



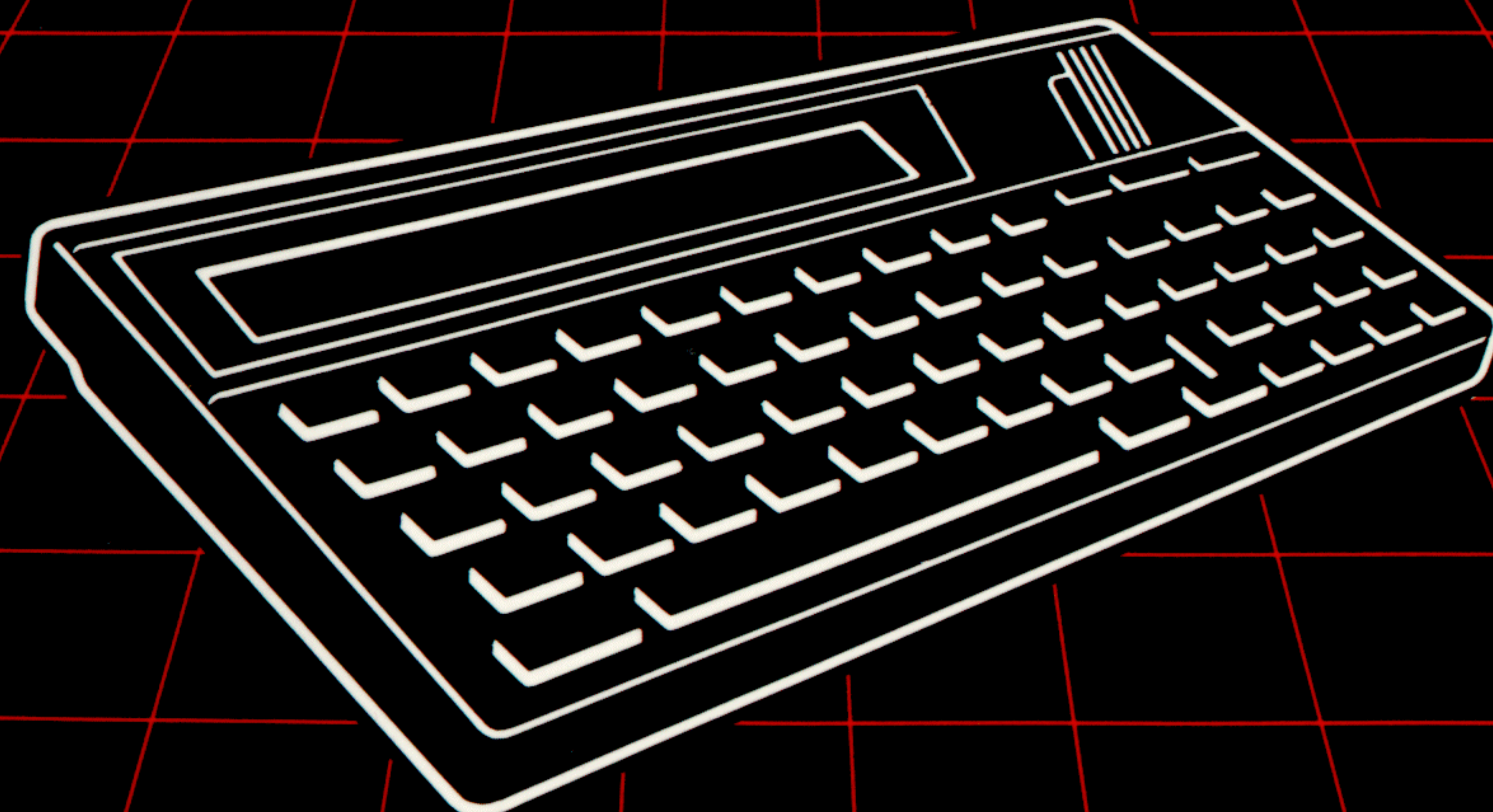
TEXAS INSTRUMENTS

TI-74 BASICALC

TEXAS INSTRUMENTS

TI-74

BASICALC



Manuel d'utilisation

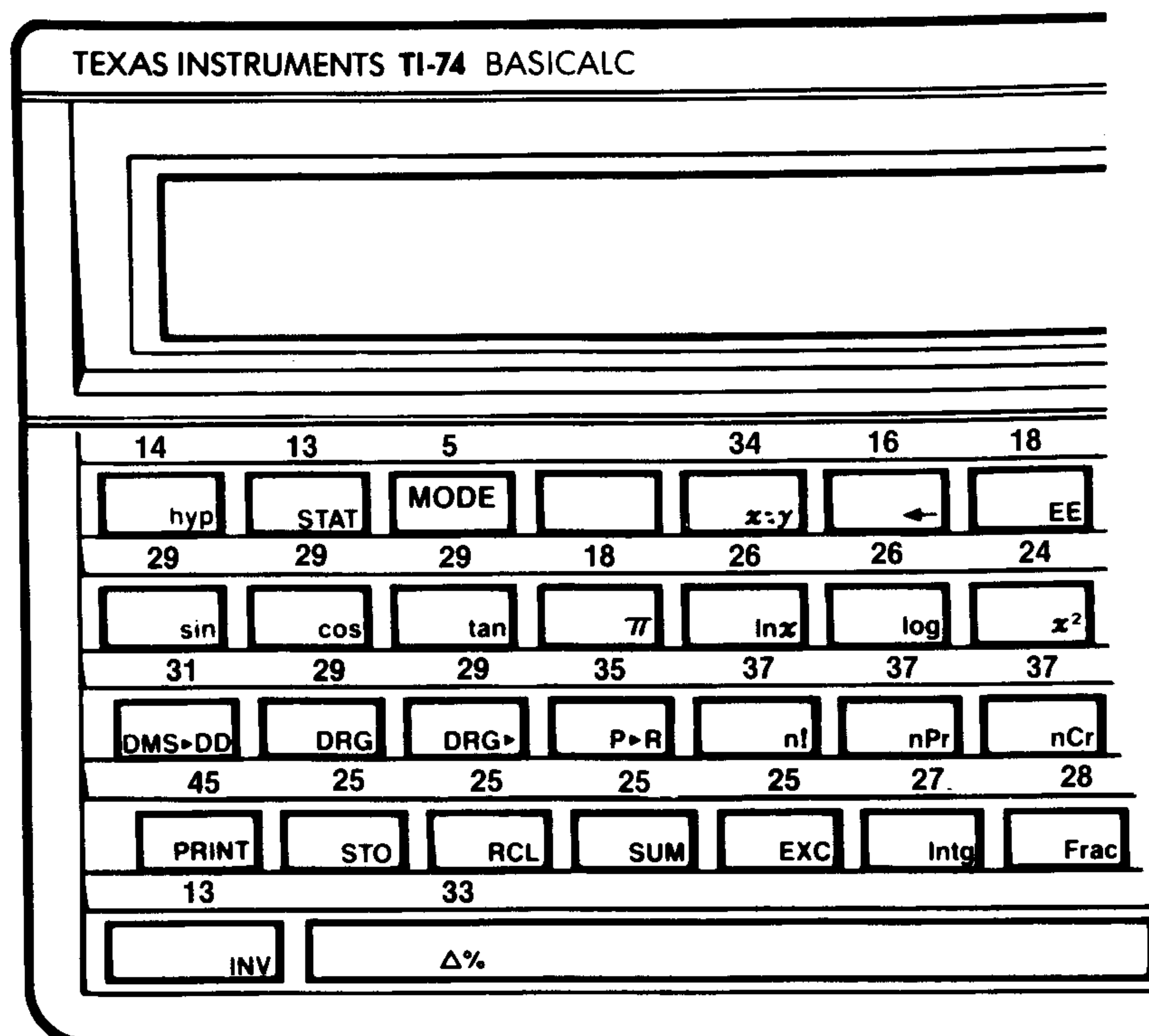


TEXAS INSTRUMENTS
TI-74 BASICALC

Manuel d'utilisation

Mode calculatrice: table de référence

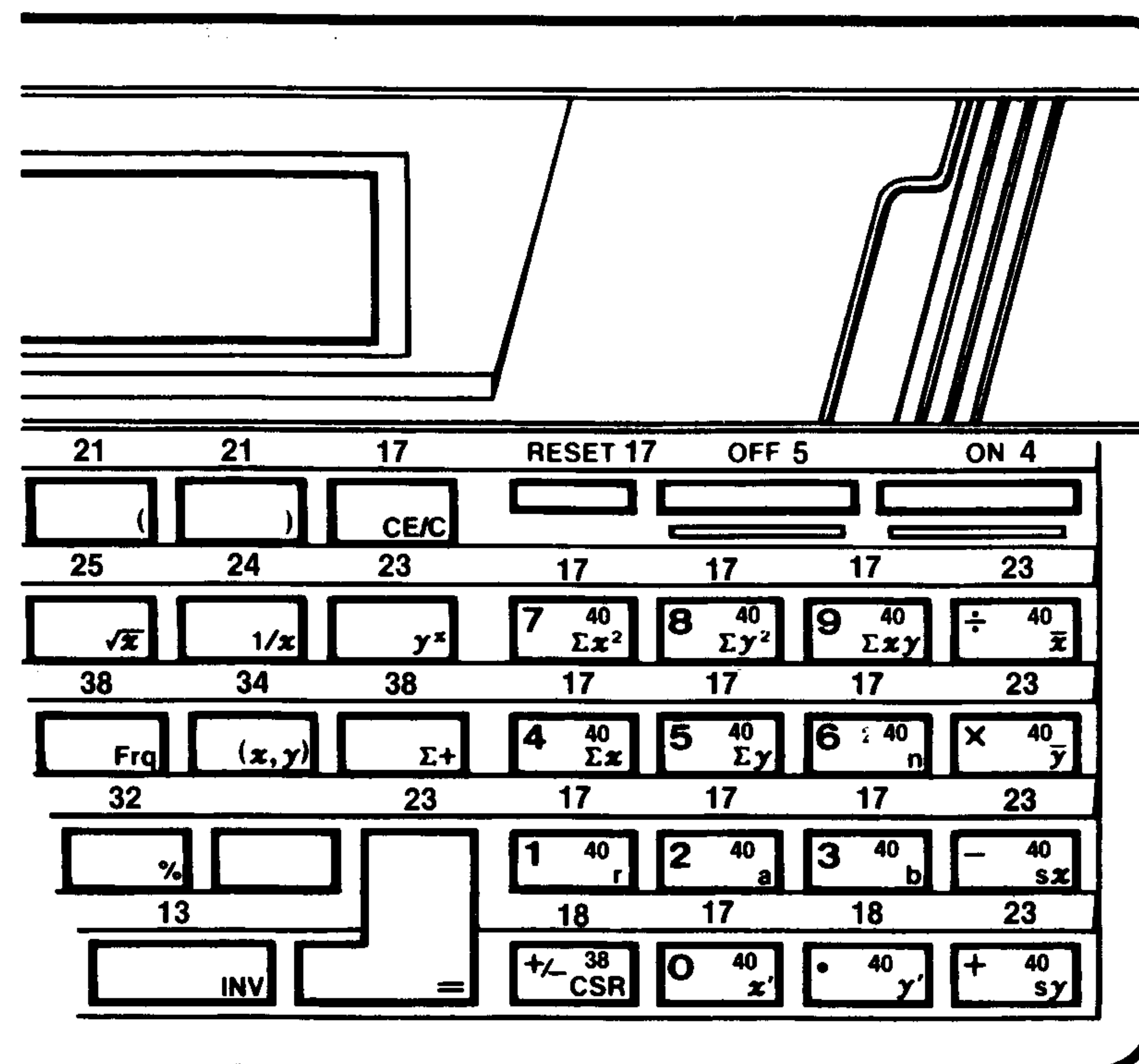
Ce schéma indique, pour chaque touche du mode calculatrice, la page du présent manuel à laquelle vous pouvez vous référer.



Ce manuel contient les instructions d'utilisation de la calculatrice Texas Instruments TI-74 BASICALC™. Vous trouverez la documentation de référence dans ce manuel et sur la carte explicative livrée avec la calculatrice.

Important

Inscrivez le numéro de série et la date d'achat de la calculatrice dans l'emplacement ci-dessous. Le numéro de série est identifié par les mots "SERIAL N°." situés au dos du boîtier. Prière de fournir ces informations dans toute correspondance concernant votre calculatrice.



TI-74

Modèle

Numéro de série

Date d'achat

TABLE DES MATIERES

CHAPITRE 1 : MISE EN ROUTE	1
Caractéristiques	2
Mise en place des piles	3
Remplacement des piles	3
Mise en marche et arrêt de la calculatrice	4
Indications lors de la mise en marche	4
Arrêt automatique et manuel	5
Mise hors service du dispositif APD	5
Mode de fonctionnement de la calculatrice	5
Fonction des touches en mode CALC	6
Abandon du mode CALC	6
Fonction des touches en mode BASIC	7
Abandon du mode BASIC	7
Les couleurs du clavier	8

CHAPITRE 2 : LE MODE CALCULATRICE

Caractéristiques du mode calculatrice	11
Caractéristiques en mode CALC	12
Fonctions secondes	12
Affichage	15
Effacement	16
Touches d'introduction de données	17
Notation scientifique	18
AOSTM (Notation algébrique directe)	19
Parenthèses	21
Correction d'erreurs d'introduction	22
Touches d'opérations arithmétiques	23
Touches inverse, carré et racine carrée	24
Utilisation des mémoires	25
Touches d'opérations logarithmiques	26
Fonctions hyperboliques	26
Touches "entier" et "fraction"	27
Sélection des unités angulaires	28
Fonctions trigonométriques	29
Conversions de deg/min/sec en degrés décimaux	30
Calculs de pourcentages	31
Entrée de couples de données	34
Conversions polaires/cartésiennes	35
Factorielles, permutations et combinaisons	36
Touches de fonctions statistiques	38
Calculs statistiques	41
Possibilités d'impression	45

TABLE DES MATIERES

CHAPITRE 3 : LE MODE BASIC	47
L'ordinateur BASIC	48
Caractéristiques en mode BASIC	48
Apprentissage de la programmation en BASIC	48
Utilisations du BASIC	48
L'affichage BASIC	49
Types d'informations visualisées	49
Texte	49
Curseur	49
Indicateurs	50
Le clavier BASIC	50
Touches primaires	50
Touches de défilement de ligne	55
Touches de modification de ligne	55
Touches de commande	56
Exécution de calculs en BASIC	57
Exécution d'instructions en BASIC	58
Exercice pratique : affectation d'un texte à une touche	59

CHAPITRE 4 : INTRODUCTION GENERALE AU BASIC

Introduction générale à la programmation	62
Affectation de valeurs à des variables	63
Affichage des informations	64
Opérations numériques	65
Opérations sur chaînes de caractères	67
Tableaux	69
Opérateurs de relation et opérateurs logiques	70
Instructions de contrôle	71
Sous-routines et sous-programmes	72
Traitement des erreurs	73
Stockage et récupération de programmes	74
Stockage et récupération de fichiers de données	75
Envoi d'informations à l'imprimante	76

CHAPITRE 5 : INDEX DE REFERENCE

Conventions utilisées dans le présent chapitre	80
Index de référence par ordre alphabétique	81
ABS	81
ACCEPT	82

TABLE DES MATIERES

ACOS	86
ACOSH	87
Sous-programme ADDMEM	88
ASC	89
ASIN	90
ASINH	91
ATN	92
ATANH	93
BREAK	94
CALL	95
CHR\$	96
CLOSE	97
CONTINUE	98
COS	99
COSH	100
DATA	101
DEG	103
DELETE	104
DIM	106
DISPLAY	107
END	110
EOF	111
Sous-programme ERR	112
EXP	114
FOR TO STEP	115
FORMAT	117
FRE	118
Sous-programme GET	119
GOSUB	120
GOTO	121
GRAD	122
IF THEN ELSE	123
IMAGE	125
INPUT au clavier	130
INPUT avec fichiers	133
INT	136
Sous-programme IO	137
Sous-programme KEY	138
KEY\$	139
LEN	140
LET	141
LINPUT	142

TABLE DES MATIERES

LIST	144
LN	145
LOG	146
NEW	147
NEXT	148
NUMBER	149
NUMERIC	150
OLD	151
ON BREAK	152
ON ERROR	154
ON GOSUB	156
ON GOTO	157
ON WARNING	158
OPEN	160
PAUSE	163
PI	165
POS	166
PRINT (à l'affichage)	167
PRINT avec fichiers	171
Sous-programme PUT	176
RAD	177
RANDOMIZE	178
READ	179
REM	180
RENUMBER	181
RESTORE	182
RETURN avec GOSUB	184
RETURN avec ON ERROR	185
RND	187
RPT\$	188
RUN	189
SAVE	191
SEG\$	192
SGN	193
SIN	194
SINH	195
SQR	196
STOP	197
STR\$	198
SUB	199
SUBEND	202
SUBEXIT	203

TABLE DES MATIERES

TAB	204
TAN	206
TANH	207
UNBREAK	208
USING	209
VAL	210
VERIFY	211

CHAPITRE 6 : UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

Conseils pour le choix de l'équipement	213
Raccordement du magnétophone à l'ordinateur	215
Messages concernant l'utilisation du magnétophone	216
Réglages du magnétophone	217
Procédure de sauvegarde de programmes	218
Vérification du stockage et récupération	222
Procédure de récupération de programmes	223
Commande et contrôle de l'imprimante à partir du BASIC	224
Accès à la mémoire du module	226
.....	230

ANNEXES

Annexe A : Symboles et messages affichés	233
Annexe B : Conditions d'erreur	235
Annexe C : Domaine des valeurs de sortie	239
Annexe D : Liste des mots réservés	241
Annexe E : Codes caractères ASCII	242
Annexe F : Opérations logiques sur les nombres	244
Annexe G : Messages d'erreur	251
Annexe H : Précision des nombres	255
Annexe I : En cas de difficultés	272
Annexe J : Index	275
Annexe K : Garantie	277
.....	287

CHAPITRE I MISE EN ROUTE

CHAPITRE 1 MISE EN ROUTE

CHAPITRE I

MISE EN ROUTE

Introduction

La calculatrice Texas Instruments TI-74 BASICALC™ est un outil de calcul à double usage qui fonctionne comme une calculatrice ou comme un ordinateur. Le manuel d'accompagnement décrit son fonctionnement.

Caractéristiques

Les caractéristiques sont les suivantes :

- Fonctions de calcul étendues.
- Langage BASIC évolué.
- Ordinateur portable. Cette machine utilise quatre piles alcalines AAA permettant une large autonomie.
- Clavier machine à écrire. Chaque touche est répétée lorsqu'elle est maintenue enfoncée.
- Connecteur permettant d'insérer un module logiciel ou un module d'extension mémoire.
- Connecteur pour périphériques permettant de raccorder des unités externes telles qu'imprimante et câble de liaison pour magnétophone.

Mise en service

Avant d'utiliser votre calculatrice, vous devez installer les piles fournies.

ATTENTION

Des décharges d'électricité statique peuvent endommager le matériel électronique. Avant d'utiliser votre calculatrice, de raccorder des périphériques ou de manipuler un module, touchez un objet métallique (par exemple une poignée de porte, une lampe de bureau), afin de vous décharger de toute électricité statique. Sans prendre cette précaution, vous risquez d'endommager le matériel.

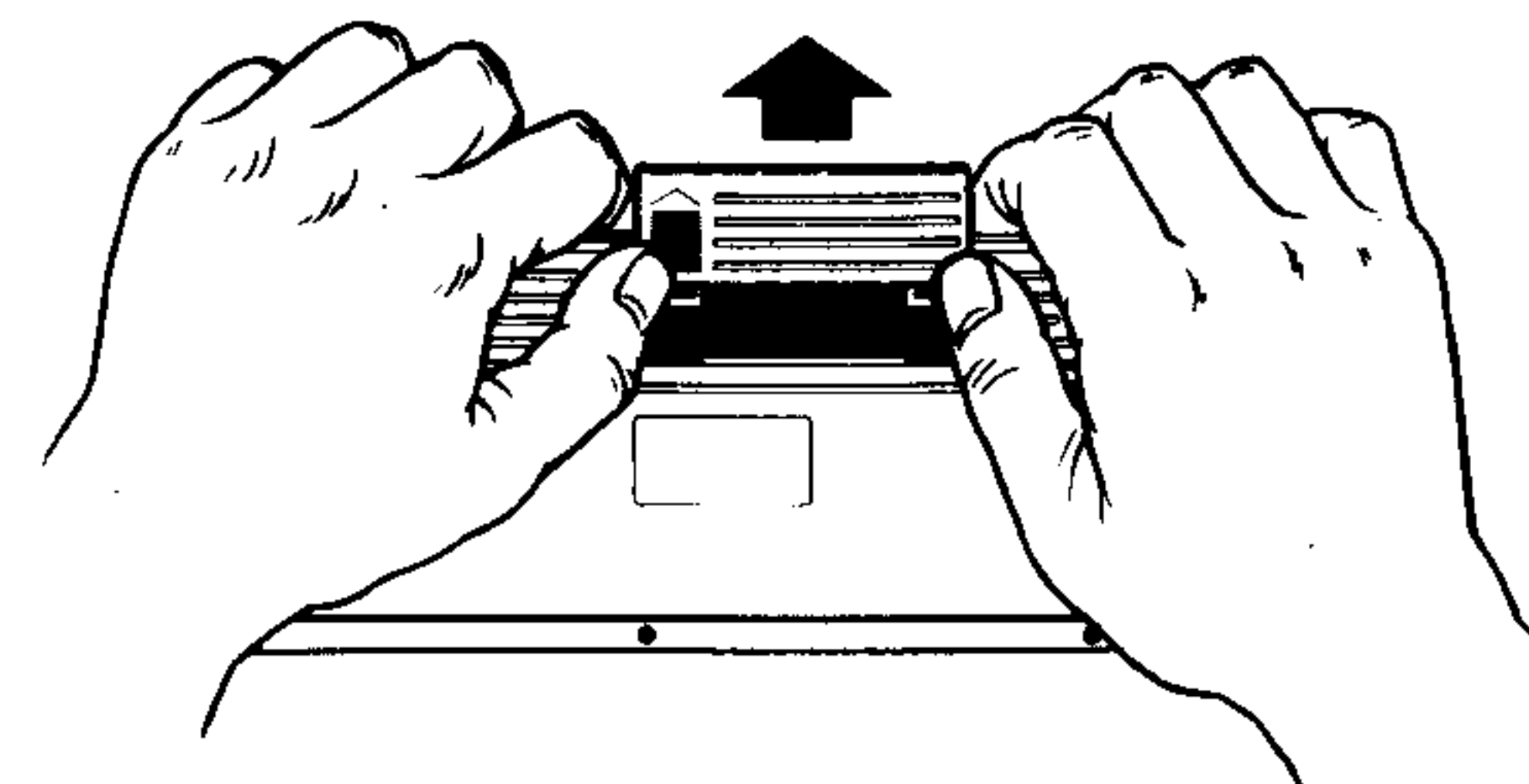
CHAPITRE I

MISE EN ROUTE

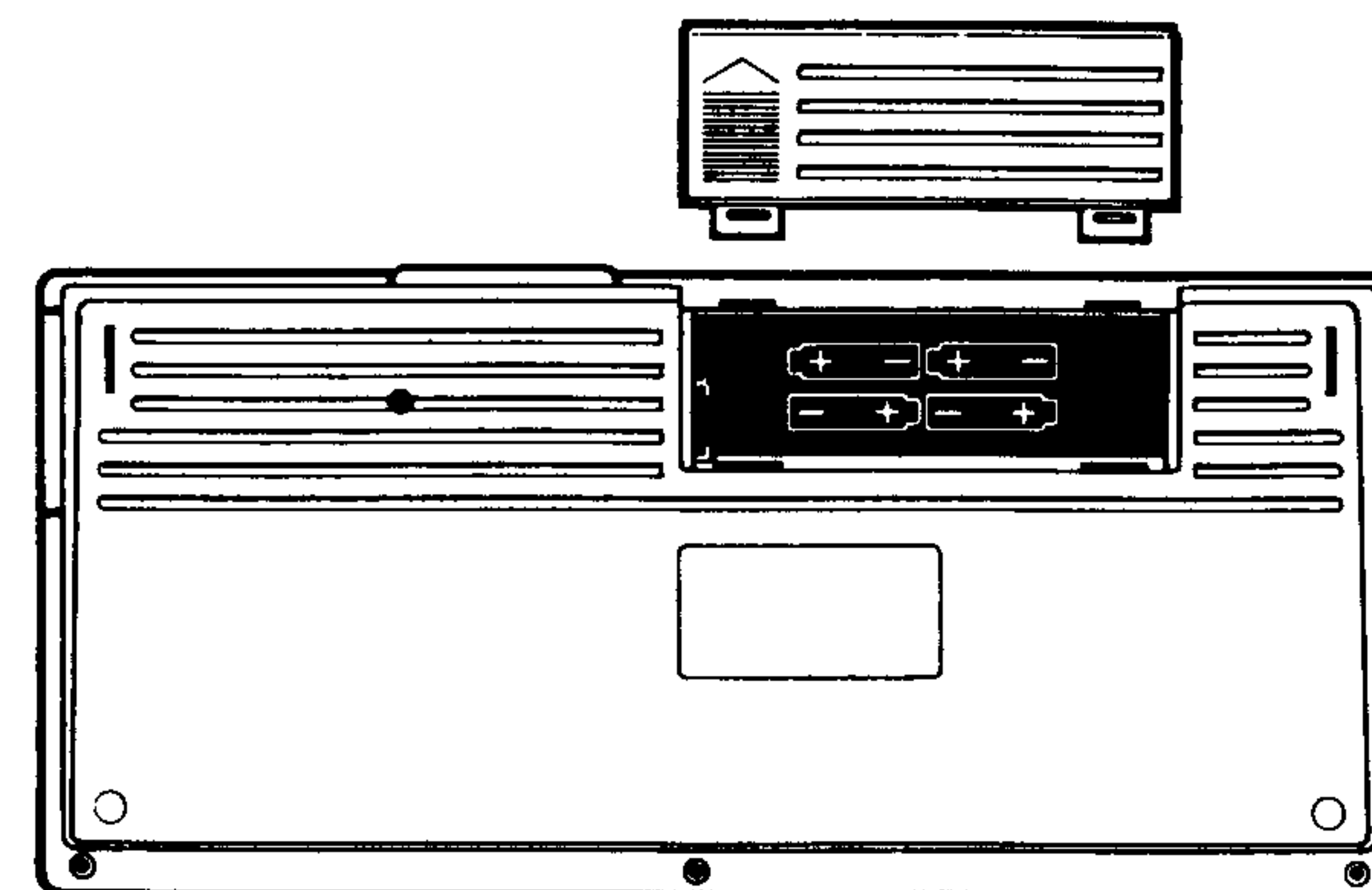
Mise en place des piles

La calculatrice est livrée avec quatre piles alcalines AAA. Pour les placer, procédez de la manière suivante :

1. Ouvrez le logement des piles en faisant glisser le couvercle comme indiqué.



2. Placez les piles neuves comme indiqué. Assurez-vous que les piles sont dans le bon sens ; vous pouvez en effet détériorer le matériel si vous placez les piles à l'envers.



3. Remettez le couvercle du logement en place.

Remplacement des piles

Lorsque les piles sont presque épuisées, l'indicateur LOW (faible) apparaît à l'affichage. Les piles doivent alors être remplacées par de nouvelles piles alcalines AAA.

CHAPITRE I MISE EN ROUTE

La mémoire permanente (Constant Memory™) garde les informations stockées pour une courte période après l'enlèvement des piles. Cependant, pour plus de précaution, vous pouvez sauvegarder des programmes importants ou des données sur une unité de stockage (telle que cassette), avant de changer les piles.

Pour remplacer les piles, procédez comme suit :

1. Si un programme BASIC est en cours d'exécution, appuyez sur la touche **[BREAK]**.
2. Arrêtez la calculatrice en appuyant sur la touche **[OFF]**.
3. Débranchez tout périphérique et tout module éventuellement installé.
4. Enlevez les piles usagées.
5. Placez les piles neuves comme décrit en page 3.

Mise en marche et arrêt de la calculatrice

A la mise en marche, les informations affichées par votre calculatrice dépendront du mode sélectionné (mode calculatrice ou mode BASIC), ainsi que des résultats de l'auto-test effectué.

Indications lors de la mise en marche

Vous mettez votre calculatrice en marche en appuyant sur la touche **[ON]**. Les indicateurs CALC ou BASIC vous indiqueront alors le mode en cours. La calculatrice vérifie ensuite s'il y a eu des changements dans le contenu de la mémoire depuis le dernier arrêt.

En cas de changement, l'un de ces deux messages s'affiche :

- Le message **W27 contents may be lost** indique que vous devez contrôler les valeurs que vous avez stockées dans les mémoires de la calculatrice, ainsi que les lignes de programme BASIC stockées. (En principe, rien n'aura changé dans les zones accessibles de la mémoire. Les zones réservées à l'usage interne sont probablement la cause de ce message).

Pour introduire des informations à partir du clavier lorsque ce message apparaît, appuyez d'abord sur la touche **[CLR]**.

CHAPITRE I MISE EN ROUTE

- Le message **W30 Initialized** signifie que la mémoire a été complètement vidée. La calculatrice ne contient pas de valeurs stockées en mémoire, ni de lignes de programme BASIC.

Pour introduire des informations à partir du clavier quand ce message apparaît, appuyez d'abord sur la touche **[CLR]**.

Arrêt automatique et manuel

Bien que la calculatrice consomme très peu d'énergie, vous pouvez économiser la durée de vie des piles en arrêtant manuellement la machine après usage. Si vous oubliez de l'arrêter, le dispositif APD (Arrêt automatique) le fait pour vous.

- Pour arrêter la calculatrice, appuyez sur la touche **[OFF]**. (Si un programme BASIC est en cours d'exécution, appuyez d'abord sur la touche **[BREAK]**).
- Le dispositif APD arrête la calculatrice automatiquement si vous n'appuyez sur aucune touche pendant environ dix minutes. (Si un programme BASIC est en cours d'exécution, le dispositif APD n'arrête pas la calculatrice).

Mise hors service du dispositif APD

Dans certains cas, vous désirez peut-être mettre le dispositif APD hors service afin de pouvoir garder des informations à l'affichage pendant plus de dix minutes.

Pour ce faire, appuyez simultanément sur les touches **[FN]**, **[CTL]**, **[SHIFT]** et **[/]**. Cette combinaison de touches fonctionne à la fois en mode CALC et en mode BASIC.

Lorsque vous arrêtez et remettez en marche votre calculatrice, le dispositif APD est réactivé.

Modes de fonctionnement de la calculatrice

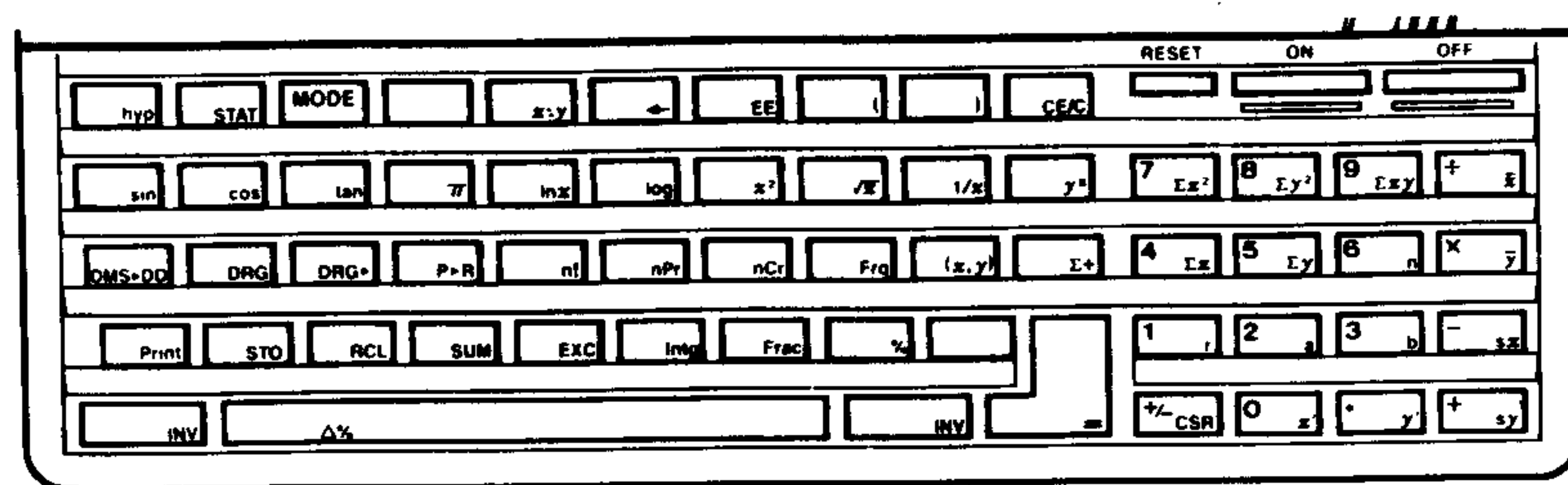
La machine fonctionne dans l'un des deux modes suivants : CALC ou BASIC, que vous sélectionnez avec la touche **[MODE]**. En mode BASIC, elle fonctionne comme un ordinateur avec langage de programmation BASIC intégré. Les fonctions disponibles sur le clavier dépendent du mode utilisé.

CHAPITRE I

MISE EN ROUTE

Fonction des touches en mode CALC

La fonction d'une touche en mode CALC est identifiée par le petit symbole gravé sur la partie inférieure. (Vous pouvez utiliser les touches numériques de 0 à 9 pour entrer des données numériques et exécuter certaines fonctions de la calculatrice). Chaque touche de la calculatrice est expliquée en détail dans le chapitre 2 du présent manuel.



Abandon du mode CALC

Lorsque vous arrêtez la calculatrice en mode CALC ou lorsque vous sortez du mode CALC en sélectionnant le mode BASIC, certaines caractéristiques du mode CALC sont maintenues et d'autres annulées.

Sont maintenus :

- Le contenu de chaque mémoire de la calculatrice
- Le mode STAT et les registres statistiques s'ils sont en service
- La dernière unité d'angle sélectionnée (que vous pouvez changer en mode BASIC).

Sont annulés :

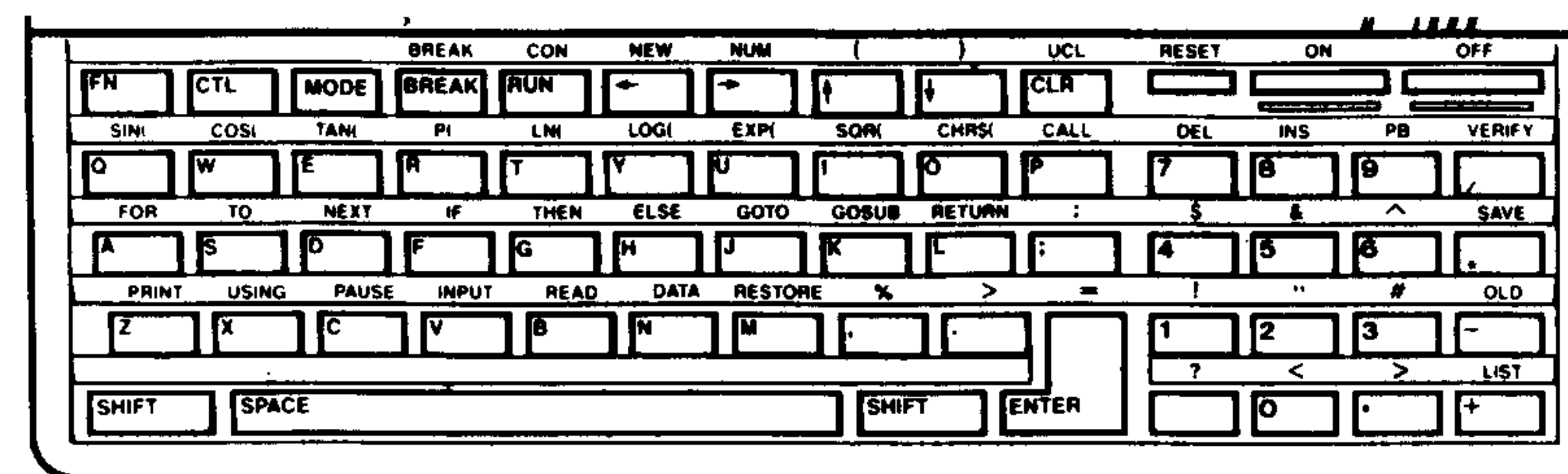
- Toute séquence de touches non terminée
- Toute opération arithmétique en attente ou partiellement introduite
- Le contenu de l'affichage.

CHAPITRE I

MISE EN ROUTE

Fonction des touches en mode BASIC

En mode BASIC, vous identifiez la fonction d'une touche par le(s) caractère(s) imprimé(s) en gros sur la touche et par le symbole imprimé au-dessus. Chaque touche de BASIC est expliquée en détail dans le chapitre 3.



Abandon du mode BASIC

Lorsque vous arrêtez la calculatrice en mode BASIC ou lorsque vous sortez du mode BASIC en sélectionnant le mode CALC, certaines caractéristiques BASIC sont maintenues et d'autres annulées. (Si un programme est en cours d'exécution, vous devez appuyer sur la touche **[BREAK]** avant d'arrêter votre calculatrice ou de sélectionner le mode CALC).

Sont maintenues :

- Toute ligne de programme stockée
- La dernière unité d'angle sélectionnée (que vous pouvez changer en mode CALC)
- Toute chaîne de caractères que vous aurez affectée à des touches numériques.

Sont annulés :

- Toute séquence de touches non terminée
- Toute affectation de variables et les dimensions des tableaux
- Toute opération utilisant une unité raccordée au connecteur de périphériques
- Le contenu de l'affichage.

CHAPITRE I

MISE EN ROUTE

Les couleurs du clavier

Les symboles du clavier sont repérés par des couleurs, et leur position vous indique la façon d'y accéder. Chaque symbole représente un caractère, un groupe de caractères ou une action que la calculatrice peut exécuter.

Touches communes à tous les modes

Les touches **[MODE]**, **[RESET]**, **[ON]** et **[OFF]** sont communes à tous les modes. Même si vous appuyez sur une touche qui sélectionne une fonction seconde, c'est l'action de ces touches qui prévaut.

Symboles en bleu

Tous les symboles imprimés en bleu correspondent à des fonctions du mode calculatrice. Pour y accéder, il n'est pas nécessaire d'appuyer sur une touche préfixe. Cependant, certaines sont associées à une seconde fonction, accessible par pression sur les touches **[INV]** ou **[hyp]**.

Symboles en jaune

Tous les symboles imprimés en jaune sont actifs en mode calculatrice. Ce sont des touches de fonctions statistiques, accessibles par pression préalable sur la touche préfixe **[STAT]**. Ces fonctions ne sont utilisables qu'avec un jeu de données statistiques.

Symboles en gris

Tous les symboles imprimés en gris correspondent à des fonctions du mode BASIC, accessibles par pression préalable sur la touche préfixe **[FN]**.

Symboles blancs au-dessus d'une touche

Les symboles imprimés en blanc au-dessus d'une touche sont utilisés en mode BASIC pour introduire un caractère ou exécuter une action. Ces fonctions sont accessibles par pression préalable sur la touche **[SHIFT]**.

CHAPITRE I

MISE EN ROUTE

Symboles blancs du clavier numérique

Les symboles imprimés en blanc sur une touche du clavier numérique sont directement accessibles, à la fois en mode calculatrice et en mode BASIC, excepté la touche **[+/-]** qui n'est pas disponible pour le mode BASIC. Il n'est pas nécessaire d'appuyer sur une touche préfixe pour utiliser l'un de ces chiffres ou de ces opérateurs.

Lettres isolées en blanc

Les lettres minuscules sont directement accessibles en mode BASIC, et les majuscules, par pression préalable sur la touche **[SHIFT]**. Pour obtenir directement une suite de majuscules, vous pouvez utiliser la séquence de touches **[SHIFT] [UCL]** (verrouillage en position majuscules).

Autres symboles en blanc

Les caractères imprimés en blanc sur les touches du clavier machine à écrire et qui ne sont pas des lettres isolées sont directement accessibles en mode BASIC.

CHAPITRE II LE MODE CALCULATRICE

CHAPITRE II LE MODE CALCULATRICE

Le mode calculatrice

Avant d'utiliser la calculatrice, sélectionnez le mode correspondant en appuyant sur la touche **[MODE]** jusqu'à ce que l'indicateur de mode **CALC** apparaisse. Ce chapitre vous permettra d'apprendre le fonctionnement de votre calculatrice.

Caractéristiques du mode calculatrice

En mode calculatrice, la machine exécute des calculs arithmétiques simples, ainsi que de nombreuses opérations complexes.

Caractéristiques en mode CALC

Dans ce mode, la machine présente les caractéristiques et possibilités suivantes :

- Précision de 13 chiffres
- Capacité numérique supérieure à beaucoup d'autres calculatrices
- Système AOS™ (notation algébrique directe)
- Fonctions logarithmiques, hyperboliques et trigonométriques
- Conversions d'angles, de coordonnées bidimensionnelles, et de degrés/minutes/secondes
- Fonctions pourcentage et écart en pourcentage
- Calculs de permutations et de combinaisons
- Statistiques à une ou deux variables, y compris l'analyse de régression.

Fonctions secondes

Une touche exécute normalement la fonction qui est indiquée en sa partie inférieure. Afin de fournir une grande variété de fonctions sans pour autant encombrer le clavier, beaucoup de touches exécutent une fonction supplémentaire lorsqu'elles sont utilisées en association avec une touche préfixe.

CHAPITRE II LE MODE CALCULATRICE

Accès aux fonctions secondes

Vous pouvez accéder à une fonction seconde en appuyant sur une touche préfixe (**[STAT]**, **[INV]** ou **[hyp]**), puis sur la touche de la fonction seconde souhaitée. Si vous appuyez sur une touche préfixe puis sur une touche qui n'a pas de fonction seconde, la calculatrice ne tient pas compte du préfixe et exécute la fonction normale de la touche.

[STAT] – Préfixe "statistiques"

Pour exécuter une fonction "statistiques", appuyez sur **[STAT]** puis sur la touche appropriée. Seules les touches du clavier numérique ont des fonctions statistiques. Notez que le nom des fonctions est inscrit dans la même couleur que celle de la touche **[STAT]**.

Bien que **[+/-]** et **[CSR]** correspondent en réalité à la même touche, ce manuel utilise le symbole représentant la fonction à exécuter. Lorsqu'une séquence de touches commence par **[STAT]**, vous pouvez situer cette fonction sur la calculatrice en repérant les symboles dont le code couleur correspond à la touche **[STAT]** sur le clavier numérique. Pour les fonctions "statistiques", voir le paragraphe "Touches de fonctions statistiques".

Nota : Si vous appuyez par inadvertance sur la touche **[STAT]**, vous pouvez :

- La remplacer par une autre présélection en appuyant sur la touche préfixe désirée.
- L'annuler en appuyant une deuxième fois sur la touche **[STAT]**.

Si une autre présélection est validée, vous pouvez la remplacer par la présélection **[STAT]** en appuyant sur la touche **[STAT]**.

[INV] – Préfixe inverse

Pour exécuter une fonction inverse, appuyez sur la touche **[INV]** puis sur la touche désirée.

Touche	Fonction	Fonction inverse
[EE]	passage en notation scientifique	annulation de la notation scientifique
[sin]	sinus	arc-sinus
[cos]	cosinus	arc-cosinus

CHAPITRE II

LE MODE CALCULATRICE

Touche	Fonction	Fonction inverse
[tan]	tangente	arc-tangente
[lnx]	logarithme népérien	e^x
[log]	logarithme décimal	10^x
[x ²]	carré de x	racine carrée de x
[√x]	racine carrée de x	carré de x
[y ^x]	y à la puissance x	racine x ^{ième} de y
[SUM]	addition en mémoire	soustraction mémoire
[DRG]	positionne sur unité angulaire suivante	repositionne sur unité précédente
[DRG→]	positionne sur unité angulaire suivante et convertit l'angle	repositionne sur unité précédente et convertit l'angle
[P→R]	conversion coordonnées polaires/cartésiennes	conversion cartés./ polaires
[DMS→DD]	conversion deg/min/sec en degrés décimaux	degrés décimaux en deg/min/sec
[Σ+]	introduction d'une donnée statistique	retrait d'une donnée statistique

Préfixe inverse

Nota : Si vous appuyez par inadvertance sur la touche [INV], vous pouvez :

- Remplacer cette présélection par une autre en appuyant sur la touche préfixe désirée.
- L'annuler en appuyant à nouveau sur la touche [INV].

[hyp] – Préfixe hyperbolique

Pour exécuter une fonction hyperbolique, appuyez sur la touche [hyp] puis sur la touche désirée. Seules les touches [sin], [cos] et [tan] ont des fonctions hyperboliques. Les touches trigonométriques exécutent aussi les fonctions inverses des fonctions hyperboliques, accessibles par pression simultanée sur les touches [INV] et [hyp], puis sur la touche désirée.

CHAPITRE II

LE MODE CALCULATRICE

Par exemple, toutes les fonctions de la touche [sin] sont accessibles de la manière suivante :

- Appuyez sur [sin] pour trouver le sinus.
- Appuyez sur [INV] [sin] pour trouver l'arc-sinus.
- Appuyez sur [hyp] [sin] pour trouver le sinus hyperbolique.
- Appuyez simultanément sur [INV] et [hyp] puis sur [sin] pour trouver l'arc-sinus hyperbolique.

Nota : Si vous appuyez par inadvertance sur la touche [hyp], vous pouvez :

- Remplacer cette présélection par une autre en appuyant sur la touche préfixe désirée.
- L'annuler en appuyant à nouveau sur la touche [hyp].

Si vous appuyez par inadvertance sur les touches [INV] [hyp], vous pouvez :

- Remplacer cette séquence par le mode "STAT" en pressant [STAT].
- L'annuler en appuyant sur la touche [hyp].
- L'annuler en appuyant sur la touche [INV].

Affichage

L'affichage est divisé en trois zones distinctes. La zone numérique affiche les entrées et les résultats des calculs. La zone de rappel de fonction affiche le type d'opérations et de fonctions en cours d'exécution. La troisième zone, au bas de l'affichage, comporte les indicateurs de la calculatrice.

Affichage des indicateurs

La calculatrice vous informe de son état en allumant des indicateurs sur l'affichage. Par exemple, CALC apparaît lorsque la machine est en mode calculatrice. D'autres indicateurs peuvent apparaître, comme décrit ci-dessous.



CHAPITRE II LE MODE CALCULATRICE

Indicateur	Signification
DEG	Les unités de mesure angulaire sont sélectionnées dans l'ordre degrés, radians, grades.
RAD	
GRAD	
STAT	La calculatrice est en mode statistique.

Sélection des unités d'angle

La sélection des unités de mesure angulaire change lorsque vous passez de degrés en radians ou en grades. Vous pouvez aussi faire cette sélection en sens inverse.

Accès au mode "statistiques"

La calculatrice passe en mode "statistiques" lorsque vous introduisez des données statistiques. Elle le quitte lorsque vous effacez ces données.

Affichage des messages

Les caractères matriciels permettent d'afficher des messages et des symboles. Les conditions d'erreur sont expliquées par des messages qui peuvent occuper la zone numérique et la zone de rappel de fonction. La frappe des touches de calcul entraîne généralement l'apparition de symboles dans la zone de rappel de fonction uniquement. La liste des messages et des symboles est indiquée en annexe A.

Effacement

En appuyant sur la touche **[←]**, sur la touche **[CE/C]** deux fois (ou une seule fois suivant les circonstances), vous pouvez obtenir différents degrés d'effacement de la zone de calcul. (Si vous voulez effacer la mémoire ou les registres statistiques, reportez-vous aux rubriques "Utilisation des mémoires" et "Touches de fonctions statistiques" du présent chapitre).

