
exp1 = _ SAUFPREMIER _ (*exp2*)
exp1 = _ SAUFDERNIER _ (*exp2*)

PREMIER ne garde que le premier caractère.

DERNIER ne garde que le dernier caractère.

SAUFPREMIER garde tout sauf le premier caractère.

SAUFDERNIER garde tout sauf le dernier caractère.

Chapitre 9 : Quelques exemples de programmes...

```
POUR _ ESSAI
ECRIS _ "DONNE UN NOMBRE
ENTREE _ N
TESTE _ N<1Ø
SIVAPPELLE _ DIX
SIFALLER _ SUP
ALALIGNE
ET _ N>5
SIVALLER _ CINQ
ECRIS _ N
ECRIS _ "EST
ALALIGNE
ECRIS _ "INFERIEUR A 5
FIN _ ESSAI
POUR _ DIX
ECRIS _ "EST INFERIEUR A 1Ø
FIN _ DIX
POUR _ CINQ
ECRIS _ "EST COMPRIS ENTRE 5 ET 1Ø
FIN _ CINQ
POUR _ SUP
ECRIS _ "EST SUPERIEUR A 1Ø
FIN _ SUP
```

```
POUR _ TABLE
ECRIS _ "QUELLE TABLE ?
ENTREE _ T
ALALIGNE
V=1
REPETE _ 1Ø
ECRIS _ T
ECRIS _ "X
ECRIS _ V
ECRIS _ "=
B= _ PRODUIT _ V _ T
ECRIS _ B
ALALIGNE
V= _ SOMME _ V _ 1
BOUCLE
FIN _ TABLE
```

```
POUR _ NOM
ECRIS _ "QUEL EST TON NOM ?
LISLISTE _ NL
V= _ Ø
REPETE _ 8
COULEURCR _ V
ALALIGNE
ECRIS _ NL
V= _ SOMME _ V _ 1
BOUCLE
FIN _ NOM
```

```
POUR _ ŒIL
ECRIS _ "RAYON ?
ENTREE _ R
APPELLE _ NCERCLE
FIXECHEVER _ Ø.5
FIXECAP _ Ø
R= _ PRODUIT _ R _ 2
APPELLE _ NCERCLE
FIN _ ŒIL
POUR _ NCERCLE
ARCD _ R _ 36Ø
FIN _ NCERCLE
```

```
POUR _ FLEUR
C=Ø
REPETE _ 4
FIXECAP _ C
ARCG _ 5Ø _ 9Ø
GAUCHE _ 9Ø
ARCG _ 5Ø _ 9Ø
C= _ SOMME _ C _ 9Ø
BOUCLE
FIXECAP _ 27Ø
AVANCE _ 8Ø
FIN _ FLEUR
```

```
POUR _ NOM
LISLISTE _ NL
REPETE _ COMPTE(NL)
BL= _ PREMIER(NL)
NL= _ SAUFPREMIER(NL)
ECRIS _ BL
ALALIGNE
BOUCLE
FIN _ NOM
```

```
POUR _ TEST
ENTREE _ N
TESTE _ N<1Ø
SIVAPPELLE _ DIX
SIFALLER _ SUP
ALALIGNE
ET _ N>5
SIVALLER _ CINQ
ECRIS _ "N<5<1Ø
FIN _ TEST
POUR _ DIX
ECRIS _ "INFERIEUR A 1Ø
FIN _ DIX
POUR _ SUP
ECRIS _ "SUPERIEUR A 1Ø
FIN _ SUP
POUR _ CINQ
ECRIS _ "SUPERIEUR A 5
FIN _ CINQ
```

```
POUR _ POLYGONES
ECRIS _ "NOMBRE DE COTES ?
ENTREE _ N
ECRIS _ "LONGUEUR DU COTE ?
ENTREE _ L
C= _ QUOTIENT _ 36Ø _ N
FIXECAP _ Ø
REPETE _ N
AVANCE _ L
GAUCHE _ C
BOUCLE
FIN _ POLYGONES
```

```
POUR _ AEXECUTER
ECRIS _ "XCOOR ?
ENTREE _ A
ECRIS _ "YCOOR ?
ENTREE _ B
ECRIS _ "ECH ?
ENTREE _ C
FIXEPOS _ A _ B
FIXECHEHOR _ C
FIXECHEVER _ C
APPELLE _ FLEUR
ALLER _ AEXECUTER
FIN _ AEXECUTER
```

```
POUR_SERPENT
ECRIS_"COMMANDE ?
LISLISTE_NL
VIDECRAN
ALLER_MOUVEMENT
FIN_SERPENT
POUR_MOUVEMENT
REPETE_COMPTE(NL)
AL=_PREMIER(NL)
NL=_SAUFPREMIER(NL)
TESTEL_AL="N
SIVAPPELLE_N
TESTEL_AL="O
SIVAPPELLE_O
TESTEL_AL="S
SIVAPPELLE_S
TESTEL_AL="E
SIVAPPELLE_E
TESTEL_AL="B
SIVAPPELLE_B
TESTEL_AL="L
SIVAPPELLE_L
BOUCLE
FIN_MOUVEMENT
POUR_N
FIXECAP_9Ø
AVANCE_5
FIN_N
POUR_O
FIXECAP_18Ø
AVANCE_5
FIN_O
POUR_S
FIXECAP_27Ø
AVANCE_5
FIN_S
POUR_E
FIXECAP_Ø
AVANCE_5
FIN_E
POUR_B
BAISSECR
FIN_B
POUR_L
LEVECR
FIN_L
```