[←] – Effacement de caractère

Cette touche permet d'éliminer le dernier caractère introduit, que ce soit un chiffre ou une virgule. Vous avez également la possibilité d'effacer tous les caractères entrés. Avant d'introduire un exposant en notation scientifique, utilisez cette touche pour éliminer les caractères de la mantisse. Lorsque vous entrez un exposant en notation scientifique, cette touche élimine l'exposant ou réintroduit la mantisse, si l'exposant est égal à 000.

CHAPITRE II LE MODE CALCULATRICE

Cette touche vous permet également de corriger le résultat d'un calcul.

[CE/C] – Effacement d'une entrée/effacement général

La touche **[CE/C]** efface toutes les opérations en attente, les entrées erronées ou les conditions d'erreur indiquées ci-dessous.

- Pour effacer une entrée numérique erronée lorsque le curseur clignote, appuyez sur **[CE/C]**. Vous pouvez ensuite introduire le nombre correct et continuer votre calcul.
- Pour effacer l'affichage et toutes les opérations en attente, appuyez sur la touche **[CE/C]** deux fois, ou une seule fois si le curseur ne clignote pas.
- Pour effacer une condition d'erreur (indiquée par un message correspondant sur l'affichage), appuyez sur **[CE/C]** une fois. La liste des conditions d'erreur est en annexe B.

[RESET] – Réinitialisation

La touche **[RESET]** est utilisée "en dernière extrémité", pour réinitialiser la calculatrice, lorsqu'un problème intervient et vous empêche d'introduire des informations à partir du clavier. Cette touche affleure le boîtier de la calculatrice, pour vous éviter de l'actionner par inadvertance. (Une pression sur la touche **[RESET]** provoque généralement l'affichage du message **W27 contents may be lost**. Ce message est expliqué à la page 4).

Touches d'introduction de données

Les touches d'introduction de données permettent d'introduire les données numériques nécessaires à l'exécution de calculs. Vous pouvez entrer un maximum de 13 chiffres pour vos calculs. Cependant, les résultats affichés ne comportent que 10 chiffres. Vous pouvez introduire des nombres en notation scientifique ou standard.

[0]-[9] – Chiffres

Les touches numériques permettent d'introduire des nombres à l'affichage. En notation standard, vous pouvez entrer 13 chiffres maximum et un point décimal. En notation scientifique, vous pouvez entrer 13 chiffres maximum, un point décimal et 3 chiffres pour l'exposant.

CHAPITRE II LE MODE CALCULATRICE

[.] – Point décimal

La touche [.] introduit un point décimal. En notation scientifique, un exposant ne peut avoir de point décimal. Si le résultat du calcul affiché est un nombre entier avec un point décimal, le nombre n'est pas réellement un nombre entier dans la machine, mais il est arrondi à un nombre entier pour l'affichage. Le point n'est pas indiqué pour des nombres entiers réels.

[+/-] – Changement de signe

La touche [+/-] change le signe du nombre affiché, facilitant ainsi l'introduction de nombres et d'exposants négatifs. Pour entrer un nombre négatif, introduisez le nombre comme une valeur positive puis appuyez sur la touche [+/-].

[π] – Pi

La touche [π] introduit une valeur de pi à 13 chiffres significatifs, à savoir 3.141592653590. La valeur de pi affichée est arrondie à dix chiffres : 3.141592654.

[EE] – Notation scientifique

La touche [EE] permet d'introduire des nombres en notation scientifique, entre $\pm 1 \times 10^{-128}$ et $\pm 9.99999999999 \times 10^{127}$.

Nota : Si le résultat d'un calcul n'est pas égal à zéro, et n'est pas compris entre -9999999999 et $-.0000000001$, ou entre $.0000000001$ et 9999999999 , la calculatrice affiche automatiquement le nombre en notation scientifique, même si vous n'avez pas appuyé sur la touche [EE].

Notation scientifique

Les calculs scientifiques portent souvent sur des nombres très grands et des nombres très petits. La notation scientifique permet de les manipuler plus facilement. Une fois que vous avez employé la notation scientifique pour un calcul, tous les résultats sont affichés sous cette forme, jusqu'à ce que vous annuliez cette notation.

CHAPITRE II LE MODE CALCULATRICE

Le tableau suivant montre des nombres en notation standard et en notation scientifique.

Notation standard	Notation scientifique
6789	6.789×10^3
0.0000021	2.1×10^{-6}
-16389043	-1.6389043×10^7
8.775	8.775×10^0

Entrée en notation scientifique

Pour introduire un nombre en notation scientifique, procédez comme suit :

1. Entrez la mantisse. Si elle est négative, appuyez sur [+/-].
2. Appuyez sur la touche [EE]. ("E + 000" apparaît à l'affichage).
3. Entrez l'exposant. S'il est négatif, appuyez sur [+/-]. Si vous appuyez sur une touche incorrecte en introduisant l'exposant, appuyez de nouveau sur les chiffres corrects. La calculatrice remplace les anciens chiffres par les trois derniers chiffres introduits.

Pour annuler la notation scientifique, procédez de la manière suivante :

- Appuyez sur [INV] [EE]. Notez que cette séquence de touches permet également de convertir un nombre en notation standard.
- Remettez la calculatrice à zéro en appuyant sur [CE/C] deux fois, ou une seule fois si le curseur ne clignote pas.

AOSTM (Notation algébrique directe)

Le système de notation algébrique directe AOSTM permet d'introduire des nombres et des opérations combinées dans la calculatrice, en séquence simple et directe. Pour garantir l'ordre d'exécution des calculs, le système de notation algébrique directe utilise les règles universellement reconnues de la hiérarchie algébrique, qui assignent des priorités aux diverses opérations mathématiques.

CHAPITRE II

LE MODE CALCULATRICE

But

Sans règles algébriques bien déterminées, un problème tel que $16 - 8 \div 2 + 6$ pourrait avoir plusieurs solutions, selon l'ordre dans lequel les opérations sont effectuées. Cependant, le système AOS™ résout ce problème en divisant d'abord ($8 \div 2$) puis en soustrayant et en additionnant, ce qui donne : $16 - 8 \div 2 + 6 = 18$.

Opérations en attente

Dans un problème tel que $4 \times 5 = 20$, la simple introduction de $4 [\times] 5$ ne permet pas d'avoir de résultat. L'opération doit être achevée par une pression sur une touche appropriée telle que $[=]$. (Notez que les multiplications peuvent être résolues par toute opération ayant une priorité inférieure ou égale dans la hiérarchie algébrique). Jusqu'à sa résolution, l'opération 4×5 reste en attente.

Nota : Les fonctions immédiates s'exécutent automatiquement ; il n'est pas nécessaire d'appuyer sur une autre touche.

Hiérarchie algébrique

La hiérarchie algébrique du système AOS résout les opérations selon les catégories suivantes, citées par ordre de priorité décroissante.

1. Toutes les fonctions immédiates telles que $[\sin]$, $[\tan]$, $[\text{Intg}]$, $[1/x]$,...
2. La fonction d'écart en pourcentage $[\Delta \%]$
3. Les fonctions de puissance et racine quelconques $[y^x]$, $[\text{INV}] [y^x]$
4. Les fonctions de multiplication et division $[\times]$, $[\div]$
5. Les fonctions d'addition et soustraction $[+]$, $[-]$
6. La touche $[=]$.

Grâce au système AOS, les opérations à faible priorité sont mises en attente jusqu'à ce que les opérations à grande priorité soient achevées. Une opération de la première catégorie s'achève automatiquement. Une opération des catégories 2, 3, 4 ou 5 est achevée par une opération de la même catégorie ou d'une catégorie inférieure. La touche $[=]$ achève toutes les opérations.

CHAPITRE II

LE MODE CALCULATRICE

Exemple

Le calcul suivant est un exemple de hiérarchie algébrique et d'opérations en attente.

$$4 + 8 \div 2^5 - 2 = 2.25$$

Procédure	Appuyer sur	Affichage
Remettre à zéro	[CE/C] [CE/C]	0
Introduire l'addition en attente	4 [+]	4
Introduire la division en attente (l'addition est aussi en attente)	8 [÷]	8
Introduire l'élévation à la puissance en attente (l'addition et la division sont aussi en attente)	2 [y ^x]	2
Achever les opérations en attente et entrer la soustraction en attente	5 [-]	4 . 25
Achever le calcul	2 [=]	2 . 25

Parenthèses

Pour certains calculs, il est nécessaire de regrouper les opérations dans un ordre différent de celui établi par le système AOS™. Une série de nombres et d'opérations placée entre parenthèses est prioritaire par rapport aux opérations situées en dehors des parenthèses. A l'intérieur de chaque niveau, la calculatrice respecte l'ordre de la hiérarchie algébrique.

[(), [)] – Parenthèses

La touche $[()]$ ouvre une parenthèse et la touche $[)]$ ferme la parenthèse. Il est possible d'utiliser un maximum de 22 niveaux de parenthèses et d'opérations.

Si vous appuyez sur la touche $[)]$ lorsqu'il n'y a pas de parenthèses en attente, la calculatrice exécute les opérations en attente comme si vous aviez appuyé sur la touche $[=]$.

CHAPITRE II LE MODE CALCULATRICE

Exemple

$$7 \times (3 + 5) = 56$$

Procédure	Appuyer sur	Affichage
Remettre à zéro	[CE/C] [CE/C]	0
Entrer la multiplication en attente	7 [×]	7
Commencer l'expression entre parenthèses	[(] 3	3
Entrer l'addition en attente	[+] 5	5
Fermer la parenthèse	[)]	8
Achever les opérations en attente	[=]	56

Si ce calcul était effectué sans les parenthèses, la hiérarchie algébrique ferait la multiplication avant l'addition, ce qui donnerait 26 comme résultat.

Correction d'erreurs d'introduction

Vous pouvez occasionnellement introduire un nombre incorrect ou appuyer sur la mauvaise touche de fonction. Pour remettre la calculatrice à zéro et recommencer, appuyez sur [CE/C] deux fois.

Dans beaucoup de cas cependant, vous pouvez corriger une erreur d'introduction sans remettre la calculatrice à zéro.

Correction de nombres et de fonctions immédiates

La touche [CE/C] a deux fonctions. Lorsque le curseur clignote, une seule pression sur cette touche permet d'effacer la donnée introduite. Appuyez deux fois sur [CE/C] ou lorsque le curseur ne clignote pas, pour effacer tous les calculs.

- Appuyez une fois sur [CE/C] pour effacer l'affichage si vous avez introduit un nombre incorrect. Introduisez ensuite le nombre correct et poursuivez votre calcul.
- Si le curseur ne clignote plus (vous avez appuyé sur une touche de fonction immédiate telle que $[x^2]$ ou $[\sin]$), appuyez sur la touche $[-]$ pour effacer l'affichage. Introduisez ensuite le nombre et la fonction corrects et poursuivez votre calcul.

CHAPITRE II LE MODE CALCULATRICE

Correction d'opérations en attente

Si vous appuyez sur une touche incorrecte concernant une opération en attente, vous ne pourrez pas toujours corriger l'opération sans effacer le calcul. Les touches d'opération en attente sont : $[\Delta\%]$, $[y^x]$, $[INV] [y^x]$, $[\div]$, $[\times]$, $[-]$ et $[+]$.

- Si l'opération incorrecte a une priorité égale ou supérieure à celle de l'opération désirée, vous pouvez appuyer sur la touche de l'opération correcte et continuer votre calcul.
- Si l'opération incorrecte a une priorité inférieure à celle de l'opération désirée, le simple fait d'appuyer sur la touche correcte ne corrigera pas le problème. S'il y a déjà des opérations en attente, en appuyant sur une touche d'opération de priorité inférieure, vous effectuez toutes les opérations précédemment en attente. Dans ce cas, vous devez utiliser la touche [CE/C] pour effacer le calcul et recommencer.

Touches d'opérations arithmétiques

Les touches d'opérations arithmétiques permettent d'effectuer les opérations mathématiques les plus courantes : addition, soustraction, multiplication, division, élévation d'un nombre à une puissance, extraction de la racine d'un nombre.

$[+]$, $[-]$, $[\times]$, $[\div]$, $[y^x]$, $[INV] [y^x]$ $[\Delta\%]$ – Fonctions arithmétiques

Toutes ces touches utilisent deux nombres. La procédure de base est la suivante :

- Introduisez une valeur.
- Appuyez sur une ou plusieurs touches d'opération.
- Introduisez une autre valeur.
- Achevez l'opération avec une opération de priorité égale ou inférieure.

La touche égale $[=]$ achève toutes les opérations en attente et prépare la calculatrice pour de nouveaux calculs. Dans la procédure décrite ci-dessus, l'introduction d'une valeur peut consister à rappeler la valeur d'une mémoire, à obtenir cette valeur par des calculs, ou simplement à introduire des chiffres à partir du clavier.

CHAPITRE II

LE MODE CALCULATRICE

[INV] [y^x] – Racine quelconque

La séquence de touches [INV] [y^x] calcule la racine xième de nombres positifs. Pour utiliser cette séquence, procédez comme suit :

1. Introduisez le nombre (y) dont vous cherchez la racine.
2. Appuyez sur la séquence [INV] [y^x].
3. Introduisez la valeur de (x).
4. Appuyez sur une touche qui achève l'opération.

Exemple

Calculez $3.871 \sqrt[3]{21.496}$

Procédure	Appuyer sur	Affichage
Remettre à zéro	[CE/C] [CE/C]	0
Introduire la valeur y	21.496 [INV] [y ^x]	21.496
Introduire la valeur x	3.871	3.871
Calculer $x\sqrt{y}$	[=]	2.208968514

Touches inverse, carré et racine carrée

Les fonctions inverse, carré et racine carrée sont fréquemment utilisées dans les calculs. Vous pouvez effectuer ces calculs en utilisant la touche puissance et racine quelconques, mais les touches inverse, carré et racine carrée réduisent le nombre de manipulations.

[1/x] – Inverse

La touche [1/x] élève la valeur affichée à la puissance - 1, ce qui équivaut à diviser 1 par le nombre. Ce nombre peut être une valeur quelconque dans les limites admises par la calculatrice.

[x²] – Fonction carré

La touche [x²] élève le nombre affiché à la puissance 2. Ce nombre peut être une valeur quelconque dont le carré se situe également dans les limites admises par la calculatrice.

CHAPITRE II

LE MODE CALCULATRICE

[√x] – Fonction racine carrée

La touche [√x] élève le nombre affiché à la puissance 1/2. Le nombre affiché doit être positif sinon, un message d'erreur apparaît. Le résultat est toujours positif.

Utilisation des mémoires

Vous pouvez utiliser une mémoire pour stocker une valeur que vous voulez comparer au résultat d'un prochain calcul ou utiliser plusieurs fois dans un calcul. Les mémoires sont numérotées de 0 à 9. Les valeurs statistiques occupent les mémoires 4 à 9 lorsque le mode STAT est sélectionné.

[STO] m – Stockage

La séquence de touches [STO] m remplace le nombre en mémoire m par le nombre affiché. Par exemple, la séquence de touches 22 [STO] 3 stocke le nombre 22 dans la mémoire 3.

Nota : Pour vider une mémoire, appuyez sur [STO] m lorsque la valeur affichée est 0.

[RCL] m – Rappel

La séquence de touches [RCL] m affiche (rappelle) le nombre stocké en mémoire m sans affecter le contenu de cette mémoire. La valeur précédemment affichée est effacée.

[EXC] m – Echange

La séquence de touches [EXC] m échange le nombre affiché contre le nombre contenu dans la mémoire m. Par exemple, la séquence 33 [EXC] 2 stocke le nombre 33 et affiche le nombre précédemment stocké en mémoire 2. Vous pouvez utiliser cette touche pour vérifier le contenu d'une mémoire sans perdre le nombre affiché.

[SUM] m – Addition en mémoire

La séquence de touches [SUM] m additionne le nombre affiché au nombre en mémoire m. Par exemple, si 5 est le nombre en mémoire 1 et si vous appuyez sur 3 [SUM] 1, le nombre en mémoire 1 devient 8. Le nombre 3 reste affiché.

[INV] [SUM] m – Soustraction de la mémoire

La séquence de touches [INV] [SUM] m soustrait le nombre affiché du nombre en mémoire m. Par exemple, si 5 est le nombre en mémoire 0 et si vous appuyez sur 3 [INV] [SUM] 0, le nombre en mémoire devient 2. Le nombre 3 reste affiché.

CHAPITRE II
LE MODE CALCULATRICE

Touches d'opérations logarithmiques

Un certain nombre de calculs techniques et théoriques requiert l'utilisation de logarithmes, qui représentent une partie importante de nombreux modèles mathématiques des phénomènes naturels. Les touches logarithmiques donnent accès aux logarithmes népériens et décimaux, ainsi qu'aux exponentielles.

[lnx], [log] – Logarithmes

La touche [lnx] calcule le logarithme népérien (base e, e = 2.718281828459) du nombre affiché. Le nombre doit être positif, sinon un message d'erreur apparaît.

La touche [log] calcule le logarithme décimal (base 10) du nombre affiché. Ce nombre doit être positif, sinon un message d'erreur apparaît.

[INV] [lnx], [INV] [log] – Exponentielles

La séquence de touches [INV] [lnx] calcule la valeur de e élevée à la puissance du nombre affiché.

La séquence de touches [INV] [log] calcule la valeur de 10 élevée à la puissance du nombre affiché.

Exemple
Trouvez le résultat de e(ln 2 x 10(log 3 + log 2)).

Table with 3 columns: Procédure, Appuyer sur, Affichage. It shows the steps to calculate e to the power of (ln 2 * 10 * (log 3 + log 2)), resulting in 64.

Fonctions hyperboliques

Les trois fonctions que les mathématiciens ont nommées hyperboliques à cause de leur similitude avec les fonctions trigonométriques sont le sinus hyperbolique, le cosinus hyperbolique et la tangente hyperbolique. Ces fonctions sont définies par rapport aux fonctions exponentielles ex. Les unités d'angle sélectionnées n'affectent en rien le résultat de ces fonctions.

CHAPITRE II
LE MODE CALCULATRICE

[hyp] [sin], [hyp] [cos], [hyp] [tan] – Fonctions hyperboliques

Les touches hyperboliques calculent la fonction hyperbolique du nombre affiché. [hyp] [sin] calcule le sinus hyperbolique [hyp] [cos] calcule le cosinus hyperbolique, et [hyp] [tan] calcule la tangente hyperbolique.

[INV] [hyp] [sin], [INV] [hyp] [cos], [INV] [hyp] [tan] – Fonctions hyperboliques inverses

Les touches hyperboliques inverses calculent les inverses des fonctions hyperboliques. [INV] [hyp] [sin] calcule l'arcsinus hyperbolique, [INV] [hyp] [cos] calcule l'arc-cosinus hyperbolique, et [INV] [hyp] [tan] calcule l'arc-tangente hyperbolique.

Nota : Vous devez appuyer en même temps sur les touches [INV] et [hyp] lorsque vous voulez accéder aux inverses des fonctions hyperboliques.

Exemple

Trouvez le résultat de sinus hyperbolique(3) et de l'arc-tangente hyperbolique(.5).

Table with 3 columns: Procédure, Appuyer sur, Affichage. It shows the steps to calculate sinh(3) and arsinh(.5), resulting in 10.01787493 and .5493061443 respectively.

Touches "entier" et "fraction"

Les touches "entier" et "fraction" éliminent la partie d'un nombre soit avant, soit après le point décimal. Elles permettent d'adapter une valeur aux fonctions qui exigent l'entrée de nombres entiers, ou d'isoler la partie décimale d'un nombre.

[Intg] – Partie entière

Cette touche tronque le nombre affiché avant le point décimal. Il en résulte un nombre entier.

CHAPITRE II LE MODE CALCULATRICE

Exemple

Trouvez l'exponentielle de 3 et donnez un nombre entier comme résultat.

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Introduction du problème	3 [INV] [log]	1000.
Affichage de la partie entière	[Intg]	999

Nota : En mode BASIC, la fonction INT(X) fonctionne différemment pour les valeurs négatives.

[Frac] – Partie décimale

Cette touche élimine les chiffres à gauche du point décimal du nombre affiché. Le résultat se situe entre -1 et 1.

Exemple

Introduisez le problème $10000 \div 7$ et éliminez la partie entière du résultat.

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Introduction du problème	10000 [÷] 7 [=]	1428.571429
Élimination de la partie entière	[Frac]	.5714285714

Les chiffres supplémentaires qui figurent dans le résultat sont les chiffres que la calculatrice utilise intérieurement pour assurer une plus grande précision.

Sélection des unités angulaires

La calculatrice traite une grande variété de calculs comprenant des angles : fonctions trigonométriques et conversions de coordonnées polaires/cartésiennes. Pour placer la calculatrice dans le mode angulaire désiré, vous pouvez sélectionner un de ces trois systèmes d'unités : degrés, radians ou grades. Vous pouvez également convertir des angles d'une unité à une autre en changeant de sélection.

CHAPITRE II LE MODE CALCULATRICE

[DRG] – Changement de mode angulaire

La touche [DRG] change les unités d'angle sans affecter l'angle affiché. Chaque fois que vous appuyez sur [DRG], les unités d'angle sont sélectionnées dans l'ordre suivant : radians (RAD est affiché), grades (GRAD est affiché), puis reviennent en degrés (DEG est affiché). En appuyant sur [INV] [DRG], vous sélectionnez les unités d'angle dans l'ordre inverse.

[DRG→] – Conversion d'angle

La touche [DRG→] change les unités d'angle et convertit l'angle affiché dans la nouvelle unité.

Par exemple, positionnez la calculatrice sur degrés et introduisez 90.

1. Appuyez une fois sur [DRG→] pour changer d'unité et passer en radians. L'angle est converti en 1.570796327 ($\pi/2$) radians.
2. Appuyez à nouveau sur [DRG→] pour changer d'unité et passer en grades. L'angle est converti en 100 grades.
3. Appuyez une troisième fois sur [DRG→] pour revenir en degrés. L'angle est de nouveau converti en 90° .

En appuyant sur [INV] [DRG→], vous changez les unités d'angle et convertissez dans l'ordre inverse.

Fonctions trigonométriques

Vous pouvez très facilement effectuer des calculs comprenant des sinus, cosinus, tangentes, et leurs inverses. Tous les angles étant interprétés dans le mode angulaire sélectionné, assurez-vous que la calculatrice est positionnée sur la bonne unité avant de commencer vos calculs.

[sin], [cos], [tan] – Fonctions trigonométriques

Les touches trigonométriques [sin], [cos] et [tan] calculent le sinus, le cosinus et la tangente de l'angle affiché. La calculatrice interprète l'angle dans les unités (DEG, RAD ou GRAD) sélectionnées par les touches [DRG] ou [DRG→].

CHAPITRE II LE MODE CALCULATRICE

[INV] [sin], [INV] [cos], [INV] [tan] – Inverse des fonctions trigonométriques

Les touches trigonométriques inverses calculent l'angle (dans les unités sélectionnées) dont les sinus, cosinus et tangente sont affichés. Les touches [INV] [sin] calculent l'arc-sinus (\sin^{-1}), les touches [INV] [COS] calculent l'arc-cosinus (\cos^{-1}) et les touches [INV] [tan] calculent l'arc-tangente (\tan^{-1}).

Les valeurs limites des angles obtenus à partir des inverses des fonctions trigonométriques sont données dans l'annexe C.

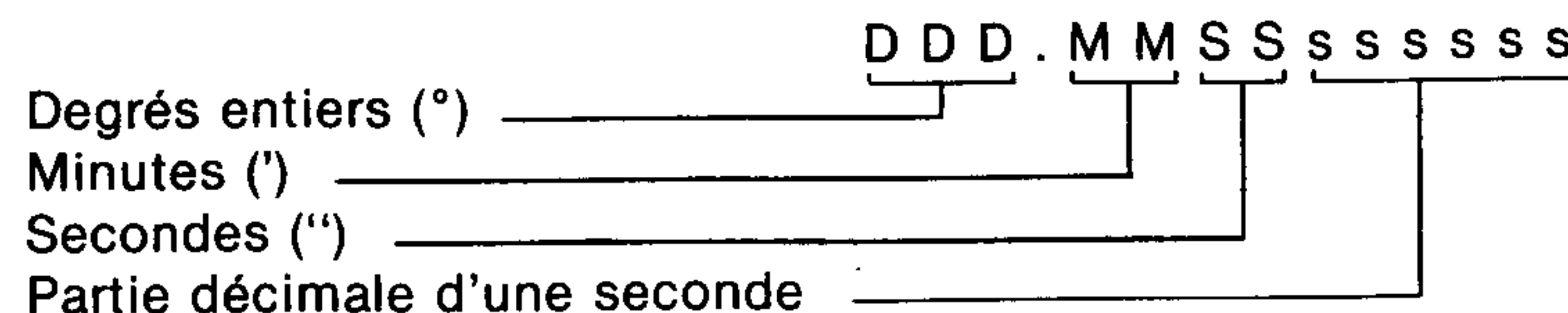
Conversions de deg/min/sec en degrés décimaux

En navigation et en astronomie, les angles sont souvent mesurés en degrés/minutes/secondes. Cependant, avant de pouvoir ajouter ces angles ou de les utiliser dans des calculs trigonométriques, vous devez les convertir en degrés décimaux. Vous pouvez également convertir les heures/minutes/secondes en heures décimales.

Introduction de degrés, minutes secondes

Pour introduire une valeur en degrés, minutes et secondes, procédez de la manière suivante :

- Introduisez à partir du clavier le nombre de degrés entiers.
- Introduisez un point décimal.
- Introduisez le nombre de minutes entières en deux chiffres.
- Introduisez le nombre de secondes entières en deux chiffres.
- Si vous devez entrer une partie décimale d'une seconde, introduisez ces chiffres comme si le point décimal pour les secondes était placé quatre chiffres à droite du point décimal des degrés.



CHAPITRE II LE MODE CALCULATRICE

Lorsque vous entrez les minutes et les secondes, n'oubliez pas d'introduire les zéros quand ils sont nécessaires, pour placer les chiffres en bonne position. (Notez que vous n'avez pas besoin d'entrer les zéros à la fin). Par exemple, l'angle $9^{\circ} 7' 50''$ est introduit sous la forme suivante : 9.075 dans le format degrés/minutes/secondes.

Pour entrer une valeur en degrés décimaux, introduisez-la à partir du clavier comme elle se présente.

[DMS→DD], [INV] [DMS→DD] – Conversion de minutes/secondes

La touche [DMS→DD] convertit les degrés/minutes/secondes en degrés décimaux.

La séquence de touches [INV] [DMS→DD] convertit des degrés décimaux en degrés/minutes/secondes. Le résultat inclut les symboles ° ' et '' pour identifier les degrés, minutes et secondes.

Nota : Bien que ces angles soient exprimés en degrés, vous n'avez pas besoin de commuter les unités d'angle sur le mode degrés lorsque vous utilisez les séquences de touches [DMS→DD] et [INV] [DMS→DD]. Vous pouvez effectuer ces conversions quel que soit le mode angulaire sélectionné.

Exemple

Convertissez $3^{\circ} 1' 30.456''$ en degrés décimaux, puis de nouveau en degrés/minutes/secondes.

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Conversion en degrés décimaux	3.0130456 [DMS→DD]	3.025126667
Conversion en deg/min/sec	[INV] [DMS→DD]	$3^{\circ} 1' 30.456''$

Les symboles ° ' '' disparaissent lorsque vous appuyez sur une nouvelle touche.

Calculs de pourcentages

Un pourcentage représente un certain nombre de centièmes. Par exemple, 50% correspond à 50/100, ce qui équivaut à 1/2. Les calculs de pourcentages sont très utiles dans la vie courante et professionnelle.

CHAPITRE II
LE MODE CALCULATRICE

[%] – Fonction pourcentage

La touche [%] divise automatiquement le nombre affiché par 100, le convertissant en un pourcentage décimal. Par exemple, si vous entrez 43.9 et appuyez sur la touche [%], le résultat affiché sera .439

Vous pouvez utiliser la touche [%] pour des calculs de pourcentages, de suppléments, de remises et de taux.

Notez que la valeur de base est le nombre affiché après que vous avez appuyé sur [x], [+], [–] ou [÷].

Exemple de pourcentage

4% de 453 = 18.12

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Calcul du pourcentage	453 [x] 4 [%] [=]	18.12

Exemple de supplément

1450 + 15% de supplément = 1667.5

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Calcul du supplément	1450 [+] 15 [%]	217.5
Calcul du résultat	[=]	1667.5

Exemple de remise

69.95 – 10% de remise = 62.955

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Calcul de la remise	69.95 [–] 10 [%]	6.995
Calcul du résultat	[=]	62.955

CHAPITRE II
LE MODE CALCULATRICE

Exemple de taux

Quel est le pourcentage de 25 représenté par 29,5?

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Calcul du résultat	29.5 [÷] 25 [%] [=]	118

[Δ%] – Fonction écart en pourcentage

Vous pouvez trouver l'écart en pourcentage entre deux valeurs en utilisant la touche [Δ%]. Le calcul s'effectue de la façon suivante :

$100 (y - x) \div x$

et vous donne la taxe réelle imputée sur les ventes ou l'amélioration en pourcentage consécutive à un réajustement. Entrez y [Δ%] x et appuyez sur une touche pour achever le calcul.

Exemple : pourcentage d'un supplément inconnu

Le coût de revient d'un achat de \$1450 est de \$1667.50. Quelle est la taxe réellement imputée? (Notez la similitude de cet exemple avec l'exemple de supplément cité plus haut).

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Calcul du résultat	1667.50 [Δ%] 1450 [=]	15

La taxe est de 15%.

Exemple : écart en % inconnu

Votre voiture consomme 8,8 litres aux 100 km avant révision et 8 litres après. Quel est le pourcentage d'amélioration?

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Calcul du résultat	8.8 [Δ%] 8 [=]	10

L'amélioration de la consommation est de 10%.

CHAPITRE II

LE MODE CALCULATRICE

Entrée de couples de données

Certaines fonctions de la calculatrice reposent sur l'introduction d'un couple de valeurs. Les deux touches $[(x,y)]$ et $[x\leftrightarrow y]$ permettent d'entrer des couples de données dans l'ordre ou dans l'ordre inverse. Reportez-vous aux rubriques "Conversions de coordonnées polaires/cartésiennes", "Factorielles, permutations et combinaisons" et "Touches de fonctions statistiques" pour comprendre comment la calculatrice utilise les couples de données.

$[(x,y)]$ – Introduction d'un couple de valeurs

La touche $[(x,y)]$ introduit la première des deux valeurs d'un couple. Si vous voulez utiliser le couple (2,6), entrez d'abord le chiffre 2 et appuyez sur la touche $[(x,y)]$. La zone de rappel de fonction affiche (x,y?) pour vous demander la deuxième valeur. Vous entrez ensuite le chiffre 6 en seconde valeur.

Nota : Vous pouvez introduire un couple de valeurs avec autant de chiffres que vous voulez, même si l'affichage ne suffit pas pour les deux nombres. Dans ce cas, l'indicateur !left! apparaît à gauche de l'affichage. Les deux nombres complets sont néanmoins pris en compte.

$[x\leftrightarrow y]$ – x remplace y

La touche $[x\leftrightarrow y]$ permet de permuter les valeurs de x et y lors de l'introduction du couple. Supposez que vous vouliez à nouveau utiliser le couple (2,6), mais le chiffre 6, résultant d'un précédent calcul, est déjà à l'affichage. Appuyez sur les touches $[(x,y)]$ 2 $[x\leftrightarrow y]$ et le couple (6,2) devient le couple (2,6).

Vous auriez également pu utiliser le chiffre 6 à l'affichage comme seconde partie du couple, en appuyant sur les touches $[x\leftrightarrow y]$ 2, qui introduisent le couple (2,6).

La touche $[x\leftrightarrow y]$ vous permet également de visualiser les deux parties d'un résultat couplé. Ce type de résultat s'utilise pour les conversions de coordonnées polaires en coordonnées cartésiennes.

Si vous introduisez un couple de valeurs sans utiliser de fonction nécessitant des couples de données, le premier nombre (x) du couple ne sera pas utilisé.

CHAPITRE II

LE MODE CALCULATRICE

Exemple

Entrez le couple (3,4) et ajoutez 5, puis vérifiez que la valeur 3 est toujours présente.

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Introduire le couple de valeurs	3 $[(x,y)]$ 4	3 , 4
Additionner 5	[+] 5 [=]	9
Vérifier l'autre partie du couple	$[x\leftrightarrow y]$	3

Conversions de coordonnées polaires/cartésiennes

Dans le système de coordonnées cartésiennes, les coordonnées (x,y) représentent la valeur d'un point x sur l'axe des x et la valeur d'un point y sur l'axe des y. Dans le système de coordonnées polaires, les coordonnées (r, θ) situent la valeur d'un point r à partir de l'origine et à un angle θ de la ligne de référence.

[P \rightarrow R] – Conversion de coordonnées polaires en coordonnées cartésiennes

Pour convertir les coordonnées polaires (r, θ) en coordonnées cartésiennes (x,y), procédez de la manière suivante :

1. Sélectionnez l'unité d'angle choisie (degrés, radians ou grades) en utilisant la touche [DRG].
2. Entrez les coordonnées polaires sous la forme r $[(x,y)]$ θ .
3. Appuyez sur [P \rightarrow R]. La coordonnée x s'affiche et x = précède le nombre .
4. Appuyez sur $[x\leftrightarrow y]$. La coordonnée y s'affiche et y = précède le nombre.

[INV] [P \rightarrow R] – Conversion de coordonnées cartésiennes en coordonnées polaires

Pour convertir les coordonnées cartésiennes (x,y) en coordonnées polaires (r, θ), procédez de la manière suivante :

1. Sélectionnez l'unité d'angle choisie (degrés, radians ou grades).

CHAPITRE II

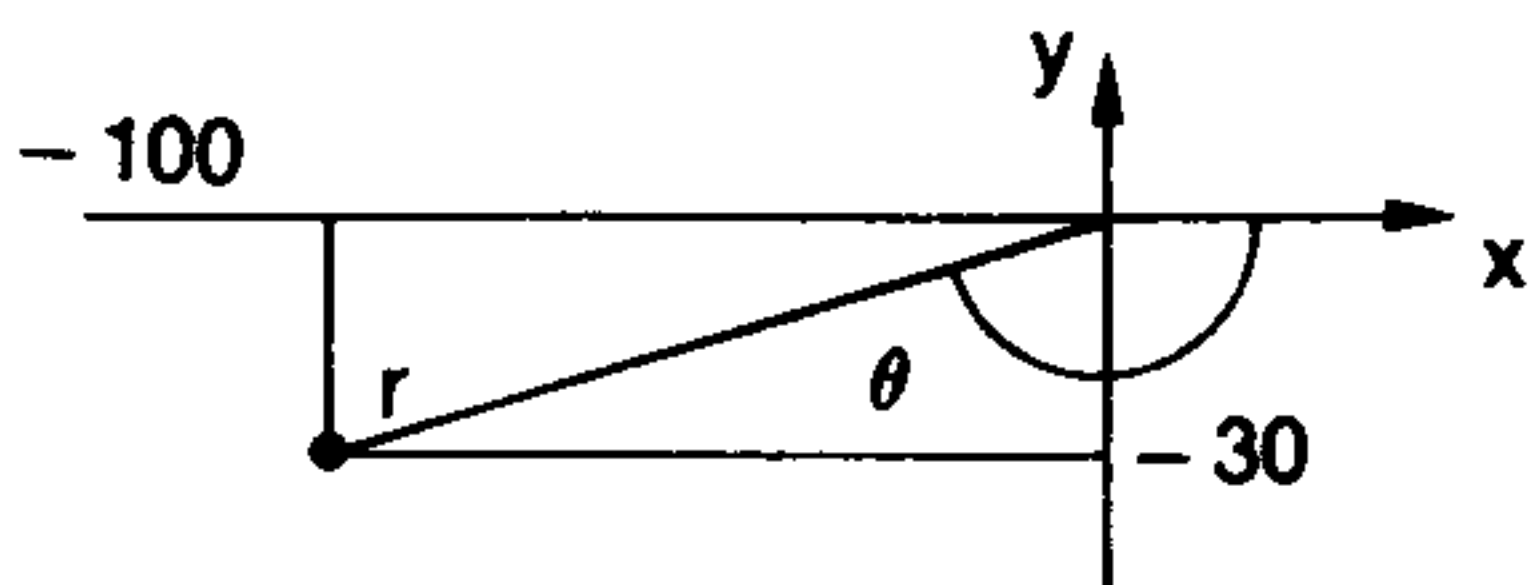
LE MODE CALCULATRICE

- 2. Entrez les coordonnées cartésiennes x [(x,y)] y.
- 3. Appuyez sur [INV] [P→R]. La coordonnée r s'affiche (dans l'unité d'angle sélectionnée) et r = précède le nombre.
- 4. Appuyez sur [x↔y]. La coordonnée θ s'affiche et θ = précède le nombre.

La séquence de touches [INV] [P→R] permet d'entrer des coordonnées cartésiennes avec des valeurs négatives pour x et y. Lorsque vous les convertissez en coordonnées polaires, la valeur de θ indique le quadrant correct. θ est compris entre -180° et +180°, -π et π radians, ou -200 et 200 grades.

Exemple

Convertissez les coordonnées cartésiennes (x = -100, y = -30) en coordonnées polaires.



Assurez-vous que la calculatrice est positionnée sur degrés avant de commencer l'opération.

Procédure	Appuyer sur	Affichage
Remettre à zéro	[CE/C] [CE/C]	0
Introduire les coordonnées x et y	100 [+/-] [(x,y)] 30 [+/-]	-100, -30
Convertir et afficher r	[INV] [P→R]	r=104.4030651
Afficher θ	[x↔y]	θ=-163.3007558

Les coordonnées cartésiennes (-100, -30) sont converties en coordonnées polaires (104.4030651, -163.3007558°).

Factorielles, permutations et combinaisons

Dans les calculs de probabilités, vous avez souvent besoin de multiplier une série de nombres entiers consécutifs ou de trouver le nombre d'arrangements possibles d'objets (permutations ou combinaisons). La touche [n!] (factorielle) vous permet de multi-

CHAPITRE II

LE MODE CALCULATRICE

plier une série de nombres entiers consécutifs ou de trouver le nombre d'arrangements possibles d'objets (permutations ou combinaisons). La touche [n!] (factorielle) vous permet de multiplier rapidement des nombres entiers consécutifs, et les touches [nPr] et [nCr] calculent les permutations et les combinaisons.

[n!] – Factorielle

La touche [n!] calcule la factorielle du nombre affiché. Le nombre affiché doit être un entier non-négatif, inférieur ou égal à 84, sinon un message d'erreur apparaît.

Nota : Un nombre dont la partie décimale est inférieure à 5×10^{-10} ressemble à un nombre entier à l'affichage. Donc, pour être sûr que l'argument de la valeur calculée est autorisé, appuyez sur [Intg] [n!] avant de trouver sa factorielle.

La factorielle de tout nombre entier (x) s'écrit x!, elle est égale au produit de tous les nombres entiers de 1 à x. Par définition, 0! est égal à 1.

[nPr] – Permutations

Pour trouver le nombre de permutations de n objets pris r à la fois, utilisez la touche [nPr]. Entrez n, appuyez sur [(x,y)], entrez r et appuyez sur [nPr].

$$nPr = n! \div (n - r)!$$

Exemple

Vous avez six grands éléments approximativement de la même largeur et vous voulez les installer sur un mur de votre cuisine, juste assez grand pour recevoir trois éléments seulement. Trouvez le nombre de possibilités d'arrangements de vos éléments sur ce mur :

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Calcul des permutations	6 [(x,y)] 3 [nPr]	120

[nCr] – Combinaisons

Pour trouver le nombre de combinaisons de n objets pris r à la fois, utilisez la touche [nCr]. Entrez n, appuyez sur [(x,y)], entrez r et appuyez sur [nCr].

$$nCr = n! \div (r!(n - r)!)$$

CHAPITRE II

LE MODE CALCULATRICE

Exemple

Une quinte flush royale est composée d'un as, d'un roi, d'une dame, d'un valet et d'un dix de la même couleur. Seules quatre mains sont possibles. Quelle est la probabilité de former une quinte flush royale à partir d'un jeu de 52 cartes?

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C]	0
Calcul de la probabilité	4 [÷] 52 [(x,y)] 5 [nCr] [=]	1.539077E-006

Les touches de fonctions statistiques

Les touches de fonctions statistiques sont d'une grande utilité pour compiler et analyser des ensembles de données. Après avoir introduit toutes les données dans les registres statistiques, vous pouvez analyser les tendances de ces données.

Ordre des opérations statistiques

Une analyse statistique s'effectue en deux étapes :

- Introduction des données.
- Utilisation de la touche préfixe [STAT] pour afficher les résultats.

[STAT] [CSR] – Effacement des registres statistiques

La séquence de touches [STAT] [CSR] permet d'effacer les registres statistiques. Vous devez toujours appuyer sur ces touches avant d'entrer de nouveaux ensembles de données, de même que lorsque vous avez terminé l'analyse des données.

[Σ +], [FRQ], [INV] [Σ +] – Introduction et retrait de données

L'utilisation des données à une variable ou à deux variables dépend du domaine d'application de ces données. Les notes scolaires sont un exemple de données à une variable. Les résultats expérimentaux d'un tracé sur un graphique à deux dimensions sont un exemple de données à deux variables.

Il est possible qu'une donnée apparaisse plusieurs fois. Il est donc préférable d'introduire les données entre parenthèses et de préciser leur fréquence à l'intérieur de ces parenthèses lorsque vous traitez un grand nombre de données. Pour s'adapter à ces situations, une option permet, lors de l'introduction des données, d'en répéter une.

CHAPITRE II

LE MODE CALCULATRICE

La séquence [Frq] "n fois" (où "n fois" correspond à un nombre n) permet de répéter n fois une entrée. Le nombre de répétitions doit être un nombre entier positif, introduit par le clavier numérique (il ne peut pas être rappelé à partir d'une mémoire). Vous ne pouvez pas introduire en même temps plus de 9,999,999,999,999 fois une donnée.

Chaque fois que vous introduisez ou retirez une donnée, l'affichage indique le nombre de données actuellement stockées dans les registres statistiques.

Introduction de données

Le tableau suivant énumère les séquences de touches servant à l'introduction des données dans les registres statistiques.

Type de données	Séquence de touches
Données à une variable, apparition unique de la donnée	x [Σ +]
Données à une variable, répétition de la donnée	x [Frq] n fois [Σ +]
Données à deux variables apparition unique de la donnée	x [(x,y)] y [Σ +]
Données à deux variables répétition de la donnée	x [(x,y)] y [Frq] n fois [Σ +]

Retrait de données

Le tableau suivant énumère les séquences de touches servant à retirer des données des registres statistiques. Si vous voulez retirer une donnée spécifique ou corriger une introduction de données, vous pouvez procéder à cette correction sans tout recommencer.

CHAPITRE II
LE MODE CALCULATRICE

Type de données	Séquence de touches
Données à une variable, apparition unique de la donnée	x [INV] [Σ +]
Données à une variable, répétition de la donnée	x [Frq] n fois [INV] [Σ +]
Données à deux variables apparition unique de la donnée	x [(x,y)] y [INV] [Σ +]
Données à deux variables répétition de la donnée	x [(x,y)] y [Frq] n fois [INV] [Σ +]

Nota : Le retrait d'une valeur de données qui n'a pas été introduite au préalable peut faire apparaître des résultats erronés.

Analyse des données

Après avoir introduit les données, vous pouvez les analyser dans les registres statistiques en effectuant l'un des calculs suivants.

Séquence de touches	Résultat
[STAT] [x]	moyenne de x
[STAT] [y]	moyenne de y
[STAT] [sx]	écart type de x
[STAT] [sy]	écart type de y
[STAT] [Σx]	somme des x
[STAT] [Σy]	somme des y
[STAT] [Σx²]	somme des carrés de x
[STAT] [Σy²]	somme des carrés de y
[STAT] [Σxy]	somme des produits xy
[STAT] [n]	nombre d'entrées de données
[STAT] [a]	ordonnée à l'origine de la droite de régression
[STAT] [b]	pente de la droite de régression
[STAT] [r]	coefficient de corrélation
y [STAT] [x']	nouvelle valeur de x pour une valeur désirée de y
x [STAT] [y']	nouvelle valeur de y pour une valeur désirée de x

CHAPITRE II
LE MODE CALCULATRICE

Calculs statistiques

Le présent chapitre explique comment effectuer des calculs statistiques. Notez que lorsque vous appuyez sur la touche [Σ +] pour introduire la première donnée, la calculatrice passe en mode "statistiques" et l'indication "STAT" apparaît à l'affichage.

Exemple : Données à une variable

Une machine conditionne des céréales pour petit déjeuner dans des emballages de 350 g. Pour vérifier la machine, vous pesez dix boîtes de céréales et vous trouvez les quantités suivantes :

355, 355, 364, 364, 361, 353, 353, 353, 350, 359

Vérifiez la moyenne et l'écart type de tout l'échantillon. Otez ensuite la plus grande et la plus petite valeur et recalculez la moyenne et l'écart type.

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C] [STAT] [CSR]	0
Introduction des quantités	355 [Frq] 2 [Σ +] 364 [Frq] 2 [Σ +] 361 [Σ +] 353 [Frq] 3 [Σ +] 350 [Σ +] 359 [Σ +]	10
Calcul de la moyenne	[STAT] [x]	356.7
Calcul de l'écart type	[STAT] [sx]	4.967673276
Retrait de la plus grande et de la plus petite valeur	364 [Frq] 2 [INV] [Σ +] 350 [INV] [Σ +]	7
Calcul de la moyenne	[STAT] [x]	355.5714286
Calcul de l'écart type	[STAT] [sx]	3.207134903
Effacement du mode "statistiques"	[STAT] [CSR]	3.207134903

CHAPITRE II
LE MODE CALCULATRICE

Régression linéaire

Dans beaucoup de domaines, les couples de données s'appliquent à un système linéaire. Un tracé de mesures du système devrait théoriquement suivre une droite. Cependant, les points ont généralement tendance à se situer quelque peu en dehors de cette droite. Les erreurs sont dues en partie à des imperfections du processus de mesure, et en partie aux propriétés non-idéales du système. La régression linéaire détermine la droite qui s'adapte le mieux aux points.

Nota : Les plus grandes et plus petites valeurs de données enregistrées définissent l'intervalle dans lequel vous pouvez utiliser la droite de régression. Même si le coefficient de corrélation (r) d'un ensemble de données se rapproche de 1 dans l'intervalle, vous ne pouvez pas être sûr que les données prises en dehors de cet intervalle auront la même relation linéaire.

Exemple

La force exercée par un ressort et sa longueur de compression et d'extension constituent un système linéaire. La force est un multiple constant de la déformation noté F = kx. Lors d'une expérience, les mesures suivantes ont été réalisées :

Table with 2 columns: Longueur du ressort (cm), Force (N). Rows: 27, 32, 23, 25, 31, 28.

Si une force est appliquée, quelle est la longueur du ressort? Quelle force faut-il pour lui donner une certaine longueur? Vous pouvez répondre à ces questions en utilisant différentes valeurs le long de la droite de régression.

Le ressort est en position repos lorsque la force appliquée est égale à zéro.

CHAPITRE II
LE MODE CALCULATRICE

Table with 3 columns: Procédure, Appuyer sur, Affichage. Rows: Remise à zéro, Entrée des données, Vérification de la corrélation, Détermination de k, Détermination de la longueur du ressort au repos.

Bien que la valeur désirée soit r = 1, le coefficient de corrélation .9697571933 indique que l'erreur n'est pas excessive.

Si vous deviez mesurer le ressort sans application d'une force, vous trouveriez une différence par rapport à la longueur calculée pour ce même ressort au repos. Cette différence est due en partie au fait que la longueur du ressort dans cette position n'entre pas dans l'intervalle des données mesurées, et en partie à l'incertitude relative d'une seule mesure.

La longueur calculée pour le ressort au repos reflète le comportement linéaire du ressort, déterminé à partir de toutes les mesures ; la longueur observée pour le ressort au repos ne représente qu'une seule mesure. Le but d'une analyse de régression est d'éviter de se fonder sur un petit nombre de mesures seulement et d'établir à la place une valeur représentative de plusieurs mesures.

Sommes statistiques

Les touches qui servent à calculer l'écart type déterminent l'écart type de l'échantillon généralement utilisé. Cependant, lorsque vous avez besoin de calculer l'écart type de la population, vous devez appliquer les équations suivantes :

sn_x = sqrt((n Σx^2 - (Σx)^2)/n^2)
sn_y = sqrt((n Σy^2 - (Σy)^2)/n^2)

CHAPITRE II
LE MODE CALCULATRICE

Exemple

Un étudiant a obtenu les notes suivantes (notation sur dix) :
8, 6, 9, 9, 10, 9, 10, 8

Calculez la moyenne et l'écart type, sachant que ces notes consti-
tuent la population totale, utilisez l'écart type de la population.

Procédure	Appuyer sur	Affichage
Remise à zéro	[CE/C] [CE/C] [STAT] [CSR]	0
Introduction des données	8 [Frq] 2 [Σ +] 9 [Frq] 3 [Σ +] 6 [Σ +] 10 [Frq] 2 [Σ +]	8
Calcul de la moyenne	[STAT] [x̄]	8.625
Calcul de l'écart type	[()][()][STAT] [n] [x] [STAT] [Σx²] [-] [STAT] [Σx] [x²] [)][÷] [STAT] [n] [x²] [)][√x]	1.218349293
Effacement du mode "statistiques"	[STAT] [CSR]	1.218349293

Définitions statistiques

Bien que la calculatrice dispose de 15 fonctions statistiques pour
les données introduites, six sont la base de toutes les autres. Ces
six fonctions seront donc définies en premier lieu.

Synthèse d'un ensemble de données

La calculatrice réduit un ensemble de données à six valeurs. Si
vous avez introduit un ensemble que vous serez susceptible d'uti-
liser ultérieurement, notez par écrit les valeurs stockées dans les
mémoires 4 à 9. Lorsque vous voudrez recréer cet ensemble de
données, stockez simplement les six valeurs dans les mémoires
correspondantes une fois le mode STAT sélectionné, au lieu
d'introduire à nouveau les données une par une.

CHAPITRE II
LE MODE CALCULATRICE

Mémoire	Contenu
4	Σx
5	Σy
6	n
7	Σx²
8	Σy²
9	Σxy

Possibilités d'impression

La touche d'impression envoie le nombre affiché et les symboles
de rappel de fonction à l'imprimante (vendue séparément), si elle
est raccordée. L'action sur la touche d'impression n'affecte pas
les calculs. Aucun symbole de rappel de fonction n'est prévu pour
la fonction impression. Le symbole de rappel de fonction de la
dernière opération exécutée est donc transmis à l'imprimante.

[Print] – Fonction impression

Lorsqu'une imprimante compatible est raccordée et prête à fonc-
tionner, votre calculatrice envoie un nombre à l'impression. Quand
l'affichage indique une valeur que vous voulez enregistrer,
appuyez sur la touche [PRINT] pour imprimer cette valeur. Les
symboles affichés dans la zone de rappel de fonction seront
imprimés en même temps. Si l'imprimante n'est pas raccordée,
cette touche est sans effet.

Pour les valeurs et les symboles nécessitant plus de 20 caractè-
res, la valeur sera imprimée en premier, et les symboles de rappel
de fonction apparaîtront sur la ligne suivante.

Reportez-vous au chapitre 6 pour de plus amples informations
concernant l'imprimante.

CHAPITRE III
LE MODE BASIC

CHAPITRE III LE MODE BASIC

Le mode BASIC

Le présent chapitre décrit l'affichage et le clavier BASIC ; il mentionne en outre de courts travaux pratiques afin de vous familiariser avec quelques opérations BASIC pouvant être effectuées avec la machine. Pour cela, sélectionnez d'abord le mode BASIC en appuyant sur la touche **[MODE]**.

L'ordinateur BASIC

Quand vous sélectionnez le mode BASIC, la machine fonctionne comme un ordinateur portatif avec langage de programmation BASIC résident.

Caractéristiques en mode BASIC

En tant qu'ordinateur BASIC, la machine présente les caractéristiques suivantes

- Clavier machine à écrire avec clavier numérique auxiliaire.
- Logement pour modules de logiciels et modules d'extension mémoire permanente RAM.
- Jeu de caractères ASCII (American Standard Code for Information Interchange) comportant les lettres majuscules et minuscules.
- Caractères spéciaux comprenant les caractères grecs, japonais et graphiques.
- 8K octets (environ 8000 caractères) de mémoire vive permanente (Constant Memory™).

Apprentissage de la programmation en BASIC

Le manuel qui accompagne l'ordinateur suppose que vous êtes déjà familiarisé avec le langage de programmation BASIC.

Utilisations du BASIC

Votre ordinateur étant en mode BASIC, vous pouvez :

- Effectuer le calcul direct d'expressions BASIC telles que $3^9 + 2$ et $\text{SIN}(X)$.
- Exécuter des commandes BASIC telles que LIST et RUN.
- Lancer des programmes BASIC. (Il peut s'agir de programmes que vous introduisez par le clavier ou de programmes BASIC accessibles par des modules enfichables et des cassettes).

CHAPITRE III LE MODE BASIC

- Conserver des programmes et des données en utilisant une unité de stockage telle que magnétophone à cassette.
- Stocker le contenu de la mémoire vive résidente sur un module à mémoire permanente.

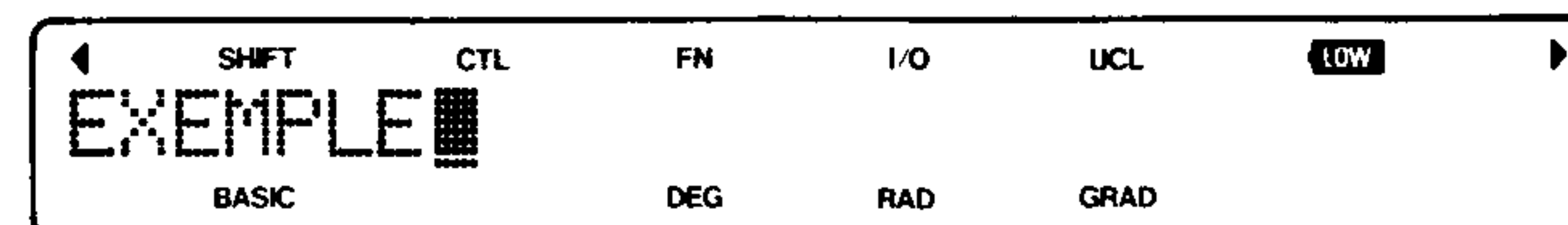
L'affichage BASIC

Lorsque l'ordinateur est en mode BASIC, l'affichage n'indique pas les mêmes informations qu'en mode calculatrice.

Types d'informations visualisées

En mode BASIC, trois types d'informations peuvent s'afficher :

- texte
- curseur
- indicateurs d'état.



Texte

Une ligne peut comporter jusqu'à 80 caractères dont 31 seulement apparaissent à l'affichage.

Curseur

Le curseur est affiché sous l'une des deux formes suivantes : un rectangle (comme représenté ci-dessus) ou un soulignement.

- Le rectangle clignote pour indiquer que vous pouvez introduire des données à l'affichage. Au fur et à mesure que vous tapez les caractères, ils s'affichent à l'emplacement du curseur et celui-ci se déplace d'une position vers la droite. Quand vous visualisez une ligne, vous pouvez en faire apparaître les parties non visibles en déplaçant le curseur vers la droite ou vers la gauche.
- Le soulignement peut apparaître en cours d'exécution d'un programme BASIC. Il signifie que l'ordinateur est

CHAPITRE III LE MODE BASIC

dans l'attente d'une commande de reprise du programme par pression sur **[ENTER]** ou sur **[CLR]**.

Indicateurs

Outre l'indicateur BASIC qui vous rappelle que l'ordinateur est en mode BASIC, il existe d'autres indicateurs qui vous informent de l'état de la machine.

Indicateurs Signification

SHIFT	Après avoir appuyé sur l'une des touches [SHIFT] , [FN] ou [CTL] , la prochaine touche que vous presserez produira l'action ou le caractère second qui y est associé, comme décrit dans les pages suivantes.
FN	
CTL	
DEG	Les unités de mesure angulaire sont affichées respectivement en degrés, radians, grades.
RAD	
GRAD	
I/O	Une opération d'échange d'informations avec une unité périphérique telle qu'une interface cassette est en cours.
UCL	Mode verrouillage en position majuscules.
LOW	Niveau des piles bas.
←	Texte disparu de l'affichage à gauche.
→	Texte disparu de l'affichage à droite.

Le clavier BASIC

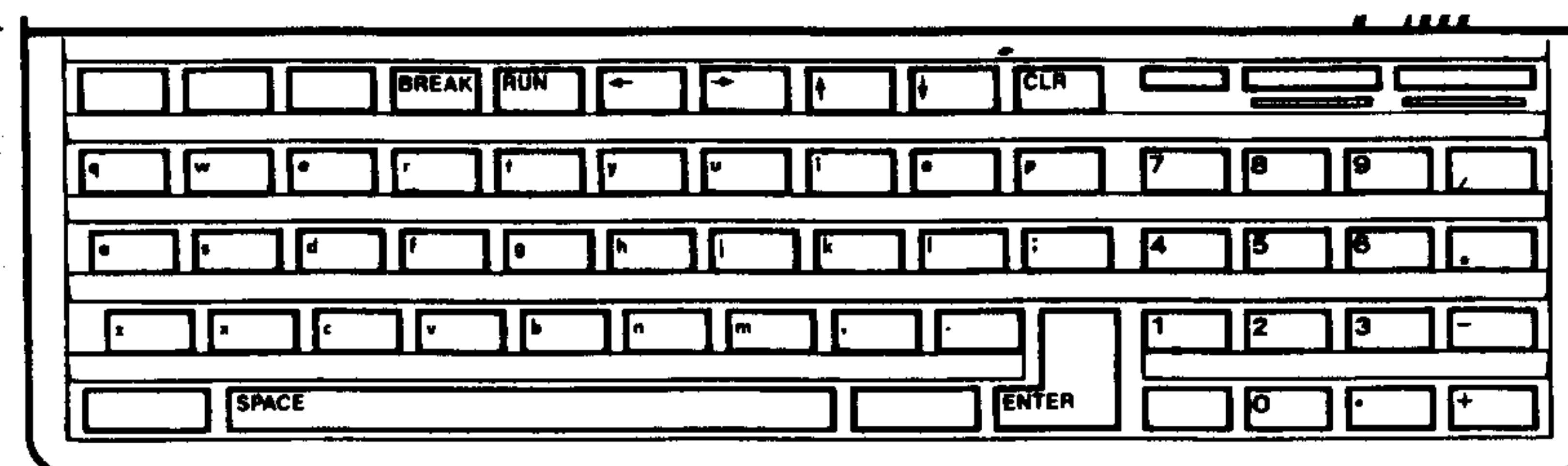
Lorsque l'ordinateur est en mode BASIC, ses touches n'ont pas la même affectation qu'en mode calculatrice. Le symbole imprimé dans l'angle supérieur gauche de chaque touche renvoie à l'affectation primaire de la touche en mode BASIC. **[SHIFT]**, **[FN]** et **[CTL]** sont des touches spéciales qui modifient le caractère ou l'action générée par d'autres touches.

Touches primaires

L'illustration ci-après montre le caractère ou l'action associée normalement à chaque touche. Le tableau d'accompagnement décrit

CHAPITRE III LE MODE BASIC

brièvement les touches qui produisent une action au lieu d'afficher un caractère.



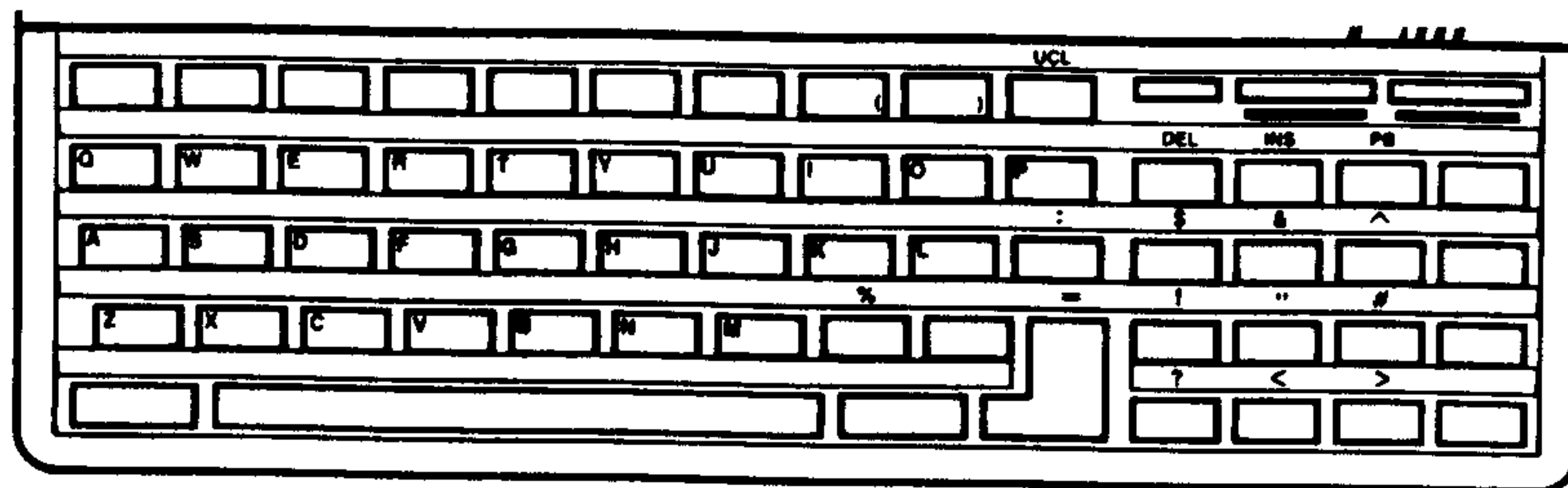
Touche	Action
[CLR]	Supprime des caractères de l'affichage
[BREAK]	Interrompt l'exécution d'un programme BASIC
[RUN]	Lance l'exécution d'un programme BASIC
[ENTER]	Commande à l'ordinateur d'accepter une ligne que vous avez tapée
[→]	Déplace le curseur vers la droite
[←]	Déplace le curseur vers la gauche
[↑]	Affiche la ligne précédente d'un programme
[↓]	Affiche la ligne suivante d'un programme

Utilisation des touches **[SHIFT]**

La touche **[SHIFT]** est utilisée en liaison avec d'autres touches. En pressant puis en relâchant l'une des deux touches **[SHIFT]**, vous provoquez l'affichage de l'indicateur SHIFT. Cela signifie que la prochaine touche sur laquelle vous appuierez produira le caractère ou l'action seconde qui y est associé. (Si vous pressez **[SHIFT]** par erreur, vous pouvez annuler l'indicateur SHIFT en réappuyant sur cette touche). Comme sur une machine à écrire, la touche **[SHIFT]** peut être verrouillée pendant que vous appuyez sur d'autres touches.

L'illustration ci-après montre les caractères et actions accessibles par la touche **[SHIFT]**. Le tableau d'accompagnement décrit brièvement les touches qui produisent une action au lieu d'afficher un caractère.

CHAPITRE III LE MODE BASIC



Touche	Action
[UCL]	Enclenche et annule le verrouillage en position majuscules
[DEL]	Supprime un caractère de l'affichage
[INS]	Permet l'insertion de caractères
[PB]	Remplace la ligne en cours d'affichage par la ligne précédemment affichée.

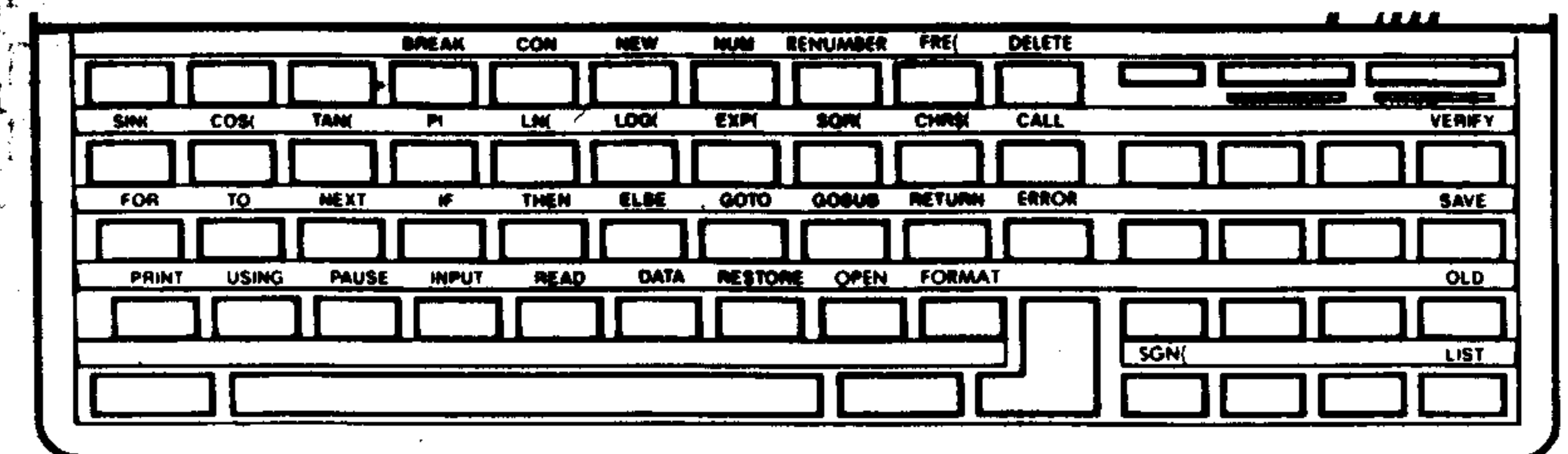
Les touches **[FN]** (Function) et **[CTL]** (Control) sont d'un emploi similaire à **[SHIFT]**, mais génèrent un ensemble différent de caractères ou d'actions seconds. Les touches utilisées avec FN entraînent l'apparition de chaînes de caractères spécifiques à l'affichage (telles que mots-clés BASIC d'usage fréquent). De même, **[CTL]** associé à des touches génère des caractères et des déplacements du curseur qui ne sont pas repérés sur le clavier.

Utilisation de la touche [FN]

[FN] est utilisé en liaison avec une autre touche pour faire apparaître une chaîne de caractères spécifiques (telle qu'un mot-clé BASIC) à l'affichage, sans l'introduire par le clavier. En pressant puis en relâchant la touche **[FN]**, vous affichez l'indicateur FN. Cela signifie que la prochaine touche sur laquelle vous appuierez provoquera l'affichage de la chaîne affectée à la touche. (Si vous pressez **[FN]** par erreur, vous pouvez annuler l'indicateur correspondant en appuyant sur la touche une deuxième fois). La touche **[FN]** peut être verrouillée pendant que vous appuyez sur d'autres touches. (Un paragraphe du présent chapitre décrit la procédure d'affectation de chaînes à d'autres touches).

CHAPITRE III LE MODE BASIC

L'illustration ci-après présente les mots-clés BASIC qui sont affectés en permanence à plusieurs touches du clavier. Le tableau d'accompagnement indique les mots-clés BASIC affectés en permanence mais non repérés sur le clavier.



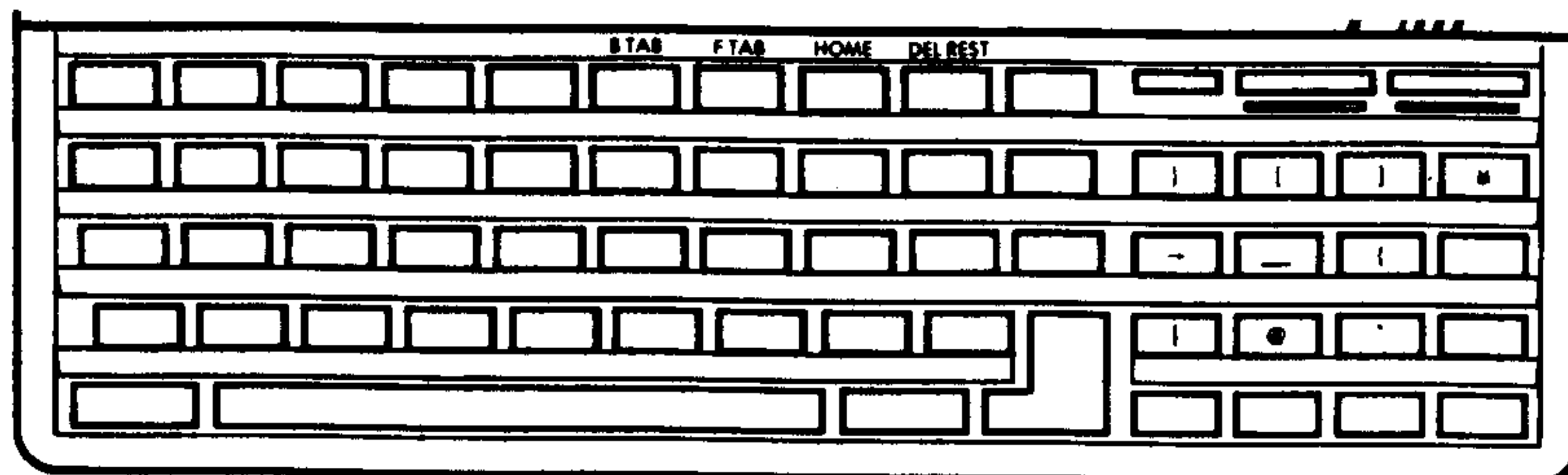
Touches	Mot-clé affecté
[FN] ;	ERROR (erreur)
[FN] .	FORMAT (format)
[FN] ,	OPEN (ouverture)
[FN] [↑]	RENUMBER (renumérotation)
[FN] [↓]	FRE((mémoire disponible)
[FN] [CLR]	DELETE (effacement)
[FN] [+/-]	SGN((signe)

Utilisation de la touche [CTL]

[CTL] est utilisé en liaison avec d'autres touches. Quand vous pressez puis relâchez la touche **[CTL]**, l'indicateur CTL s'affiche. Cela signifie que la prochaine touche sur laquelle vous appuierez générera le caractère ou l'action second qui y est associé. (Si vous pressez **[CTL]** par erreur, vous pouvez annuler l'indicateur correspondant en appuyant sur la touche une deuxième fois). La touche **[CTL]** peut être verrouillée pendant que vous utilisez d'autres touches.

L'illustration ci-après montre les caractères et actions que l'on peut générer avec la touche **[CTL]**. Le tableau d'accompagnement décrit brièvement les touches qui produisent une action au lieu d'afficher un caractère. (Les caractères et actions associés à **[CTL]** ne sont pas repérés sur le clavier).

CHAPITRE III LE MODE BASIC



Touches	Action
[CTL] [←]	Déplace le curseur jusqu'à l'arrêt de tabulation suivant
[CTL] [→]	Fait reculer le curseur jusqu'à l'arrêt de tabulation précédent
[CTL] [↑]	Ramène le curseur en début de ligne
[CTL] [↓]	Efface le texte à partir de la position du curseur

Majuscules

La touche **[UCL]** (**Verrouillage en position majuscules**) est utile lorsque vous voulez taper une série de lettres majuscules sans maintenir la touche **[SHIFT]** enfoncée. Dans certains cas, le BASIC transpose automatiquement les lettres d'une ligne de programme que vous introduisez, de minuscules en majuscules.

Utilisation de la touche [UCL]

Appuyez successivement sur **[SHIFT]** et **[UCL]** pour mettre le Verrouillage en position majuscules en fonction. L'indicateur UCL s'affiche pour rappeler que le mode Majuscules est en fonction.

Une fois terminée la frappe en majuscules, annulez UCL en appuyant de nouveau sur **[SHIFT] [UCL]**.

Passage automatique en majuscules

Certaines des lettres minuscules que vous tapez dans une ligne de programme BASIC sont automatiquement passées en majuscules. Il s'agit :

- de lettres de mots-clés BASIC telles que PRINT ou GOTO
- de noms de variables BASIC tels que X ou D\$.

CHAPITRE III LE MODE BASIC

Touches de défilement de ligne

Quand vous travaillez avec une ligne qui comporte plus de 31 caractères, vous pouvez éventuellement avoir besoin d'en visualiser une partie qui n'est pas visible à l'affichage. Vous avez alors la possibilité d'utiliser les touches de déplacement du curseur décrites dans le tableau ci-dessous, pour afficher n'importe quelle partie d'une ligne.

[→]

Déplace le curseur d'une position vers la droite. Lorsqu'il se trouve sur le côté droit de l'affichage, les caractères affichés sont déplacés vers la gauche par maintien de la pression sur **[→]**, jusqu'à ce qu'ils atteignent l'extrémité de la ligne de 80 caractères.

[←]

Déplace le curseur d'une position vers la gauche. (Si le curseur se trouve en début de ligne, **[←]** n'a aucune incidence).

[CTL] [→] (Tabulation avant)

Déplace le curseur jusqu'à l'arrêt de tabulation suivant. (Les arrêts de tabulation se trouvent aux colonnes 1, 25 et 50). La flèche à gauche apparaît alors pour indiquer que le texte a disparu de l'affichage à gauche.

[CTL] [←] (Tabulation arrière)

Fait reculer le curseur jusqu'à l'arrêt de tabulation précédent. (Les arrêts de tabulation se trouvent aux colonnes 1, 25 et 50).

[CTL] [↑] (début de ligne)

Ramène le curseur en début de ligne.

[SHIFT] [PB] (rappel)

Remplace la ligne en cours d'affichage par la ligne précédemment affichée. (Cette dernière ligne peut être une ligne que vous avez tapée ou une ligne qui était affichée par un programme BASIC).

Touches de modification de ligne

Vous pouvez éventuellement avoir besoin de modifier le contenu d'une ligne, soit pour corriger une erreur, soit pour obtenir un résultat différent. Pour ce faire, utilisez les touches d'édition-correction présentées ci-après.

CHAPITRE III LE MODE BASIC

[SHIFT] [INS]

Permet l'insertion de caractères. Au fur et à mesure que vous ajoutez un caractère, il s'insère à l'emplacement du curseur, et tous les caractères situés à droite de l'insertion sont décalés vers la droite. L'opération se poursuit avec chaque caractère que vous tapez, jusqu'à ce que vous appuyiez sur [ENTER], [CLEAR], [BREAK], [→] ou [←]. (En cas d'insertion faisant dépasser la limite de 80 caractères à la ligne, le dernier caractère est perdu).

[SHIFT] [DEL]

Supprime le caractère situé sous le curseur. Tous les caractères qui se trouvent à sa droite sont déplacés d'une position vers la gauche.

[CTL] [↓]

Efface le reste de la ligne à partir du caractère indiqué par le curseur.

[CLR]

Supprime des caractères de l'affichage.

- Quand aucun programme n'est en cours d'exécution, [CLR] supprime tous les caractères de l'affichage.
- Quand un programme attend une entrée au clavier, seuls les caractères que vous venez de taper sont effacés.

Touches de commande

Parmi les touches du clavier, quatre servent à commander l'exécution d'un programme BASIC, ainsi que le fonctionnement de l'ordinateur.

[ENTER]

Commande à l'ordinateur de traiter la ligne qui se trouve à l'affichage.

- Quand l'affichage comporte une commande BASIC, [ENTER] provoque son exécution.
- Quand l'affichage comporte une ligne de programme BASIC, [ENTER] stocke la ligne en mémoire.
- Quand un programme est dans l'attente de vos informations, [ENTER] provoque l'acceptation de l'information que vous tapez.

CHAPITRE III LE MODE BASIC

- Quand vous utilisez la commande LIST pour visualiser des lignes d'un programme, [ENTER] permet d'afficher la ligne suivante du programme.

[RUN]

Fait apparaître le mot RUN à l'affichage. Normalement, une pression sur [RUN] [ENTER] a pour effet de faire démarrer le programme en partant du plus petit numéro de ligne. ([RUN] peut également être associé à d'autres options comme il est décrit dans l'Index de référence BASIC).

[BREAK]

Interrompt un programme en cours. (En fonction des changements que vous apportez pendant l'interruption du programme, vous pouvez éventuellement poursuivre son exécution en utilisant la commande CONTINUE).

[RESET]

Réinitialise l'ordinateur. Cette touche est utilisée en "dernière extrémité", en cas de problème empêchant d'introduire des informations à partir du clavier. Elle affleure le boîtier de façon à ne pas pouvoir être actionnée involontairement. (Généralement, [RESET] fait apparaître le message W27 contents may be lost à l'affichage. Ce message est décrit en page 4).

Exécution de calculs en BASIC

Quand l'ordinateur est en mode BASIC et que le curseur rectangulaire clignote, vous pouvez, soit effectuer des calculs, taper des instructions devant être immédiatement exécutées, soit introduire une ligne que vous voulez stocker comme ligne de programme BASIC. Les calculs effectués en mode BASIC sont semblables à ceux du mode calculatrice.

Avantages relatifs

De même que le mode calculatrice, le mode BASIC vous permet de calculer le résultat d'une expression mathématique. Toutefois, chacun de ces modes présente des avantages relatifs par rapport à l'autre.

En mode BASIC, vous pouvez :

- Corriger des erreurs contenues dans une expression à plusieurs termes avant d'appuyer sur [ENTER] pour calculer le résultat.

CHAPITRE III

LE MODE BASIC

- Utiliser constantes et variables dans une même expression.

En mode calculatrice, vous pouvez :

- Vérifier des résultats intermédiaires tout en introduisant les termes de l'expression.
- Utiliser les légendes des touches qui sont les mêmes que celles d'une calculatrice classique.

Exemple d'exécution de calculs

La démonstration ci-après vous montre un exemple de procédure de calcul en mode BASIC.

1. Appuyez sur **[CLR]** pour effacer l'affichage.
2. Tapez l'expression suivante :
 $48 + 275 - 34$.
3. Appuyez sur **[ENTER]** pour calculer le résultat.
L'affichage indique 289 (48 plus 275 moins 34).
4. Tapez $-EXP(3)$ pour soustraire e^3 de la valeur affichée.
5. Appuyez sur **[ENTER]** pour afficher le nouveau résultat.
L'affichage indique 268.9144631 (289 moins e^3).

Exécution d'instructions en BASIC

Il est possible d'exécuter des instructions en BASIC en les tapant puis en appuyant sur la touche **[ENTER]**. Cette possibilité permet de vérifier rapidement l'effet produit par une instruction particulière avant de l'introduire dans un programme.

Exemple d'exécution d'instructions

Lorsque vous appuyez sur **[ENTER]** après avoir tapé une ligne constituée d'une instruction BASIC (ou de plusieurs séparées par deux points), l'ordinateur recherche d'éventuelles erreurs dans la ligne. S'il n'y en a pas, les instructions sont exécutées.

La démonstration ci-après vous montre un exemple d'exécution d'instructions en BASIC. (Cet exemple permet de comparer rapidement les inverses des nombres 20 à 25 à l'aide d'une boucle FOR/NEXT).

CHAPITRE III

LE MODE BASIC

1. Si nécessaire, appuyez sur **[CLR]** pour effacer l'affichage.
2. Tapez la ligne suivante :
`FOR X=20 TO 25:PRINT 1/X:NEXT X`
3. Appuyez sur **[ENTER]**.
L'affichage indique .05, inverse du nombre 20.
4. Appuyez une deuxième fois sur **[ENTER]**.
L'affichage indique .0476190476, inverse du nombre 21.
5. Si vous voulez achever la boucle FOR/NEXT, maintenez la touche **[ENTER]** enfoncée jusqu'à ce que tous les inverses soient affichés et que le curseur réapparaisse.
Si vous voulez interrompre l'exécution de la boucle FOR/NEXT, pressez **[BREAK]**. Appuyez ensuite sur **[CLR]** pour effacer l'affichage.

Exercice pratique : affectation d'un texte à une touche

Une particularité de l'ordinateur vous permet d'affecter une suite spécifique de caractères telle que nombres ou expressions fréquemment utilisés, à chacune des touches numériques 0 à 9. Après avoir procédé à cette opération, vous pouvez réafficher le texte mémorisé en pressant seulement deux touches.

Affectation de la chaîne de caractères

En affectant un texte à une touche, vous pouvez immédiatement afficher des informations telles que numéros de téléphone, expressions que vous utilisez fréquemment dans vos programmes, ou commandes qui ne sont pas affectées en permanence à des touches. Bien qu'il ne soit possible de réafficher la chaîne mémorisée qu'en mode BASIC, elle est cependant conservée lorsque vous éteignez l'ordinateur ou que vous sélectionnez le mode calculatrice.

Supposons que vous ayez mémorisé un programme comportant des numéros de ligne inférieurs à ceux du programme de démonstration. Pour travailler avec ce dernier programme, vous auriez à taper très souvent RUN 100 de façon à ce que l'exécution commence au numéro de ligne correct. Vous pouvez gagner du temps en affectant la chaîne RUN 100 à une touche.

CHAPITRE III LE MODE BASIC

Pour affecter RUN 100 à la touche numérique 6, procédez de la manière suivante :

1. Si nécessaire, appuyez sur **[CLR]** pour effacer l'affichage.
2. Tapez RUN 100 (ne pas appuyer sur **[ENTER]**).
3. Maintenez **[SHIFT]** enfoncé et pressez **[FN]**

Les deux indicateurs SHIFT et FN apparaissent à l'affichage.

4. Relâchez les touches **[SHIFT]** et **[FN]**.
5. Appuyez sur 6 pour affecter la chaîne affichée à la touche numérique 6.

L'affichage s'efface et les indicateurs SHIFT et FN s'éteignent.

Vous pouvez maintenant réafficher la commande RUN 100 chaque fois que vous désirez passer rapidement le programme de démonstration.

Réaffichage d'une chaîne mémorisée

Après avoir affecté un texte à une touche numérique, vous pouvez le réafficher à tout moment en appuyant sur **[FN]** puis sur la touche à laquelle vous l'avez attribué. La chaîne mémorisée apparaît alors à l'écran, à l'endroit où se trouve le curseur.

Si vous avez par exemple affecté une chaîne à la touche **[6]**, procédez comme suit :

1. Si nécessaire, effacez l'affichage en appuyant sur **[CLR]**.
2. Pressez **[FN] [6]**.

L'affichage indique RUN 100.

3. Pour exécuter le texte affiché comme une commande BASIC, appuyez sur **[ENTER]**.

L'ordinateur exécute le programme de démonstration en commençant à la ligne 100.

CHAPITRE IV INTRODUCTION GENERALE AU BASIC

CHAPITRE IV INTRODUCTION GENERALE AU BASIC

CHAPITRE IV INTRODUCTION GENERALE AU BASIC

Introduction générale au BASIC

Consultez ce chapitre si vous connaissez le type d'opération que vous voulez exécuter sans connaître le mot-clé correspondant. Après avoir localisé les mots-clés qui s'appliquent au type d'opération envisagé, reportez-vous au chapitre 5 pour la description détaillée de chaque mot-clé, illustrée par un exemple d'utilisation.

Introduction générale à la programmation

Le présent chapitre décrit brièvement quelques conventions et caractéristiques du BASIC de la série TI-70.

Numérotation des lignes

Chaque ligne de programme doit commencer par un numéro de ligne. Ce numéro doit être un nombre entier compris entre 1 et 32766. Les mots-clés suivants vous permettent de créer de nouveaux numéros de ligne et de modifier les numéros existants.

NUM Crée automatiquement des numéros de ligne quand vous introduisez un programme.

REN Renumérote les lignes d'un programme.

Duplication d'une ligne de programme

Vous pouvez dupliquer une ligne de programme en rappelant une ligne existante et en tapant un nouveau numéro de ligne à la place de l'ancien. Ceci permet de créer une nouvelle ligne sans affecter la ligne originale.

Lignes à instructions multiples

En général, vous pouvez introduire plus d'une instruction dans une ligne de programme. Ces instructions doivent être séparées par deux points.

```
100 X=3:PRINT X:PAUSE
```

Les lignes à instructions multiples sont soumises aux restrictions suivantes :

- Vous ne pouvez pas utiliser IMAGE ou DATA dans une ligne à instructions multiples.
- Si vous utilisez une instruction SUB dans une ligne à instructions multiples, cette instruction doit être la première de la ligne.

CHAPITRE IV INTRODUCTION GENERALE AU BASIC

- Si vous utilisez une remarque dans une ligne à instructions multiples, cette remarque doit être la dernière instruction de la ligne.
- Si vous utilisez une instruction DIM dans une ligne à instructions multiples, cette instruction doit être la dernière de la ligne. (Vous ne pouvez utiliser qu'une instruction DIM par ligne).

Remarques

Vous pouvez documenter vos programmes en utilisant des remarques. Une remarque représente un aide-mémoire pour le programmeur ; l'ordinateur n'en tient pas compte durant l'exécution d'un programme.

REM Signifie que le reste de la ligne de programme est une remarque.

! Peut remplacer le mot-clé REM.

Gestion de la mémoire

Le BASIC de la série TI-70 contient deux mots-clés pour vous aider à gérer l'espace-mémoire.

ADDMEM Ajoute la mémoire vive d'un module à la mémoire résidente de l'ordinateur. (Utilisez les mots-clés CALL ADDMEM, car il s'agit d'un sous-programme système).

FRE Indique la quantité totale de mémoire disponible pour le stockage de programmes et de données, ou bien signale la quantité de mémoire utilisée par le programme en cours.

Affectation de valeurs à des variables

Vous pouvez affecter des valeurs à des variables à partir des données contenues dans le programme ou introduites par le clavier durant l'exécution du programme.

Noms de variables

Pour que le nom d'une variable soit autorisé, il doit être conforme aux règles ci-dessous :

- Un nom de variable peut comporter 15 caractères au maximum.

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

- Le premier caractère doit être une lettre, un trait de soulignement ou le symbole @. Les autres caractères peuvent être des lettres, des nombres, des soulignements ou des symboles @.
- Le nom d'une variable-chaîne doit se terminer par le symbole \$. (Vous ne pouvez pas utiliser ce symbole \$ pour un nom de variable numérique).

Données contenues dans un programme

Les instructions suivantes vous permettent d'affecter des valeurs à des variables en utilisant les données contenues dans un programme.

READ	Affecte des valeurs à des variables en lisant par séquence les valeurs des données à partir des instructions DATA.
DATA	Fournit la liste des valeurs qui seront affectées aux variables par des instructions READ.
RESTORE	Détermine la prochaine instruction DATA qui sera lue.

Données introduites par le clavier

Les instructions suivantes suspendent l'exécution d'un programme pour vous permettre d'introduire des données à partir du clavier.

INPUT	Attend que vous introduisiez une ou plusieurs valeurs. INPUT peut afficher un message.
LINPUT	Attend que vous introduisiez une chaîne. LINPUT accepte votre entrée comme vous l'avez tapée, y compris avec les virgules et les guillemets. LINPUT peut afficher un message.
ACCEPT	Attend que vous introduisiez une valeur. ACCEPT comporte plusieurs options qui ne sont pas valables pour INPUT ou LINPUT.

Affichage d'informations

Les mots-clés de ce chapitre vous permettent d'afficher des informations.

Affichage non formaté

Les mots-clés suivants vous permettent d'afficher des valeurs numériques et des chaînes de caractères.

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

PRINT	Affiche une constante, la valeur d'une variable, ou le résultat d'une expression. (Vous utilisez normalement une PAUSE à la suite d'une instruction PRINT).
DISPLAY	Similaire à PRINT, excepté que DISPLAY comporte plusieurs options qui le rendent plus polyvalent. (Vous utilisez normalement une PAUSE à la suite d'une instruction DISPLAY).
PAUSE	Suspend l'exécution d'un programme pour une durée déterminée, ce qui vous permet de visualiser les informations à l'affichage. Sans PAUSE, les résultats s'affichent trop rapidement pour être visualisés.

INPUT et LINPUT vous permettent également d'afficher un message ou des informations relatives à l'entrée souhaitée.

Affichage formaté

Vous pouvez utiliser plusieurs options avec les instructions PRINT et DISPLAY pour mieux contrôler la présentation des informations à l'affichage.

USING	Définit un format de chaîne pour l'information affichée, ou cite en référence un format de chaîne contenu dans une instruction IMAGE séparée.
IMAGE	Place un format de chaîne sur une ligne différente, pour pouvoir la citer en référence dans plusieurs instructions PRINT USING ou DISPLAY USING
TAB	Positionne l'information dans une colonne déterminée de l'affichage.

Opérations numériques

Le BASIC de la série TI-70 comprend une grande variété d'opérateurs et de fonctions arithmétiques.

Opérateurs arithmétiques

Dans une expression numérique, vous pouvez utiliser les opérateurs suivants.

A + B	Additionne A et B.
A - B	Soustrait B de A.
A * B	Multiplie A par B.
A / B	Divise A par B.
A ^ B	Elève A à la puissance B.

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

+ et - sont également utilisés pour indiquer des valeurs positives et négatives, telles que +4 ou -6.

L'ordinateur contient aussi des opérateurs de relation et des opérateurs logiques qui permettent de comparer deux expressions numériques. Pour avoir plus de détails concernant ces opérateurs, reportez-vous à la page 70.

Hiérarchie des opérateurs

La liste ci-après vous indique l'ordre à respecter lors de l'exécution des opérations arithmétiques.

signe +, signe -

^

*, /

+, -

Une expression comprise entre parenthèses est prioritaire par rapport aux opérations en dehors des parenthèses.

Fonctions trigonométriques

Les mots-clés suivants sélectionnent un mode angulaire pour les calculs trigonométriques. Tous les angles sont interprétés selon le mode angulaire choisi.

DEG Sélectionne le mode degrés

RAD Sélectionne le mode radians.

GRAD Sélectionne le mode grades.

Avant de commencer des calculs trigonométriques, sélectionnez toujours le mode angulaire approprié. Vous pouvez alors utiliser les fonctions trigonométriques suivantes.

ACOS Calcule l'arc-cosinus d'un nombre.

ASIN Calcule l'arc-sinus d'un nombre.

ATN Calcule l'arc-tangente d'un nombre.

COS Calcule le cosinus d'un angle.

SIN Calcule le sinus d'un angle.

TAN Calcule la tangente d'un angle.

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

Fonctions hyperboliques

Le BASIC de la série TI-70 comprend les fonctions hyperboliques suivantes.

ACOSH Calcule l'arc-cosinus hyperbolique.

ASINH Calcule l'arc-sinus hyperbolique

ATANH Calcule l'arc-tangente hyperbolique.

COSH Calcule le cosinus hyperbolique.

SINH Calcule le sinus hyperbolique.

TANH Calcule la tangente hyperbolique.

Autres fonctions numériques

Les autres fonctions numériques disponibles dans le BASIC de la série TI-70 sont mentionnées ci-dessous.

ABS Calcule la valeur absolue.

EXP Calcule e élevé à une puissance.

INT Donne la partie entière d'un nombre.

LN Calcule le logarithme népérien.

LOG Calcule le logarithme décimal.

PI Donne la valeur de pi égale à 3.141592654.

RANDOMIZE Permet d'initialiser le générateur de nombres aléatoires, en donnant une valeur de départ pour qu'un nombre aléatoire reste imprévisible.

RND Donne un nombre aléatoire.

SGN Teste un nombre pour savoir s'il est positif, négatif ou nul.

SQR Calcule la racine carrée d'un nombre.

Opérations sur chaînes de caractères

Les manipulations de chaînes vous permettent d'inclure des messages et des informations dans vos programmes et d'écrire des programmes qui traitent des expressions-chaîne.

Constantes-chaîne

Lorsque vous introduisez une constante-chaîne à partir du clavier, mettez-la entre guillemets. Ces guillemets identifient la chaîne

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

mais n'en font pas partie. Si vous voulez introduire des guillemets à l'intérieur d'une chaîne, mettez deux paires consécutives de guillemets.

Frappe

"Essai avec guillemets"
"Essai""avec guillemets"
""Essai""""avec guillemets""""

Affichage

Essai avec guillemets
Essai""avec guillemets
""Essai""""avec guillemets""

Pour introduire une chaîne vide, ouvrez et fermez des guillemets ne contenant aucun caractère à l'intérieur.

Concaténation de chaînes

L'opérateur de concaténation (&) combine deux chaînes en une seule. La chaîne résultante ne doit pas comporter plus de 255 caractères.

L'extrait du programme suivant vous montre un exemple de concaténation de chaînes. (Notez que le premier caractère de ligne CH2\$ est un espace, assurant un espacement correct de la chaîne).

```
CH1$="CECI EST UNE"  
CH2$=" CHAINE CONCATENEE"  
CH3$=CH1$ & CH2$  
PRINT CH3$
```

Le segment de programme affiche :

```
CECI EST UNE CHAINE CONCATENEE
```

L'ordinateur contient également des opérateurs de relation et des opérateurs logiques qui permettent de comparer deux expressions-chaîne. Pour avoir plus de détails concernant ces opérateurs, reportez-vous à la page 70.

Fonctions-chaîne

Les fonctions-chaîne suivantes traitent des expressions-chaînes et les convertissent en valeur numérique.

ASC	Convertit le premier caractère d'une chaîne en son code ASCII correspondant.
NUMERIC	Teste une chaîne pour déterminer s'il s'agit d'une représentation autorisée d'une valeur numérique.
VAL	Donne la valeur numérique d'une expression-chaîne.

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

LEN	Calcule la longueur d'une chaîne.
POS	Recherche la première occurrence d'une sous-chaîne dans une chaîne.
Les fonctions-chaîne suivantes traitent des expressions numériques et les convertissent en chaînes.	
CHR\$	Convertit un caractère en code ASCII en son caractère d'affichage correspondant.
STR\$	Convertit un nombre en chaîne.
Les fonctions-chaîne suivantes traitent des expressions-chaîne et les convertissent en chaîne.	
RPT\$	Crée un chaîne en répétant une chaîne de départ un certain nombre de fois.
SEG\$	Donne une sous-chaîne d'une chaîne.

Tableaux

Le BASIC de la série TI-70 permet de manipuler des tableaux à une, deux ou trois dimensions.

Détermination de la taille d'un tableau

Un tableau exige de la place en mémoire. Vous pouvez définir un tableau à l'aide de l'instruction DIM ; vous avez également la possibilité d'accepter la valeur définie par défaut par l'ordinateur.

DIM Définit le nombre de dimensions et le nombre d'éléments dans chaque dimension d'un tableau.

Si vous désignez un tableau qui n'a pas été défini par une instruction DIM, l'ordinateur crée automatiquement un tableau à 11 éléments (numérotés de 0 à 10) pour chaque dimension.

Par exemple, si vous utilisez l'instruction suivante sans dimensionner au préalable le tableau NOMDETABLEAU, le tableau aura deux dimensions par défaut, chacune pouvant contenir onze éléments.

```
NOMDETABLEAU(0,0)=123
```

Si votre programme fait référence à un élément de tableau non compris dans les limites définies pour le tableau, l'ordinateur renvoie un message d'erreur. Si vous introduisez un nombre non-entier comme indice, l'ordinateur arrondit ce nombre. Si vous

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

entrez un entier négatif comme indice, l'ordinateur renvoie un message d'erreur.

Opérateurs de relation et opérateurs logiques

Une expression de relation compare deux valeurs numériques ou deux chaînes, puis vérifie si la relation déterminée est vraie ou fausse. Une expression logique relie deux expressions de relation.

Opérateurs de relation

Le tableau suivant vous montre des exemples utilisant les six opérateurs de relation. Dans chaque exemple, l'opérateur compare les valeurs de A et B, A et B représentant soit une expression numérique, soit une expression-chaîne.