Bon de Garantie

A renvoyer par l'acheteur dans les 8 jours suivant la date d'achat.

LOGICIEL :

CASSETTE ☐
CARTOUCHE ☐
DISQUETTE ☐

NOM DE l'ACHETEUR :

_____

ADRESSE COMPLÈTE :

_____

_____

TITRE DU LOGICIEL :

_____

DATE D'ACHAT

JOUR	MOIS	ANNÉE

CACHET DU REVENDEUR

Pour bénéficier de la garantie, EDIL-BELIN doit être en possession de ce volet complété.
Remarque : tous les renseignements contenus sur ce bon de garantie seront transférés sur un fichier informatisé.

Volet retour

A joindre obligatoirement au logiciel défectueux renvoyé à BELIN.

DÉFAUT CONSTATÉ :

- ☐ Bande sourde;
- ☐ Défaut d'enregistrement;
- ☐ Erreur d'étiquetage;
- ☐ Cassette bloquée;
- ☐ Défaut technique (à préciser) :

NOM DE L'ACHETEUR :

ADRESSE COMPLÈTE :

DATE D'ACHAT

CACHET DU REVENDEUR

JOUR	MOIS	ANNÉE

Conditions de Garantie
Logiciel Éducatif EDIL-BELIN

1 - CONDITIONS GÉNÉRALES

De cette cassette, cartouche ou disquette protégée par copyright, toute reproduction directe ou indirecte par quelque moyen électronique, électrique, magnétique, optique, laser, acoustique ou toutes autres technologies similaires existantes ou à venir est strictement interdite sous peine de poursuite.

2 - CONDITIONS DE GARANTIE

Cette garantie couvre les défauts de fabrication des composants physiques de la cassette, ou de la disquette, et les erreurs éventuelles de duplication des programmes.

ÉCHANGE STANDARD DU LOGICIEL :

– gratuitement pendant un mois à compter de la date d'acquisition pour les cassettes et les disquettes;
– contre une somme de 30 F pour les cassettes et 70 F pour les disquettes, payable par chèque à l'ordre de BELIN, pendant 1 an à compter de la date de cessation de la garantie gratuite.

3 - POUR BÉNÉFICIER DE CETTE GARANTIE, DEUX CONDITIONS :

– envoi par l'acquéreur du volet « bon de garantie » à BELIN. 8, rue Férou, 75278 PARIS Cedex 06, dans les 8 jours de la date d'acquisition (date d'achat et cachet du vendeur figurant obligatoirement sur ce volet);
– retour de la cassette, ou de la disquette dans son emballage d'origine, en recommandé avec accusé de réception.

Le non-respect par l'acquéreur de l'une quelconque des conditions énoncées emportera la déchéance immédiate de la présente garantie.

```
POUR _ PALINDROME
ECRIS _ "DONNEZ-MOI UN TEXTE
LISLISTE _ NL
APPELLE _ RENVERSE
TESTEL _ NL=ML
SIVALLER _ OUI
ECRIS _ "LE TEXTE DONNE N'EST PAS UN PALINDROME
FIN _ PALINDROME
POUR _ RENVERSE
OL= _ NL
REPETE _ COMPTE(NL)
AL= _ PREMIER(OL)
OL= _ SAUFPREMIER(OL)
METD _ AL _ ML
BOUCLE
ECRIS"L'INVERSE DE CE TEXTE EST
ECRIS _ ML
ALALIGNE
FIN _ RENVERSE

POUR _ RACINE
ECRIS _ "NOMBRE ?
ENTREE _ N
ALALIGNE
V= _ 1
REPETE _ N
W= _ PRODUIT _ V _ V
TESTE _ W>N
SIVALLER _ SUP
TESTE _ W=N
SIVALLER _ EG
V= _ SOMME _ V _ 1
BOUCLE
FIN _ RACINE
POUR _ SUP
V= _ DIFFERENCE _ V _ 1
ALLER _ EG
FIN _ SUP
POUR _ EG
ECRIS _ "LA RACINE ENTIERE DE
ECRIS _ N
ECRIS _ "EST
ECRIS _ V
FIN _ EG
```

Imp. BULOT - 357.20.79