$A < B$	Vrai si A est inférieur à B.
$A \leq B$	Vrai si A est inférieur ou égal à B.
$A > B$	Vrai si A est supérieur à B.
$A \geq B$	Vrai si A est supérieur ou égal à B.
$A = B$	Vrai si A est égal à B.
$A < > B$	Vrai si A est différent de B.

Nota : Dans la plupart des programmes, la seule chose importante est que la condition soit vraie ou fausse. Par ailleurs, l'ordinateur renvoie la valeur -1 si la condition est vraie, et 0 si la condition est fausse ; certains programmes peuvent tirer parti de cette caractéristique.

Comparaisons de chaînes

Lorsqu'une expression de relation compare deux chaînes, l'ordinateur prend un à un les caractères des deux chaînes et compare leur code ASCII. Les blancs en début et en fin de chaîne sont pris en considération. (Voir la liste complète des codes ASCII en annexe E).

- Si les codes ASCII diffèrent, la chaîne ayant le code le plus bas est inférieure à celle ayant le code le plus élevé.
- Si tous les codes ASCII sont identiques, les chaînes sont égales.
- Si une des chaînes est plus longue, la comparaison s'effectue sur autant de caractères que ce qu'en comporte la chaîne la plus courte. Si tous les codes ASCII sont

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

identiques, la chaîne la plus longue est considérée comme étant la plus grande.

- Une chaîne vide ("") est inférieure à toute autre chaîne.

Opérateurs logiques

Le tableau suivant vous montre des exemples utilisant les quatre opérateurs logiques.

A AND B (et)	Vrai si et seulement si A et B sont vrais.
A OR B (ou)	Vrai si soit A soit B est vrai, ou si A et B sont vrais.
A XOR B (ou exclusif)	Vrai si soit A soit B est vrai, mais faux si A et B sont vrais.
NOT A (non)	Vrai si et seulement si A est faux.

Les opérateurs logiques peuvent également servir à manipuler des données numériques. Reportez-vous à l'annexe F pour plus d'informations.

Instructions de contrôle

Les instructions de contrôle servent à modifier l'ordre d'exécution des instructions d'un programme. Les instructions de contrôle permettent des renvois conditionnels ou inconditionnels, ou bien des boucles.

Renvoi inconditionnel

Une instruction de renvoi inconditionnel transfère toujours le contrôle à un numéro de ligne déterminé.

GOTO	Transfère le contrôle à un et un seul numéro de ligne.
------	--

Renvoi conditionnel

Une instruction de renvoi conditionnel permet à un programme de sélectionner une des différentes voies possibles, selon certaines conditions incluses dans le programme.

IF/THEN/ELSE	Utilise des opérateurs de relation et des opérateurs logiques pour tester une condition dans un programme, et détermine l'instruction à exécuter.
--------------	---

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

ON GOTO Transfère le contrôle à un des différents numéros de ligne, selon la valeur de l'expression numérique.

ON GOSUB Transfère le contrôle à une des différentes sous-routines, selon la valeur de l'expression numérique.

Boucle

Une boucle FOR/NEXT répète les instructions dans la boucle un certain nombre de fois. Les mots-clés suivants vous permettent de définir une boucle FOR/NEXT.

FOR/TO/STEP Repère le début d'une boucle FOR/NEXT.

NEXT Repère la fin d'une boucle FOR/NEXT. Lorsque l'instruction NEXT est exécutée, le contrôle est renvoyé à l'instruction qui suit immédiatement l'instruction FOR/TO/STEP.

Sous-routines et sous-programmes

Lorsque votre programme se complique, il peut être nécessaire d'utiliser le même groupe de lignes en plusieurs endroits. En écrivant ces lignes sous forme de sous-routine ou de sous-programme, vous supprimez la nécessité de les reproduire à chaque endroit. Un programme peut accéder à une sous-routine ou à un sous-programme à partir de n'importe quelle ligne.

Utilisation d'une sous-routine

Les mots-clés suivants vous permettent d'utiliser des sous-routines.

GOSUB Transfère le contrôle d'un programme principal à une sous-routine, à un numéro de ligne déterminé.

ON GOSUB Transfère le contrôle à une parmi plusieurs sous-routines, suivant la valeur de l'expression numérique.

RETURN Repère la fin de la sous-routine et renvoie le contrôle à l'instruction qui succède à une instruction GOSUB ou ON GOSUB.

Il est courant de placer des sous-routines après la principale séquence d'un programme. Cependant, il n'est pas possible d'en placer après une instruction END.

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

Utilisation d'un sous-programme

Les mots-clés suivants vous permettent d'utiliser des sous-programmes.

CALL Transfère le contrôle d'un-programme principal à un nom de sous-programme déterminé.

SUB Annonce le début d'un sous-programme.

SUBEND Marque la fin d'un sous-programme.

SUBEXIT Termine l'exécution d'un sous-programme. Cette instruction est utilisée quand vous voulez sortir du sous-programme avant l'instruction SUBEND.

Un sous-programme doit être placé après la dernière instruction d'un programme principal.

Sous-programmes système

L'ordinateur met à votre disposition six sous-programmes système : ADDMEM, ERR, GET, IO, KEY et PUT. Pour accéder à ces sous-programmes système, n'oubliez pas d'utiliser l'instruction CALL. Par exemple, utilisez CALL ADDMEM pour accéder au sous-programme ADDMEM.

Traitement des erreurs

L'ordinateur possède différentes instructions pour vous aider à localiser des erreurs dans un programme. Certaines instructions vous aident à mettre au point un programme, et d'autres vous permettent de l'interrompre lorsqu'une erreur apparaît durant l'exécution du programme.

Mise au point d'un programme

Lors de la mise au point d'un programme, vous pouvez utiliser les mots-clés suivants pour interrompre son exécution et tester ainsi la valeur des variables du programme.

BREAK Crée des points d'arrêt à l'intérieur du programme.

Nota : Vous pouvez également appuyer sur la touche **[BREAK]** pour interrompre l'exécution du programme.

CON Reprend l'exécution du programme après une interruption.

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

ON BREAK	Détermine le comportement du programme en cas d'interruption.
UNBREAK	Supprime les points d'arrêt créés par l'instruction BREAK.

Traitement des erreurs et des avertissements

Grâce au traitement des erreurs et des avertissements, un programme peut souvent résoudre des problèmes, avant qu'ils n'affectent son exécution.

ON ERROR	Détermine le comportement du programme en cas d'erreur à l'exécution.
ON WARNING	Détermine le comportement du programme en cas d'avertissement à l'exécution.

Stockage et récupération de programmes

Vous pouvez sauvegarder votre programme pour une utilisation future, en le stockant sur une unité externe telle que magnétophone à cassette ou module de mémoire vive. Une fois stocké, le programme peut facilement être récupéré à n'importe quel moment pour être rechargé dans l'ordinateur. (Reportez-vous au chapitre 6 pour plus de détails concernant l'utilisation des cassettes).

Stockage d'un programme

Les mots-clés suivants vous permettent de stocker un programme chargé dans la mémoire de l'ordinateur.

FORMAT	Initialise le support physique de stockage monté sur une unité externe. (Les cassettes n'ont pas besoin d'être formatées). <i>Attention</i> : Si un support contenant déjà des programmes est reformaté, ces programmes sont effacés.
SAVE	Fait une copie du programme sur une unité de stockage externe telle que magnétophone à cassette.
VERIFY	Contrôle la copie stockée sur l'unité externe pour vérifier qu'elle est conforme à l'original.

CHAPITRE IV

INTRODUCTION GENERALE AU BASIC

PUT	Fait une copie du programme sur un module de mémoire vive. (Comme il s'agit d'un sous-programme système, utilisez l'instruction CALL PUT).
-----	--

Récupération d'un programme

Les mots-clés suivants permettent de récupérer la copie d'un programme stockée sur une unité externe.

Attention : Lorsque vous récupérez un programme stocké sur une unité externe, tout programme chargé est alors effacé. Si vous désirez sauvegarder le programme en cours, assurez-vous que vous l'avez stocké avant de récupérer un nouveau programme.

OLD	Récupère un programme stocké sur une unité externe telle que magnétophone à cassette.
GET	Récupère un programme stocké dans un module de mémoire vive. (Comme il s'agit d'un sous-programme système, utilisez l'instruction CALL GET).

Stockage et récupération de fichiers de données

- Lorsqu'un programme traite une grande quantité de données, il est généralement plus facile de stocker ces données sur une unité externe telle que magnétophone à cassette.

Dénomination d'un fichier de données

La validation d'un nom de fichier ouvert sur une unité externe dépend de la conception du périphérique. Pour les fichiers sur cassette, le nom du fichier peut comporter 1 à 18 caractères. Ce nom doit commencer par une lettre, mais le reste du nom peut être constitué de n'importe quel caractère, à l'exception d'une virgule ou d'un point. Les noms de fichiers sur cassette peuvent inclure un suffixe .NM pour supprimer l'affichage des messages.

Utilisation de fichiers de données

Les mots-clés suivants vous permettent de stocker et de récupérer des enregistrements de données. Vous pouvez aussi effacer des fichiers de données.

CHAPITRE IV INTRODUCTION GENERALE AU BASIC

FORMAT	Initialise le support physique de stockage monté sur une unité externe. (Les cassettes n'ont pas besoin d'être formatées). <i>Attention :</i> Si vous reformatez un support contenant déjà des programmes, ces programmes sont effacés.
OPEN #	Etablit un lien de communication entre l'ordinateur et l'unité externe.
CLOSE #	Coupe le lien de communication entre l'ordinateur et l'unité externe.
PRINT #	Stocke un enregistrement dans un fichier de données ouvert.
INPUT #	Récupère un enregistrement stocké d'un fichier de données ouvert.
EOF	Teste le fichier de données pour vérifier s'il reste des enregistrements dans le fichier.
RESTORE	Sélectionne le prochain enregistrement de données à introduire. (Vous ne pouvez pas utiliser l'instruction RESTORE pour un magnétophone à cassette).
DELETE	Efface un fichier de données spécifique stocké sur une unité externe.

Envoi d'informations à une imprimante

En reliant une imprimante (équipement optionnel) à l'ordinateur, vous pouvez imprimer la liste d'un programme chargé en mémoire. Vous pouvez également inclure des instructions dans votre programme afin d'envoyer des informations à une imprimante.

Listage d'un programme

Vous pouvez lister un programme avec un seul mot-clé. Vous n'avez ni à établir (OPEN) ni à couper (CLOSE) le lien de communication avec l'imprimante pour le listage.

LIST <i>nom-de-périphérique</i>	Imprime la liste d'un programme ligne par ligne. LIST doit être utilisé comme une commande, et ne peut pas figurer à l'intérieur d'un programme.
---------------------------------	--

CHAPITRE IV INTRODUCTION GENERALE AU BASIC

Utilisation d'une imprimante à partir d'un programme

Les mots-clés suivants peuvent être utilisés comme instructions à l'intérieur d'un programme.

OPEN #	Etablit un lien de communication entre l'ordinateur et l'imprimante.
CLOSE #	Coupe le lien de communication entre l'ordinateur et l'imprimante.
PRINT #	Envoie un nombre ou une chaîne à l'imprimante.
IO	Exécute des opérations de contrôle supplémentaires qui ne sont pas intégrées au BASIC de la série TI-70. (Comme il s'agit d'un sous-programme système, utilisez l'instruction CALL IO.) Les opérations compatibles avec les sous-programme IO dépendent de la conception de votre imprimante.

Reportez-vous au chapitre 6 de ce manuel, "Contrôle et commande de l'imprimante à partir du BASIC", pour les informations concernant l'utilisation de l'imprimante thermique PC-324.

CHAPITRE V
INDEX DE REFERENCE

CHAPITRE V

INDEX DE REFERENCE

Index de référence

Le présent chapitre décrit chaque commande, instruction et fonction du BASIC de la série TI-70. Les mots-clés sont présentés par ordre alphabétique.

Conventions utilisées dans le présent chapitre

Toutes les descriptions des mots-clés BASIC suivent le même ordre de présentation. La disposition et les conventions employées dans les descriptions sont expliquées ci-dessous.

Disposition

Le but du mot-clé est d'abord énoncé.

La partie Syntaxe donne l'articulation complète du mot-clé.

La Description expose l'utilisation et la fonction du mot-clé et comporte les options que celui-ci peut utiliser.

Le Renvoi désigne des mots-clés similaires et complémentaires, s'il y a lieu.

La partie Exemples donne des exemples d'utilisation des mots-clés, s'il y a lieu.

Conventions afférentes à la syntaxe

Les paragraphes concernant la syntaxe observent les conventions suivantes :

- Les MOTS-CLES sont en majuscules.
- Les *variables* sont en italiques.
- Les éléments facultatifs sont entre [crochets].
- Toutes les parenthèses sont obligatoires. Les parenthèses comprises dans un élément facultatif doivent figurer quand celui-ci est utilisé.
- Les éléments pouvant être répétés sont suivis de points de suspension (...).

CHAPITRE V

INDEX DE REFERENCE

ABS

Syntaxe

ABS (*expression-numérique*)

Description

La fonction ABS donne la valeur absolue d'une *expression-numérique*.

- Si l'*expression-numérique* est positive ou nulle, ABS répond par la valeur de l'*expression-numérique*.
- Si l'*expression-numérique*, est négative, ABS donne son opposé. Le résultat donné par ABS est toujours positif ou nul.

Renvoi

SGN

Exemples

```
370 PRINT ABS(42.3):PAUSE
```

Affiche 42.3.

```
140 VV=ABS(-6.124)
```

Met VV à 6,124.

ACCEPT

Syntaxe

ACCEPT [[AT(*colonne*)] [SIZE(*expression-numérique*)] [ERASE ALL]
[VALIDATE(*type-de-donnée*,...)] [NULL(*expression*)],]*variable*

Description

La forme générale de l'instruction ACCEPT

ACCEPT *variable*

affecte la donnée introduite à partir du clavier à la *variable*. La variable peut être soit un numérique soit une chaîne, selon le type de données à introduire. Vous pouvez introduire un maximum de 80 caractères et tous les espaces de fin de ligne sont ignorés.

L'affichage s'efface depuis la position du curseur jusqu'à la fin de la ligne de 80 colonnes. L'entrée commence en colonne 1, à moins qu'une instruction d'entrée/sortie en attente ne positionne le curseur dans une autre colonne.

Note : Pour introduire une chaîne comportant une virgule, des guillemets, des espaces en début ou en fin de ligne, vous devez ouvrir et fermer des guillemets. Dans une chaîne, les guillemets sont représentés par deux guillemets adjacents.

Quand une instruction ACCEPT est dans l'attente de données, [CLR] n'efface que le champ d'entrée ; [CTL] [↑] (début) et [CTL] [←] (tabulation arrière) déplacent le curseur jusqu'au début du champ d'entrée, et [CTL] [→] est sans effet.

Options

Vous pouvez utiliser l'une des options suivantes ou plus et dans n'importe quel ordre, avec l'instruction ACCEPT. Tapez un espace avant chacune d'elles (à moins qu'elle ne soit précédée de la parenthèse fermée ") d'une clause antérieure), et assurez-vous que vous avez bien placé une virgule après la dernière, et avant la *variable*.

- AT(*colonne*) place le curseur à la *colonne* spécifiée par la valeur arrondie de *colonne*. Le champ d'entrée s'étend de la position du curseur jusqu'à la fin de la ligne de 80 colonnes.

Les valeurs de *colonne* doivent obligatoirement être comprises entre 1 et 31. Si la valeur arrondie se trouve en dehors de ces limites, le message d'erreur *Bad value* (valeur incorrecte) s'affiche.

ACCEPT

- SIZE(*expression-numérique*) limite le nombre maximum de caractères qui peuvent être introduits, ce nombre pouvant aller jusqu'à la valeur absolue de l'*expression-numérique*.

Si l'*expression-numérique* est positive, le champ d'entrée est effacé avant que l'instruction ne soit acceptée.

Si l'*expression-numérique* est négative, le champ d'entrée n'est pas effacé, ce qui permet d'utiliser une instruction PRINT OU DISPLAY antérieure pour introduire une valeur par défaut dans le champ d'entrée.

Le curseur est laissé à la première position suivant le champ d'entrée en vue des instructions d'entrées/sorties suivantes.

Notez que l'option SIZE ne spécifie que le nombre maximum de caractères pouvant être introduits. Quelle que soit la longueur spécifiée, le champ d'entrée ne peut aller au-delà de la ligne de 80 colonnes.

- ERASE ALL efface la totalité de l'affichage avant d'accepter une donnée.
- VALIDATE(*type-de-données*) permet de n'accepter que certains caractères. S'il y a plus d'un *type-de-données* spécifié, un caractère de n'importe lequel d'entre eux est acceptable.

Les symboles suivants sont également des *types de données* possibles :

Type	Entrée autorisée
ALPHA	Toutes les lettres
UALPHA	Uniquement les lettres majuscules
DIGIT	Tous les chiffres de 0 à 9
NUMERIC	Tous les chiffres (0 à 9), le point décimal (.), le signe plus (+), le signe moins (−) et la lettre majuscule E
ALPHANUM	Toutes les lettres et tous les chiffres
UALPHANUM	Uniquement les lettres majuscules et les chiffres

ACCEPT

Nota : Quand on utilise les types UALPHA et UALPHANUM, toutes les lettres minuscules introduites sont automatiquement acceptées sous forme de majuscules.

Type-de-données peut également être une expression-chaîne, auquel cas, tout caractère ou combinaison de caractères mentionné dans la chaîne constitue une entrée acceptable.

- **NULL(expression)** donne la valeur par défaut spécifiée par *expression*. Si vous appuyez sur **[ENTER]** avec un champ d'entrée vierge (ou nul), la valeur par défaut est automatiquement affectée à la *variable*. Notez que la valeur par défaut n'est pas touchée par la clause **VALIDATE**.

Méthodes supplémentaires d'introduction

Il existe en plus deux méthodes qui permettent d'introduire des données pendant l'exécution d'une instruction **ACCEPT**. Vous pouvez en effet :

- Utiliser la touche **[FN]** pour entrer des mots-clés ou des chaînes affectées aux touches utilisateur.
- Introduire une expression numérique si la *variable* est numérique. L'expression est évaluée et le résultat est affecté à la *variable*.

Renvoi

INPUT, LINPUT

Exemples

```
100 ACCEPT AT(5) ERASE ALL, T
```

Efface l'affichage et accepte la donnée commençant en colonne 5, jusqu'à la fin de la ligne. La donnée est affectée à la variable T.

```
320 ACCEPT VALIDATE ("ON") SIZE(1), A$
```

Accepte un champ de un caractère, soit O soit N, pour la variable A\$.

ACCEPT

```
430 ACCEPT AT(3) SIZE(-5) VALIDATE(DIGIT, "+-"), X
```

Accepte jusqu'à 5 caractères pour la variable X, en commençant à la colonne 3. Les caractères introduits doivent être des chiffres ou les caractères + ou -. En raison de la spécification négative de la clause **SIZE**, le champ d'entrée n'est pas effacé avant l'introduction.

```
570 ACCEPT NULL(PI), C
```

Accepte une donnée pour la variable C. Si aucune donnée n'est introduite lorsque vous appuyez sur **[ENTER]**, la valeur PI est affectée à la variable C.

```
210 PRINT "ADDRESS:"; :ACCEPT AT(10), ADDR$
```

Affiche **ADDRESS :** et positionne le curseur après le message.

Accepte une donnée pour la variable ADDR\$. Notez qu'une pause n'est pas nécessaire avec **DISPLAY** du fait que le point-virgule crée une condition d'affichage en attente.

ACOS

Syntaxe

ACOS(*expression-numérique*)

Description

La fonction ACOS (arccosinus) donne l'angle dont le cosinus est l'*expression-numérique*. Le résultat est calculé avec l'unité (RAD, DEG, ou GRAD) préalablement sélectionnée.

Les domaines de valeurs de la fonction ACOS pour les trois unités d'angle sont les suivants :

Unités	Domaines de valeurs
Degrés	$0 \leq \text{ACOS}(X) \leq 180$
Radians	$0 \leq \text{ACOS}(X) \leq \text{PI}$
Grades	$0 \leq \text{ACOS}(X) \leq 200$

Renvoi

ASIN, ATN, COS, DEG, GRAD, RAD, SIN, TAN.

Exemples

```
100 DEG
Choisit DEG comme unite d'angle.
110 PRINT ACOS(1):PAUSE
Affiche 0.
220 RAD
Choisit RAD comme unité d'angle.
230 T=ACOS(0.75)
Met T égal à 0,7227342478.
```

ACOSH

Syntaxe

ACOSH(*expression-numérique*)

Description

La fonction ACOSH (arc-cosinus hyperbolique) donne le nombre dont le cosinus hyperbolique est l'*expression-numérique*. L'arc-cosinus hyperbolique est défini comme suit :

$$\text{ACOSH}(X) = \text{LN}(X + \text{SQR}(X^2 - 1))$$

Renvoi

ASINH, ATANH, COSH, SINH, TANH

Exemples

```
100 PRINT ACOSH(1):PAUSE
Affiche 0 (arc-cosinus hyperbolique de 1).
230 A=ACOSH(4/3)
Met A égal à .7953654609 (arc-cosinus hyperbolique de 4/3).
```

Syntaxe

CALL ADDMEM

Description

Le sous-programme ADDMEM permet d'ajouter à la mémoire résidente la mémoire vive (RAM) contenue dans le module *Extension Mémoire*, une fois celui-ci mis. L'ordinateur considère alors la mémoire combinée comme une mémoire résidente en continu.

On ne peut utiliser le sous-programme ADDMEM dans un programme, il ne peut être employé que comme commande.

La capacité mémoire résultant de l'adjonction de la mémoire d'un module à la mémoire résidente est de 16K octets.

Le message d'erreur E31 No RAM (pas de module RAM) s'affiche s'il n'y a pas de module *Extension Mémoire* en place à l'exécution de CALL ADDMEM.

Une fois que la mémoire d'un module *Extension Mémoire* a été ajoutée à la mémoire résidente, le système conserve les lignes de programme BASIC stockées. Toutefois, tout programme en langage assembleur se trouvant alors en mémoire est effacé.

Avant l'exécution d'un sous-programme ADDMEM, l'ordinateur identifie la mémoire du module *Extension Mémoire* comme étant une zone mémoire séparée. Le contenu de la mémoire résidente peut être stocké dans le module par recours à la commande PUT. Cependant, après l'exécution d'un sous-programme ADDMEM, l'ordinateur considère le module comme faisant partie intégrante de la mémoire RAM, et la mémoire du module ne peut plus être utilisée comme zone mémoire séparée.

La mémoire d'un module *Extension Mémoire* reste annexée à la mémoire résidente jusqu'à l'exécution de la commande NEW ALL, la mise en marche de l'ordinateur sans module, le retrait des piles, ou la réinitialisation du système.

Renvoi

GET, PUT

Syntaxe

ASC (*expression-chaîne*)

Description

La fonction ASC donne le code ASCII du premier caractère de l'*expression-chaîne*. Le message Bad argument (argument incorrect) s'affiche si l'*expression-chaîne* est une chaîne vide. Une liste des codes ASCII figure à l'annexe E. La fonction ASC est l'inverse de la fonction CHR\$.

Renvoi

CHR\$

Exemples

```
100 PRINT ASC("A"):PAUSE
```

Affiche 65.

```
130 B = ASC("1")
```

Met B égal à 49

```
790 DISPLAY ASC("HUMOUR"):PAUSE
```

Affiche 72.

ASIN

Syntaxe

ASIN(*expression-numérique*)

Description

La fonction ASIN (arc-sinus) donne l'angle dont le sinus est l'*expression-numérique*. Le résultat est calculé avec l'unité d'angle (RAD, DEG, ou GRAD) préalablement sélectionnée.

Les domaines de valeurs de la fonction ASIN pour les trois unités d'angle sont les suivants :

Unités	Domaines de valeurs
Degrés	$-90 \leq \text{ASIN}(X) \leq 90$
Radians	$-\pi/2 \leq \text{ASIN}(X) \leq \pi/2$
Grades	$-100 \leq \text{ASIN}(X) \leq 100$

Renvoi

ACOS, ATN, COS, DEG, GRAD, RAD, SIN, TAN

Exemples

```
140 DEG
```

Choisit DEG comme unité d'angle.

```
150 PRINT ASN(1):PAUSE
```

Affiche 90.

```
240 RAD
```

Choisit RAD comme unité d'angle.

```
250 B=ASN(.9)
```

Met B égal à 1,119769515.

ASINH

Syntaxe

ASINH(*expression-numérique*)

Description

La fonction ASINH (arc-sinus hyperbolique) calcule le nombre dont l'arc-sinus hyperbolique est l'*expression-numérique*. L'arc-sinus hyperbolique est défini comme suit :

$$\text{ASINH}(X) = \text{LN}(X + \text{SQR}(X^2 + 1))$$

Renvoi

ACOSH, ATANH, COSH, SINH, TANH

Exemples

```
100 PRINT ASINH(0):PAUSE
```

Affiche 0.

```
230 A=ASINH(4/3)
```

Met A égal à 1.098612289 (arc-sinus hyperbolique de 4/3).

ATANH

Syntaxe

ATANH(*expression-numérique*)

Description

La fonction ATANH (arc-tangente hyperbolique) calcule le nombre dont la tangente est l'*expression-numérique*. L'arc-tangente hyperbolique est défini comme suit :

$$\text{ATANH}(X) = .5 * \text{LN}((1 + X)/(1 - X))$$

Renvoi

ACOSH, ASINH, COSH, SINH, TANH

Exemples

```
100 PRINT ATANH(0):PAUSE
```

Affiche 0.

```
230 A=ATANH(4/7)
```

Met A égal à .6496414921 (arc-tangente hyperbolique de 4/7).

ATN

Syntaxe

ATN(*expression-numérique*)

Description

La fonction ATN (arctangente) donne l'angle dont la tangente est l'*expression-numérique*. Le résultat est calculé avec l'unité d'angle (RAD, DEG, ou GRAD) préalablement sélectionnée. Les domaines de valeurs de la fonction ATN pour les trois unités d'angle sont les suivants :

Unités	Domaines de valeurs
Degrés	$-90 < \text{ATN}(X) < 90$
Radians	$-\text{PI}/2 < \text{ATN}(X) < \text{PI}/2$
Grades	$-100 < \text{ATN}(X) < 100$

Renvoi

ACOS, ASIN, COS, DEG, GRAD, RAD, SIN, TAN

Exemples

```
130 GRAD
```

Choisit GRAD comme unité d'angle.

```
140 PRINT ATN(30):PAUSE
```

Affiche 97.87871952

```
810 RAD
```

Choisit RAD comme unité d'angle.

```
820 Q=ATN(2.5)
```

Met Q à 1,19028995.

BREAK

Syntaxe

BREAK
BREAK [*liste-de-lignes*]

Description

L'instruction BREAK sert à interrompre un programme en des points spécifiques, appelés points d'arrêt. On peut spécifier les points d'arrêt de deux façons : si aucune *liste-de-lignes* n'est donnée avec l'instruction BREAK, une interruption intervient lors de l'exécution de l'instruction elle-même. Si une *liste-des-lignes* est donnée, le(s) point(s) d'arrêt est/sont posé(s) immédiatement avant la/les ligne(s) dont le numéro figure dans la *liste-de-lignes*. La touche **[BREAK]** provoque également une interruption du programme, comme si l'instruction BREAK avait été exécutée.

A chaque interruption, le message BREAK est affiché.

Un point d'arrêt posé devant une ligne de programme reste dans le programme jusqu'à ce que l'instruction UNBREAK soit utilisée pour le retirer, ou que la ligne soit corrigée ou supprimée.

BREAK s'avère utile lors de la mise au point d'un programme. Quand l'exécution d'un programme s'interrompt à un point d'arrêt, on peut imprimer des variables et effectuer des calculs afin de savoir pour quelle raison le programme ne s'exécute pas correctement. On utilise la commande CONTINUE pour poursuivre l'exécution du programme.

Renvoi

CONTINUE, ON BREAK, UNBREAK

Exemples

150 BREAK

Interrompt l'exécution du programme

100 BREAK 120,130

Pose des points d'arrêt avant l'exécution des lignes 120 et 130.

BREAK 10,400,130

Pose des points d'arrêts avant l'exécution des lignes 10, 400 et 130.

CALL

Syntaxe

CALL *nom-de-sous-programme*
CALL *nom-de-sous-programme* (*liste-d'arguments*)

Description

Les deux syntaxes de l'instruction CALL sont décrites ci-dessous.

- CALL *nom-de-sous-programme* transfère le contrôle du programme au premier sous-programme trouvé ayant le *nom-de-sous-programme* donné. Le *nom-de-sous-programme* doit être le nom d'un sous-programme existant. Faute de quoi, le message d'erreur E13 Not found (non trouvé) s'affiche.
- CALL *nom-de-sous-programme* (*liste-d'arguments*) utilise la *liste-d'arguments* pour transmettre des données à un sous-programme ou à partir d'un sous-programme. La liste peut être constituée d'un ou plusieurs arguments, séparés par des virgules. Le nombre et les types d'arguments de la *liste-d'arguments* doivent correspondre aux paramètres de la *liste-de-paramètres* du sous-programme, sinon, le message d'erreur E23 Bad argument (argument incorrect) ou E1 Syntax s'affiche.

Ordre de priorité sous-programmes

L'ordre dans lequel l'ordinateur recherche un sous-programme spécifié est le suivant :

1. Sous-programmes système tels que ADDMEM, ERR, IO et KEY
2. Sous-programmes BASIC définis par l'instruction SUB
3. Sous-programmes de modules de logiciel.

Chaque sous-programme système est expliqué sous son propre nom dans le présent chapitre. Les sous-programmes BASIC sont traités dans le chapitre 4 et dans le présent chapitre à la rubrique SUB.

Renvoi

ADDMEM, ERR, GET, IO, KEY, PUT, SUB, SUBEND, SUBEXIT

Exemple

100 CALL KEY (K,S)

Appelle le sous-programme système KEY.

CHR\$

Syntaxe

CHR\$(*expression-numérique*)

Description

La fonction CHR\$ donne le caractère correspondant au code ASCII spécifié dans l'*expression-numérique*. La fonction CHR\$ est l'inverse de la fonction ASC. Une liste des codes ASCII correspondant à chaque caractère du jeu standard figure à l'annexe E.

On utilise CHR\$ avec les instructions PRINT et DISPLAY pour afficher les caractères spéciaux ou ceux du jeu supplémentaire, non-disponibles au clavier (voir annexe E). On peut se servir de CHR\$ pour des opérations de contrôle de périphériques, par exemple pour faire avancer l'imprimante d'une page.

Renvoi

ASC

Exemples

```
840 PRINT CHR$(72):PAUSE
```

Affiche H

```
900 X$ = CHR$(33)
```

Met X\$ = !.

CLOSE

Syntaxe

CLOSE #*numéro-de-fichier*

CLOSE #*numéro-de-fichier* [,DELETE]

Description

L'instruction CLOSE met un terme à l'association entre un fichier et son *numéro-de-fichier*. Le fichier ou le périphérique n'est plus disponible à partir du programme à moins qu'il ne soit réouvert. Une fois le fichier fermé, le *numéro-de-fichier* peut être attribué à un autre fichier ou périphérique. Si vous essayez d'utiliser l'instruction CLOSE avec un fichier qui n'est pas ouvert, il y a erreur.

Tous les fichiers ouverts sont fermés par l'une des manœuvres suivantes :

- Edition d'un programme ou d'un sous-programme.
- Introduction d'une commande NEW, RENUMBER, RUN, OLD, SAVE ou VERIFY.
- Listage du programme sur un périphérique.
- Appel du sous-programme ADDMEM.
- Arrêt ou réinitialisation de l'ordinateur.
- Pression sur la touche **[MODE]** pour passer en mode CALC.

La fin normale d'un programme ferme également tous les fichiers ouverts.

Pour les périphériques qui le permettent, on peut supprimer un fichier après l'avoir fermé, en ajoutant DELETE à l'instruction CLOSE. Le manuel de chaque périphérique explique l'emploi de DELETE.

Renvoi

DELETE, OPEN

Exemple

```
790 CLOSE #6
```

Ferme le fichier #6.

CONTINUE

Syntaxe

CONTINUE
CONTINUE [numéro-de-ligne]

Description

La commande CONTINUE (ou CON) sert à poursuivre l'exécution après une interruption. On peut continuer l'exécution d'un programme ou d'un sous-programme à partir de la ligne spécifiée par le *numéro-de-ligne*. Celui-ci doit faire référence à une ligne du programme principal si c'est ce dernier qui est arrêté. Si c'est un sous-programme qui est arrêté, le *numéro-de-ligne* doit concerner une ligne de ce sous-programme. L'emploi d'un *numéro-de-ligne* impropre provoque des résultats imprévisibles.

Dans les situations suivantes, on ne peut pas poursuivre l'exécution par une commande CONTINUE :

- Edition d'un programme ou d'un sous-programme.
- Introduction d'une commande NEW, OLD, RENUMBER, RUN, SAVE ou VERIFY.
- Listage d'un programme sur un périphérique.
- Appel du sous-programme ADDMEM.
- Arrêt ou réinitialisation de l'ordinateur.
- Pression sur la touche **[MODE]** pour passer en mode CALC.

Renvoi

BREAK

COS

Syntaxe

COS(*expression-numérique*)

Description

La fonction COS donne le cosinus de l'*expression numérique*. La fonction est calculée avec l'unité d'angle (RAD, DEG ou GRAD) préalablement sélectionnée. Voir annexe B pour une explication sur les limites de l'*expression numérique*.

Renvoi

ACOS, ASIN, ATN, DEG, GRAD, RAD, SIN, TAN

Exemples

140 GRAD

Choisit GRAD comme unité d'angle.

150 PRINT COS(30):PAUSE

Affiche .8910065242.

240 RAD

Choisit RAD comme unité d'angle.

300 T=COS(PI)

Met T égal à -1.

COSH

Syntaxe

COSH(*expression-numérique*)

Description

La fonction COSH (cosinus hyperbolique) calcule le cosinus hyperbolique de l'*expression numérique*. Le cosinus hyperbolique est défini comme suit :

$$\text{COSH}(X) = .5 * (\text{EXP}(X) + \text{EXP}(-X))$$

Renvoi

ACOSH, ASINH, ATANH, SINH, TANH

Exemples

```
100 PRINT COSH(0):PAUSE
```

Affiche 1.

```
230 T=COSH(0.75)
```

Met T égal à 1.294683285.

DATA

Syntaxe

DATA *liste-de-données*

Description

On utilise l'instruction DATA associée à l'instruction READ pour affecter des valeurs à des variables. A l'exécution de l'instruction READ, les valeurs de la *liste-de-données* sont affectées aux variables spécifiées dans la *liste-de-variables* de l'instruction READ.

La *liste-de-données* comporte des constantes numériques ou des constantes-chaîne, séparées par des virgules. Les espaces de début et de fin de ligne sont ignorés. Une constante-chaîne comportant des virgules ou des espaces en début ou en fin de chaîne doit être mise entre guillemets. Un guillemet à l'intérieur d'une chaîne déjà entre guillemets doit être représenté par un double guillemet. Une chaîne vide est représentée par deux virgules contiguës.

Une instruction DATA doit être la seule instruction d'une ligne. Elle peut se situer n'importe où dans un programme ou un sous-programme. Si un programme comporte plusieurs instructions DATA, celles-ci sont exploitées en séquence, en partant de la ligne de plus petit numéro.

L'instruction RESTORE permet d'exploiter à nouveau toutes les instructions DATA ou de modifier l'ordre dans lequel elles sont exploitées.

Renvoi

READ, RESTORE

Exemple

Le programme suivant lit et affiche plusieurs constantes numériques et constantes-chaîne :

```
100 FOR A=1 TO 5
110 READ B,C
120 PRINT B;C:PAUSE 1.1
130 NEXT A
```

Les lignes 100 à 130 lisent 5 jeux de données et impriment leur valeur, à raison de deux par ligne.

DATA

```
140 DATA 2,4,6,7,8
150 DATA 1,2,3,4,5
160 DATA """"CELLE-CI A DES GUILLEMETS""""
170 DATA "PAS DE GUILLEMETS ICI"
180 DATA PAS DE GUILLEMETS ICI NON PLUS
190 FOR A=1 TO 7
200 READ B$
210 PRINT B$:PAUSE 1.1
220 NEXT A
```

Les lignes 190 à 200 lisent sept jeux de données et les impriment chacune sur une ligne.

```
230 DATA 1,NOMBRE,,TI
```

DEG

Syntaxe

DEG

Description

L'instruction DEG sélectionne le degré comme unité d'angle. Après DEG, tous les angles et les calculs qui s'y rapportent sont exprimés en degrés. L'exécution de NEW ALL ou la réinitialisation de l'ordinateur remet l'unité d'angle au radian RAD.

Renvoi

GRAD, RAD

Exemples

```
100 DEG
```

Sélectionne l'unité d'angle DEG.

```
200 DEG:PRINT COS(90):PAUSE
```

Affiche 0 (cosinus de 90 degrés).

DELETE

Syntaxe

DELETE { *groupe-de-lignes* [, *groupe-de-lignes* ...] }
 "nom-de-fichier-périphérique"

Description

L'instruction DELETE (ou DEL) sert à ôter des lignes d'un programme en mémoire ou à supprimer un fichier d'un périphérique. Le *groupe-de-lignes* désigne les lignes de programme à supprimer et peut comporter les éléments suivants :

Groupe-de-lignes	Effet
Un numéro de ligne unique	Efface cette ligne.
numéro de ligne —	Efface cette ligne et toutes les suivantes.
— numéro de ligne	Efface cette ligne et toutes les précédentes.
numéro de ligne — numéro de ligne	Efface le groupe de lignes comprises entre ces numéros, bornes comprises.

On ne peut pas utiliser DELETE *groupe-de-lignes* dans une ligne de programme.

Si un *groupe-de-lignes* désigne un numéro de ligne unique qui n'existe pas, le message W11 Line number error (erreur de numéro de ligne) est affiché. Toutefois, tous les *groupes-de-lignes* restants sont supprimés quand on appuie sur la touche [ENTER]. Si la première ligne d'un groupe n'existe pas, la ligne de numérotation suivante est considérée comme étant la ligne initiale. Si la dernière ligne n'existe pas, la ligne précédente est considérée comme la ligne finale.

Le *périphérique.nom-de-fichier* sert à supprimer un fichier stocké sur un périphérique. Le *périphérique* est le numéro correspondant au dispositif concerné, il se situe entre 1 et 255. Le *nom-de-fichier* désigne un fichier particulier. Le *périphérique.nom-de-fichier* peut être une expression-chaîne. Consultez les manuels des périphériques pour connaître le numéro de chacun d'entre eux et pour

DELETE

obtenir des informations spécifiques quant au *nom-de-fichier*.

On peut également supprimer des fichiers de données sur certains périphériques en utilisant DELETE dans l'instruction CLOSE. Reportez-vous au manuel du périphérique concerné pour plus d'informations.

Renvoi

CLOSE

Exemples

DELETE 10-50, 90, 110-120

Supprime les lignes de 10 à 50, 90, et de 110 à 120.

DELETE 900-

Supprime les lignes 900 et suivantes jusqu'à la fin du programme.

DELETE -500, 750

Supprime toutes les lignes jusqu'à la ligne 500 et la ligne 750.

DELETE "1.fichier"

Supprime le "fichier" du périphérique 1.

Syntaxe

DIM *nom-de-tableau*(*entier1* [, *entier2*] [, *entier3*]) [...]

Description

L'instruction DIM précise les caractéristiques d'un tableau et lui réserve en mémoire l'espace nécessaire. Le *nom-de-tableau* est un nom de variable-chaîne ou de variable numérique. Le nombre des valeurs entre parenthèses suivant le *nom-de-tableau* détermine le nombre de dimensions. On peut manipuler des tableaux jusqu'à trois dimensions. Les valeurs entre parenthèses désignent les bornes supérieures des indices dans chaque dimension.

La borne inférieure d'un indice est zéro. En conséquence, le nombre d'éléments dans chaque dimension est supérieur de 1 à la borne supérieure. Par exemple, un tableau défini par DIM A(6) est un vecteur de 7 éléments, de A(0) à A(6). Si un tableau n'est pas défini par une instruction DIM, la borne supérieure de chaque indice est 10.

Au début de l'exécution d'un programme, chaque élément d'un tableau numérique est mis à zéro, et chaque élément d'un tableau de chaînes est initialisé à vide.

On ne peut fixer les dimensions d'un tableau qu'une seule fois. L'instruction DIM doit apparaître dans le programme sur une ligne de numéro inférieur à toute autre référence au tableau. Les remarques (REM) et les remarques de fin de ligne (!) sont les seules instructions pouvant apparaître après une instruction DIM dans une ligne à instructions multiples. L'instruction DIM ne peut apparaître dans une instruction IF THEN ELSE.

Exemples

```
120 DIM X$(30)
```

Réserve un espace dans la mémoire de l'ordinateur pour les 31 éléments d'un tableau appelé X\$. Chaque élément est initialisé à une valeur vide.

```
430 DIM D(100), B(10,9)
```

Réserve un espace dans la mémoire de l'ordinateur pour les 101 éléments du tableau appelé D et 110 (11 fois 10) éléments de la matrice appelée B. Chaque élément de chaque matrice est initialisé à zéro.

Syntaxe

DISPLAY [[AT(*colonne*)] [ERASE ALL] [SIZE(*expression-numérique*)]
[USING *numéro-de-ligne*,]]*liste-d'affichage*

Description

L'instruction DISPLAY donne le format et affiche la (ou les) valeur(s) comprise(s) dans la *liste-d'affichage*.

La *liste-d'affichage* peut être constituée d'une ou plusieurs expressions numériques et expressions-chaîne, séparées par des virgules ou des points-virgules. Reportez-vous à la section PRINT du présent manuel pour connaître la façon dont virgules et points-virgules affectent l'espacement entre les valeurs affichées.

Options

Les options possibles avec DISPLAY permettent de contrôler le format de l'information affichée. Elles peuvent être utilisées dans n'importe quel ordre et doivent être précédées d'un espace (à moins de n'être précédées par la parenthèse fermée ")" d'une spécification antérieure).

- AT(*colonne*) définit la position du premier caractère de l'information affichée à la colonne spécifiée par la valeur arrondie de *colonne*.

Les valeurs autorisées pour *colonne* sont 1 à 80. Si *colonne* est supérieure à 80, l'ordinateur affiche le message d'erreur E4 Bad value (valeur incorrecte).

L'évaluation de la fonction TAB et des virgules séparatrices se réfère à la position de départ spécifiée par la clause AT. Toutefois, si les caractères à afficher dépassent la colonne 80, l'affichage commence à la colonne 1, et non à la colonne spécifiée par la clause AT, la fonction TAB, ou une virgule séparatrice.

L'action de l'option AT peut être affectée par l'option SIZE. Reportez-vous à la page suivante pour plus de précisions concernant l'incidence de SIZE sur l'information affichée.

DISPLAY

Quand la clause AT n'est pas utilisée, la sortie commence à la position 1, à moins d'une instruction d'entrée/sortie en attente, et la fonction TAB et les virgules séparatrices se réfèrent à la colonne 1.

- ERASE ALL efface la ligne avant que les données ne soient affichées.
- SIZE(*expression-numérique*) limite le nombre total des caractères à la valeur absolue de l'*expression-numérique*.

Si l'*expression-numérique* est supérieure au nombre de positions restantes, le champ d'affichage s'étend de la position du curseur à la fin de la ligne de 80 colonnes. La longueur du champ d'affichage ainsi définie devient la nouvelle longueur d'enregistrement destinée à l'évaluation de la fonction TAB et des virgules séparatrices dans la *liste-d'affichage*. Le champ spécifié est toujours effacé avant l'affichage des données.

En l'absence de clause SIZE, et si les caractères à afficher dépassent la colonne 80, vous ne pouvez visualiser l'information qu'en exécutant une instruction PAUSE ALL avant la ligne d'affichage. Vous pouvez alors visualiser ligne par ligne en pressant [ENTER] ou [CLR] pour afficher la ligne suivante. Cependant, en cas d'utilisation de la clause SIZE lorsque les caractères dépassent la colonne 80, le texte est ramené à cette colonne ou au nombre de caractères spécifié par SIZE, s'il est plus réduit.

En l'absence de clause SIZE, l'affichage n'est pas effacé (à moins que ERASE ALL n'ait été spécifié) avant l'affichage des données. S'il n'y a pas de séparateur de fin de ligne après la *liste-d'affichage*, la fin de l'instruction DISPLAY efface l'affichage depuis le dernier élément affiché jusqu'à la fin de la ligne de 80 colonnes.

- L'emploi de USING permet de spécifier le format exact de l'impression. Si USING est spécifié, il doit apparaître en dernier dans la liste des options. Consultez IMAGE et USING pour avoir la description de la définition du format et ses effets sur l'impression de l'instruction DISPLAY.

DISPLAY

Renvoi

IMAGE, PAUSE, PRINT, TAB, USING

Exemples

120 DISPLAY AT(7),Y

Affiche la valeur de Y en commençant à la colonne 7 et efface tout ce qui suit le nombre. La valeur apparaît effectivement dans la colonne 8 puisque le signe précède le nombre.

150 DISPLAY N

Affiche la valeur de N en colonne de 1 et efface le reste de l'affichage.

190 DISPLAY ERASE ALL,B

Efface la totalité de la ligne avant d'afficher la valeur de B.

370 DISPLAY AT(C) SIZE(19),X

Efface 19 caractères à partir de la position de C et affiche la valeur de X à partir de cette même position C.

Syntaxe

END

Description

L'instruction END termine un programme et peut être utilisée indifféremment avec l'instruction STOP. Bien que l'instruction END puisse apparaître n'importe où, elle figure habituellement à la dernière ligne d'un programme. Cette instruction n'est pas nécessaire. Un programme s'arrête automatiquement après la ligne de numérotation la plus élevée.

L'instruction END ferme tous les fichiers ouverts.

Renvoi

STOP

Syntaxe

EOF(*numéro-de-fichier*)

Description

La fonction EOF permet de savoir s'il y a un autre enregistrement à lire dans un fichier. La valeur du *numéro-de-fichier* désigne le fichier à tester et doit correspondre au numéro d'un fichier ouvert. EOF donne une valeur qui indique la position courante dans un fichier, de la façon suivante :

Valeur	Position
--------	----------

0	Pas de fin de fichier
---	-----------------------

-1	Fin de fichier logique
----	------------------------

La fin de fichier logique intervient quand tous les enregistrements du fichier ont été traités.

Quand on utilise une instruction INPUT en attente, EOF n'indique pas si les données en attente de traitement restent en mémoire.

Renvoi

INPUT (avec fichiers)

Exemples

140 PRINT EOF(3):PAUSE

Imprime -1 si le fichier #3 a atteint la fin de fichier, et 0 si ce n'est pas le cas.

710 IF EOF(27) THEN 1150

Transfère le contrôle à la ligne 1150 si la fin de fichier a été atteinte pour le fichier #27.

CHAPITRE V INDEX DE REFERENCE

SOUS-PROGRAMME

ERR

Syntaxe

CALL ERR(*code-d'erreur*, *type-d'erreur* [, *numéro-de-fichier*,
numéro-de-ligne])

Description

Le sous-programme ERR donne le code d'erreur, le type d'erreur et, facultativement, le numéro de fichier et le numéro de ligne de la dernière erreur non effacée. Quand une erreur se produit, on peut appeler une sous-routine (voir ON ERROR) contenant CALL ERR. L'erreur est effacée quand la sous-routine de traitement des erreurs se termine par un RETURN.

Les *codes-d'erreur* s'échelonnent de 0 à 127. La signification de chacun de ces codes est donnée à l'annexe G.

Le *type-d'erreur* est toujours 0 sauf si le code-d'erreur est 0, ce qui représente une erreur d'entrée/sortie (I/O). Quand il s'agit d'une erreur d'I/O, le *type-d'erreur* est un code d'erreur d'I/O lié au périphérique. Les codes-d'erreur d'I/O s'échelonnent entre 1 et 255.

Le *numéro-de-fichier* est 0 sauf si l'erreur est une erreur d'I/O. Quand il s'agit d'une erreur d'I/O, le *numéro-de-fichier* est le numéro utilisé dans l'instruction d'I/O qui a provoqué l'erreur. Le *numéro-de-ligne* est le numéro de la ligne dans laquelle l'erreur s'est produite lors de l'exécution. La ligne n'est pas toujours à l'origine du problème puisqu'une erreur peut se produire à cause de valeurs générées ou d'actions effectuées ailleurs dans un programme.

Si aucune erreur ne s'est produite, CALL ERR donne toutes les valeurs égales à 0.

Renvoi

ON ERROR, RETURN (avec ON ERROR)

Exemples

170 CALL ERR(A,B)

Donne A égal au *code-d'erreur* et B égal au *type-d'erreur* de la plus récente erreur non effacée.

CHAPITRE V INDEX DE REFERENCE

SOUS-PROGRAMME

ERR

390 CALL ERR(W,X,Y,Z)

Donne W égal au *code-d'erreur*, X égal au *type-d'erreur*, Y égal au *numéro-de-fichier*, et Z égal au *numéro-de-ligne* de la plus récente erreur non effacée.

EXP

Syntaxe

EXP(*expression-numérique*)

Description

La fonction EXP donne le résultat de e^x , où x est l'*expression-numérique*. La valeur de e est de 2,71828182846.

Renvoi

LN

Exemple

```
150 Y=EXP(7)
```

Affecte à Y la valeur de e élevé à la puissance 7, soit 1096,633158429.

```
390 L=EXP(4.394960467)
```

Affecte à L la valeur de e élevé à la puissance 4,394960467, soit 81,04142688867.

FOR TO STEP

Syntaxe

FOR *variable-de-commande* = *valeur-initiale* TO *limite* [STEP *pas*]

Description

On utilise l'instruction FOR TO STEP avec l'instruction NEXT pour former une boucle qui constitue une série d'instructions effectuées un nombre spécifique de fois. La *variable-de-commande* est une variable numérique non-indicée qui agit en tant que compteur pour la boucle. La *valeur-initiale*, la *limite* et le *pas* sont des expressions numériques.

A l'exécution de l'instruction FOR, la *valeur-initiale* est affectée à la *variable-de-commande*. Dans le cas d'un *pas* positif, si la *valeur-initiale* dépasse la *limite*, l'ordinateur saute la boucle et l'exécution se poursuit par l'instruction suivant l'instruction NEXT. Autrement, les instructions succédant à l'instruction FOR sont exécutées jusqu'à l'exécution de l'instruction NEXT correspondante. Le *pas* est alors ajouté à la *variable-de-commande*. Si la *variable-de-commande* n'est pas supérieure à la *limite*, l'exécution retourne à l'instruction suivant l'instruction FOR.

Quand la *variable-de-commande* devient supérieure à la *limite*, le contrôle est transféré à l'instruction succédant à l'instruction NEXT. la *variable-de-commande* est alors égale à la valeur qu'elle avait lors du dernier passage dans la boucle, additionnée de la valeur du *pas*.

On dit d'une boucle entièrement contenue par une autre boucle qu'elle est imbriquée. Les boucles imbriquées doivent utiliser des variables de commande différentes. On peut sortir d'une boucle à l'aide des instructions GOTO, GOSUB ou IF THEN ELSE, puis revenir dans la boucle.

Si une instruction NEXT est exécutée avant l'instruction FOR correspondante, une erreur se produit.

STEP spécifie le *pas* qui est ajouté à la *variable-de-commande* à chaque exécution de la boucle. En l'absence de clause STEP, le *pas* est 1. Si le *pas* est négatif, la *variable-de-commande* diminue à chaque exécution de la boucle, et la *limite* doit être inférieure à la *valeur-initiale*. Si la *valeur-initiale* est inférieure à la *limite* la boucle est sautée. Dans le cas contraire, la boucle est exécutée

FOR TO STEP

jusqu'à ce que la *variable-de-commande* soit inférieure à la *limite*.

Renvoi

NEXT

Exemples

```
140 FOR A=1 TO 5 STEP 2
```

```
.  
. .  
. .
```

```
190 NEXT A
```

Exécute les instructions entre FOR et NEXT A trois fois, avec pour A les valeurs 1, 3, et 5. Après la fin de la boucle, A a pour valeur 7.

```
250 FOR J=7 TO -5 STEP -.5
```

```
.  
. .  
. .
```

```
350 NEXT J
```

Exécute les instructions entre FOR et NEXT J 25 fois, J ayant pour valeurs 7, 6.5, 6, ..., -4, -4.5, et -5. Après la fin de la boucle, J a pour valeur -5.5.

```
700 FOR X=1 TO 2 STEP -1
```

```
.  
. .  
. .
```

```
780 NEXT X
```

N'exécute pas la boucle puisque le *pas* est négatif et que la *valeur-initiale* est déjà inférieure à la *limite*.

FORMAT

Syntaxe

FORMAT *périphérique*

Description

L'instruction FORMAT initialise le support utilisé sur un périphérique externe. Cette opération détruit les données préalablement enregistrées.

Le *périphérique* est un numéro correspondant à un périphérique et peut se situer entre 1 et 255. Consultez les manuels des périphériques pour connaître le numéro de code de chacun d'entre eux.

Nota : Cette instruction n'est pas valable pour des cassettes utilisées avec un magnétophone.

Exemple

```
140 FORMAT 2
```

Initialise le support qui se trouve dans le périphérique appelé périphérique numéro 2. Toutes les données préalablement enregistrées sur ce support sont détruites.

Syntaxe

FRE(*expression-numérique*)

Description

La fonction FRE donne des informations sur l'espace mémoire disponible et sur l'espace déjà utilisé :

- place disponible pour le stockage d'un programme et de données.
- Place occupée par le programme actuellement en mémoire.

La valeur de l'*expression-numérique* spécifie le type d'informations désirées de la façon suivante :

Valeur	Signification
0	Mémoire disponible pour le stockage d'un programme et de données (espace mémoire <i>non</i> réservé au fonctionnement de l'ordinateur).
1	Espace total occupé par le programme actuellement en mémoire, y compris 11 octets de prise en charge.

Exemple

300 A=FRE(1)

Etablit A égal au nombre d'octets nécessaire au stockage du programme en cours.

Syntaxe

CALL GET(*numéro-d'image*)

Description

Le sous-programme GET permet de récupérer la copie d'une image mémoire RAM système d'un module *Extension Mémoire*, y compris les lignes de programme, les variables et l'espace inutilisé.

Le *numéro-d'image* peut être 1 ou -1. Avec le numéro 1, l'image du module est copiée en mémoire, et avec le numéro -1, les images de la mémoire sont échangées. Cette option permet de stocker le programme à partir de la mémoire pendant que vous récupérez un programme en module.

Au fur et à mesure que le contenu du module est dupliqué ou échangé, l'ordinateur vérifie que le module contient bien une image de la mémoire système. Si ce n'est pas le cas, l'ordinateur renvoie un message d'erreur.

Renvoi

PUT, ADDMEM

Exemples

CALL GET(1)

Duplique l'image du module dans la mémoire système RAM.

CALL GET(-1)

Remplace l'image du module par la mémoire système RAM.

GOSUB

Syntaxe

GOSUB *numéro-de-ligne*

Description

L'instruction GOSUB transfère le contrôle à la sous-routine commençant à la ligne de numéro *numéro-de-ligne*. Les instructions d'une sous-routine sont exécutées jusqu'à la rencontre d'une instruction RETURN. Une instruction RETURN renvoie à l'instruction suivant immédiatement l'instruction GOSUB.

Les sous-routines peuvent être appelées un nombre quelconque de fois à l'intérieur d'un programme, s'appeler elles-mêmes ou appeler d'autres sous-routines. On ne peut utiliser l'instruction GOSUB pour se transférer vers un sous-programme, ou en sortir.

Renvoi

ON GOSUB, RETURN

Exemple

```
100 GOSUB 200
```

Transfère le contrôle à la ligne 200. L'instruction de cette ligne et toutes les instructions suivantes sont exécutées jusqu'à la rencontre de RETURN. RETURN renvoie à l'instruction suivant l'instruction GOSUB.

GOTO

Syntaxe

GOTO *numéro-de-ligne*

Description

L'instruction GOTO renvoie sans condition à une autre ligne de programme. A l'exécution de l'instruction GOTO, le contrôle est transféré à la première instruction de la ligne spécifiée par le *numéro-de-ligne*.

On ne peut utiliser l'instruction GOTO pour se transférer vers un sous-programme ou en sortir.

Exemple

```
100 GOTO 300
```

Transfère le contrôle à la ligne 300.

GRAD

Syntaxe

GRAD

Description

L'instruction GRAD choisit comme unité de calcul angulaire le grade. Après qu'on ait sélectionné le GRAD, toutes les valeurs d'angle, introduites ou calculées sont exprimées en grades. L'unité est remplacée par RAD après l'introduction de NEW ALL ou lors de l'initialisation de l'ordinateur.

Renvoi

DEG, RAD

Exemples

100 GRAD

Sélectionne l'unité d'angle GRAD.

200 GRAD:PRINT COS(100):PAUSE

Affiche 0 (arc-cosinus de 100 degrés).

IF THEN ELSE

Syntaxe

IF *condition* THEN *action1* (ELSE *action2*)

Description

L'instruction IF THEN ELSE effectue l'une ou l'autre de deux actions, suivant une certaine condition. Si la *condition* est vraie, l'*action1* est effectuée. Si la *condition* est fausse, l'*action2* est effectuée. En l'absence de clause ELSE, et si la *condition* est fausse, le contrôle est transféré à la ligne suivante.

La *condition* peut être une expression logique ou une expression numérique. Lors de l'évaluation d'une expression logique, le résultat est 0 si elle fausse et -1 si elle est vraie. Lors de l'évaluation d'une expression numérique, une valeur 0 est considérée fausse et une valeur non nulle est considérée vraie.

Les *action1* et *action2* peuvent être des numéros de ligne, des instructions ou des groupes d'instructions séparés par deux points. Si un numéro de ligne est mentionné, le contrôle est transféré à cette ligne. Si des instructions figurent, celles-ci sont exécutées.

L'instruction IF THEN ELSE se termine à la fin de la ligne, et doit donc tenir dans cette ligne. Ce type d'instruction peut être imbriqué, en plaçant une instruction IF THEN ELSE dans l'*action1* ou l'*action2*. Si un ensemble de IF THEN ELSE imbriquées ne contient pas le même nombre de clauses THEN et ELSE, chaque ELSE est appariée au THEN le plus proche.

Les instructions IF THEN ELSE ne peuvent contenir des instructions DIM, IMAGE, SUB ou SUBEND.

Exemples

100 IF Y < 5 THEN 150

Si la valeur de Y est inférieure à 5, l'instruction 150 est exécutée. Si Y est supérieur ou égal à 5, l'instruction suivante est exécutée.

140 IF MBB=0 THEN 200

150 PRINT "NON-NUL":PAUSE 2

IF THEN ELSE

Si MBB est égal à zéro, le contrôle passe à ligne 200. Si MBB n'est pas égal à zéro, NON-NUL est affiché et l'exécution du programme est interrompu pendant 2 secondes avant de passer à l'instruction suivante.

```
230 IF X > 5 THEN GOSUB 300 ELSE X=X+5
```

Si la valeur de X est supérieure à 5, GOSUB 300 est exécuté. Quand la sous-routine est achevée, le contrôle retourne à la ligne suivant l'instruction IF THEN ELSE. Si X est égal ou inférieur à 5, X est augmenté de 5 et le contrôle passe à la ligne suivante.

```
250 IF Q THEN C=C+1:GOTO 500 ELSE L=L/C:GOTO 300
```

Si Q n'est pas égal à zéro (vrai) C est augmenté de 1 et le contrôle est transféré à la ligne 500. Si Q est égal à zéro (faux), L est divisé par C et le contrôle est transféré à la ligne 300.

```
290 IF A$="O" THEN COUNT=COUNT+1:DISPLAY  
AT(4),"ON RECOMMENCE!":GOTO 400
```

Si A\$ est égal à "O", COUNT est augmenté de 1, un message s'affiche, et le contrôle est transféré à la ligne 400. Si A\$ n'est pas égal à "O", le contrôle passe à la ligne suivante.

```
350 IF HRS <= 40 THEN PAYE =HRS*SALAIRE ELSE PAYE  
=HRS*SALAIRE+.5*SALAIRE*(HRS-40):OT=1
```

Si HRS est inférieur ou égal à 40, PAYE est égal à HRS*SALAIRE et le contrôle passe à la ligne suivante. Si HRS est supérieur à 40, PAYE est alors égal à HRS*SALAIRE + 0,5*SALAIRE*(HRS - 40), OT est mis à 1, et le contrôle passe à la ligne suivante.

```
700 IF A=1 THEN IF B=2 THEN C=3 ELSE D=4
```

Si A est égal à 1 et B est égal à 2, C est mis à 3 et le contrôle passe à la ligne suivante. Si A est égal à 1 et B n'est pas égal à 2, D est mis à 4 et le contrôle passe à la ligne suivante. Si A n'est pas égal à 1 le contrôle passe à la ligne suivante.

IMAGE

Syntaxe

IMAGE *constante-chaîne*

Description

L'instruction IMAGE sert à définir le format d'une sortie. On utilise le format en introduisant le numéro de ligne de l'instruction IMAGE dans l'option USING des instructions PRINT ou DISPLAY (voir USING dans le présent chapitre). La *constante-chaîne* peut être entre guillemets. Si ce n'est pas le cas, les blancs en début et en fin de ligne sont ignorés.

L'instruction IMAGE doit être la seule dans une ligne de programme et doit apparaître dans le programme ou le sous-programme qui l'utilise. A la rencontre d'une instruction IMAGE, l'exécution continue immédiatement avec la ligne suivante du programme.

Une définition de format (voir ci-dessous) est décomposée en champs qui sont, soit des modèles, soit des littéraux. Lorsqu'une instruction PRINT ou une instruction DISPLAY utilise un format, les modèles sont remplacés par les valeurs des éléments à afficher et les littéraux sont affichés tels qu'ils apparaissent dans la définition du format.

Définition de format

Les trois caractères pouvant intervenir dans un modèle sont le signe dièse (#), le point décimal (.), et le symbole de puissance (^). Les signes dièse représentent chacun une position de caractères dans le modèle. Il sont remplacés par les caractères issus de la représentation ASCII de la valeur de l'élément à imprimer. Le point décimal est utilisé pour préciser la position du point décimal. Le symbole de puissance (^) sert le cas échéant à préciser le nombre de positions sur lequel doit être imprimé l'exposant. Tous les autres caractères du format sont alphabétiques, et constituent des littéraux.

Les cinq types de champs dans la définition d'un format sont les modèles d'entiers, de nombres décimaux, de notations scientifiques et de chaînes, et les littéraux. Les règles s'appliquant à chacun d'entre eux sont présentées ci-dessous.

Modèle d'entier

- On peut spécifier jusqu'à 14 chiffres significatifs.
- Un modèle d'entier est composé exclusivement de signes dièse.
- Quand le nombre à sortir ne remplit pas le champ, il est cadré à droite.
- Quand le nombre à sortir dépasse le champ, des astérisques (*) sont sortis à sa place.
- Les valeurs non entières sont arrondies au nombre entier le plus proche.
- Quand un nombre à sortir est négatif, le signe moins remplace un signe dièse.

Modèle de nombre décimal

- On peut spécifier jusqu'à 14 chiffres significatifs.
- Un modèle de nombre décimal est composé de signes dièse et d'un seul point décimal. Le point décimal peut apparaître n'importe où dans le modèle.
- Le nombre à sortir est muni de point décimal à la position spécifiée.
- Quand la partie entière du nombre est plus longue que la partie entière du format, des astérisques (*) sont sorties à sa place.
- Le nombre à sortir est arrondi au nombre de positions spécifiées à droite du point décimal.
- Quand le nombre est négatif, au moins un signe dièse doit précéder le point décimal pour servir de signe moins.

Modèle de notation scientifique

- On peut spécifier jusqu'à 14 chiffres significatifs.
- Le modèle se compose d'un modèle décimal ou entier, qui définit la mantisse, suivi de 4 ou 5 symboles de puissance qui définissent l'exposant. A moins de 4, ceux ci sont considérés comme des littéraux. A plus de 5, les cinq

premiers servent à définir l'exposant, les autres sont considérés comme littéraux.

- Le nombre à sortir est arrondi conformément à la définition de la mantisse.
- Quand la définition de la mantisse comporte des positions à gauche du point décimal, une de ces positions est toujours utilisée pour le signe, qui est un signe moins si le nombre à sortir est négatif et un espace s'il est positif.

Modèle de chaîne

- La dimension du modèle est seulement limitée par celle de la chaîne qui définit le format.
- Le modèle est de type entier, décimal ou notation scientifique. En plus des signes dièse, le point décimal et les symboles de puissance représentent chacun une position de caractère.
- Quand la chaîne à sortir est plus courte que le modèle, elle est cadrée à gauche.
- Quand la chaîne est plus longue que le modèle, des astérisques (*) sont sorties à sa place.

Littéraux

- La dimension d'un littéral n'est limitée que par celle de la chaîne qui définit le format.
- Un littéral se compose de caractères qui ne sont pas des caractères de modèle. Cependant, des points décimaux et des symboles de puissance peuvent y apparaître.
- Les champs de caractères apparaissent en sortie tels qu'ils se présentent dans la définition du format.

IMAGE

Renvoi

DISPLAY, PRINT, USING

Exemples

Le programme suivant imprime deux nombres par ligne en utilisant l'instruction IMAGE.

```
100 FOR COMPTE=1 TO 6
110 READ A,B
120 PRINT USING 150;A,B:PAUSE
130 NEXT COMPTE
140 DATA -99,-9.99,-7,-3.459,0,0,14.8,12.75,
795,852,-984,64.7
150 IMAGE LES REPOSES SONT ### ET ##.##
```

Les données et les résultats se présentent de la façon suivante :

Valeurs		Affichage	
99	-9.99	LES REPOSES SONT	99 ET -9.99
-7	-3.459	LES REPOSES SONT	-7 ET -3.46
0	0	LES REPOSES SONT	0 ET .00
14.8	12.75	LES REPOSES SONT	15 ET 12.75
795	852	LES REPOSES SONT	795 ET *****
-984	64.7	LES REPOSES SONT	*** ET 64.70

Un programme analogue permet d'afficher des chaînes de caractères, avec comme format : IMAGE CHER #####. Le tableau ci-dessous indique les résultats obtenus avec certaines données.

Valeur	Affichage
JEAN,	CHER JEAN,
ANDRE,	CHER ANDRE,
FREDERIC,	CHER *****

IMAGE

Le programme présenté ci-dessous illustre une autre utilisation de l'instruction IMAGE. Il lit et affiche sept sommes et leur total. Les sommes sont affichées avec les points décimaux.

```
100 IMAGE F####.##
110 IMAGE "####.##"
```

Les lignes 100 et 110 établissent les formats. Elles sont identiques, au signe F près. Pour remplacer celui ci par un espace, la *constante-chaîne* de la ligne 110 est placée entre guillemets.

```
120 DATA 233.45,-147.95,8.4,37.263,-51.299,
85.2,464
130 TOTAL=0
140 FOR A=1 TO 7
150 READ SOMME
160 TOTAL=TOTAL+SOMME
170 IF A=1 THEN PRINT USING 100, SOMME:PAUSE ELSE
PRINT USING 110,SOMME:PAUSE
```

Affiche les valeurs en employant les instructions IMAGE.

```
180 NEXT A
190 PRINT USING "F####.##", TOTAL:PAUSE
```

Utilise le format directement dans l'instruction PRINT.

Syntaxe

INPUT [*message-guide*] *liste-de-variables* [,*message-guide* *liste-de-variables*] [...]

Description

L'instruction INPUT sert à introduire des données au clavier. Quand cette instruction est exécutée, l'exécution du programme est interrompue jusqu'à l'entrée de données.

Le *message-guide* est une expression-chaîne qui doit être suivie d'un point-virgule. S'il s'agit d'une constante chaîne, elle doit être placée entre guillemets. Le *message-guide* est affiché à partir de la position courante du curseur tel qu'il a été laissé par les précédentes instructions d'entrée sortie. En l'absence de *message-guide*, un point d'interrogation suivi d'un espace est affiché à sa place.

Le curseur clignotant est affiché à la suite du message. S'il se trouve que la position du curseur se situe au-delà de 31, l'affichage est effacé et le curseur se place en colonne 1 avant l'affichage du message. Quand le *message-guide* comporte plus de 30 caractères, il est tronqué après 30 caractères.

La *liste-de-variables* comporte des variables séparées par des virgules. Il peut s'agir de variables numériques ou de variables-chaînes, indicées ou non indicées. Quand plus d'une variable succède au *message-guide*, le message n'est affiché que pour la première variable. Ensuite, figure un point d'interrogation jusqu'à ce qu'un autre *message-guide* soit rencontré. Chaque valeur est affectée à la variable correspondante avant que l'ordinateur ne demande la valeur suivante.

Lors de l'entrée de variables numériques, on peut introduire une expression numérique au lieu d'une constante. L'expression est évaluée et le résultat est affecté à la variable. A l'entrée de variables-chaînes, les espaces de début et de fin de ligne sont ignorés. En conséquence, si une chaîne de valeurs comporte des virgules, des espaces de début ou de fin de ligne, elle doit figurer entre guillemets. Les guillemets à l'intérieur de guillemets doivent être représentés par des doubles guillemets.

S'il se produit une erreur pendant l'introduction des données, un message d'erreur est affiché. L'instruction INPUT est reprise dès que l'on fait [ENTER] ou [CLR], et l'on peut à nouveau introduire les données sous la forme correcte.

Quand une donnée est introduite, sa validité est contrôlée de la façon suivante :

- Si plus d'une valeur est introduite à la fois, le message E1 at line number Syntax (syntaxe incorrecte au numéro de ligne) est affiché, les données doivent alors être réintroduites, une à une.
- Si on introduit une constante-chaîne comme variable numérique, le message E3 at line number Mismatch (défaut de concordance au numéro de ligne) est affiché et une valeur numérique doit être introduite.
- Si un nombre dont la valeur absolue est supérieure à 9,99999999999999E + 127 est introduit, le message W25 at line number Overflow (dépassement de capacité au numéro de ligne) est affiché, la valeur doit alors être réintroduite.
- Si un nombre dont la valeur absolue est inférieure à 1E - 128 est introduit, la valeur est remplacée par 0, aucun message n'est affiché.

Note : Quand une instruction INPUT est dans l'attente de données, [CLR] n'efface que le champ d'entrée, [CTL] [↑] et [CTL] [←] déplacent le curseur vers le début de ce champ, et [CTL] [→] est sans effet.

Renvoi

ACCEPT, INPUT (avec fichiers), LINPUT

Exemples

100 INPUT X

Entraîne l'affichage d'un point d'interrogation, l'ordinateur attendant l'introduction d'une valeur. Quand on appuie sur [ENTER], la valeur introduite est mémorisée dans la variable X.

INPUT

```
100 INPUT X$,Y,"INTRODUIRE Z";Z(A)
```

Entraîne l'affichage d'un point d'interrogation, l'ordinateur attendant l'introduction d'une valeur pour X\$. Quand on appuie sur **[ENTER]**, la valeur introduite est affectée à X\$. Le point d'interrogation est à nouveau affiché et l'ordinateur attend qu'une valeur soit introduite pour Y. Ensuite, **INTRODUIRE Z** est affiché et l'ordinateur attend l'introduction d'une valeur pour Z(A). L'indice est évalué pour Z(A) avant que la valeur de la donnée soit mémorisée.

INPUT

Syntaxe

INPUT #*numéro-de-fichier* [, *REC expression-numérique*],
liste-des-variables

Description

L'instruction **INPUT** sert à lire des données dans des fichiers qui ont été ouverts en mode **INPUT** ou **UPDATE**. Une valeur du fichier est affectée à chaque variable de la *liste-des-variables*.

Le *numéro-de-fichier* est un nombre entre 0 et 255 désignant un fichier ouvert ou un périphérique. Le numéro de fichier 0 désigne le clavier et l'affichage, il est toujours ouvert. Voir **INPUT** (au clavier). Le *numéro-de-fichier* est arrondi au nombre entier le plus proche.

La *liste-des-variables* contient des variables séparées par des virgules. Il peut s'agir de variables numériques, de variables-chaîne, indicées ou non indicées. Les données des variables de l'enregistrement courant sont affectées aux variables de la liste. Si l'enregistrement courant ne contient pas suffisamment de données, un autre enregistrement est lu. Des enregistrements successifs sont lus jusqu'à ce qu'une valeur soit affectée à chacune des variables, ou qu'intervienne une fin de fichier.

L'ordinateur interprète différemment les données en lecture, selon leur appartenance au type **DISPLAY** ou **INTERNAL**. Voir "PRINT avec fichiers" au chapitre 5.

Une donnée de type **DISPLAY** a la même forme qu'une donnée introduite à partir du clavier. Les valeurs de chaque enregistrement sont séparées par des virgules. Les espaces de début et de fin de ligne sont ignorés à moins qu'ils ne fassent partie d'une chaîne de caractères entre guillemets. Un guillemet à l'intérieur d'une chaîne entre guillemets doit être représenté par un double guillemet. Quand l'instruction **INPUT** rencontre deux virgules contiguës, une chaîne vide est affectée à la variable. Chaque élément est vérifié pour s'assurer que les valeurs numériques sont affectées à des variables numériques et chaînes à des variables-chaîne.

CHAPITRE V INDEX DE REFERENCE

AVEC FICHIERS

INPUT

Une donnée de type INTERNAL est en format binaire, celui utilisé lors d'une exécution interne. Chaque valeur est précédée de sa longueur. L'instruction INPUT utilise les longueurs pour séparer et affecter les valeurs à des variables. La seule validation effectuée par l'instruction INPUT consiste à veiller à ce que la donnée numérique ait une longueur de 2 à 8 octets.

Quand une instruction INPUT se termine, toutes les valeurs de données restant dans l'enregistrement courant sont en général ignorées. L'instruction INPUT suivante relative au même fichier lit un autre enregistrement. Cependant, quand la *liste-des-variables* se termine par une virgule, l'entrée est laissée en attente. C'est-à-dire que les valeurs restantes de l'enregistrement courant sont gardées. La prochaine instruction INPUT relative au même fichier affecte la prochaine valeur disponible.

S'il existe encore des données en attente d'entrée quand une instruction PRINT, RESTORE ou CLOSE accède au même fichier, ces données sont perdues. S'il y a des données en attente d'impression quand l'instruction INPUT intervient, ces données sont imprimées avant l'exécution de l'instruction INPUT.

REC *expression-numérique* est utilisée quand le *numéro-de-fichier* désigne un fichier à accès direct. L'*expression-numérique* désigne l'enregistrement à lire dans le fichier. Le premier enregistrement d'un fichier est l'enregistrement zéro. Voir "Utilisation des Périphériques Externes" au chapitre 6 et consulter les manuels des périphériques pour une plus ample information sur les fichiers à accès direct et l'utilisation de la clause REC.

Renvoi

CLOSE, INPUT (au clavier), LINPUT, OPEN, PRINT, RESTORE

Exemples

```
100 INPUT #1, X$
```

Affecte dans X\$ la prochaine chaîne lue dans le fichier ouvert #1.

```
250 INPUT #23, X, A, LL$
```

Affecte à X, A et LL\$ les trois prochaines valeurs lues dans le fichier ouvert #23.

CHAPITRE V INDEX DE REFERENCE

AVEC FICHIERS

INPUT

```
320 INPUT #3, A, B, C,
```

Affecte à A, B et C les trois prochaines valeurs lues dans le fichier ouvert #3. La virgule après C crée une condition d'entrée en attente.

```
100 OPEN #, "1.MONFICHIER", DISPLAY, UPDATE
110 FOR A=1 TO 5
120 READ SORTIEDONNEE
130 PRINT #1, SORTIEDONNEE
```

Les lignes 110 à 130 lisent cinq enregistrements de l'instruction DATA et les écrivent dans le fichier #1.

```
140 PRINT SORTIEDONNEE; "EST ECRIT DANS FICHIER
      #1.": PAUSE 1.5
```

```
150 NEXT A
160 RESTORE #1
170 FOR B=1 TO 5
180 INPUT #1, ENTREEDONNEE
190 PRINT ENTREEDONNEE; "DANS
      ENREGISTREMENT #"; B: PAUSE 1.5
200 NEXT B
```

Les lignes 170 à 200 lisent les cinq enregistrements qui ont été écrits sur le fichier #1 puis affichent leurs valeurs.

```
210 CLOSE #1, DELETE
```

Supprime le fichier.

```
220 DATA 15, 30, 72, 36, 94
```

Syntaxe

INT(*expression-numérique*)

Description

La fonction INT donne le plus grand nombre entier inférieur ou égal à l'*expression-numérique*.

Exemples

```
250 P=INT(3.999999999)
```

Met P égal à 3.

```
470 DISPLAY AT(7),INT(4.0):PAUSE
```

Affiche 4 en colonne 8.

```
610 K=INT(-3.0000001)
```

Met K égal à -4.

Nota : En mode calculatrice, la fonction Intg exerce une action différente pour les valeurs négatives.

Syntaxe

CALL IO (*périphérique*, *commande* [, *variable-d'état*])

Description

Le sous-programme IO permet d'adresser un périphérique en vue d'effectuer une commande ou une opération de contrôle spéciales qui ne sont pas disponibles en TI-74 BASIC. Les commandes et opérations de contrôle disponibles avec un périphérique sont fonction de la configuration de ce dernier.

L'utilisation correcte de ce sous-programme suppose une connaissance des structures de données d'entrée/sortie (I/O) et des capacités particulières des périphériques. Consultez les exemples d'utilisation du sous-programme IO dans le manuel du périphérique pour de plus amples informations.

Le *périphérique* est le numéro correspondant au périphérique, ce peut être un nombre compris entre 1 et 255.

La *commande* est le code numérique spécifiant l'opération à effectuer sur le périphérique.

La *variable-d'état* est une variable numérique dans laquelle des données concernant le résultat de l'opération sont stockées. Tant qu'il n'y a pas d'erreur d'I/O, la *variable-d'état* reste à zéro. Si une erreur d'I/O se produit, la *variable-d'état* comporte le code d'erreur correspondant.

L'inclusion d'une *variable-d'état* agit sur la réponse de l'ordinateur en cas d'erreur d'I/O. Si une telle erreur survient quand la *variable d'état* est précisée, aucun message d'erreur n'est affiché et il est impossible de traiter l'erreur à l'aide de l'instruction ON ERROR. Si une erreur survient en l'absence de la *variable-d'état*, le message est affiché et l'erreur peut être traitée par ON ERROR.

Renvoi

ON ERROR

Exemple

```
140 CALL IO(1,1)
```

Ferme le périphérique 1 (l'opération CLOSE est le code de commande 1).

CHAPITRE V INDEX DE REFERENCE

SOUS-PROGRAMME

KEY

Syntaxe

CALL KEY(*variable-de-retour*, *variable-d'état*)

Description

Le sous-programme KEY affecte le code ASCII de la touche du clavier sur laquelle on a appuyé à la *variable-de-retour*. Si l'on n'a appuyé sur aucune touche, la *variable-de-retour* est mise égale à 255. Voir la liste des codes ASCII à l'annexe E.

La *variable-d'état* sert à mémoriser une valeur qui représente l'état de la touche sur laquelle on a appuyé. Une valeur 1 signifie que l'on a appuyé sur une nouvelle touche depuis la dernière exécution de CALL KEY. Une valeur - 1 signifie que l'on a appuyé sur la même touche que celle donnée dans le précédent CALL KEY. Une valeur 0 signifie qu'aucune touche n'a été actionnée.

Exemple

```
340 CALL KEY(K,S)
350 IF S=0 THEN 340
360 PRINT K;CHR$(K)
370 PAUSE
```

Affecte à K le code ASCII de la touche sur laquelle on a appuyé, et à S une valeur indiquant l'état de cette touche.

Le programme ci-dessous montre comment utiliser la fonction KEY pour s'assurer que les réponses à des questions successives correspondent à plusieurs frappes de touches et non à l'appui continu sur une même touche.

```
500 PRINT "AUTRES DONNEES? (O OU N)"
510 CALL KEY(K,S)
520 IF S < > 1 THEN 510
530 IF K=ASC("O") OR K=ASC("o") THEN PRINT
"ENTRER...":PAUSE 1 GOTO nnn, RENVOI A LA LIGNE
nnn
540 PRINT "FIN? (O ou N)"
550 CALL KEY(K,S)
560 IF S < > 1 THEN 550
570 IF K=ASC("N") OR K=ASC("n") THEN 500
```

CHAPITRE V INDEX DE REFERENCE

KEY\$

Syntaxe

KEY\$

Description

La fonction KEY\$ interrompt l'exécution d'un programme jusqu'à ce que l'on appuie sur une autre touche. A ce moment là, l'exécution du programme reprend et KEY\$ fournit la chaîne d'un caractère correspondant à la touche enfoncée. Consultez la liste des codes de caractères ASCII à l'annexe E.

Si on appuie sur [BREAK] alors que KEY\$ est en attente d'une réponse, l'interruption a lieu normalement.

Exemple

Le programme suivant se poursuit si l'on appuie sur O et s'arrête, si on appuie sur N.

```
100 PRINT "APPUYEZ SUR :":PAUSE 1:PRINT"O POUR
CONTINUER, N POUR ARRETER"
110 A$=KEY$
120 IF A$="O" OR A$="o" THEN 140
130 IF A$="N" OR A$="n" THEN 150 ELSE 110
140 PRINT "CONTINUER":PAUSE 1.5:GOTO 100
150 PRINT "Stop":PAUSE
```


LEN

Syntaxe

LEN(*expression-chaîne*)

Description

La fonction LEN donne le nombre de caractères d'une *expression-chaîne*. Un espace est considéré comme un caractère.

Exemples

```
170 PRINT LEN("ABCDE"):PAUSE
```

Imprime 5.

```
230 X=LEN("CECI EST UNE PHRASE.")
```

Donne X égal à 20.

```
910 DISPLAY LEN(""):PAUSE
```

Affiche 0.

LET

Syntaxe

[LET] *variable-numérique*[,*variable-numérique* ...] =
expression-numérique

[LET] *variable-chaîne* [,*variable-chaîne* ...] = *expression-chaîne*

Description

L'instruction LET affecte la valeur d'une expression à la/les variable(s) spécifiée(s). L'ordinateur évalue l'expression de droite et place le résultat dans la/les variable(s) de gauche. Si plus d'une variable est spécifiée, elles doivent être séparées par des virgules. Le mot LET est facultatif, il ne figure pas dans les exemples du présent manuel. Tous les indices de gauche sont évalués avant les affectations.

Exemples

```
110 LET T=4
```

Met T égal à 4.

```
170 X,Y,Z=12.4
```

Met X, Y et Z égaux à 12.4.

```
200 A=3 < 5
```

Met A égal à -1 puisqu'il est vrai que 3 est inférieur à 5.

```
350 L$,D$,B$="B"
```

Met L\$, D\$, et B\$ égaux à "B".

LINPUT

Syntaxe

LINPUT { [*message-guide* ;]*variable-chaine*
[#*numéro-de-fichier*, [REC *expression-numérique*,]]
variable-chaine }

Description

L'instruction LINPUT affecte la totalité d'un enregistrement d'entrée ou la partie restante d'un enregistrement d'entrée en attente dans une *variable-chaine*. Contrairement à INPUT, LINPUT n'interprète pas les données. Ainsi, tous les caractères, y compris les virgules, les espaces de début et de fin de ligne, les points-virgules et les guillemets sont placés dans la *variable-chaine*.

Le *message-guide* est une chaîne de caractères qui doit être suivie d'un point-virgule. S'il s'agit d'une constante-chaîne, elle doit figurer entre guillemets. Un *message-guide* est affiché en commençant à la position courante du curseur tel qu'il a été laissé par les précédentes instructions d'entrée sortie. En cas d'absence du *message-guide*, un point d'interrogation suivi d'un espace en tient lieu.

Le curseur clignotant s'affiche après le message. Si sa position dépasse alors 31, l'affichage est effacé et le curseur est ramené en colonne 1 avant que le message ne soit affiché. Quand le *message-guide* comporte plus de 30 caractères, il est tronqué à partir du 30^e.

LINPUT peut également servir à lire des données de type DISPLAY à partir d'un fichier ou d'un périphérique. Le *numéro-de-fichier* est celui d'un fichier ouvert. Si le fichier spécifié a une entrée en attente, la partie restante de l'enregistrement en attente est lue. Le message E7 Bad data (donnée incorrecte) s'affiche si l'enregistrement ou l'enregistrement partiel comporte plus de 255 caractères.

On peut utiliser la clause facultative REC avec des périphériques supportant des fichiers à accès direct. L'*expression-numérique* désigne l'enregistrement à saisir. Consultez le manuel du périphérique concerné pour plus d'informations sur les fichiers à accès direct. Remarquez que les fichiers relatifs ne peuvent être utilisés avec un magnétophone à cassette.

LINPUT

Renvoi

INPUT (avec fichiers), OPEN, CLOSE, PRINT, RESTORE

Exemples

300 LINPUT L\$

Entraîne l'affichage du message point d'interrogation et mémorise les données introduites dans L\$.

470 LINPUT "NOM: ";NM\$

Entraîne l'affichage de NOM: et mémorise les données introduites dans NM\$.

CHAPITRE V

INDEX DE REFERENCE

LIST

Syntaxe

LIST { [groupe-de-lignes]
["nom-de-périphérique"]
["nom-de-périphérique", groupe-de-lignes] }

Description

La commande LIST sert à lister les lignes d'un programme. Si le *groupe-de-lignes* ne figure pas, la totalité du programme est listée. Quand un *groupe-de-lignes* est mentionné, seules ces lignes sont listées. Le *groupe-de-lignes* peut être l'un des quelconques types de lignes ci-dessous.

Groupe-de-lignes	Effet
un numéro de ligne unique	Fait lister cette ligne.
numéro de ligne —	Fait lister cette ligne et les suivantes.
— numéro de ligne	Fait lister cette ligne et les précédentes.
numéro de ligne — numéro de ligne	Fait lister les lignes comprises entre ces numéros, bornes comprises.

Quand un *nom-de-périphérique* est mentionné, les lignes sont listées sur le périphérique désigné. Sinon, elles sont affichées sur la console et peuvent être modifiées au fur et à mesure. Pour interrompre un listage sur un périphérique, appuyez sur n'importe quelle touche et maintenez-la jusqu'à ce que le listage s'arrête. Une nouvelle pression sur la touche permet de poursuivre le listage. Pour terminer un listage, appuyez sur [BREAK], ou sur [↑] quand il s'agit de la console.

Exemples

LIST 100

Affiche la ligne 100.

LIST 100-200

Affiche toutes les lignes de 100 à 200.

LIST "50"

Liste tout un programme sur le périphérique 50.

LIST "50.R=C", -200

Liste toutes les lignes jusqu'à la ligne 200 comprise, sur le périphérique 50.

CHAPITRE V

INDEX DE REFERENCE

LN

Syntaxe

LN(expression-numérique)

Description

La fonction LN calcule le logarithme naturel (népérien) de l'*expression-numérique*. Celle-ci doit être supérieure à 0, sinon le message E23 Bad argument (argument incorrect) est affiché. La fonction LN est l'inverse de la fonction EXP.

Renvoi

EXP, LOG

Exemples

710 PRINT LN(3.4): PAUSE

Affiche le logarithme naturel de 3,4, c'est-à-dire 1.223775432.

850 X=LN(EXP(2.7))

Affecte à X le logarithme naturel de e élevé à la puissance 2,7; X est égal à 2,7.

910 S=LN(SQR(T))

Affecte à S le logarithme naturel de la racine carrée de T.

LOG

Syntaxe

LOG(*expression-numérique*)

Description

La fonction LOG calcule le logarithme ordinaire (à base 10) de l'*expression-numérique*. Celle-ci doit être supérieure à 0, sinon le message d'erreur E23 Bad argument (argument incorrect) est affiché.

Renvoi

LN

Exemples

```
150 PRINT LOG(3.4):PAUSE
```

Affiche le logarithme ordinaire de 3,4, c'est-à-dire 0.531478917

```
230 S=LOG(SQR(T))
```

Affecte à S le logarithme ordinaire de la racine carrée de T.

NEW

Syntaxe

NEW [ALL]

Description

La commande NEW prépare l'ordinateur à l'introduction d'un nouveau programme, en supprimant le programme et les variables déjà en mémoire. Tous les fichiers ouverts sont fermés.

La commande NEW ALL supprime le programme et les variables en mémoire, et réinitialise les touches utilisateur et efface les sous-programmes en langage assembleur. Elle annule aussi toute extension de mémoire ajoutée par CALL ADDMEM, efface tous les indicateurs d'affichage, sélectionne l'unité d'angle RAD et ferme tous les fichiers ouverts.

NEXT

Syntaxe

NEXT [*variable-de-commande*]

Description

L'instruction NEXT est toujours associée à l'instruction FOR TO STEP pour la construction d'une boucle. Si une *variable-de-commande* est mentionnée, elle doit être la même que celle de l'instruction FOR TO STEP. En l'absence de *variable-de-commande*, NEXT est associé à la plus récente instruction FOR TO STEP non appariée. Il est recommandé d'inclure une *variable-de-commande* dans chaque instruction NEXT.

Quand des boucles FOR TO STEP...NEXT sont imbriquées, l'instruction NEXT de la boucle interne doit figurer avant l'instruction NEXT de la boucle externe.

Voir la manipulation des boucles à l'instruction FOR TO STEP.

Renvoi

FOR TO STEP

Exemple

Le programme ci-dessous illustre l'utilisation de l'instruction NEXT. Les valeurs imprimées sont 30 et -2.

```
100 TOTAL=0
110 FOR COMPTE=10 TO 0 STEP -2
120 TOTAL=TOTAL+COMPTE
130 NEXT COMPTE
140 PRINT TOTAL,COMPTE:PAUSE
```

NUMBER

Syntaxe

NUMBER [*ligne-initiale*] [,*pas*]

Description

La commande NUMBER (ou NUM) génère des numéros de ligne en séquence. Ils sont affichés avec un espace pour faciliter l'introduction des lignes de programme. Il ne reste à taper que la/les instructions. Lorsqu'on a appuyé sur **[ENTER]**, la ligne est mémorisée et le numéro de ligne suivant est affiché.

Si la *ligne-initiale* et le *pas* ne sont pas spécifiés, les numéros de ligne commencent à 100 et augmentent de 10 en 10. Sinon, les lignes sont numérotées depuis la *ligne-initiale* et selon le *pas* spécifiés. Si une ligne existe déjà, elle est affichée et peut être remplacée ou modifiée au moyen des touches d'édition. Si le numéro de ligne est modifié, la séquence des numéros générés se poursuit à partir du nouveau numéro de ligne.

Pour terminer la numérotation, appuyez sur **[ENTER]** à la suite d'une ligne sans instruction, ou sur **[BREAK]** après l'affichage de n'importe quelle ligne.

Renvoi

RENUMBER

Exemples

NUM 110

Donne l'ordre à l'ordinateur de numéroté à partir de 110 par pas de 10.

NUM 105,5

Donne l'ordre à l'ordinateur de numéroté à partir de la ligne 105 par pas de 5.

NUMERIC

Syntaxe

NUMERIC(*expression-chaine*)

Description

La fonction NUMERIC détermine la validité de l'*expression-chaine* en tant que représentation d'une constante numérique. NUMERIC donne la valeur - 1 (vrai) si l'*expression-chaine* est une constante numérique autorisée, et 0 (faux) dans le cas contraire.

Les espaces de début et de fin de ligne de l'*expression-chaine* sont ignorés. On peut utiliser NUMERIC pour savoir si la fonction VAL agira correctement sur une chaîne supposée représenter un nombre.

Renvoi

VAL

Exemple

La séquence de programme suivante détermine si une introduction effectuée au clavier correspond à une constante numérique autorisée. Si ce n'est pas le cas, un message d'erreur est affiché jusqu'à la réintroduction de la donnée. S'il s'agit bien d'une constante numérique, celle-ci est affectée à la variable A.

```
100 LINPUT "INTRODUIRE LA VALEUR:";A$
110 IF NOT NUMERIC(A$) THEN LINPUT "ERREUR,
    REINTRODUIRE";A$:GOTO 110
120 A=VAL(A$)
```

OLD

Syntaxe

OLD "*périphérique.nom-de-fichier*"

Description

La commande OLD ferme tous les fichiers ouverts et retire le programme déjà en mémoire avant de charger le nouveau programme. On peut stocker un programme BASIC sur un *périphérique.nom-de-fichier* à l'aide de la commande SAVE.

Périphérique.nom-de-fichier identifie le périphérique sur lequel le programme est stocké ainsi que le nom du fichier. *Périphérique* correspond au numéro du périphérique, il se situe entre 1 et 255. Le *nom-de-fichier* désigne un fichier particulier. Consultez le manuel du périphérique correspondant pour le numéro de code de chacun d'entre eux ainsi que pour toute information spécifique concernant la forme du *nom-de-fichier*. Se reporter au chapitre 6 pour l'utilisation d'un magnétophone à cassette.

Nota : La commande OLD ne permet pas de récupérer les données d'un module. Il est également impossible de charger un fichier de données avec cette commande. Si le *nom-de fichier* spécifie un fichier de données plutôt qu'un fichier de programmes, il peut être nécessaire d'appuyer sur [RESET].

Renvoi

GET, INPUT (avec fichiers), PUT, SAVE, VERIFY

Exemples

OLD "1.PROGRAMME1"

Charge dans la mémoire de l'ordinateur le programme "PROGRAMME1" stocké dans le périphérique 1.

OLD "1."

Charge le fichier suivant dans la mémoire à partir du périphérique 1, à savoir le magnétophone à cassette. Si le fichier suivant n'est pas un fichier de programmes, l'ordinateur affiche une erreur.

ON BREAK

Syntaxe

ON BREAK STOP
ON BREAK NEXT
ON BREAK ERROR

Description

L'instruction ON BREAK détermine le comportement du programme en cas d'interruption. Après l'exécution de cette instruction, les interruptions sont traitées selon l'option sélectionnée.

ON BREAK STOP rétablit la fonction normale de BREAK, qui consiste à arrêter l'exécution du programme et à afficher le message standard d'interruption. Cette option est celle qui prévaut lors du lancement d'un programme.

Avec ON BREAK NEXT, les interruptions sont ignorées. A la rencontre d'un point d'arrêt précédant immédiatement un numéro de ligne, le point d'arrêt est ignoré et cette ligne de programme est exécutée. La touche **[BREAK]** est également ignorée. Cependant, une instruction BREAK sans *liste-de-numéros-de-ligne* arrête le programme, même si ON BREAK NEXT y figure. On peut également se servir de ON BREAK NEXT pour ignorer les points d'arrêt préalablement spécifiés dans un programme afin de détecter des erreurs.

Note : Dans la mesure où la touche **[BREAK]** est ignorée, il faut appuyer sur le bouton de réinitialisation pour arrêter un programme qui ne s'arrête pas normalement.

Avec ON BREAK ERROR, les interruptions sont traitées comme des erreurs, ce qui permet d'utiliser l'instruction ON ERROR pour y donner suite. Pour plus ample information, voir ON ERROR.

L'option ON BREAK reste effective jusqu'à ce qu'une autre instruction ON BREAK la modifie. A la fin d'un sous-programme, le mode ON BREAK, effectif lors de l'appel de ce sous-programme, redevient en fonction.

Renvoi

BREAK, ON ERROR

ON BREAK

Exemples

Le programme ci-dessous illustre l'utilisation de l'instruction ON BREAK. Quand le message W29 a t 120 B r e a k (interruption à la ligne 120) apparaît à l'affichage, reprendre l'exécution avec la commande CON.

```
100 BREAK 140
```

Pose un point d'arrêt à la ligne 140.

```
110 ON BREAK NEXT
```

Les interruptions seront ignorées.

```
120 BREAK
```

Un point d'arrêt intervient à la ligne 120 malgré l'instruction de la ligne 110. Appuyez sur **[CLR]** et tapez CONTINUE.

```
130 FOR A=1 TO 500
```

```
140 PRINT "(BREAK) HORS SERVICE"
```

```
150 NEXT A
```

La touche **[BREAK]** est sans effet pendant que les lignes 130 à 150 sont exécutées.

```
160 ON BREAK STOP
```

Restaure l'utilisation normale de **[BREAK]**.

```
170 FOR A=1 TO 500
```

```
180 PRINT "MAINTENANT, (BREAK) EN SERVICE"
```

```
190 NEXT A
```

La touche **[BREAK]** fonctionne de nouveau pendant que les lignes 170 à 190 sont exécutées.

ON ERROR

Syntaxe

ON ERROR STOP *numéro-de-ligne*

Description

L'instruction ON ERROR détermine le comportement du programme en cas d'erreur à l'exécution. Après l'exécution de l'instruction ON ERROR, toutes les erreurs sont traitées selon l'option sélectionnée.

ON ERROR STOP rétablit le traitement normal des erreurs, qui consiste à interrompre l'exécution du programme et à afficher un message explicatif. Cette option prévaut lors du lancement d'un programme.

ON ERROR *numéro-de-ligne* transfère le contrôle de l'exécution à la ligne désignée lorsque survient une erreur. Le *numéro-de-ligne* doit correspondre au début d'une sous-routine de traitement des erreurs. Lorsqu'une erreur s'est produite et que le contrôle a été transféré, le traitement des erreurs revient à ON ERROR STOP. Si l'option ON BREAK ERROR a été sélectionnée, elle est ramenée à ON BREAK NEXT. Une instruction ON ERROR *numéro-de-ligne* doit à nouveau être exécutée pour que toute nouvelle erreur soit traitée par la sous-routine appropriée.

L'option ON ERROR reste effective jusqu'à ce qu'une autre instruction ON ERROR la modifie. Si un sous-programme se termine sans erreur, le mode ON ERROR effectif lors de l'appel de ce programme redevient en fonction. Quand une erreur s'est produite dans un sous-programme, toute modification du mode de traitement des erreurs effectuée par la sous-routine correspondante, reste effective quand le sous-programme se termine.

Le programme principal et les sous-programmes peuvent partager la même sous-routine de traitement des erreurs. Les sous-routines appelées au moyen de GOSUB ne peuvent être partagées.

Renvoi

ON BREAK, ON WARNING, RETURN (avec ON ERROR)

ON ERROR

Exemple

Le programme ci-dessous illustre l'utilisation de l'instruction ON ERROR.

```
100 ON ERROR 150
```

En cas d'erreur, le contrôle passe à la ligne 150.

```
110 X$="A"
```

```
120 X=VAL(X$)
```

Provoque une erreur.

```
130 PRINT X;"CARRE:";X*X:PAUSE 2
```

```
140 STOP
```

```
150 REM SOUS-ROUTINE DE TRAITEMENT DES ERREURS
```

```
160 ON ERROR 220
```

L'erreur *suivante* sera traitée en ligne 220.

```
170 CALL ERR(CODE,TYPE,FILE,LINE)
```

Détermine l'erreur en utilisant CALL ERR.

```
180 IF LINE > 120 THEN RETURN 220
```

Passe en 220 si l'erreur n'est pas à la ligne prévue.

```
190 IF CODE > 29 THEN RETURN 220
```

Passe en 220 si l'erreur n'est pas celle prévue.

```
200 X$="5"
```

Affecte à X\$ une valeur acceptable.

```
210 RETURN
```

Revient à la ligne dans laquelle l'erreur s'est produite.

```
220 REM ERREUR INCONNUE
```

```
230 PRINT "ERREUR":CODE;"A LA LIGNE";LINE:PAUSE
```

Mentionne la nature de l'erreur imprévue et arrête le programme.

ON GOSUB

Syntaxe

ON *expression-numérique* GOSUB *numéro-de-ligne1*
[, *numéro-de-ligne*] [...]

Description

L'instruction ON GOSUB détermine quelle sous-routine utiliser, suivant la valeur de l'*expression-numérique*. Si la valeur de l'*expression-numérique* est 1, la sous-routine commençant au *numéro-de-ligne1* est exécutée, si la valeur est 2, la sous-routine commençant au *numéro-de-ligne2* est exécutée, et ainsi de suite. Chaque numéro de ligne doit faire référence à la première instruction d'une sous-routine. Si l'*expression-numérique* est 0, négative, ou supérieure au nombre de numéros de ligne, le message d'erreur E4 Bad value (valeur incorrecte) est affiché. Si l'*expression-numérique* est un nombre décimal, il est automatiquement arrondi.

Après l'exécution de l'instruction RETURN de la sous-routine, le contrôle revient à l'instruction succédant à ON GOSUB. On ne peut pas utiliser ON GOSUB pour se transférer à l'intérieur ou à l'extérieur d'un sous-programme ou en sortir.

Renvoi

GOSUB, RETURN (avec GOSUB)

Exemples

```
140 ON GOSUB 1000,2000,300
```

Transfère le contrôle aux lignes 1000 si X vaut 1, 2000 si X vaut 2, et 300 si X vaut 3.

```
240 ON P-4 GOSUB 200,250,300,800,170
```

Transfère le contrôle aux lignes 200 si P-4 vaut 1 (P vaut 5), 250 si P-4 vaut 2, 300 si P-4 vaut 3, 800 si P-4 vaut 4 et 170 si P-4 vaut 5.

ON GOTO

Syntaxe

ON *expression-numérique* GOTO *numéro-de-ligne1*
[,*numéro-de-ligne2*] [...]

Description

L'instruction ON GOTO détermine à quelle ligne transférer le contrôle, suivant la valeur de l'*expression-numérique*. Si la valeur de l'*expression-numérique* est 1, on passe au *numéro-de-ligne1*, si la valeur est 2, on passe au *numéro-de-ligne2*, et ainsi de suite. Si l'*expression-numérique* est 0, négative ou supérieure au nombre de numéros de ligne, le message d'erreur E4 Bad value (valeur incorrecte) est affiché. Si l'*expression-numérique* est un nombre décimal, il est automatiquement arrondi.

On ne peut pas utiliser ON GOTO pour se transférer à l'intérieur d'un sous-programme, ou en sortir.

Renvoi

GOTO

Exemples

```
130 ON X GOTO 1000,2000,300
```

Transfère le contrôle aux lignes 1000 si X vaut 1, 2000 si X vaut 2, et 300 si X vaut 3. L'instruction équivalente en cas d'utilisation de IF THEN ELSE est 130 IF X=1 THEN 1000 ELSE IF X=2 THEN 2000 ELSE IF X=3 THEN 300 ELSE PRINT "Bad value" : PAUSE : STOP, qui comporte plus de 80 caractères.

```
21 ON P-4 GOTO 200,250,300,800,170
```

Transfère le contrôle aux lignes 200 si P-4 vaut 1 (P vaut 5), 250 si P-4 vaut 2, 300 si P-4 vaut 3, 800 si P-4 vaut 4, et 170 si P-4 vaut 5.

ON WARNING

Syntaxe

ON WARNING PRINT
ON WARNING NEXT
ON WARNING ERROR

Description

L'instruction ON WARNING détermine le comportement du programme en cas d'avertissement à l'exécution. Après l'exécution de l'instruction ON WARNING, tout avertissement est traité selon l'option ON WARNING sélectionnée.

ON WARNING PRINT rétablit l'utilisation normale des avertissements qui consiste à imprimer des messages d'avertissements explicatifs et à continuer l'exécution du programme après que l'on ait appuyé sur les touches [ENTER] ou [CLR]. Cette option prévaut lors du lancement d'un programme.

Avec ON WARNING NEXT, l'exécution du programme continue sans l'impression d'aucun message.

Avec ON WARNING ERROR, un avertissement est traité comme une erreur, ce qui permet d'y donner suite à l'aide d'instructions ON ERROR.

L'option ON WARNING reste effective jusqu'à ce qu'une autre instruction ON WARNING la modifie. Quand un sous-programme se termine, le mode ON WARNING effectif lors de l'appel de ce sous-programme, redevient en fonction.

Renvoi

ON ERROR

Exemples

Le programme ci-dessous illustre l'utilisation de ON WARNING.

100 ON WARNING NEXT

Le traitement des avertissements consiste à passer à l'instruction suivante.

110 PRINT 110,5/0:PAUSE

Affiche le résultat sans aucun message.

ON WARNING

120 ON WARNING PRINT

Rétablit le traitement normal des avertissements, qui consiste à imprimer un message et à permettre la poursuite ultérieure de l'exécution.

130 PRINT 130,5/0:PAUSE

Affiche l'avertissement. Quand on appuie sur [ENTER] ou [CLR], il s'affiche 130 suivi de la valeur de 5/0.

140 ON WARNING ERROR

Les avertissements doivent être traités comme des erreurs.

150 PRINT 150,5/0:PAUSE

Affiche un message et traite l'avertissement comme une erreur.

160 PRINT 160:PAUSE

N'est pas exécutée car l'exécution s'arrête à la ligne 150.

Nota : En effaçant le message d'erreur, vous effacez en même temps l'affichage. Néanmoins, pour voir la valeur 150 affichée, vous pouvez appuyer sur [SHIFT] [PB].

OPEN

Syntaxe

OPEN #*numéro-de-fichier*, "*périphérique.nom-de-fichier*"
[,*organisation-de-fichier*] [,*type-de-fichier*]
[,*longueur-d'enregistrement*] [,*mode-d'ouverture*]

Description

L'instruction OPEN permet d'utiliser, dans un programme BASIC, des fichiers de données et des périphériques en fournissant un lien entre le *numéro-de-fichier* et un fichier ou un périphérique déterminé. En établissant ce lien, l'instruction OPEN spécifie comment le fichier ou le périphérique peut être utilisé (pour une entrée ou une sortie) et comment le fichier est organisé.

L'instruction OPEN doit être exécutée avant toute instruction BASIC essayant d'utiliser un fichier ou un périphérique nécessitant un numéro de fichier. Si une instruction OPEN fait référence à un fichier existant déjà, les attributs *organisation-de-fichier*, *type-de-fichier*, et *longueur-d'enregistrement* doivent être les mêmes que ceux du fichier. Si une instruction OPEN fait référence à un fichier déjà ouvert, une erreur se produit.

Le *numéro-de-fichier* est un nombre compris entre 1 et 255 que l'instruction OPEN associe à un fichier ou à un périphérique. Ce *numéro-de-fichier* sert à toutes les instructions d'entrée/sortie qui accèdent le fichier ou le périphérique. Le *numéro-de-fichier* est arrondi au nombre entier le plus proche. Le *numéro-de-fichier* 0 correspond au clavier et à l'affichage de l'ordinateur. On ne peut l'utiliser pour d'autres fichiers et il est toujours ouvert.

Le *périphérique.nom-de-fichier* représente un numéro de périphérique réel et d'autres informations concernant le périphérique. Le *périphérique.nom-de-fichier* peut être une expression-chaîne. Le *périphérique* est le numéro de l'unité physique et, et peut se situer entre 1 et 255. Le *nom-de-fichier* donne des informations au périphérique pour traiter l'instruction OPEN. Par exemple, avec une unité de stockage externe, le *nom-de-fichier* précise le nom du fichier. Avec d'autres périphériques, le *nom-de-fichier* spécifie des options telles que la parité, le classement des données, etc. Consultez les manuels des périphériques pour connaître le code de chacun d'entre eux et obtenir des informations spécifiques concernant la forme du *nom-de-fichier*.

OPEN

Attributs de fichier

Les attributs de fichier énumérés ci-dessous peuvent se présenter dans n'importe quel ordre ou être absents. En l'absence d'un attribut, des valeurs par défaut sont utilisées.

L'*organisation-de-fichier* peut être SEQUENTIAL ou RELATIVE. Les enregistrements dans un fichier séquentiel sont lus ou écrits en séquence, du début à la fin. Les enregistrements dans un fichier RELATIVE (d'accès direct) peuvent être lus ou écrits dans n'importe quel ordre, y compris séquentiel. En l'absence d'*organisation-de-fichier*, l'organisation est supposée séquentielle, on précisera donc RELATIVE pour des fichiers d'accès direct.

Le *type-de-fichier* peut être DISPLAY ou INTERNAL. DISPLAY spécifie que la donnée est écrite en format ASCII. INTERNAL spécifie que la donnée est écrite en format binaire. Les enregistrements binaires occupent moins d'espace, sont traités plus rapidement par l'ordinateur et sont plus efficaces pour l'enregistrement de données sur des unités de stockage externe. Cependant, on doit utiliser le format DISPLAY si les informations doivent être imprimées ou affichées pour être lues. En l'absence du *type-de-fichier*, le mode DISPLAY est en service.

La *longueur-d'enregistrement* est composée du mot VARIABLE suivi d'une expression numérique précisant au fichier la longueur maximale de l'enregistrement. La longueur d'enregistrement maximale tolérée dépend du périphérique utilisé. En l'absence de la longueur d'enregistrement, l'unité périphérique spécifie une *longueur-d'enregistrement* par défaut.

Le *mode-d'ouverture* indique à l'ordinateur de traiter le fichier selon le mode UPDATE, INPUT, OUTPUT ou APPEND. UPDATE spécifie que les données peuvent être lues ou écrites dans celui-ci. INPUT spécifie que les données ne peuvent qu'être lues. OUTPUT spécifie que les données ne peuvent qu'être écrites dans le fichier. APPEND spécifie que les données ne peuvent être écrites qu'à la fin du fichier. En l'absence du *mode-d'ouverture*, le mode UPDATE est en fonction.

Nota : Si un fichier existe déjà sur un périphérique externe, l'emploi du mode OUTPUT a pour résultat la surimpression des nouvelles données sur les données existantes.

OPEN

Ouverture de fichiers sur un magnétophone à cassette

L'interface cassette est appelée périphérique 1 dans une instruction OPEN. Les attributs d'un fichier magnétophone à cassette sont les suivants.

- Les fichiers doivent être séquentiels.
- La longueur d'enregistrement par défaut est de 256 octets.
- Les fichiers doivent disposer d'un *mode-d'ouverture* INPUT ou OUTPUT.
- Le *type-de-fichier* doit être DISPLAY.

Nota : Lorsque des fichiers se trouvent sur un magnétophone à cassette, vous ne pouvez pas utiliser les commandes RESTORE ou DELETE, le *mode-d'ouverture* APPEND, ni le *type-de-fichier* INTERNAL.

Se reporter au chapitre 6 sur l'utilisation d'un magnétophone à cassette pour le stockage et la récupération de fichiers.

Renvoi

CLOSE, DELETE, INPUT, LINPUT, PRINT, RESTORE

Exemples

```
100 OPEN #23, "2.X", INTERNAL, UPDATE
```

Ouvre le fichier appelé "X" du périphérique 2 et permet à n'importe quelle instruction d'entrée/sortie d'accéder au fichier en utilisant le numéro 23. Le type de fichier est INTERNAL. Le fichier étant ouvert selon le mode UPDATE, les données peuvent être lues du fichier et écrites vers celui-ci.

```
150 OPEN #243, A$&" .ABC", INTERNAL
```

Si A\$ est égal à "2", cette ligne ouvre le fichier numéro 2 et le nomme ABC. Le type de fichier est INTERNAL, le mode UPDATE est en fonction, et le périphérique spécifie la longueur de l'enregistrement par défaut.

```
200 OPEN #1, "1.DATA1", DISPLAY, OUTPUT
```

Ouvre le fichier appelé "DATA1" sur un magnétophone à cassette. Le type de fichier est DISPLAY. Le fichier étant ouvert selon le mode OUTPUT, les données ne peuvent être qu'écrites.

PAUSE

Syntaxe

PAUSE [*expression-numérique*]

PAUSE

PAUSE ALL

Description

L'instruction PAUSE interrompt l'exécution d'un programme soit pendant un nombre de secondes spécifié soit jusqu'à ce que l'on appuie sur [CLR] ou [ENTER]. En l'absence d'*expression-numérique*, le curseur souligné s'affiche en colonne 1 pour indiquer qu'une pause indéfinie est en cours. Les touches spéciales d'édition peuvent alors servir à visionner le contenu d'une ligne de 80 colonnes de la ligne. L'exécution du programme reprend quand on appuie sur [ENTER] ou [CLR].

Avec une *expression-numérique*, PAUSE suspend l'exécution du programme pendant un nombre de secondes correspondant à la valeur absolue de l'*expression-numérique*. Si l'*expression-numérique* est positive, on peut écourter le temps de pause en appuyant sur [ENTER] ou [CLR]. Si elle est négative, le temps de pause ne peut être écourté. Le temps de résolution est d'environ un dixième de seconde. Si l'*expression-numérique* est inférieure à 0,1, le programme ne marque pas de pause. Lors d'une pause limitée, le curseur n'est pas affiché et l'affichage ne peut pas défiler.

Pause (sans *expression-numérique* ni ALL) suspend l'exécution pendant une durée indéfinie. Le curseur est affiché en colonne 1 pour indiquer qu'il se produit une pause indéfinie. Les touches de commande du curseur peuvent alors être utilisées pour visualiser la ligne complète. L'exécution reprend lorsqu'on appuie sur [ENTER] ou [CLR].

L'instruction PAUSE ALL suspend l'exécution d'un programme chaque fois qu'une ligne complète de sortie est affichée. L'exécution reprend lorsqu'on appuie sur [CLR] ou [ENTER]. PAUSE ALL reste en fonction jusqu'à ce qu'une PAUSE limitée de durée zéro soit exécutée.

PAUSE ALL reste en fonction quand un sous-programme est appelé. Si PAUSE ALL est modifiée dans un sous-programme, elle est remise en fonction quand le sous-programme se termine.

PAUSE

Renvoi

DISPLAY, PRINT

Exemples

120 PAUSE 2.2

Interrompt l'exécution pendant 2,2 secondes jusqu'à ce que l'on appuie sur [CLR] ou [ENTER].

190 PAUSE

Interrompt l'exécution jusqu'à ce que l'on appuie sur [CLR] ou [ENTER].

Le programme suivant convertit des degrés Fahrenheit en degrés Celsius.

100 PRINT "INTRODUIRE DEG: ";

Affiche le message INTRODUIRE DEG: . Le message en attente créé par le point-virgule à la fin de l'instruction PRINT entraîne l'affichage du message jusqu'à l'introduction d'une donnée.

110 ACCEPT DG

120 PRINT DG;"DEG F=";(DG-32)*5/9;
"DEGRES C":PAUSE

Affiche la réponse. L'instruction PAUSE qui succède à l'instruction PRINT entraîne l'affichage de la réponse jusqu'à ce que l'on appuie sur [ENTER] ou [CLR].

PI

Syntaxe

PI

Description

La fonction PI donne la valeur de pi, c'est-à-dire, 3,14159265359.

Exemple

130 VOLUME=4/3*PI*R^3

Affecte à VOLUME les quatre tiers de PI x par le rayon élevé au cube, ce qui représente le volume d'une sphère de rayon R.

CHAPITRE V INDEX DE REFERENCE

POS

Syntaxe

POS(*chaîne1*, *chaîne2*, *expression-numérique*)

Description

La fonction POS donne la position de la première apparition de la *chaîne2* dans la *chaîne1*. La recherche commence à la position spécifiée par l'*expression-numérique*. Si aucune coïncidence n'est trouvée, la fonction vaut zéro.

Exemples

```
110 X=POS("PAN","A",1)
```

Met X égal à 2 parce que A est la seconde lettre de PAN.

```
140 Y=POS("APAN","A",2)
```

Met X égal à 3 parce que le A en troisième position dans APAN est la première apparition de cette lettre dans la portion examinée.

```
170 Z=POS("PAN","A",3)
```

Met Z égal à zéro parce que A n'est pas dans la partie de PAN examinée.

```
290 R=POS("PABNAN","AN",1)
```

Met R égal à 5 parce que la première apparition de AN commence avec le A à la cinquième position dans PABNAN.

CHAPITRE V INDEX DE REFERENCE

PRINT

A L'AFFICHAGE

Syntaxe

PRINT [USING *numéro-de-ligne*
expression-chaîne'] [*liste*]

Description

L'instruction PRINT sert à conférer un format à des données et à les afficher. On utilise USING pour spécifier le format de chaque élément de la *liste*. Reportez-vous à IMAGE et USING pour connaître la définition d'un format et ses effets sur l'instruction PRINT. En l'absence de *liste*, l'instruction PRINT efface l'affichage.

La *liste* comporte les éléments à afficher et leurs séparateurs. Les éléments sont des expressions numériques et des expressions-chaînes dont les valeurs sont à afficher, ainsi que des fonctions TAB qui commandent la disposition de l'affichage.

Les séparateurs sont des virgules ou des points-virgules qui spécifient la position relative des éléments à afficher.

Éléments affichés

Lors de l'impression d'une instruction PRINT, les valeurs des expressions de la *liste* sont affichées dans l'ordre, de gauche à droite aux positions déterminées par les séparateurs et les fonctions TAB.

- Les *expressions-chaîne* sont évaluées de façon à produire un résultat sous forme de chaîne. Les constantes-chaînes doivent être entre guillemets. Les espaces de début et de fin sont ignorés. Pour afficher un espace avant ou après une chaîne, faites-le figurer dans la chaîne ou insérez-le séparément avec des guillemets.
- Les *expressions-numériques* sont évaluées et affichées avec un espace à la suite. Les valeurs positives sont précédées d'un espace (remplaçant le signe plus) et les nombres négatifs sont précédés du signe moins.
- La fonction TAB spécifie la position de départ pour l'affichage du prochain élément de la *liste*. Reportez-vous à la section TAB pour plus d'informations.

Séparateurs

Vous devez placer au moins un séparateur entre chaque élément à afficher. Les séparateurs multiples placés à l'intérieur d'une instruction PRINT sont évalués de gauche à droite.

- Le point-virgule affiche le prochain élément de la *liste* immédiatement après le précédent, sans espace supplémentaire entre les valeurs.
- La virgule imprime le prochain élément au début du champ suivant. Les champs d'impression comportent 15 caractères et se trouvent aux colonnes 1, 16, 31, 46, 61 et 76 pour une ligne de 80 colonnes. Si la position courante se situe au-delà du début du dernier champ, la virgule provoque l'affichage du prochain élément à la ligne suivante.

Si un élément est plus long que le reste de la ligne courante, il est affiché à partir du début de la ligne suivante. Si une valeur numérique peut entrer dans la ligne courante sans son espace de fin de ligne, elle est affichée sur cette ligne. Si un élément comporte plus de 80 caractères, les 80 premiers caractères sont imprimés sur une ligne et les autres sur les lignes suivantes, par 80 caractères.

Affichage en attente

Si la *liste* n'est pas suivie d'une virgule ou d'un point virgule, la partie restante de la ligne de 80 colonnes est effacée. En conséquence, la prochaine instruction d'entrée/sortie doit commencer sur une nouvelle ligne.

L'utilisation de virgules et de points-virgules en fin de *liste* crée un affichage en attente, qui empêche l'effacement de la partie restante de la ligne. Le système introduit des espaces jusqu'au début du prochain champ si l'instruction PRINT est terminée par une virgule, ou rien si l'instruction est terminée par un point-virgule. La prochaine instruction d'I/O affiche ou accepte les données en partant de la position courante ainsi définie, sauf si l'instruction modifie cette position.

Un affichage en attente peut servir à générer des messages d'entrée pour les instructions ACCEPT, INPUT (à l'affichage) ou LINPUT. La prochaine instruction INPUT ou LINPUT place son propre message guide à la suite. consultez les sections ACCEPT, INPUT (à l'affichage) et LINPUT pour plus d'informations.

Formats numériques

Les nombres sont affichés soit sous la forme décimale normale, soit en notation scientifique. On utilise la notation scientifique quand elle permet d'afficher un plus grand nombre de chiffres significatifs.

Quand un nombre est affiché sous forme décimale normale, les conventions suivantes sont observées :

- Les nombres entiers sont affichés sans point décimal.
- Les nombres non entiers sont affichés avec un point décimal. Les zéros qui suivent la partie fractionnaire sont ignorés. Si le nombre comporte plus de dix chiffres significatifs, la valeur est arrondie à dix chiffres.
- Un nombre dont la valeur absolue est inférieure à 1 est affiché sans zéro à gauche du point décimal.

Un nombre affiché en notation scientifique se présente sous la forme suivante : mantisse E exposant.

Quand un nombre est affiché en notation scientifique, les conventions suivantes sont observées :

- La mantisse est affichée avec 7 chiffres ou moins dont un à la gauche du point décimal.
- Les zéros qui suivent la partie fractionnaire de la mantisse sont ignorés.
- L'exposant est affiché avec un signe plus ou moins suivi d'un nombre de deux ou trois chiffres.
- Quand l'exposant comporte deux chiffres, la mantisse est limitée à sept chiffres. Quand il comporte trois chiffres, la mantisse est limitée à six. Si nécessaire, la mantisse est arrondie au nombre de chiffres approprié.

Renvoi

ACCEPT, DISPLAY, IMAGE, INPUT, LINPUT, PAUSE, TAB, USING

Exemples

```
100 PRINT
```

Affiche une ligne vierge.

```
210 PRINT "LA REPONSE EST";A:PAUSE
```

Affiche LA REPONSE EST immédiatement suivi de la valeur de A.

```
320 PRINT X,Y/2:PAUSE
```

Affiche la valeur de X, puis la valeur de Y/2 dans le champ suivant.

```
450 PRINT "NOM:";
```

```
460 ACCEPT N$
```

Imprime NOM : et accepte l'introduction après ce message-guide.

Syntaxe

PRINT #*numéro-de-fichier* [,REC *expression-numérique*]

[,USING *numéro-de-ligne*
expression-de-chaîne [,*liste*]

Description

L'instruction PRINT sert à conférer un format à des données et à les écrire sur un fichier ou un périphérique. Le *numéro-de-fichier* est un nombre de 0 à 255 qui désigne un fichier ou un périphérique ouvert. Le fichier doit avoir été ouvert selon un mode OUTPUT, UPDATE, ou APPEND. Le *numéro-de-fichier* 0 désigne l'affichage, qui est toujours ouvert. Le *numéro-de-fichier* est arrondi au nombre entier le plus proche.

REC *expression-numérique* ne peut apparaître que lorsque le *numéro-de-fichier* fait référence à un fichier à accès direct (en mode RELATIVE). Consultez le chapitre 6 et les manuels des périphériques pour obtenir des informations sur les fichiers en mode d'enregistrement RELATIVE et l'utilisation correcte de REC. L'*expression-numérique* est évaluée pour désigner le numéro d'enregistrement dans lequel on veut écrire.

On peut utiliser USING pour spécifier le format exact des données dans un fichier de type DISPLAY. Consultez les sections IMAGE et USING pour la définition du format et ses effets sur l'instruction PRINT. Le fait d'utiliser USING avec un fichier de données de type INTERNAL, provoque une erreur.

La *liste* comporte les éléments à écrire et leurs séparateurs. Les éléments sont des expressions numériques et des expressions-chaînes dont les valeurs sont à écrire, ainsi que des fonctions TAB qui commandent la disposition de l'écriture. Les séparateurs sont des virgules et des points-virgules qui indiquent la position relative des éléments écrits.

La *liste* est interprétée dans l'ordre, de gauche à droite. La forme du résultat dépend du type de fichier ou de périphérique (DISPLAY ou INTERNAL). Consultez OPEN pour avoir une description du *type-de-fichier*.

En l'absence de *liste* et lorsqu'il n'y a pas d'enregistrement en attente, le résultat dépend du type de fichier. S'il s'agit du type DISPLAY, l'instruction PRINT écrit un enregistrement blanc (longueur zéro). Si le fichier est de type INTERNAL, une erreur se produit du fait que ce type de fichier n'admet pas d'enregistrement de longueur zéro.

Fichiers de type DISPLAY

Lors de l'exécution d'une instruction PRINT faisant référence à un fichier ou un périphérique de type DISPLAY, la *liste* est évaluée de la façon suivante :

- Les *expressions-chaîne* sont évaluées de façon à produire un résultat sous forme de chaîne. Les constantes-chaînes doivent être entre guillemets. Les espaces de début et de fin sont ignorés. Pour écrire un espace avant et après une chaîne, faites-le figurer à l'intérieur de la chaîne ou insérez-le avec des guillemets.
- Les *expressions-numériques* sont évaluées et écrites avec un espace à la suite. Les valeurs positives sont précédées d'un espace (remplaçant le signe plus), et les nombres négatifs sont précédés du signe moins.
- La fonction TAB spécifie la position de début dans la ligne pour le prochain élément de la *liste*. Consultez TAB pour plus d'informations.

Vous devez placer au moins un séparateur entre les éléments à écrire. Les séparateurs multiples placés à l'intérieur d'une instruction PRINT sont évalués de gauche à droite.

- Le point-virgule écrit le prochain élément de la *liste* immédiatement après le précédent, sans espace supplémentaire.
- La virgule écrit le prochain élément au début du prochain champ. Les champs comportent 15 caractères et se trouvent aux colonnes 1, 16, 31, et ainsi de suite. Si la position courante de la colonne se situe au delà du début du dernier champ, la virgule provoque l'écriture du prochain élément dans l'enregistrement suivant.

Si un élément est plus long que la partie restante de l'enregistrement courant, il est écrit au début de l'enregistrement suivant. Si une valeur numérique peut entrer dans l'enregistrement courant sans son espace final, elle est écrite dans cet enregistrement. Si un élément est plus grand que la longueur d'enregistrement, il est divisé en segments d'une longueur égale à la longueur d'enregistrement, jusqu'à ce que le dernier segment soit d'une longueur égale ou inférieure. Les segments sont alors écrits dans les enregistrements successifs correspondants.

Fichiers de type INTERNAL

Lors de l'exécution de l'instruction PRINT faisant référence à un fichier ou à un périphérique de type INTERNAL, la *liste* est évaluée de la façon suivante :

- Les expressions-chaînes sont évaluées et écrites dans l'enregistrement sous forme de chaîne INTERNAL.
- Les expressions numériques sont évaluées et écrites dans l'enregistrement sous forme numérique INTERNAL.
- La fonction TAB provoque une erreur si on l'utilise avec un fichier de type INTERNAL.

Vous devez placer au moins un séparateur entre chaque élément à écrire. Les séparateurs d'impression multiples d'une instruction PRINT sont évalués de gauche à droite.

- Le point-virgule imprime le prochain élément de la *liste* immédiatement après le précédent, sans espace supplémentaire entre les valeurs.
- La virgule fonctionne de la même façon que le point-virgule.

Si l'élément à écrire est plus long que la partie restante de l'enregistrement courant, il est écrit au début de l'enregistrement suivant. Si l'élément est d'une longueur supérieure à celle de l'enregistrement, il se produit une erreur.

Ecriture en attente

Si la *liste* se termine sans virgule ni point-virgule, l'enregistrement est immédiatement inclus dans le fichier. La prochaine instruction d'entrée/sortie qui accède au fichier commence un nouvel enregistrement.

CHAPITRE V INDEX DE REFERENCE

AVEC FICHIERS

PRINT

L'utilisation d'une virgule ou d'un point virgule après la *liste* crée une écriture en attente. Dans ce cas, l'enregistrement courant n'est pas inclus dans le fichier. Le système introduit des espaces jusqu'au début du prochain champ si une virgule termine l'instruction PRINT, et rien du tout si un point-virgule termine cette instruction. La prochaine instruction de sortie qui accède à ce fichier écrit les données sur ce même enregistrement, en commençant à la position courante ainsi définie, à moins que l'instruction ne modifie cette position.

En l'absence d'une *liste* s'il y a un enregistrement de sortie en attente, l'instruction PRINT inclus cet enregistrement dans le fichier. S'il n'y a pas d'enregistrement en attente, le résultat dépend du type de fichier. Si le fichier est de type DISPLAY, l'instruction PRINT écrit un enregistrement vierge (de longueur zéro). Si le fichier est de type INTERNAL, une erreur se produit étant donné que ce type de fichier n'admet pas d'enregistrement de longueur zéro.

Renvoi

IMAGE, INPUT (avec fichiers), OPEN, TAB, USING

Exemples

```
150 PRINT #32,A,B,C,
```

Entraîne l'écriture des valeurs A, B, et C dans le prochain enregistrement du fichier ouvert sous le numéro 32. La virgule finale crée une condition d'écriture en attente. La prochaine instruction PRINT accédant au fichier #32 écrira dans le même enregistrement.

Le programme suivant écrit des données dans un fichier.

```
100 OPEN #5,"1.MONFICHIER",DISPLAY,OUTPUT
```

Ouvre le fichier numéro 5. MONFICHIER est créé s'il n'existe pas déjà sur le périphérique numéro 1.

```
110 DIM A(50)
```

Dimensionne un tableau pour 51 valeurs.

CHAPITRE V INDEX DE REFERENCE

AVEC FICHIERS

PRINT

```
120 B=0
```

Initialise le total de contrôle.

```
130 FOR J=1 TO 50
```

Les lignes 130 à 180 commandent l'introduction des données.

```
140 PRINT "INTRODUIRE VALEUR";
```

```
150 ACCEPT A(J)
```

```
160 B=B+A(J)
```

```
170 PRINT #5,A(J);
```

La valeur de A(J) est écrite dans le fichier.

```
180 NEXT J
```

```
190 PRINT #5,B
```

La valeur du total de contrôle est écrite dans le fichier.

```
200 CLOSE #5
```


CHAPITRE V INDEX DE REFERENCE

SOUS-PROGRAMME

PUT

Syntaxe

CALL PUT (*numéro-d'image*)

Description

Le sous-programme PUT mémorise une copie de l'image RAM système dans le module *Extension Mémoire*. Le terme "image" s'applique à tous les contenus de la mémoire RAM 8K, y compris les lignes de programme, les variables et l'espace inutilisé.

Le *numéro-d'image* peut être 1 ou -1. Avec le numéro 1, l'image mémoire est copiée dans le module, et avec le numéro -1, les images mémoire sont échangées. Cette option permet de stocker le programme à partir de la mémoire pendant que vous récupérez un programme en module.

Au fur et à mesure que le contenu du module est remplacé par la mémoire, l'ordinateur vérifie que le module contient bien une image de la mémoire système. Si ce n'est pas le cas, l'ordinateur renvoie un message d'erreur.

Renvoi

GET, ADDMEM

Exemples

CALL PUT(1)

Copie l'image de RAM système dans le module.

CALL PUT(-1)

Remplace l'image du module par la mémoire système RAM.

CHAPITRE V INDEX DE REFERENCE

RAD

Syntaxe

RAD

Description

L'instruction RAD sélectionne le radian comme unité d'angle. Après sélection du mode RAD, toutes les données angulaires sont exprimées en radians. La sélection du mode RAD est automatique après introduction de NEW ALL ou réinitialisation de l'ordinateur.

Renvoi

DEG, GRAD

Exemples

100 RAD

Sélectionne l'unité d'angle RAD.

200 RAD:PRINT COS(PI/2):PAUSE

Affiche 0 (le cosinus de $\pi/2$ radians).

RANDOMIZE

Syntaxe

RANDOMIZE (*expression-numérique*)

Description

L'instruction RANDOMIZE réinitialise le générateur de nombres aléatoires, en vue d'une nouvelle séquence imprévisible.

Si RANDOMIZE est suivi d'une *expression-numérique*, la même séquence de nombres aléatoires est générée à chaque exécution de l'instruction avec cette valeur. Différentes valeurs entraînent différentes séquences.

Renvoi

RND

Exemple

Le programme ci-dessous illustre l'utilisation de l'instruction RANDOMIZE. Il accepte une valeur comme *expression-numérique* et imprime les dix premiers nombres obtenus à l'aide de la fonction RND. Appuyez sur **[BREAK]** pour arrêter le programme.

```
100 INPUT "BASE: ";S
110 RANDOMIZE S
120 FOR A=1 TO 10:PRINT A;RND:PAUSE 1.1
130 NEXT A
140 GOTO 100
```

READ

Syntaxe

READ *liste-de-variables*

Description

L'instruction READ est utilisée avec l'instruction DATA pour affecter des valeurs à des variables. La *liste-de-variables* se compose de variables-chaînes et de variables numériques, indicées ou non indicées, séparées par des virgules. La valeur lue dans l'instruction DATA doit correspondre au type de variables à laquelle elle est affectée dans l'instruction READ. Note : tout nombre est une chaîne autorisée. Quand deux virgules contigües figurent dans DATA, une chaîne vide est lue.

L'instruction READ commence la lecture depuis la première instruction DATA du programme ou sous-programme courant, et continue avec la suivante quand ses données sont épuisées. La même instruction READ peut lire dans plusieurs instructions DATA, et plusieurs instructions READ peuvent lire dans la même instruction DATA. Si un READ n'épuise pas toutes les données d'une instruction DATA, le READ suivant commence par la première donnée non lue. Toute tentative de lire des données après que toutes celles du programme ou sous-programme courant aient été épuisées, entraîne une erreur.

On peut utiliser l'instruction RESTORE pour modifier l'ordre dans lequel les instructions DATA sont exploitées.

READ ne peut lire des données qu'à partir de l'instruction DATA située dans le même programme ou sous-programme. A chaque appel d'un sous-programme, les données sont lues à partir de la première instruction DATA.

Renvoi

DATA, RESTORE

REM

Syntaxe

REM [*chaîne-de-caractères*]

Description

L'instruction REM vous permet d'introduire des remarques explicatives dans un programme. Les remarques peuvent donner toutes sortes d'informations, mais habituellement elles expliquent le fonctionnement d'une partie de programme. La *chaîne-de-caractères* peut comporter n'importe quel caractère affichable.

Les remarques ne sont pas exécutées, mais elles prennent de la place en mémoire. Tout caractère succédant à REM, y compris le symbole séparateur d'instructions (:) est considéré comme faisant partie de la remarque. En conséquence, si REM se trouve dans une ligne à instructions multiples, elle doit être la dernière instruction de la ligne.

Le point d'exclamation (!) est appelé symbole de remarque final. On peut l'utiliser au lieu du mot REM. Le point d'exclamation peut figurer en tant que première instruction d'une ligne, ou après la dernière instruction d'une ligne à instructions multiples. Si le point d'exclamation figure après une instruction, le séparateur d'instructions (:) n'est pas nécessaire. L'utilisation du symbole de remarque final économise de la place dans le listage du programme, aussi bien que sous sa forme interne.

Exemple

```
150 REM DEBUT SOUS-ROUTINE
```

Identifie le point de départ d'une sous-routine.

```
270 SUBTOTAL=L+B ! Calculer sous-total
```

Identifie les instructions effectuant un calcul spécifique.

RENUMBER

Syntaxe

RENUMBER [*ligne-initiale*] [,*pas*]

Description

La commande RENUMBER (ou REN) change les numéros de lignes d'un programme. En l'absence de *ligne-initiale*, la nouvelle numérotation commence à 100. Si le *pas* n'est pas précisé, un *pas* de 10 est automatiquement appliqué.

REN modifie également toutes les références aux numéros de lignes, de façon à ce qu'ils désignent les mêmes lignes que précédemment. Si une instruction désigne une ligne qui n'existe pas, un message d'avertissement est affiché et le numéro de ligne est remplacé par 32767, qui n'est pas un numéro de ligne autorisé.

Si les valeurs introduites pour la *ligne-initiale* et le *pas* ont pour résultat la création de numéros de lignes supérieurs à 32766, un message d'erreur E11 Line number error (numéro de ligne incorrect) est affiché et le programme reste inchangé.

Renvoi

NUMBER

Exemples

```
REN
```

Re-numérote toutes les lignes en partant de 100, avec un incrément par 10.

```
REN ,100
```

Re-numérote toutes les lignes en partant de 100, avec un incrément par 100

```
REN 10000,5
```

Re-numérote toutes les lignes en partant de 10000, avec un incrément par 5.

RESTORE

Syntaxe

RESTORE { [numéro-de-ligne]
[#numéro-de-fichier [, REC expression-numérique]] }

Description

L'instruction RESTORE sert à contrôler l'ordre dans lequel les données sont lues à partir d'instructions DATA ou à partir d'un fichier.

L'instruction RESTORE spécifie que la prochaine instruction READ exécutée doit lire le premier élément de l'instruction DATA spécifiée par le *numéro-de-ligne*. Celle-ci doit être dans le même programme ou sous-programme que l'instruction RESTORE. En l'absence de *numéro-de-ligne*, l'instruction DATA ayant la ligne de numérotation la plus faible dans le programme ou le sous-programme courant est utilisée. Si le *numéro-de-ligne* n'est pas celui d'une instruction DATA, la prochaine instruction DATA lui faisant suite est utilisée.

RESTORE #*numéro-de-fichier* place ce fichier en position de lecture du premier enregistrement. La prochaine instruction d'entrée/sortie faisant référence au *numéro-de-fichier* accède à cet enregistrement. Toute donnée de sortie en attente est écrite dans le fichier avant l'exécution de l'instruction RESTORE. Toutes données d'entrée est ignorée. Un *numero-de-fichier* 0 fait référence à une instruction DATA comme il est indiqué ci-dessus.

Nota : L'instruction RESTORE ne peut être utilisée avec des fichiers stockés par un magnétophone à cassette.

On peut utiliser REC avec un périphérique supportant un fichier d'enregistrement de type RELATIVE (accès direct). L'*expression-numérique* spécifie le numéro de l'enregistrement sur lequel on se place. La prochaine instruction d'entrée/sortie faisant référence à ce fichier accède à cet enregistrement. Consultez les manuels des périphériques pour de plus amples informations sur les fichiers de type RELATIVE.

Note : Le premier enregistrement d'un fichier est l'enregistrement zéro.

RESTORE

Renvoi

DATA, READ

Exemples

150 RESTORE

Détermine que la prochaine instruction DATA utilisée sera la première du programme.

200 RESTORE 130

Détermine que la prochaine instruction DATA utilisée sera celle de ligne 130 ; si la ligne 130 ne comporte pas d'instruction DATA, ce sera la prochaine instruction DATA qui suit la ligne 130.

230 RESTORE #1

Remet le fichier #1 en position de lecture du premier enregistrement, c'est-à-dire l'enregistrement numérique 0.

AVEC GOSUB

RETURN

Syntaxe

RETURN

Description

RETURN utilisé avec GOSUB permet de se transférer à l'instruction succédant la dernière instruction GOSUB ou ON GOSUB exécutée. Une sous-routine peut contenir plus d'une instruction RETURN.

Renvoi

GOSUB, ON GOSUB

AVEC ON ERROR

RETURN

Syntaxe

RETURN { [NEXT]
[numéro-de-ligne] }

Description

RETURN termine une sous-routine de traitement des erreurs. Celle-ci est appelée quand une erreur se produit après qu'une instruction ON ERROR *numéro-de-ligne* ait été exécutée. La sous-routine de traitement des erreurs peut contenir n'importe quelle instruction BASIC, y compris une autre instruction ON ERROR.

En l'absence d'option, RETURN transfère le contrôle à l'instruction dans laquelle l'erreur s'est produite et l'instruction est de nouveau exécutée.

RETURN NEXT transfère le contrôle à l'instruction succédant à celle dans laquelle l'erreur s'est produite.

RETURN *numéro-de-ligne* transfère le contrôle à la ligne spécifiée. Celle-ci doit se trouver dans le même programme ou sous-programme que la sous-routine de traitement des erreurs, bien que l'erreur ait pu se produire dans un autre sous-programme.

Renvoi

ON ERROR

Exemples

Le programme ci-dessous illustre l'utilisation de RETURN avec ON ERROR.

```
100 ON ERROR 150
```

Permet de se transférer à la ligne 150 quand une erreur se produit.

```
110 X=VAL("D")
```

Provoque une erreur, il y a donc transfert à la ligne 150.

```
120 PRINT "FAIT":PAUSE 2
```

Affiche FAIT.

AVEC ON ERROR

RETURN

```
130 STOP
140 REM TRAITEMENT ERREURS
150 IF A>3 THEN 200
```

Teste si l'erreur s'est produite quatre fois et transfère à la ligne 200 si c'est le cas.

```
160 A=A+1
```

Augmente le compteur des erreurs de un.

```
170 PRINT A;"erreur(s)":PAUSE 2
```

Affiche le nombre d'erreurs qui se sont produites.

```
180 ON ERROR 150
```

Permet à nouveau de traiter les erreurs en se transférant à la ligne 150.

```
190 RETURN
```

Retourne à la ligne qui a provoqué l'erreur et l'exécute de nouveau.

```
200 PRINT "Derniere erreur":PAUSE 2:RETURN NEXT
```

N'est exécutée qu'après que l'erreur se soit produite quatre fois. Affiche Derniere erreur et retourne à la ligne succédant à celle qui a provoqué l'erreur.

RND

Syntaxe

RND

Description

La fonction RND donne le prochain nombre pseudo-aléatoire de la séquence courante. Ce nombre est supérieur ou égal à zéro et inférieur à un. RND génère la même séquence à chaque lancement d'un programme à moins que l'instruction RANDOMIZE ne soit utilisée pour générer une nouvelle séquence imprévisible.

Renvoi

RANDOMIZE

Exemple

```
10 PRINT 10*RND:PAUSE
```

Affiche un nombre aléatoire supérieur ou égal à zéro et inférieur à dix.

Syntaxe

RPT\$(*expression-chaîne*,*expression-numérique*)

Description

La fonction RPT\$ donne une chaîne constituée de l'*expression-numérique* répétitions de l'*expression-chaîne*. Si RPT\$ produit une chaîne comportant plus de 255 caractères, les caractères en excès sont rejetés et le message d'avertissement W28 Truncation (chaîne tronquée) est affiché.

Exemples

```
100 M$=RPT$("ABCD",4)
```

Affecte à M\$ la chaîne "ABCDABCDABCDABCD".

```
100 PRINT USING RPT$("#",40);X$:PAUSE
```

Affiche la valeur de X\$ en utilisant un format de 40 signes dièse.

Syntaxe

RUN [*numéro-de-ligne*]

RUN [*"nom-de-programme"*]

RUN [*"périphérique.nom-de-fichier"*]

Description

L'instruction RUN lance l'exécution d'un programme. En l'absence d'option, RUN exécute le programme en mémoire, en commençant par la ligne de numérotation la plus basse.

RUN *numéro-de-ligne* exécute le programme en mémoire à partir du *numéro-de-ligne* spécifié.

RUN *"nom-de-programme"* cherche le *nom-de-programme* dans la cartouche Solid State Software™ et l'exécute après l'avoir trouvé. Si le *nom-de-programme* n'existe pas ou s'il fait référence à un sous-programme, une erreur se produit. Une expression-chaîne peut être utilisée pour spécifier un *nom-de-programme*.

RUN *"périphérique.nom-de-fichier"* supprime le programme actuellement en mémoire, charge à sa place et exécute le contenu du *nom-de-fichier* à partir du *périphérique*. Une expression-chaîne peut être utilisée pour spécifier un *périphérique.nom-de-fichier*.
Note : Si le *nom-de-fichier* désigne un fichier de données plutôt qu'un fichier de programme, il peut être nécessaire de réinitialiser l'ordinateur.

La procédure suivante précède l'exécution d'un programme :

- Les variables sont initialisées. Les variables numériques sont mises à zéro et les variables-chaîne sont affectée d'une chaîne vide.
- Certaines erreurs, telle qu'une instruction FOR sans instruction NEXT ou une référence à une ligne inaccessible, sont détectées.
- Tous les fichiers qui étaient ouverts sont fermés.
- ON BREAK STOP, ON WARNING PRINT et ON ERROR STOP sont sélectionnés.
- L'unité d'angle reste inchangée.

Exemples

RUN

Entraîne l'exécution du programme en mémoire, en commençant par la ligne de numérotation la plus basse.

RUN 20

Entraîne l'exécution du programme en mémoire en commençant à la ligne 20.

RUN "1.PR3"

Entraîne le chargement du programme du fichier PR3 du périphérique 1, puis son exécution.

RUN "STAT"

Exécute le programme STAT du module *Solid State Software*

Le programme ci-dessous illustre l'utilisation de l'instruction RUN pour exécuter un programme à partir d'un programme. Il s'agit d'établir un "menu" permettant à l'utilisateur du programme principal de choisir quel autre programme lancer. Les autres programmes doivent relancer le menu plutôt que de se terminer de la façon habituelle, de manière à ce que le menu soit à nouveau proposé dès qu'ils ont fini de s'exécuter.

```
100 PRINT "CHOISISSEZ UN N° DE PROGRAMME":PAUSE 2
110 PRINT "... OU FAITES 4 POUR ARRETER":PAUSE 2
120 INPUT "VOTRE SELECTION 1,2 ou 3:";C
130 IF C=1 THEN RUN "1.PR1"
140 IF C=2 THEN RUN "1.PR2"
150 IF C=3 THEN RUN "1.PR3"
150 IF C=4 THEN STOP
170 GOTO 100
```

Syntaxe

SAVE "*périphérique.nom-de-fichier*" [,PROTECTED]

Description

La commande SAVE vous permet de copier le programme BASIC en mémoire sur un périphérique externe. Vous pouvez par la suite rappeler le programme en mémoire au moyen de la commande OLD.

Avant de mémoriser le programme, SAVE efface de l'ordinateur toutes variables inutilisées dans le programme.

Périphérique.nom-de-fichier identifie le périphérique sur lequel le programme doit être stocké ainsi que le nom du fichier.

Périphérique est un numéro correspondant à un périphérique et se situe entre 1 et 255. Le *nom-de-fichier* identifie le fichier contenant le programme.

Si PROTECTED est spécifié, le programme en mémoire demeure non protégé, mais la copie sur le périphérique externe est sauvegardée sous forme protégée. Un programme protégé ne peut être ni listé, ni modifié, ni sauvegardé.

Nota: La commande SAVE ne permet pas de mémoriser des données dans un module.

Renvoi

GET, OLD, PRINT (avec fichiers), PUT, VERIFY

Exemples

SAVE "1.PR1"

Sauvegarde le programme en mémoire dans le périphérique 1 sous le nom PR1.

SAVE "1.PR2" ,PROTECTED

Sauvegarde le programme en mémoire, sur le périphérique 1 sous le nom PR2. Le programme peut être chargé en mémoire et lancé, mais il ne peut ni être édité, ni défilé, ni être sauvegardé de nouveau.

SEG\$

Syntaxe

SEG\$(*expression-de-chaîne*, *position*, *longueur*)

Description

La fonction SEG\$ fournit une sous-chaîne. La sous-chaîne obtenue commence au caractère *position* de l'*expression-chaîne* et se poursuit sur la *longueur* de caractères spécifiée. Si la *position* se situe après la fin de l'*expression-chaîne*, on obtient une chaîne vide (""). Si la *longueur* dépasse la fin de l'*expression-chaîne*, seuls les caractères situés en deçà sont fournis.

Exemples

```
100 X$=SEG$("PRENOM NOM",1,6)
```

Affecte à X\$ la chaîne "PRENOM".

```
200 X$=SEG$("PRENOM NOM",8,3)
```

Affecte à X\$ la chaîne "NOM".

```
240 Z$=SEG$("PRENOM NOM",7,1)
```

Affecte à Z\$ la chaîne " ".

```
280 PRINT SEG$(A$,B,C):PAUSE
```

Affiche le segment d'A\$ commençant au caractère numéro B et s'étendant sur C caractères.

SGN

Syntaxe

SGN(*expression-numérique*)

Description

La fonction SGN donne le signe mathématique de l'*expression-numérique*. On obtient 1 si celle-ci est positive, 0 si elle est nulle et -1 si elle est négative.

Renvoi

ABS

Exemples

```
140 IF SGN(A)=1 THEN 300 ELSE 400
```

Transfère le contrôle à la ligne 300 si A est positif et à la ligne 400 si A est nul ou négatif.

```
790 ON SGN(X)+2 GOTO 200,300,400
```

Transfère le contrôle à la ligne 200 si X est négatif, à la ligne 300 si X est nul et à la ligne 400 si X est positif.

CHAPITRE V

INDEX DE REFERENCE

SIN

Syntaxe

SIN(*expression-numérique*)

Description

La fonction sinus donne le sinus de l'*expression-numérique*. Celle-ci est interprétée en radians, degrés ou grades selon l'unité d'angle en service (voir DEG, GRAD et RAD). Consultez la description des limites de l'*expression-numérique* à l'annexe B.

Renvoi

ACOS, ASIN, ATN, COS, DEG, RAD, GRAD, TAN

Exemple

```
150 DEG  
160 PRINT SIN(3*21.5+4):PAUSE
```

Affiche .930417568.

CHAPITRE V

INDEX DE REFERENCE

SINH

Syntaxe

SINH(*expression-numérique*)

Description

La fonction SINH (sinus hyperbolique) calcule le sinus hyperbolique de l'*expression-numérique*. Le sinus hyperbolique se définit comme suit :

$$\text{SINH}(X) = .5 * (\text{EXP}(X) - \text{EXP}(-X))$$

Renvoi

ACOSH, ASINH, ATANH, COSH, TANH

Exemples

```
100 PRINT SINH(0):PAUSE
```

Affiche 0.

```
230 T=SINH(0.75)
```

Met T égal à .8223167319

SQR

Syntaxe

SQR(*expression-numérique*)

Description

La fonction SQR donne la racine carrée positive de l'*expression-numérique*. SQR(X) est équivalent à $X^{(1/2)}$. L'*expression-numérique* ne doit pas être un nombre négatif.

Exemples

```
150 PRINT SQR(4):PAUSE
```

Affiche 2.

```
780 X=SQR(2.57E5)
```

Affecte à X la racine carrée de 257000, qui est 506,9516742.

STOP

Syntaxe

STOP

Description

L'instruction STOP arrête l'exécution d'un programme. On peut l'utiliser indifféremment à la place de l'instruction END, sauf après un sous-programme.

Renvoi

END

Exemple

Le programme ci-dessous illustre l'utilisation de l'instruction STOP. Ce programme ajoute les nombres de 1 à 100.

```
100 TOT=0
110 NUMB=1
120 TOT=TOT+NUMB
130 NUMB=NUMB+1
140 IF NUMB > 100 THEN PRINT TOT:PAUSE 2:STOP
150 GOTO 120
```

STR\$

Syntaxe

STR\$(*expression-numérique*)

Description

La fonction STR\$ donne la valeur de l'*expression-numérique* sous forme de chaîne sans espace de début ou de fin. La fonction STR\$ est l'inverse de la fonction VAL.

Renvoi

NUMERIC, VAL

Exemples

150 NUM\$=STR\$(78.6)

Affecte à NUM\$ la chaîne "78,6".

220 LL\$=STR\$(3E15)

Affecte à LL\$ la chaîne "3,E + 15".

330 J\$=STR\$(A*4)

Affecte à J\$ la chaîne représentant valeur obtenue quand A est multiplié par 4. Par exemple, si A est égal à -8, J\$ est égal à "- 32".

SUB

Syntaxe

SUB *nom-de-sous-programme* [(*liste-des-paramètres*)]

Description

L'instruction SUB est la première dans un sous-programme et doit être la première instruction de la ligne. Un sous-programme est un groupe d'instructions distinct du programme principal, qui par exemple sert à effectuer la même tâche en plusieurs endroits, sans avoir à reproduire les instructions en ces différents endroits.

On accède à un sous-programme par CALL *nom-de-sous-programme*[(*liste-d'arguments*)]. Le *nom-de-sous-programme* comporte de 1 à 15 caractères. Le premier caractère doit être une lettre ou un soulignement. Les suivants peuvent être des lettres, des chiffres ou des soulignements. L'instruction CALL recherche un sous-programme dans un ordre spécifique (voir cet ordre sous CALL) et exécute le premier sous-programme trouvé ayant le *nom-de-sous-programme*. Si le nom d'un de vos sous-programmes est le même que celui d'un sous-programme système, c'est ce dernier qui est exécuté.

La *liste-de-paramètres* définit l'information à transmettre au sous-programme. Un paramètre peut être une variable-chaîne simple, une variable numérique simple ou un tableau. Pour qu'un tableau serve de paramètre, il faut écrire son nom suivi de parenthèses. Un tableau unidimensionnel s'écrit M(), une matrice bidimensionnelle M(,), et une matrice tridimensionnelle s'écrit M(,,).

L'information est communiquée au sous-programme par l'intermédiaire de la *liste-d'arguments* de l'instruction CALL. Les arguments de cette liste et les paramètres de la *liste-de-paramètres* n'ont pas nécessairement les mêmes noms. Cependant, le nombre et le type des arguments de la *liste-d'arguments* doivent coïncider avec le nombre et le type des paramètres de la *liste-de-paramètres* de l'instruction SUB.

Les données sont communiquées à un sous-programme, par leur nom ou par leur valeur. Si un argument est transmis par son nom, le sous-programme utilise la variable correspondante du programme appelant. Si le paramètre dans le sous-programme est modifié, la variable du programme appelant est également modifiée. Une variable simple, un élément de tableau, ou un tableau

SUB

figurant dans la *liste-d'arguments* sont transmis par leur nom. Les tableaux sont toujours communiqués par leur nom.

Si un argument est transmis par sa valeur, seule cette valeur est communiquée au sous-programme. Si le paramètre correspondant dans le sous-programme est modifié, cela n'affecte pas l'argument du programme appelant. Toute expression de la *liste-d'arguments* est évaluée et communiquée par sa valeur au sous-programme. On peut transmettre des variables simples par leur valeur en les faisant figurer entre parenthèses.

Toutes les variables utilisées à l'intérieur d'un sous-programme autres que celles de la *liste-de-paramètres*, sont locales à ce sous-programme, de façon à ce que les mêmes noms de variables puissent servir dans le programme principal et dans d'autres sous-programmes. Le changement des valeurs des variables locales d'un programme ou d'un sous-programme n'affecte pas les valeurs des variables locales d'autres programmes ou sous-programmes. Toutes les variables locales d'un sous-programme sont initialisées chaque fois que celui-ci est appelé.

L'exécution des instructions SUBEXIT ou SUBEND termine celle du sous-programme correspondant. Le contrôle revient à l'instruction suivant l'instruction CALL.

Les sous-programmes figurent après le programme principal. Un sous-programme ne peut contenir d'autre sous-programme. Quand l'instruction SUB est rencontrée dans un programme principal, celui-ci est terminé comme si l'instruction STOP avait été exécutée. Seules des remarques et des instructions END peuvent figurer entre le SUBEND d'un programme et le SUB du sous-programme suivant.

Les instructions ON BREAK, ON WARNING, ON ERROR et PAUSE ALL en fonction lors de l'exécution d'un CALL le demeurent pendant l'exécution du sous-programme. Si on change une ou plusieurs de ces instructions dans le sous-programme, elles retrouvent leur forme initiale quand le sous-programme est terminé. Les sous-programmes ne peuvent partager de sous-routines, à l'exception de celles traitant les erreurs.

SUB

Renvoi

CALL, ON BREAK, ON ERROR, ON WARNING, RETURN, SUBEND, SUBEXIT

Exemples

100 SUB MENU

Indique le début d'un sous-programme. Aucun paramètre n'est transmis ni restitué.

220 SUB MENU (COMPTE, SELECTION)

Indique le début d'un sous-programme. Les variables COMPTE et SELECTION peuvent être utilisées et/ou leurs valeurs changées dans le sous-programme ainsi que leurs arguments correspondants dans l'instruction d'appel.

330 SUB TEST (D, Q, S, P, T (,))

Indique le début d'un sous-programme. Les variables D, S, P, et la matrice à deux dimensions T peuvent être utilisées et/ou leurs valeurs changées dans le sous-programme ainsi que leurs arguments correspondants dans l'instruction d'appel.

SUBEND

Syntaxe

SUBEND

Description

L'instruction SUBEND indique la fin d'un sous-programme. Après son exécution, le contrôle est transféré à l'instruction succédant à celle qui a appelé le sous-programme. L'instruction SUBEND doit toujours être la dernière d'un sous-programme et ne peut figurer dans une instruction IF THEN ELSE. Seules des remarques et des instructions END peuvent figurer entre une instruction SUBEND et la prochaine instruction SUB.

Renvoi

SUB, SUBEXIT

SUBEXIT

Syntaxe

SUBEXIT

Description

L'instruction SUBEXIT termine l'exécution d'un sous-programme. Après son exécution, le contrôle est transféré à l'instruction succédant à celle qui a appelé le sous-programme. L'instruction SUBEXIT peut figurer autant de fois que cela est nécessaire dans un sous-programme.

Renvoi

SUB, SUBEND

TAB

Syntaxe

TAB(*expression-numérique*)

Description

La fonction TAB utilisée dans une instruction PRINT ou DISPLAY sert à préciser la position d'un élément à afficher ou à écrire. Si l'*expression-numérique* est inférieure ou égale à zéro, la colonne de départ est 1. Si l'*expression-numérique* est supérieure à la longueur d'enregistrement pour le périphérique utilisé, elle est alors réduite autant de fois que nécessaire, de la longueur d'enregistrement jusqu'à ce qu'elle lui soit inférieure ou égale.

Si la position courante est inférieure ou égale à la position spécifiée, la fonction TAB ménage un espace jusqu'à la position spécifiée. Si la position courante est supérieure à la position spécifiée, la fonction TAB continue jusqu'à l'enregistrement suivant et ménage un espace jusqu'à la position spécifiée.

La fonction TAB est traitée comme un *élément à afficher* ou à *écrire*, et doit être séparée des autres éléments par un séparateur. Le séparateur situé avant TAB est évalué avant la fonction TAB et celui situé à la suite de TAB est évalué après cette fonction. On utilise normalement des points-virgules avant et après TAB.

Dans une instruction DISPLAY, la fonction TAB concerne le début du champ d'affichage. Si AT est utilisé, la fonction TAB concerne la position de colonne spécifiée. Si plus d'une ligne de résultats est affichée, les lignes suivantes commencent en colonne un. Les fonctions TAB concernent alors la colonne un.

Si l'on utilise SIZE, la valeur spécifiée est alors la limite absolue du nombre de caractères affichés. Cette limite est la longueur d'enregistrement utilisée lors de l'évaluation de toutes les fonctions TAB.

TAB

Renvoi

DISPLAY, PRINT (avec affichage), PRINT (avec fichiers)

Exemples

```
100 PRINT TAB(12);35:PAUSE
```

Affiche le nombre 35 en commençant à la colonne 13.

```
190 PRINT 356;TAB(18);"NOM":PAUSE
```

Affiche 356 au début de la ligne et NOM en commençant à la colonne 18.

```
710 DISPLAY AT(10) SIZE(20),"MGB";TAB(10);  
"ADR":PAUSE
```

Affiche MGB en commençant à la colonne 10 et ADR en commençant à la colonne 19.

TAN

Syntaxe

TAN(*expression-numérique*)

Description

La fonction TAN (tangente) donne la tangente de l'*expression numérique*. Celle-ci est interprétée en radians, degrés ou grades selon l'unité d'angle sélectionnée (voir DEG, GRAD, et RAD). Une description des limites de l'*expression-numérique* figure à l'annexe B.

Renvoi

ACOS, ASIN, ATN, COS, DEG, RAD, GRAD, SIN

Exemple

```
250 RAD
260 PRINT TAN(20):PAUSE

Affiche 2.237160944.
```

TANH

Syntaxe

TANH(*expression-numérique*)

Description

La fonction TANH donne la tangente hyperbolique de l'*expression-numérique*. La tangente hyperbolique se définit comme suit :

$$\text{TANH}(X) = (\text{EXP}(X) - \text{EXP}(-X)) / (\text{EXP}(X) + \text{EXP}(-X))$$

Renvoi

ACOSH, ASINH, ATANH, COSH, SINH

Exemples

```
100 PRINT TANH(0):PAUSE

Affiche 0.

230 T=TANH(0.75)

Met T égal à .6351489524
```

UNBREAK

Syntaxe

UNBREAK [*liste-de-lignes*]

Description

L'instruction UNBREAK retire tous les points d'arrêt. Si une *liste-de-lignes* est spécifiée, seuls les points d'arrêt assignés à ces lignes sont retirés.

Renvoi

BREAK

Exemples

UNBREAK

Retire tous les points d'arrêt.

400 UNBREAK 100,130

Retire les points d'arrêt posés avant les lignes 100 et 130.

USING

Syntaxe

DISPLAY USING *numéro-de-ligne*

DISPLAY USING *expression-chaîne*

PRINT USING *numéro-de-ligne*

PRINT USING *expression-chaîne*

Description

USING peut figurer dans une instruction PRINT ou DISPLAY pour conférer un format au résultat. Si un *numéro-de-ligne* est fourni, le format est spécifié dans cette ligne par une instruction IMAGE. Le *numéro-de-ligne* doit désigner une ligne du programme ou sous-programme courant. Voir IMAGE. Si une *expression-de-chaîne* est précisée, le format est défini par USING.

En présence de USING, les modifications suivantes interviennent dans l'évaluation de la *liste* de PRINT ou de DISPLAY.

- Les virgules séparatrices sont traitées comme des points-virgules.
- L'emploi de la fonction TAB provoque une erreur.
- Les éléments à écrire sont disposés conformément aux modèles de la définition de format. Si le nombre des éléments de la *liste* dépasse le nombre de modèles du format, l'enregistrement entièrement mis en forme est écrit. Les valeurs restantes sont écrites dans l'enregistrement suivant, en utilisant de nouveau la définition du format, depuis le commencement. Le format est utilisé autant de fois que cela est nécessaire pour terminer la *liste*. Un nouvel enregistrement est généré à chaque utilisation du format. Quand le nombre d'éléments est inférieur au nombre de modèles de la définition, les résultats s'arrêtent au premier modèle pour lequel il n'y a pas d'élément à écrire.
- Si un élément mis en forme est trop long pour l'enregistrement courant, il est divisé en segments. Le premier segment remplit la partie restante de l'enregistrement en cours, et les autres sont écrits dans le prochain enregistrement.

Renvoi

DISPLAY, IMAGE, PRINT

VAL

Syntaxe

VAL(*expression-chaîne*)

Description

La fonction VAL donne la valeur numérique de l'*expression-chaîne*. Les espaces de début et de fin sont ignorés. La fonction VAL est l'inverse de la fonction STR\$.

Si l'*expression-chaîne* n'est pas la représentation autorisée d'un nombre, une erreur se produit. Pour l'éviter, on peut vérifier l'*expression-chaîne* à l'aide de la fonction NUMERIC.

Renvoi

NUMERIC, STR\$

Exemples

```
170 NUM=VAL("78.6")
```

Met NUM à 78,6.

```
190 LL=VAL("3E15")
```

Met LL à 3,E + 15.

```
300 PRINT VAL("$3.50"):PAUSE
```

Provoque une erreur puisque la chaîne ne représente pas une constante numérique autorisée.

VERIFY

Syntaxe

VERIFY "*périphérique.nom-de-fichier*" [,PROTECTED]

Description

La commande VERIFY vérifie que les données ont été sauvegardées sur une unité de stockage externe ou correctement chargées en mémoire. On utilise VERIFY après les commandes SAVE ou OLD pour comparer le programme en mémoire à celui du périphérique. Si une différence est constatée, un message d'erreur est affiché. Les messages d'erreur d'entrée/sortie 12 et 24 indiquent une erreur lors de la vérification.

Périphérique.nom-de-fichier identifie le périphérique et le fichier dans lesquels le programme est stocké. *Périphérique* est un numéro correspondant à une unité physique, il se situe entre 1 et 255. Le *nom-de-fichier* identifie le fichier.

Tout comme SAVE, VERIFY retire tous les noms de variables qui ne sont pas utilisés dans le programme. Si le programme est protégé, on doit alors spécifier PROTECTED dans la commande VERIFY.

Renvoi

OLD, SAVE

Exemples

```
SAVE "1.MONPROG"
```

Sauvegarde le fichier MONPROG dans le périphérique 1.

```
VERIFY "1.MONPROG"
```

Vérifie si le fichier a été correctement stocké.

```
OLD "1.STAT"
```

Lit en mémoire le fichier nommé STAT à partir du périphérique 1.

```
VERIFY "1.STAT"
```

Vérifie que le fichier a été lu correctement.

CHAPITRE VI
UTILISATION DE
L'ORDINATEUR AVEC DES
EQUIPEMENTS OPTIONNELS

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

Utilisation de l'ordinateur avec des équipements optionnels

Le présent chapitre traite des équipements périphériques optionnels : magnétophone à cassette, imprimante et module d'extension mémoire. Ces équipements vous permettent de sauvegarder des données en les enregistrant sur cassette, d'imprimer des résultats, ou d'augmenter la capacité de la mémoire.

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

Conseils pour le choix de l'équipement

Les performances de votre système de stockage sur cassette dépendent du type de magnétophone et de cassette choisi. Il est recommandé de tenir compte des conseils qui suivent lorsque vous choisirez votre magnétophone à cassette et ses accessoires. Le câble de liaison, le magnétophone à cassette et la cassette se transforment en une unité complète de stockage d'informations pour l'ordinateur.

Choix d'un magnétophone

L'ordinateur est compatible avec la plupart des magnétophones à cassette standard. Toutefois, pour obtenir les meilleurs résultats possibles, choisissez un appareil présentant les caractéristiques suivantes :

- Réglage du volume sonore et de la tonalité
- Prise pour microphone
- Prise pour télécommande
- Prise pour écouteurs ou haut-parleur externe
- Compteur numérique de déroulement de bande (vous permettant de repérer rapidement un endroit précis sur la bande lorsque vous stockez plus d'un programme ou ensemble de données sur la même bande)
- Adaptateur-secteur optionnel (évitant les problèmes dus aux piles déchargées, tels que perturbation dans le transfert des données)

Choix des cassettes

Pour une capacité optimale de stockage, suivez les conseils ci-après lorsque vous choisissez des cassettes.

- Optez pour des bandes à faible niveau de bruit. Il n'est pas utile de prendre des cassettes à grande réponse en fréquence, telles que bandes digitales, qui coûtent plus cher que les cassettes classiques pour enregistrement sonore.
- Choisissez des bandes de qualité. Les bandes de mauvaise qualité ont tendance à bourrer et à se casser.
- Utilisez le type et la longueur de bande recommandés par le fabricant de votre magnétophone.

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

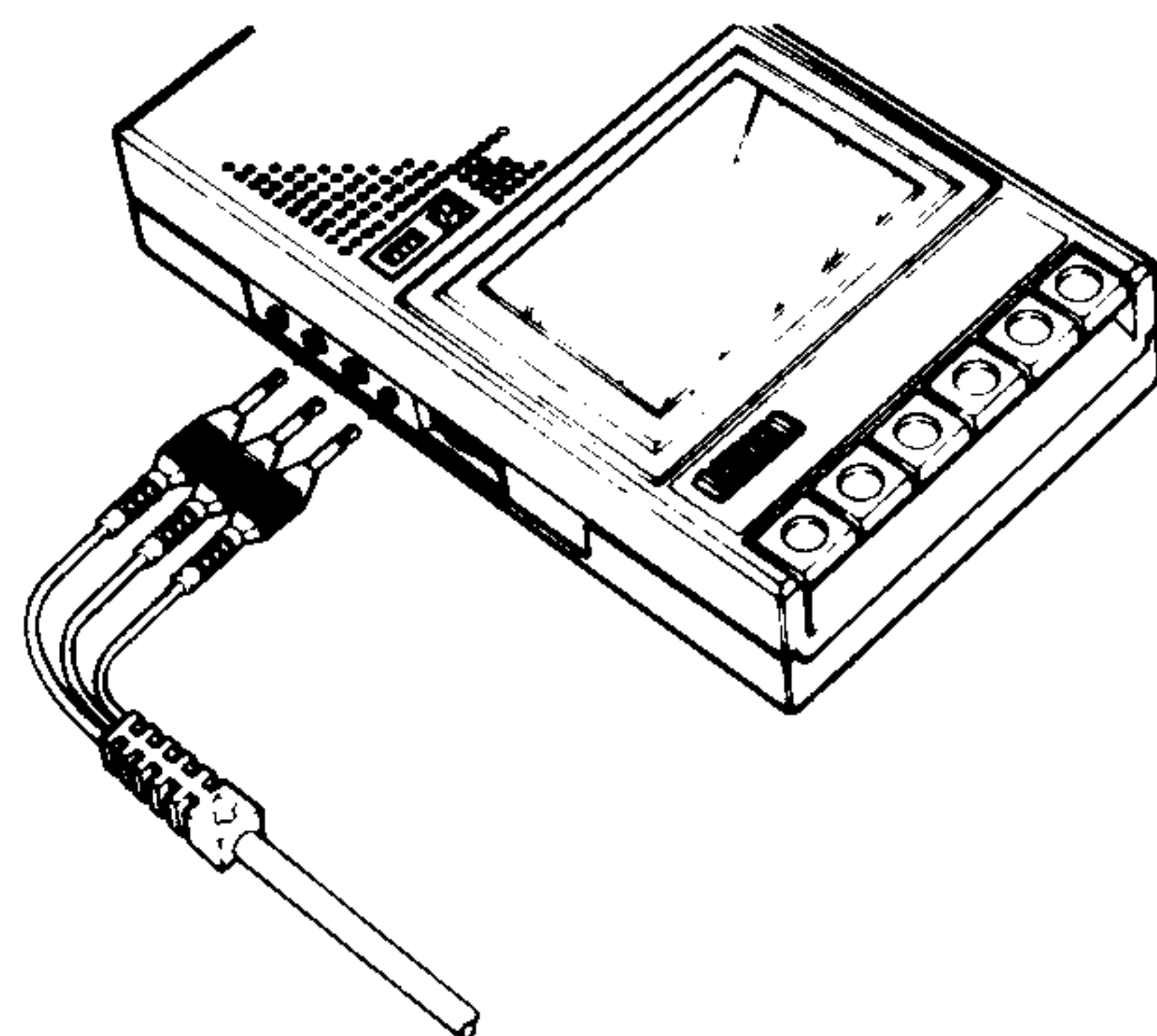
Raccordement du magnétophone à l'ordinateur

Le câble d'interface cassette permet à l'ordinateur d'échanger des données avec le magnétophone. La connexion du câble d'interface à l'ordinateur et au magnétophone se fait de façon très simple, selon les étapes décrites ci-après.

Raccordement du magnétophone

Assurez-vous que l'ordinateur est éteint avant de connecter le câble. Pour raccorder le magnétophone à l'ordinateur, procédez comme suit :

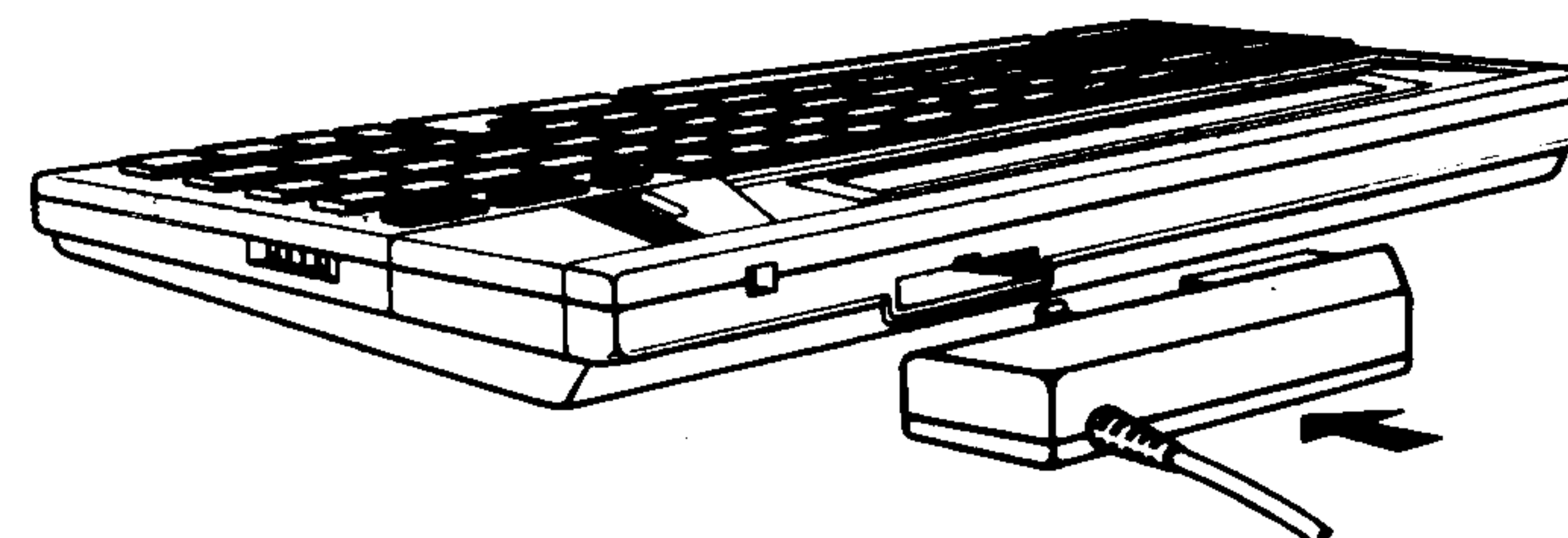
1. Branchez la fiche rouge sur la prise microphone du magnétophone (habituellement marquée MIC).



2. Branchez la fiche grise sur la prise casque, moniteur ou haut-parleur externe du magnétophone (habituellement marquée EAR ou MONITOR).

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

3. Branchez la petite fiche noire sur la prise de télécommande de votre magnétophone (habituellement marquée REM).
4. Branchez l'autre extrémité du câble de liaison cassette sur le connecteur de périphériques situé à l'arrière de l'ordinateur.



Messages concernant l'utilisation du magnétophone

Quand vous sauvegardez et récupérez des programmes ou des données, vous devez faire fonctionner le magnétophone manuellement. Des messages apparaissent à l'affichage pour vous aider dans la manipulation de l'appareil et pour vous informer du déroulement de la procédure d'enregistrement.

Messages affichés automatiquement

Au cours de l'enregistrement, deux types de messages apparaissent à l'affichage.

- Messages directifs : ils vous indiquent la marche à suivre pour commander manuellement le magnétophone et l'ordinateur. Par exemple, le message `Position tape ; then press ENTER` (positionnez la bande puis appuyez sur ENTER) vous dit ce que vous devez faire à ce moment précis.

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

- Messages d'information : ils vous indiquent où en est le processus d'enregistrement. Par exemple, le message `Reading...` (lecture) vous informe que l'ordinateur est en train de récupérer les données d'un fichier sur cassette.

Enregistrement sans messages

Une fois familiarisé avec les opérations d'enregistrement, vous avez la possibilité d'accélérer le processus en éliminant l'affichage automatique des messages. Pour ce faire, tapez le suffixe `“.NM“` ("no messages", "pas de messages") à la fin du nom du fichier. Si vous invalidez les messages afin de commencer une opération d'entrée ou de sortie, assurez-vous que la bande est correctement positionnée.

Par exemple, si vous faites `RUN "1.TEST.NM"` et appuyez sur **[ENTER]** puis sur le bouton **PLAY** de votre magnétophone, l'ordinateur recherche le fichier `TEST` sur la bande. Une fois qu'il l'a localisé, vous pouvez charger et lancer le programme sans recevoir de messages.

Réglages du magnétophone

Avant de commencer à enregistrer vos programmes et fichiers de données, vous pouvez exécuter le programme d'essai ci-après. Ce test d'enregistrement vous permet de régler correctement le volume et la tonalité du magnétophone, de vérifier la compatibilité de l'appareil et de la bande, et de vous familiariser avec l'enregistrement d'un fichier de programmes.

Procédure

1. Si votre magnétophone dispose d'un bouton de réglage de tonalité, positionnez-le sur une valeur intermédiaire.
2. Réglez le volume au niveau maximum.

Commentaire

Si ce réglage ne donne pas la précision requise, reprenez éventuellement le programme de test qui suit en réduisant progressivement le niveau de réglage.

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

Procédure

3. Introduisez une cassette dans le magnétophone. Revenez en début de bande et mettez le compteur à zéro. Faites avancer la bande pour la positionner après l'amorce (004 ou plus).
4. Assurez-vous qu'aucune touche du magnétophone n'est enfoncée. Mettez l'ordinateur en marche.
5. Vérifiez que l'ordinateur est en mode **BASIC**. Entrez le programme de test suivant :

```
100 A$="TEST"
110 FOR B=1 TO 3
120 FOR C=1 TO 4
130 DISPLAY SEG$
(A$,1,C):PAUSE .3
140 NEXT C:NEXT B:END
```
6. Tapez `SAVE "1.TESTPROG"` et appuyez sur **[ENTER]**.
7. Lorsque le message `Position tape; then press ENTER` apparaît, vérifiez que la bande est bien positionnée après l'amorce. Notez le chiffre indiqué par le compteur et appuyez sur **[ENTER]**.

Commentaire

Pour enregistrer des données, la bande doit se trouver après une des bandes-amorces. L'amorce est le segment non-magnétique situé à chaque extrémité d'une bande.

Permet d'accéder au périphérique 1 (le magnétophone) et affecte un nom à cet enregistrement d'essai. L'indicateur d'entrée/sortie **I/O** s'affiche.

Le chiffre indiqué par le compteur vous aide à repérer le début du fichier lorsque vous voulez le récupérer.

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

Procédure

8. Une fois que le message **Press RECORD; then ENTER** est affiché, appuyez sur le(s) bouton(s) d'enregistrement du magnétophone, puis sur **[ENTER]**.
9. Après affichage de **Press STOP, then ENTER**, pressez le bouton STOP du magnétophone. Notez le chiffre indiqué par le compteur et appuyez sur **[ENTER]**.
10. Tapez **VERIFY "1.TESTPROG"** (vérification 1^{er} programme de test) et pressez **[ENTER]**.
11. Après apparition du message **Positiontape; then press ENTER**, utilisez les boutons du magnétophone pour repérer le début du fichier. Appuyez ensuite sur **[ENTER]**.
12. Lorsque le message **Press PLAY; then ENTER** s'affiche, appuyez sur le bouton de lecture du magnétophone, puis sur **[ENTER]**.

Commentaire

Au bout de quelques secondes, le magnétophone commence à enregistrer le fichier, et le message **Writing...** (enregistrement en cours) s'affiche.

Votre fichier est sauvegardé. L'ordinateur peut maintenant mémoriser un autre fichier ou repasser votre enregistrement, et vous savez où commence et finit le fichier d'après les chiffres du compteur.

Vous pouvez maintenant lire le fichier.

Après quelques secondes, le magnétophone restitue le fichier et le message **Reading...** apparaît à l'affichage.

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

Procédure

13. Si l'un des messages suivants s'affiche, suivez les instructions qui vous concernent.
 - Si **Press STOP; then ENTER** s'affiche, appuyez sur le bouton STOP du magnétophone. Le fichier est sauvegardé sur la bande.
 - Si **E0 I/O error 3 "1"** s'affiche, passez à l'opération n° 14.
 - Si **E0 I/O error 23 "1"** s'affiche, passez à l'opération n° 14.
 - Si aucun message d'erreur n'apparaît au bout d'un certain temps et que l'ordinateur continue de chercher le fichier, vérifiez l'état des piles. Passez ensuite à l'opération n° 6.
14. Rebobinez la bande jusqu'au début du fichier et réduisez le volume d'environ 10 pour cent. Recommencez l'opération n° 10 pour procéder à la vérification.

Commentaire

Votre programme de test est vérifié. Notez les niveaux de réglage du volume et de la tonalité que vous avez utilisés pour votre enregistrement d'essai.

L'ordinateur n'a pas trouvé le programme de test. Il se peut que vous ayez essayé d'enregistrer sur l'amorce de la bande, que le réglage du volume soit incorrect, ou encore que vous n'ayez pas commencé en début de fichier.

Le fichier est bien présent, mais un problème technique empêche sa lecture par l'ordinateur. Ou bien le volume n'est pas correctement réglé, ou bien le magnétophone est incompatible.

Des piles trop faibles peuvent perturber le transfert des données. Remplacez les éventuellement, ou branchez le magnétophone sur secteur en utilisant un adaptateur.

Poursuivez la lecture du segment à différents niveaux de volume sonore jusqu'à ce que vous trouviez le fichier. Si vous ne le trouvez pas bien que la bande soit correctement positionnée, reprenez l'opération n° 6 pour essayer à nouveau de sauvegarder le programme.

CHAPITRE VI

UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

Procédure de sauvegarde de programmes

Vous pouvez utiliser un magnétophone à cassette pour enregistrer tous les programmes que vous avez développés sur votre ordinateur. Vous pourrez les récupérer plus tard en rechargeant les informations dans la mémoire de l'ordinateur. Assurez-vous d'avoir bien suivi la procédure "test d'enregistrement" servant à déterminer les réglages du volume et de la tonalité avant de commencer à sauvegarder des fichiers.

Utilisation de la commande SAVE

La commande SAVE sert à copier (ou à sauvegarder) un programme de la mémoire de l'ordinateur sur une cassette.

Pour sauvegarder le programme en mémoire, procédez de la manière suivante :

1. Entrez par le clavier SAVE "1.nom-de-fichier" et appuyez sur [ENTER].

Le chiffre 1 est le numéro attribué au magnétophone et nom-du-fichier est le nom de votre fichier.

2. Lorsque le message `Position tape; then press ENTER` apparaît, faites avancer la bande pour la positionner sur une zone vierge, tout en prenant garde de laisser suffisamment d'espace pour l'enregistrement du programme.

Laissez un espace entre le dernier fichier enregistré et le début du nouveau fichier.

Si votre magnétophone est équipé d'un compteur numérique, rebobinez jusqu'au début de la bande, et mettez le compteur à zéro avant de la faire avancer.

3. Appuyez sur la touche [ENTER].
4. Lorsque le message `Press RECORD; then ENTER` apparaît, appuyez sur la touche RECORD (enregistrement) de votre magnétophone, puis sur [ENTER].

Après quelques secondes, le message `Writing...` apparaît. L'ordinateur est en train de transférer les informations sur le fichier cassette.

Nota : Si vous devez interrompre le processus, appuyez sur la touche [RESET] de l'ordinateur.

CHAPITRE VI

UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

5. Lorsque le message `Press STOP; then ENTER` apparaît, appuyez sur la touche STOP de votre magnétophone, puis sur la touche [ENTER].

Le programme est maintenant sauvegardé sur une cassette. Si vous désirez vérifier que l'enregistrement est correct, utilisez la commande VERIFY.

Vérification du stockage et récupération de programmes

La commande VERIFY compare le fichier de programmes et le programme en mémoire. Il est bon d'utiliser cette commande après avoir sauvegardé ou chargé un programme.

Utilisation de la commande VERIFY

Pour vérifier que le stockage ou le chargement d'un fichier s'est effectué correctement, procédez de la manière suivante :

1. Entrez par le clavier VERIFY "1.nom-de-fichier" et appuyez sur la touche [ENTER].
2. Lorsque le message `Position tape; then press ENTER` apparaît, procédez comme suit :
 - Si vous connaissez la position de votre fichier sur la cassette, appuyez sur les touches FAST FORWARD (avance rapide) ou REWIND (rebobinage) pour trouver le début du fichier.
 - Si vous ne connaissez pas la position exacte de votre fichier, rebobinez la bande jusqu'au début.
3. Appuyez sur [ENTER].
4. Lorsque le message `Press PLAY; then ENTER` apparaît, appuyez sur la touche PLAY (lecture) de votre magnétophone, puis sur [ENTER].

Lorsque le fichier est localisé, le message `Reading...` apparaît. L'ordinateur compare le fichier enregistré sur la cassette avec les informations contenues dans sa mémoire.

5. Si le message `Press STOP; then ENTER` apparaît, cela signifie que les contenus sont identiques. Appuyez sur la touche STOP de votre magnétophone, puis sur la touche [ENTER].

CHAPITRE VI

UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

Si un message d'erreur apparaît, cela signifie que les contenus ne sont pas identiques. Dans ce cas, vous avez deux possibilités d'intervention :

- Si vous avez sauvegardé un programme, vérifiez les réglages de votre magnétophone, puis sauvegardez à nouveau votre programme en suivant la procédure décrite à la page précédente.
- Si vous avez récupéré un programme, vérifiez les réglages de votre magnétophone, puis tentez de récupérer votre programme une seconde fois, en suivant la procédure décrite à la page précédente.

Procédure de récupération de programmes

Vous pouvez récupérer des fichiers que vous avez sauvegardés sur cassette en les rechargeant dans la mémoire de l'ordinateur. La commande OLD permet de récupérer un fichier et de le charger en mémoire ; la commande VERIFY vous permet de vérifier que le chargement s'est effectué correctement, et l'instruction RUN récupère et exécute un programme en une seule étape.

Utilisation de la commande OLD

La commande OLD permet de charger dans l'ordinateur un fichier sauvegardé sur une cassette. Si vous introduisez par inadvertance une mauvaise commande, vous pouvez annuler l'opération en appuyant sur la touche [BREAK].

Pour utiliser la commande OLD, procédez de la manière suivante :

1. Entrez par le clavier une des commandes suivantes, puis appuyez sur [ENTER] :
 - OLD "1.nom-de-fichier" si vous voulez récupérer un fichier spécifique.
 - OLD "1." si vous voulez récupérer le fichier suivant d'une bande. Si le fichier suivant n'est pas un fichier de programmes, un message d'erreur s'affiche.

Notez qu'un fichier de données ne peut pas être chargé par la commande OLD. Par contre, un programme peut lire les données d'un fichier une par une, à l'aide de l'instruction INPUT.

CHAPITRE VI

UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

2. Lorsque le message Position tape; then press ENTER apparaît, procédez comme suit :
 - Si vous connaissez la position de votre fichier sur la cassette, appuyez sur les touches FAST FORWARD et REWIND pour trouver le début.
 - Si vous ne connaissez pas la position exacte de votre fichier, rebobinez la bande jusqu'au début.
3. Appuyez sur la touche [ENTER].
4. Lorsque le message Press PLAY; then ENTER apparaît, appuyez sur la touche PLAY de votre magnétophone, puis sur [ENTER].

L'ordinateur recherche le début du fichier. Lorsque le fichier est localisé, le message Reading... apparaît. L'ordinateur est en train de récupérer le fichier de la cassette et de charger les informations en mémoire.

Nota : Si vous devez interrompre le processus, appuyez sur la touche [RESET].

5. Lorsque le message Press STOP; then ENTER apparaît, appuyez sur la touche STOP de votre magnétophone, puis sur la touche [ENTER].

Le programme est maintenant chargé en mémoire. Si vous désirez vérifier que la lecture du programme s'est effectuée correctement, utilisez la commande VERIFY en suivant la procédure décrite à la page 223.

Si le programme a été chargé correctement, vous pouvez l'éditer ou le lancer.

Utilisation de l'instruction RUN

L'instruction RUN peut également récupérer et exécuter immédiatement un programme sauvegardé sur une cassette.

Pour utiliser l'instruction RUN, entrez les commandes suivantes par le clavier, puis appuyez sur [ENTER].

- RUN "1.nom-de-fichier" si vous désirez récupérer un fichier spécifique.
- RUN "1." si vous désirez récupérer le fichier suivant d'une bande. Si le fichier suivant n'est pas un fichier de programmes, un message d'erreur apparaît.

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

L'ordinateur utilise les mêmes messages que pour la commande OLD, pour simplifier la manipulation du magnétophone. Cependant, lorsque le programme est chargé, il est immédiatement exécuté.

Commande et contrôle de l'imprimante à partir du BASIC

Ce chapitre traite des commandes applicables lorsque l'imprimante PC-324 est connectée à l'ordinateur. Vous pouvez sélectionner les options "retour de chariot" et "interligne". L'imprimante peut également recevoir des instructions sous forme de sous-programmes IO ou de chaînes de caractères.

Code périphérique

Pour vous adresser à l'imprimante à partir du BASIC, vous devez utiliser le code 12, qui correspond à l'imprimante.

Options retour de chariot

Après avoir utilisé l'instruction OPEN pour créer un lien de communication avec l'imprimante, vous pouvez utiliser l'instruction PRINT, qui permettra l'impression d'un élément. Lorsque l'élément est imprimé, les options "retour de chariot" entraînent :

- Soit un retour de chariot automatique après l'impression de l'élément : vous sélectionnez cette option en ajoutant R=L à l'instruction OPEN ou en ne précisant pas R.
- Soit l'impression de l'élément suivant sur la même ligne : vous sélectionnez cette option en ajoutant R=N à l'instruction OPEN.

Ces options peuvent être utilisées avec une instruction OPEN ou une commande LIST.

Exemple

Lorsque le programme suivant est lancé :

```
100 !R=N Option
110 OPEN #1,"12.R=N",OUTPUT
120 PRINT #1"un ";"deux ";"trois ";"quatre "
```

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

la sortie sur imprimante se présente comme suit :

un deux trois quatre

Lorsque le programme suivant est lancé :

```
100 !R=L Option
110 OPEN #1,"12.R=L",OUTPUT
120 PRINT #1"un ";"deux ";"trois ";"quatre "
```

la sortie sur imprimante se présente comme suit :

un
deux
trois
quatre

Options interligne

Les options "interligne" permettent d'obtenir des interlignes simples ou doubles.

- Il est possible d'imprimer chaque ligne de la feuille de papier. Vous sélectionnez cette option en ajoutant L=S à l'instruction OPEN, ou en ne précisant pas L.
- Il est possible d'imprimer une feuille de papier toutes les deux lignes. Vous sélectionnez cette option en ajoutant L=D à l'instruction OPEN.

Ces options peuvent être utilisées avec l'instruction OPEN et la commande LIST.

Exemple

Lorsque le programme suivant est lancé :

```
100 !L=S Option
110 OPEN #1,"12.L=S",OUTPUT
120 PRINT #1"un ";"deux ";"trois ";"quatre "
```

la sortie sur imprimante se présente comme suit :

un
deux
trois
quatre

CHAPITRE VI
UTILISATION DE L'ORDINATEUR
AVEC DES EQUIPEMENTS OPTIONNELS

Lorsque le programme suivant est lancé :

```
100 !L=D Option
110 OPEN #1,"12.L=D",OUTPUT
120 PRINT "un ";"deux ";"trois ";"quatre "
```

la sortie imprimante se présente comme suit :

un
deux
trois
quatre

Commandes d'entrée/sortie

La PC-324 peut recevoir certaines commandes périphériques fournies par le sous-programme IO. Ces commandes sont énumérées ci-dessous.

Code	Commande	Résultat
80	Self Test #0	Rappelle le numéro de version de la mémoire ROM de l'imprimante. Lorsque l'instruction CALL IO(12,80,X) est exécutée, X devient 100 pour la version 1, 101 pour la version 2, etc...
81	Self Test #1	Exécute une démonstration d'impression. Lorsque l'instruction CALL IO(12,81,X) est exécutée, X devient 6 si une erreur de périphérique intervient, et 1 si la démonstration s'exécute normalement.
82	Self Test #2	Compare l'imprimante avec les valeurs à imprimer. Ce test est similaire à celui effectué par la calculatrice lors de la mise en route, pour vérifier les éventuels changements intervenus dans le contenu des mémoires. Toutefois les valeurs testées étant en mémoire morte (ROM), le test échoue uniquement si l'imprimante est endommagée. Lorsque l'instruction CALL IO(12,82,X) est exécutée, si X prend la valeur 0 le test est satisfaisant. La valeur 80 indique que le test a échoué.

CHAPITRE VI
UTILISATION DE L'ORDINATEUR
AVEC DES EQUIPEMENTS OPTIONNELS

Exemple

Le programme suivant utilise le sous-programme IO pour tester l'imprimante.

```
110 X=0:Y=0
120 CALL IO(12,80,X)
130 CALL IO(12,82,Y)
140 X=X-99
150 PRINT USING"VERSION # EST DANS
    L'IMPRIMANTE";X:PAUSE
160 IF Y=0 THEN 170 ELSE 180
170 PRINT "LA MEMOIRE MORTE DE L'IMPRIMANTE EST
    SATISFAISANTE":PAUSE:STOP
180 PRINT "L'IMPRIMANTE EST EN PANNE"
    :PAUSE:STOP
```

Codes de commande de l'imprimante

Vous pouvez commander à l'imprimante d'exécuter une des actions suivantes en lui transmettant le code correspondant.

Code	Action
CHR\$(13)	Retour chariot
CHR\$(17)	Simple interligne
CHR\$(18)	Double interligne

Comme le retour chariot comprend un changement de ligne automatique, le code ASCII (10) destiné au changement de ligne est ignoré.

Exemple

Le programme suivant vous permet d'entrer des lignes par l'intermédiaire du clavier et de les envoyer à l'imprimante.

```
110 ACCEPT ERASE ALL,A$
120 DISPLAY "Double interligne ? (O/N)"
130 S$=KEY$
140 OPEN #1,"12"
150 IF S$="O" OR S$="o" THEN PRINT #1,CHR(18):GOTO 170
160 PRINT #1,CHR$(17)
170 PRINT #1,A$
180 CLOSE #1
190 GOTO 110
```


CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

Accès à la mémoire du module

L'emploi du module d'extension mémoire vous offre deux possibilités : extension de la capacité mémoire disponible pour la programmation en BASIC, ou stockage du contenu de la mémoire pour récupération ultérieure.

Adjonction de la mémoire du module au BASIC

Une fois que le module d'extension mémoire est placé dans le connecteur réservé à cet effet, vous devez exécuter une commande CALL ADDMEM si vous désirez utiliser la mémoire du module comme quantité de mémoire disponible supplémentaire.

ATTENTION : Si vous exécutez une commande CALL ADDMEM afin d'ajouter la mémoire du module au BASIC, vous effacez le contenu de la mémoire de ce module. La mémoire de l'ordinateur reste cependant intacte.

Reportez-vous au chapitre 5 pour toute information concernant le sous-programme ADDMEM et la fonction FRE.

Annulation de l'extension mémoire

Après l'exécution de CALL ADDMEM, la mémoire du module reste disponible pour le BASIC, à moins de la réalisation d'une des conditions suivantes :

- Exécution d'une commande NEW ALL.
- Pression sur la touche [RESET].
- Retrait du module pendant que l'ordinateur est allumé.
- Piles déchargées.

En cas d'occurrence de l'une de ces conditions, la mémoire du module est séparée de la mémoire BASIC disponible, et celle de l'ordinateur est effacée.

CHAPITRE VI UTILISATION DE L'ORDINATEUR AVEC DES EQUIPEMENTS OPTIONNELS

Emploi d'un module pour le stockage

Le module d'extension mémoire peut être utilisé pour stocker tout le contenu de la mémoire de l'ordinateur, connu sous le nom d'image de la mémoire. Reportez-vous au chapitre 5 pour toute information concernant les sous-programmes GET et PUT qui permettent de transférer des données sur un module.

Le module conserve une image de la mémoire tant que :

- une autre image n'y est pas introduite,
- CALL ADDMEM n'est pas exécuté,
- les piles ne sont pas déchargées.

Nota : En cas d'utilisation du module comme extension de mémoire, vous devez annuler cette extension avant d'employer le module pour le stockage.

ANNEXES

Annexes

Les annexes contiennent les informations dont vous pourrez avoir besoin pour approfondir certains aspects d'un programme. Notez qu'un paragraphe "Difficultés" est inclus dans les annexes, pour vous aider à résoudre certains problèmes. Ce manuel contient également les informations concernant la garantie et le service après-vente de Texas Instruments.

Symboles et messages affichés - Mode Calculatrice

L'affichage de la calculatrice indique non seulement des nombres, mais également des symboles et des messages. Aux différentes opérations effectuées par la machine correspondent des symboles d'identification, affichés à droite de l'entrée concernée. Ceci permet de conserver la trace des calculs pendant que vous les exécutez.

Symboles affichés	Séquence de touches	Signification
acos	[INV] [cos]	Arc-cosinus
acosh	[INV] [hyp] [cos]	Arc-cosinus hyperbolique
asin	[INV] [sin]	Arc-sinus
asinh	[INV] [hyp] [sin]	Arc-sinus hyperbolique
atan	[INV] [tan]	Arc-tangente
atanh	[INV] [hyp] [tan]	Arc-tangente hyperbolique
CE	[CE/C]	Effacement (si pressé une fois quand le curseur clignote)
CLR	[CE/C]	Effacement
-	[←]	Suppression de caractère
cos	[cos]	Cosinus
cosh	[hyp] [cos]	Cosinus hyperbolique
CSR	[STAT] [CSR]	Effacement registres statistiques
DMS→DD	[DMS→DD]	Degrés décimaux
DD→DMS**	[INV] [DMS→DD]	Degrés/minutes/secondes
DRG	[DRG]	Sélection de l'unité angulaire suivante
INV DRG	[INV] [DRG]	Sélection de l'unité angulaire précédente
DRG→	[DRG→]	Conversion vers l'unité angulaire suivante
←DRG	[INV] [DRG→]	Conversion vers l'unité angulaire précédente
EE	[EE]	Introduction exposant

ANNEXE A SYMBOLES ET MESSAGES AFFICHES

Symboles affichés	Séquence de touches	Signification
INV EE	[EE]	Suppression notation scientifique
e^x	[INV] [lnx]	Exponentielle e^x
EXC	[EXC]	Echange mémoire
Frac	[Frac]	Partie décimale
hyp	[hyp]	Préfixe hyperbolique
Intg	[Intg]	Partie entière
INV	[INV]	Préfixe inverse
INVhyp	[INV] [hyp]	Préfixe hyperbolique inverse
lnx	[lnx]	Logarithme népérien
log	[log]	Logarithme décimal
n!	[n!]	Factorielle
%	[%]	Pourcentage
$\Delta\%$	[$\Delta\%$]	Ecart en pourcentage
RCL	[RCL]	Rappel mémoire
sin	[sin]	Sinus
sinh	[hyp] [sin]	Sinus hyperbolique
STAT	[STAT]	Préfixe statistique
STO	[STO]	Stockage en mémoire
SUB	[INV] [SUM]	Soustraction en mémoire
SUM	[SUM]	Addition en mémoire
tan	[tan]	Tangente
tanh	[hyp] [tan]	Tangente hyperbolique
10^x	[INV] [log]	Exponentielle décimale
x^2	[x^2]	Carré
\sqrt{x}	[\sqrt{x}]	Racine carrée
π	[π]	Valeur de pi
$1/x$	[1/x]	Inverse
$y^x:x?$	[y^x]	Puissance quelconque
$^x\sqrt{y}:x?$	[INV] [y^x]	Racine quelconque
$x:x\rightleftharpoons y$	[$x\rightleftharpoons y$]	Permutation (x,y) avant introduction du couple
$x\rightleftharpoons y$	[$x\rightleftharpoons y$]	Permutation (x,y) après introduction du couple

ANNEXE A SYMBOLES ET MESSAGES AFFICHES

Symboles affichés	Séquence de touches	Signification
\div	[/]	Division
*	[*]	Multiplication
-	[-]	Soustraction
+	[+]	Addition
=	[=]	Egale
([(]	Ouverture parenthèse
)	[)]	Fermeture parenthèse
+/-	[+/-]	Changement de signe
(x,y?)	[x,y]	Entrée couple (x,y)
$P \rightarrow R$	[$P \rightarrow R$]	Conversion polaires/cartésiennes
y =	[$x\rightleftharpoons y$]	Coordonnée y
x =	[$x\rightleftharpoons y$]	Coordonnée x
$R \rightarrow P$	[INV] [$P \rightarrow R$]	Conversion cartésiennes/polaires
$\theta =$	[$x\rightleftharpoons y$]	Coordonnée θ
r =	[$x\rightleftharpoons y$]	Coordonnée rayon
nPr	[nPr]	Permutations
nCr	[nCr]	Combinaisons
$\Sigma +$	[$\Sigma +$]	Sigma plus
$\Sigma -$	[INV] [$\Sigma +$]	Sigma moins
Frq	[FRQ]	Fréquence
Σx	[STAT] [Σx]	Somme de valeurs x
Σy	[STAT] [Σy]	Somme de valeurs y
Σxy	[STAT] [Σxy]	Somme de produits xy
Σx^2	[STAT] [Σx^2]	Somme de valeurs x^2
Σy^2	[STAT] [Σy^2]	Somme de valeurs y^2
\bar{x}	[STAT] [\bar{x}]	Moyenne de valeurs x
\bar{y}	[STAT] [\bar{y}]	Moyenne de valeurs y
σx	[STAT] [Δx]	Ecart type de x
σy	[STAT] [Δy]	Ecart type de y
n	[STAT] [n]	Nombre de points de données
r	[STAT] [r]	Coeff.de corrélation
a	[STAT] [a]	Intersection de y par la droite de régression
b	[STAT] [b]	Pente de la droite de régression
x'	[STAT] [x']	x pour une valeur de y
y'	[STAT] [y']	y pour une valeur de x

** L'affichage indique le résultat accompagné des symboles ° ' ".

ANNEXE A

SYMBLES ET MESSAGES AFFICHES

Messages affichés en mode calculatrice

Les messages qui apparaissent à l'affichage ont la signification suivante.

E23 Bad argument (argument incorrect)	La calculatrice a décelé une entrée non conforme à la fonction recherchée, par exemple 6 [+/-] [lnx] ou .5 [n!].
E4 Bad value (valeur incorrecte)	La méthode de calcul de la machine a révélé une entrée non conforme telle que [(] 6 [+/-] [y ^x] .5 [)].
E2 Complex (complexe)	Un calcul a engendré plus de 22 niveaux d'imbrication de parenthèses et d'opérations en attente.
E26 Division by zero (division par zéro)	Un calcul a inséré une division par zéro telle que [(] 4 [/] 1 [log] [)], 0 [1/x], ou 0 [(x,y)] 0 [INV] [P→R].
E25 Overflow (dépassement de capacité)	Le résultat du calcul se trouve hors des limites de la calculatrice, par exemple 400 [INV] [log] ou [(] 150 [y ^x] 150 [)].

Messages système

Les messages ci-après apparaissent dans l'un des modes CALC ou BASIC.

W27 contents may be lost (le contenu peut être à la mise sous tension, altéré)	l'ordinateur a décelé que le contenu de la mémoire permanente n'était pas le même que lors du dernier arrêt.
W30 Initialized (réinitialisé)	Certaines conditions ont imposé l'initialisation complète du système.

ANNEXE B

CONDITIONS D'ERREUR

Conditions d'erreur

L'affichage indique une condition d'erreur en cas de dépassement de capacité, ou lorsque vous tentez une opération avec une entrée non conforme. Dans ce cas, aucune introduction par le clavier n'est acceptée. En pressant [CE/C], vous effacez le message d'erreur ainsi que toutes les opérations en attente. Vous devez alors déterminer la cause de l'erreur et refrapper l'entrée pour résoudre le problème.

Types de conditions d'erreur

Le paragraphe suivant énumère les conditions d'erreur générales. Elles concernent les deux modes, CALC et BASIC, à moins que la fonction ne soit disponible qu'en mode CALC. La section qui traite des erreurs consécutives à des opérations statistiques énumère les erreurs que vous pouvez éventuellement rencontrer en introduisant des données statistiques. Une condition d'erreur n'a aucune incidence sur les valeurs qui se trouvent en mémoire, ni sur les données statistiques qui ont été introduites.

Conditions d'erreur générales

1. Le nombre introduit ou le résultat du calcul (y compris la mémoire) est en dehors des limites comprises entre $-9.999999999999 \times 10^{127}$ et $9.999999999999 \times 10^{127}$.
2. Division d'un nombre par zéro
3. Calcul de la valeur inverse de zéro ou de la racine 0 d'un nombre quelconque.
4. Elévation d'un nombre négatif à une puissance décimale.
5. Extraction d'une racine quelconque d'un nombre négatif.
6. Calcul de la tangente de 90° ou de 270°, de $\pi/2$ radians ou de $3\pi/2$ radians, de 100 grades ou de 300 grades, ou de leurs multiples rotationnels, par exemple 450°.
7. Imbrication de parenthèses ouvertes et d'opérations en attente sur plus de 22 niveaux.
8. Calcul de conversions cartésiennes/polaires où les valeurs attribuées à x et y sont telles que la somme de leurs carrés dépasse la limite supérieure de la calculatrice, ou bien lorsque x et y sont égaux à zéro.
9. Utilisation d'un argument hors des limites données par le "domaine des valeurs d'entrée" pour les fonctions énumérées sur la page suivante.

ANNEXE B CONDITIONS D'ERREUR

Domaine des valeurs d'entrée

Quelques fonctions pouvant donner comme résultat une valeur beaucoup plus grande que la valeur d'entrée, il peut se produire un dépassement de capacité même pour une entrée qui semble moyenne.

Vous trouverez ci-après les limites dans lesquelles doit se trouver la valeur affichée lors du calcul de certaines fonctions.

Fonction	Limite
$\sin x, \cos x, \tan x$	$0 \leq x < 4.5 \times 10^9$ degrés $0 \leq x < 78539816$ radians $0 \leq x < 5 \times 10^9$ grades
$\sin^{-1} x, \cos^{-1} x$	$-1 \leq x \leq 1$
$\sinh x, \cosh x$	$0 \leq x < 295.1040919$
$\cosh^{-1} x$	$1 \leq x < 10^{128}$
$\sinh^{-1} x$	$0 \leq x < 10^{128}$
\tanh^{-1}	$-1 < x < 1$
$\ln x, \log x$	$10^{-128} \leq x < 10^{128}$
e^x	$ x \leq 294.7308919$
10^x	$-128 \leq x < 128$
$n!$	$0 \leq n \leq 84$ où n est un entier

Conditions d'erreur statistique

1. Introduction d'une valeur telle que $|x| \geq 1 \times 10^{64}$.
2. Introduction d'une série de données telles que la somme de leurs carrés se trouve hors des limites supérieure ou inférieure de la calculatrice.
3. Introduction de couples de données tels que la somme de leurs produits se trouve hors des limites supérieure ou inférieure de la calculatrice.
4. Introduction de 10^{14} ou plus de données.

Nota : Vous pouvez provoquer l'effacement des registres statistiques en supprimant des données avec **[INV] [Σ+]** jusqu'à ce qu'il y ait zéro donnée. Si vous tentez de supprimer plus de données que ce que vous n'en avez introduit, la calculatrice ne supprime que les données existantes.

ANNEXE C DOMAINE DES VALEURS DE SORTIE

Domaine des valeurs de sortie

Les définitions mathématiques de fonctions trigonométriques inverses offrent la possibilité de nombreuses valeurs de sortie pour une même entrée. Toutefois, la calculatrice fournit des valeurs comprises dans des domaines prédéterminés pour les fonctions trigonométriques inverses.

Fonctions trigonométriques inverses

Les possibilités de résultat des fonctions trigonométriques inverses se présentent comme suit :

Fonction arc arc-sinus x	Domaine de valeurs de l'angle résultant 0 à 90°, $\pi/2$ radians ou 100 grades (premier quadrant)
arc-sinus $-x$	0 à -90°, $-\pi/2$ radians ou -100 grades (quatrième quadrant)
arc-cosinus x	0 à 90°, $\pi/2$ radians ou 100 grades (premier quadrant)
arc-cosinus $-x$	90° à 180°, $\pi/2$ à π radians ou 100 grades à 200 grades (deuxième quadrant)
arc-tangente x	0 à 90°, $\pi/2$ radians ou 100 grades (premier quadrant)
arc-tangente $-x$	0 à -90°, $-\pi/2$ radians ou -100 grades (quatrième quadrant)

ANNEXE D

LISTE DES MOTS RESERVES

Liste des mots réservés

La liste suivante présente tous les mots réservés du BASIC de la série TI-70. Un mot réservé ne peut être utilisé comme nom de variable, mais peut constituer une partie de celui-ci. Un nom de variable peut également être une partie d'un mot réservé à l'exception des abréviations des mots réservés.

Commandes et instructions

La plupart des instructions peuvent aussi bien être exécutées immédiatement que dans une ligne de programme. Les mots que vous ne pouvez utiliser que dans une instruction de programme sont suivis d'un P et les commandes que vous ne pouvez utiliser qu'en dehors d'un programme sont suivies d'un °.

ABS	ELSE	NUMERIC
ACCEPT ^P	END	OLD [°]
ACOS	EOF	ON
ACOSH	ERASE	OPEN
ALL	ERROR	OR
ALPHA	EXP	OUTPUT
ALPHANUM	FOR	PAUSE
AND	FORMAT	PI
APPEND	FRE	POS
ASC	GOSUB ^P	PRINT
ASIN	GOTOP ^P	PROTECTED
ASINH	GRAD	RAD
AT	IF	RANDOMIZE
ATANH	IMAGE ^P	READ ^P
ATN	INPUT ^P	REC
BREAK	INT	RELATIVE
CALL	INTERNAL	REM
CHR\$	KEY\$	REN [°]
CLOSE	LEN	RENUMBER [°]
CON [°]	LET	RESTORE
CONTINUE [°]	LINPUT ^P	RETURN ^P
COS	LIST [°]	RND
COSH	LN	RPT\$
DATA	LOG	RUN
DEG	NEW [°]	SAVE [°]
DEL	NEXT	SEG\$
DELETE	NOT	SGN
DIGIT	NULL	SIN
DIM	NUM [°]	SINH
DISPLAY	NUMBER [°]	SIZE

ANNEXE D

LISTE DES MOTS RESERVES

SQR	TANH	VALIDATE
STEP	THEN	VARIABLE
STOP	TO	VERIFY [°]
STR\$	UALPHA	WARNING
SUB ^P	UALPHANUM	XOR
SUBEND ^P	UNBREAK	
SUBEXIT ^P	UPDATE	
TAB	USING	
TAN	VAL	

Sous-programmes

Vous pouvez utiliser les noms de sous-programmes système comme noms de variables. Cependant, les sous-programmes que vous écrirez ne devront pas porter des noms de sous-programmes système.

CALL ADDMEM[°]
CALL ERR
CALL GET[°]
CALL IO
CALL KEY
CALL PUT[°]

ANNEXE E
CODES CARACTERES ASCII

Codes caractères ASCII

Le tableau suivant donne la liste des codes caractères ASCII en notation décimale et hexadécimale.

Tableau ASCII

Les codes ASCII produits et/ou le(s) caractère(s) affiché(s) quand on appuie sur la/les touche(s) correspondante(s) figurent dans la colonne intitulée Caractères. Les caractères affichables au moyen de la fonction CHR\$ figurent dans la colonne intitulée Affichage avec CHR\$. Les touches sur lesquelles il faut appuyer pour obtenir un code ASCII donné figurent dans la colonne intitulée Séquence de touches.

Les codes réservés pour le système (0-15)et les touches utilisateur (codes 128-137) sont présentés avec deux astérisques (**).

Code DEC	ASCII HEX	Caractère	Affichage avec CHR\$	Séquence de touches
00	00	NULL	**	[CTL] 0
01	01	SOH	**	[CTL] A
02	02	STX	**	[CTL] B
03	03	ETX	**	[CTL] C
04	04	EOT	**	[CTL] D
05	05	ENQ	**	[CTL] E
06	06	ACK	**	[CTL] F
07	07	BEL	**	[CTL] G
08	08	BS	**	[CTL] H
09	09	HT	**	[CTL] I
10	0A	LF	**	[CTL] J
11	0B	VT	**	[CTL] K
12	0C	FF	**	[CTL] L
13	0D	CR	**	[CTL] M ou [ENTER]
14	0E	SO	**	[CTL] N
15	0F	SI	**	[CTL] O
16	10	DLE		[CTL] P
17	11	DC1		[CTL] Q
18	12	DC2		[CTL] R
19	13	DC3		[CTL] S
20	14	DC4		[CTL] T
21	15	NAK		[CTL] U
22	16	SYN		[CTL] V
23	17	ETB		[CTL] W
24	18	CAN		[CTL] X

ANNEXE E
CODES CARACTERES ASCII

Code DEC	ASCII HEX	Caractère	Affichage avec CHR\$	Séquence de touches
25	19	EM		[CTL] Y
26	1A	SUB		[CTL] Z
27	1B	ESC		[CTL] [CLR]
28	1C	F\$		[CTL] + / -
29	1D	GS		[CTL] ;
30	1E	RS		[CTL] .
31	1F	US		[CTL] ,
32	20	Espace	Espace	Espace
33	21	!	!	[SHIFT] !
34	22	"	"	[SHIFT] "
35	23	≠	≠	[SHIFT] ≠
36	24	\$	\$	[SHIFT] \$
37	25	%	%	[SHIFT] %
38	26	&	&	[SHIFT] &
39	27	,	,	[SHIFT] Space
40	28	(([SHIFT] [↑]
41	29))	[SHIFT] [↓]
42	2A	*	*	*
43	2B	+	+	+
44	2C	,	,	,
45	2D	-	-	-
46	2E	.	.	.
47	2F	/	/	/
48	30	0	0	0
49	31	1	1	1
50	32	2	2	2
51	33	3	3	3
52	34	4	4	4
53	35	5	5	5
54	36	6	6	6
55	37	7	7	7
56	38	8	8	8
57	39	9	9	9
58	3A	:	:	[SHIFT] :
59	3B	;	;	;
60	3C	<	<	[SHIFT] <
61	3D	=	=	[SHIFT] [ENTER]
62	3E	>	>	[SHIFT] >
63	3F	?	?	[SHIFT] ?
64	40	@	@	[CTL] 2
65	41	A	A	[SHIFT] A
66	42	B	B	[SHIFT] B

ANNEXE E
CODES CARACTERES ASCII

Code DEC	ASCII HEX	Caractère	Affichage avec CHR\$	Séquence de touches
67	43	C	C	[SHIFT] C
68	44	D	D	[SHIFT] D
69	45	E	E	[SHIFT] E
70	46	F	F	[SHIFT] F
71	47	G	G	[SHIFT] G
72	48	H	H	[SHIFT] H
73	49	I	I	[SHIFT] I
74	4A	J	J	[SHIFT] J
75	4B	K	K	[SHIFT] K
76	4C	L	L	[SHIFT] L
77	4D	M	M	[SHIFT] M
78	4E	N	N	[SHIFT] N
79	4F	O	O	[SHIFT] O
80	50	P	P	[SHIFT] P
81	51	Q	Q	[SHIFT] Q
82	52	R	R	[SHIFT] R
83	53	S	S	[SHIFT] S
84	54	T	T	[SHIFT] T
85	55	U	U	[SHIFT] U
86	56	V	V	[SHIFT] V
87	57	W	W	[SHIFT] W
88	58	X	X	[SHIFT] X
89	59	Y	Y	[SHIFT] Y
90	5A	Z	Z	[SHIFT] Z
91	5B	[[[CTL] 8
92	5C	¥	¥	[CTL] /
93	5D]]	[CTL] 9
94	5E	^	^	[SHIFT] ^
95	5F	_	_	[CTL] 5
96	60	`	`	[CTL] 3
97	61	a	a	A
98	62	b	b	B
99	63	c	c	C
100	64	d	d	D
101	65	e	e	E
102	66	f	f	F
103	67	g	g	G
104	68	h	h	H
105	69	i	i	I
106	6A	j	j	J
107	6B	k	k	K
108	6C	l	l	L

ANNEXE E
CODES CARACTERES ASCII

Code DEC	ASCII HEX	Caractère	Affichage avec CHR\$	Séquence de touches
109	6D	m	m	M
110	6E	n	n	N
111	6F	o	o	O
112	70	p	p	P
113	71	q	q	Q
114	72	r	r	R
115	73	s	s	S
116	74	t	t	T
117	75	u	u	U
118	76	v	v	V
119	77	w	w	W
120	78	x	x	X
121	79	y	y	Y
122	7A	z	z	Z
123	7B	{	{	[CTL] 6
124	7C			[CTL] 1
125	7D	}	}	[CTL] 7
126	7E	→	→	[CTL] 4
127	7F	DEL	←	[SHIFT] [←]
128	80	**		[FN] O
129	81	**		[FN] 1
130	82	**		[FN] 2
131	83	**		[FN] 3
132	84	**		[FN] 4
133	85	**		[FN] 5
134	86	**		[FN] 6
135	87	**		[FN] 7
136	88	**		[FN] 8
137	89	**		[FN] 9
138	8A			
139	8B			
140	8C			
141	8D			[SHIFT] /
142	8E			[SHIFT] *
143	8F			[SHIFT] -
144	90			[SHIFT] +
145	91			[CTL] *
146	92			[CTL] -
147	93			[CTL] +
148	94	NEW		[FN] [→]
149	95	NUMBER		[FN] [←]
150	96	RENUMBER		[FN] [↑]

ANNEXE E CODES CARACTERES ASCII

Code DEC	ASCII HEX	Caractère	Affichage avec CHR\$	Séquence de touches
151	97	FRE([FN] [↓]
152	98	VERIFY		[FN] /
153	99	SAVE		[FN] *
154	9A	OLD		[FN] -
155	9B	LIST		[FN] +
156	9C	FORMAT		[FN] .
157	9D	OPEN		[FN] ,
158	9E	ERROR		[FN] ;
159	9F	SGN([FN] +/-
160	A0	DELETE		[FN] [CLR]
161	A1	FOR	「	[FN] A
162	A2	READ	」	[FN] B
163	A3	PAUSE	「	[FN] C
164	A4	NEXT	」	[FN] D
165	A5	TAN(「	[FN] E
166	A6	IF	」	[FN] F
167	A7	THEN	「	[FN] G
168	A8	ELSE	」	[FN] H
169	A9	SQR(「	[FN] I
170	AA	GOTO	」	[FN] J
171	AB	GOSUB	「	[FN] K
172	AC	RETURN	」	[FN] L
173	AD	RESTORE	「	[FN] M
174	AE	DATA	」	[FN] N
175	AF	CHR\$(「	[FN] O
176	B0	CALL	」	[FN] P
177	B1	SIN(「	[FN] Q
178	B2	PI	」	[FN] R
179	B3	TO	「	[FN] S
180	B4	LN(」	[FN] T
181	B5	EXP(「	[FN] U
182	B6	INPUT	」	[FN] V
183	B7	COS(「	[FN] W
184	B8	USING	」	[FN] X
185	B9	LOG(「	[FN] Y
186	BA	PRINT	」	[FN] Z
187	BB	BREAK	「	[FN] [BREAK]
188	BC	STOP	」	[SHIFT] [RUN]
189	BD		「	
190	BE	CONTINUE	」	[FN] [RUN]
191	BF	RUN	「	[RUN]
192	C0		」	

ANNEXE E CODES CARACTERES ASCII

Code DEC	ASCII HEX	Caractère	Affichage avec CHR\$	Séquence de touches
193	C1		チ	
194	C2		ツ	
195	C3		テ	
196	C4		ト	
197	C5		+	
198	C6		-	+/-
199	C7		ニ	
200	C8		ヌ	
201	C9		ネ	
202	CA		ノ	
203	CB		ハ	
204	CC		ヘ	
205	CD		フ	
206	CE		ン	
207	CF		ホ	
208	D0		マ	[SHIFT] [FN] 0
209	D1		ミ	[SHIFT] [FN] 1
210	D2		ム	[SHIFT] [FN] 2
211	D3		メ	[SHIFT] [FN] 3
212	D4		モ	[SHIFT] [FN] 4
213	D5		ユ	[SHIFT] [FN] 5
214	D6		ヨ	[SHIFT] [FN] 6
215	D7		ラ	[SHIFT] [FN] 7
216	D8		リ	[SHIFT] [FN] 8
217	D9		ル	[SHIFT] [FN] 9
218	DA		レ	
219	DB		ロ	
220	DC		ク	
221	DD		コ	
222	DE		ケ	
223	DF		カ	
224	E0		「	
225	E1		」	[SHIFT] [→]
226	E2		「	
227	E3		」	
228	E4		「	
229	E5	PB	「	[SHIFT] 9
230	E6	OFF	」	[OFF]
231	E7	BREAK	「	[BREAK]
232	E8	UP	」	[↑]
233	E9	DOWN	「	[↓]
234	EA		」	

ANNEXE E
CODES CARACTERES ASCII

Code DEC	ASCII HEX	Caractère	Affichage avec CHR\$	Séquence de touches
235	EB		*	
236	EC		¢	
237	ED		£	
238	EE		¤	
239	EF		¥	
240	F0	MODE	␣	[MODE]
241	F1		␣	
242	F2		␣	
243	F3		␣	
244	F4		␣	
245	F5		␣	
246	F6	DEL	␣	[SHIFT] 7
247	F7	INS	␣	[SHIFT] 8
248	F8	HOME	␣	[CTL] [↑]
249	F9	SKIP	␣	[CTL] [↓]
250	FA	CLR	␣	[CLR]
251	FB	BTAB	␣	[CTL] [→]
252	FC	u	␣	[→]
253	FD	FTAB	␣	[CTL] [←]
254	FE	o	␣	[←]
255	FF			

ANNEXE F
OPERATIONS LOGIQUES SUR LES NOMBRES

Opérations logiques sur les nombres

Les opérateurs logiques AND, OR, NOT et XOR (ET, OU, NON et OU-exclusif) peuvent être utilisés sur des nombres entiers allant de - 32768 à 32767. La présente annexe décrit brièvement le système de numérotation binaire, la conversion des nombres décimaux en leurs équivalents binaires et le fonctionnement des opérateurs logiques.

Numérotation binaire

La numérotation binaire (base 2) est une autre façon d'exprimer la valeur d'un nombre. Notre système habituel, la numérotation décimale (base 10), utilise une combinaison de dix chiffres, de zéro à neuf. Les nombres écrits en binaire n'utilisent que deux chiffres, zéro et un. Chaque position occupée par un chiffre binaire (0 ou 1) est appelée bit.

En numérotation décimale, chaque chiffre d'un nombre représente une puissance de 10. Par exemple, le nombre 2408 en décimal peut se développer de la façon suivante :

$(2 \times 10^3) + (4 \times 10^2) + (0 \times 10^1) + (8 \times 10^0)$

Ce qui est bien égal à 2408 :

$$\begin{array}{r} 2 \times 10^3 = 2 \times 1000 = 2000 \\ 4 \times 10^2 = 4 \times 100 = 400 \\ 0 \times 10^1 = 0 \times 10 = 0 \\ 8 \times 10^0 = 8 \times 1 = 8 \\ \hline 2408 \end{array}$$

En binaire, chaque chiffre représente un puissance de deux. Par exemple, le nombre binaire 101101 peut s'écrire sous la forme :

$(1 \times 2^5) + (0 \times 2^4) + (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0)$

A titre de référence, les puissances de deux et leurs valeurs décimales sont les suivantes :

...	2 ⁷	2 ⁶	2 ⁵	2 ⁴	2 ³	2 ²	2 ¹	2 ⁰
...	128	64	32	16	8	4	2	1

ANNEXE F

OPERATIONS LOGIQUES SUR LES NOMBRES

L'équivalent décimal de 101101 peut se calculer de la façon suivante :

$$\begin{array}{rcl} 1 \times 2^5 & = & 1 \times 32 = 32 \\ 0 \times 2^4 & = & 0 \times 16 = 0 \\ 1 \times 2^3 & = & 1 \times 8 = 8 \\ 1 \times 2^2 & = & 1 \times 4 = 4 \\ 0 \times 2^1 & = & 0 \times 2 = 0 \\ 1 \times 2^0 & = & 1 \times 1 = 1 \\ & & \underline{\hspace{1cm}} \\ & & 45 \end{array}$$

Pour convertir un nombre décimal en binaire, soustrayez-lui autant de fois que c'est nécessaire la plus grande puissance de 2 qui lui est au plus égal jusqu'à ce qu'il n'y ait plus de reste.

Par exemple, le nombre décimal 77 se convertit en binaire de la façon suivante :

La plus grande puissance de 2 contenue dans le nombre 77 est 64(2⁶). Un 1 est placé en cette position du nombre binaire comme nous le montrons ci-dessous :

128	64	32	16	8	4	2	1
0	1	0	0	0	0	0	0

La soustraction de 64 à 77 donne 13. La plus grande puissance de 2 contenue dans 13 est 8(2³) et un 1 est placé en cette position. La soustraction de 8 à 13 donne 5. La plus grande puissance de 2 contenue dans 5 est 4(2²) et un 1 est placé en cette position. La soustraction de 4 à 5 donne un reste de 1. Placez un 1 à la position 2⁰. Le nombre décimal 77 s'écrit donc en binaire :

128	64	32	16	8	4	2	1
0	1	0	0	1	1	0	1

Vous pouvez vérifier l'exactitude de la conversion de la façon suivante :

$$\begin{array}{rcl} 1 \times 2^6 & = & 1 \times 64 = 64 \\ 0 \times 2^5 & = & 0 \times 32 = 0 \\ 0 \times 2^4 & = & 0 \times 16 = 0 \\ 1 \times 2^3 & = & 1 \times 8 = 8 \\ 1 \times 2^2 & = & 1 \times 4 = 4 \\ 0 \times 2^1 & = & 0 \times 2 = 0 \\ 1 \times 2^0 & = & 1 \times 1 = 1 \\ & & \underline{\hspace{1cm}} \\ & & 77 \end{array}$$

ANNEXE F

OPERATIONS LOGIQUES SUR LES NOMBRES

Opérations logiques

Pour exécuter une opération logique sur des nombres entiers appartenant au domaine de valeurs autorisé, l'ordinateur convertit d'abord ces nombres en leur équivalent binaire sur 16 bits. L'opération s'effectue ensuite bit à bit, et le nombre binaire résultant est reconverti en décimal.

Le bit le plus à gauche étant réservé au signe (0 = positif ; 1 = négatif), le plus grand nombre pouvant être représenté (sur les 15 bits restants) est 32767.

Si un nombre décimal avec partie fractionnaire intervient dans l'opération, il est préalablement arrondi.

Les quatre opérateurs logiques obéissent aux règles suivantes :

Opérateurs	Règles
AND	Si les 2 bits sont des 1, le résultat est 1. Si l'un des bits est 0, le résultat est 0.
OR	Si l'un des bits est 1, le résultat est 1. Si les 2 bits sont des 0, le résultat est 0.
XOR	Si l'un des bits, mais non les deux, est 1, le résultat est 1. Si les deux bits sont identiques, le résultat est 0.
NOT	Si le bit est 0, le résultat est 1. Si le bit est 1, le résultat est 0.

Le tableau suivant donne le résultat des quatre opérations logiques avec toutes les combinaisons possibles de bits.

AND	Premier bit	0	0	1	1
	Second bit	0	1	0	1
	Résultat	0	0	0	1
OR	Premier bit	0	0	1	1
	Second bit	0	1	0	1
	Résultat	0	1	1	1
XOR	Premier bit	0	0	1	1
	Second bit	0	1	0	1
	Résultat	0	1	1	0
NOT	Bit	0	0		
	Résultat	1	0		

ANNEXE F

OPERATIONS LOGIQUES SUR LES NOMBRES

Par exemple, lorsque des opérations logiques sont effectuées sur les nombres 77 et 67, ceux-ci sont d'abord convertis en binaire. Le nombre 77 est représenté sur 16 bits sous la forme 0000000001001101, le nombre 67 sous la forme 0000000001000011. Les résultats de l'exécution d'un AND, d'un OR et d'un XOR sur ces deux valeurs sont présentés ci-dessous :

AND		OR	
(77)	0000000001001101	(77)	0000000001001101
(67)	0000000001000011	(67)	0000000001000011
(65)	0000000001000001	(79)	0000000001001111
XOR			
(77)	0000000001001101		
(67)	0000000001000011		
(14)	0000000000001110		

On peut obtenir les résultats des opérations AND, OR et XOR sur 77 et 67 en introduisant l'expression suivante dans l'ordinateur :

PRINT 77 AND 67;77 OR 67; 77 XOR 67

L'effet de l'opérateur logique NOT sur 77 et 67 est le suivant :

NOT 77		NOT 67	
(77)	0000000001001101	(67)	0000000001000011
(-78)	111111110110010	(-68)	111111110111100

Pour afficher les résultats de NOT 77 et NOT 67, frappez l'expression suivante :

PRINT NOT 77;NOT 67

Remarquez que les résultats de NOT 77 et NOT 67 ont un 1 au niveau du bit le plus à gauche, ce qui dénote un nombre négatif. Dans l'ordinateur un nombre binaire négatif est représenté comme le complément à 2 de sa valeur absolue.

Pour obtenir le complément à 2 d'un nombre binaire, remplacez chaque bit 0 par un 1 et chaque 1 par un 0. Ajouter ensuite 1 à ce nombre modifié. Par exemple, le complément de 2 de 77 est obtenu comme suit :

77 en binaire	0000000001001101
Changez chaque bit	111111110110010
Ajoutez	1
-77 sous la forme d'un complément à 2	11111111010011

Le propos de cette annexe n'est pas de donner une description plus approfondie de l'arithmétique binaire. Consultez un ouvrage de référence à ce sujet pour plus ample information.

ANNEXE G

MESSAGES D'ERREUR

Messages d'erreur

Les listes suivantes décrivent l'origine de chacun des messages d'erreur générés par l'ordinateur. La première liste donne des informations détaillées sur la cause probable des erreurs, par ordre alphabétique. La seconde liste, par ordre de code d'erreur, sert à retrouver le message associé à un code d'erreur particulier.

Traitement d'erreurs

Une fois qu'un message d'erreur a été affiché, les touches [←], [→],[↑],[↓] et [SHIFT] [PB] permettent d'obtenir des informations complémentaires sur l'erreur et de corriger une ligne erronée.

[SHIFT] [PB] est utilisé lors d'une erreur à l'introduction d'une ligne, pour afficher la ligne en question, la modifier et l'introduire de nouveau.

[↑], [↓] sont utilisés lors d'une erreur à l'exécution pour afficher la ligne de programme en cours quand l'erreur s'est produite.

ON ERROR and CALL ERR permettent le traitement des erreurs dans un programme. Consultez le chapitre 5 pour plus ample information.

Liste des messages par ordre alphabétique

Code	Message/Cause
23	Bad argument (argument incorrect) <ul style="list-style-type: none">• Introduction d'un argument non autorisé pour une fonction standard numérique, chaîne, ou de fichier telle que LOG, CHR\$ ou EOF.• Introduction d'un argument non autorisé pour une des clauses facultatives d'une instruction d'entrée/sortie telle que AT, SIZE, VALIDATE et TAB.• Les arguments d'une instruction CALL ne satisfont pas aux spécifications du sous-programme appelé.
07	Bad data (données incorrectes) <ul style="list-style-type: none">• Introduction de plus d'une donnée à la fois pour une instruction INPUT ou ACCEPT.• Données non autorisées d'un fichier, pour une instruction INPUT ou LINPUT.

ANNEXE G

MESSAGES D'ERREUR

Code Message/Cause

- 08 **Bad dimension** (dimension incorrecte)
- Une dimension d'un tableau est négative ou n'est pas une constante numérique.
 - Eléments trop nombreux pour un tableau.
 - Plus de trois dimensions pour un tableau.
 - Absence de virgule entre les dimensions d'un tableau ou absence de parenthèses autour des dimensions.
 - Valeur de l'indice trop grande.
 - Virgule absente entre les indices, ou absence de parenthèses autour de ceux-ci.
 - Nombre d'indices incorrect.
- 14 **Bad program type** (type de programme incorrect)
- Introduction d'une ligne de programme BASIC alors qu'un programme en langage PASCAL, ou non BASIC en général, est en mémoire.
 - Introduction d'une commande SAVE, VERIFY, BREAK liste-de-lignes, UNBREAK liste-de-lignes, NUMBER, RENUMBER, LIST, CONTINUE numéro-de-ligne, RUN numéro-de-ligne, ou DELETE groupe-de-lignes, alors qu'un programme en langage PASCAL, ou non BASIC en général, est en mémoire.
 - Tentative d'appeler (CALL) un programme principal ou de lancer (RUN) un sous-programme.
- 04 **Bad value** (valeur incorrecte)
- Le numéro utilisé dans une instruction ON GOTO ou ON GOSUB est zéro ou supérieur au nombre de lignes existantes.
 - Elévation d'une valeur négative à une puissance fractionnaire.
 - Valeur non autorisée pour une clause facultative dans une instruction d'entrée/sortie, comme AT, SIZE, REC et VARIABLE.
 - Tentative d'opération logique (AND, OR, XOR ou NOT) avec une valeur inférieure à - 32768 ou supérieure à 32767.

ANNEXE G

MESSAGES D'ERREUR

Code Message/Cause

- 29 **Break** (interruption)
- Un point d'arrêt a été rencontré ou la touche BREAK a été pressée.
- 10 **Can't do that** (ne peut se faire)
- Tentative d'effectuer une opération sur les chaînes en tant que calcul immédiat.
 - Introduction de la commande CONTINUE alors qu'il n'y a pas eu rencontre d'un point d'arrêt.
 - Rencontre d'une instruction SUBEXIT ou SUBEND alors qu'aucun sous-programme n'a été appelé. Par exemple, CONTINUE numéro-de ligne a désigné une ligne de sous-programme après que le programme principal s'est arrêté à un point d'arrêt.
- 02 **Complex** (trop complexe)
- Trop de fonctions, d'opérateurs, de niveaux d'imbrication de parenthèses, l'expression doit être simplifiée ou exécutée en deux ou plusieurs étapes dans des instructions séparées.
- 27 **Contents may be lost** (le contenu peut être altéré)
- A la mise en marche du système, le contenu de la mémoire permanente n'est pas le même que lors du dernier arrêt. Cependant, certaines données du système étant correctes, l'altération de la mémoire ne peut être importante. Ce message apparaît souvent quand on appuie sur la touche de réinitialisation [RESET], l'ordinateur étant en marche.
- 32 **DATA error** (erreur données)
- Données épuisées dans le programme ou sous-programme en cours.
 - Liste des données incorrecte dans une instruction DATA. Par exemple, absence de virgule entre les éléments.
 - Lors d'une tentative de lecture d'un élément numérique, la donnée lue n'est pas une représentation autorisée d'une constante numérique.

ANNEXE G

MESSAGES D'ERREUR

Code	Message/Cause
------	---------------

26	Division by zero (division par zéro)
----	--------------------------------------

- L'évaluation d'une expression numérique comporte une division par zéro ; le résultat est remplacé par 9.999999999999E + 127 accompagné du signe algébrique approprié.

24	Extension (extension)
----	-----------------------

- Tentative d'exécution d'une instruction ou d'une commande BASIC étendue non prévue dans le système.
- Peut également se produire quand le contenu de la mémoire a été modifié par erreur. (Voir Erreur système).

19	File error (erreur fichier)
----	-----------------------------

- Le numéro-de-fichier spécifié dans une instruction OPEN désigne un fichier déjà ouvert.
- Le numéro-de-fichier dans une instruction d'entrée/sortie autre que OPEN, ne désigne pas de fichier ouvert.
- Le numéro-de-fichier ou le numéro-de-périphérique dans une instruction d'entrée/sortie est supérieur à 255.
- Tentative d'exécution d'une instruction INPUT ou LINPUT à partir d'un fichier ouvert en OUTPUT ou APPEND.
- Tentative d'exécution d'une instruction LINPUT à partir d'un fichier de type INTERNAL.
- Tentative d'exécution de l'instruction PRINT en direction d'un fichier ouvert en mode INPUT.
- Utilisation de la clause REC dans une instruction d'entrée/sortie qui accède à un fichier de type SEQUENTIAL.
- Absence de point ou de virgule après le numéro d'appareil dans la spécification d'un périphérique ou d'un nom-de-fichier.

ANNEXE G

MESSAGES D'ERREUR

Code	Message/Cause
------	---------------

06	FOR/NEXT error (erreur FOR/NEXT)
----	----------------------------------

- Plus d'instructions FOR que d'instructions NEXT dans un programme ou un sous-programme. Nota : le numéro de ligne affiché est celui de la dernière ligne du programme ou sous-programme courant, et non celui de la ligne contenant un FOR non apparié.
- Plus d'instructions NEXT que d'instructions FOR dans un programme ou un sous-programme.
- La variable-de-commande dans une instruction NEXT ne correspond pas à la variable-de-commande dans une instruction FOR correspondante.
- Une instruction NEXT exécutée sans avoir exécuté au préalable une instruction FOR correspondante.
- Trop de niveaux de boucles FOR NEXT imbriquées.
- Même variable de commande utilisée dans des boucles FOR NEXT imbriquées.

18	IMAGE error (erreur format)
----	-----------------------------

- Introduction d'une chaîne vide comme format.
- Spécification d'un modèle numérique avec plus de 14 chiffres significatifs.
- La liste comporte un élément à sortir alors que la chaîne format n'a que des modèles littéraux.

16	In use (actif)
----	----------------

- Les sous-programmes ne peuvent s'appeler eux-mêmes, directement ou indirectement.

30	Initialized (réinitialisé)
----	----------------------------

- S'affiche quand les circonstances imposent une réinitialisation complète du système. C'est le cas lorsque se produit l'une des situations suivantes à la mise en marche :
 - l'ordinateur constate que le contenu de la mémoire a été détruit (par exemple, après remplacement des piles).

ANNEXE G MESSAGES D'ERREUR

Code Message/Cause

30 **Initialized** (réinitialisé) (*Cont.*)

- l'ordinateur constate que l'extension de RAM préalablement ajoutée (par l'intermédiaire du sous-programme ADDMEM) n'est plus en service.
- Ce message peut également apparaître quand on appuie sur le bouton de réinitialisation, puisque l'on effectue alors une grande part de la vérification de la mémoire.

00 **I / O error** (erreur d'entrée/sortie)

- Une erreur a été constatée par un périphérique lors d'une instruction ou d'une commande d'entrée/sortie (I/O), ou au moment de l'utilisation de la fonction EOF. Un code d'I/O spécial est transmis par le périphérique et affiché à la suite. Les codes d'erreur I/O les plus courants figurent dans la section CODE D'ERREUR I/O de la présente annexe.

Le code d'erreur est suivi du numéro-de-fichier ou du numéro-de-périphérique, selon l'instruction ou commande en cours d'exécution. Le signe dièse # indique un numéro-de-fichier, des guillemets indiquent un numéro-de-périphérique. Les codes d'erreurs courantes ainsi que les erreurs I/O propres aux périphériques sont décrits dans les manuels correspondants.

11 **Line number error** (erreur de numérotation de ligne)

- Ligne spécifiée dans BREAK, CONTINUE, DELETE, GOSUB, GOTO, ON ERROR, USING, RESTORE, RUN ou BREAK, non trouvée.
- RENUMBER n'a pas trouvé une ligne référencée. La référence est remplacée par 32767, qui n'est pas un numéro de référence autorisé.
- Une instruction BASIC fait référence à un numéro de ligne inférieur au premier numéro de ligne (ou supérieur au dernier) du programme ou sous-programme en cours.

ANNEXE G MESSAGES D'ERREUR

Code Message/Cause

11 **Line number error** (*Cont.*)
(erreur de numérotation de ligne)

- Le numéro de ligne spécifié dans une instruction ou une commande est inférieur à 1 ou supérieur à 32766.
- La commande RENUMBER a généré un numéro de ligne supérieur à 32766.

127 **Memory full** (mémoire pleine)

- Espace insuffisant pour ajouter, insérer ou modifier une ligne de programme.
- Espace insuffisant pour allouer des variables à un programme ou un sous-programme.
- Mémoire insuffisante pour affecter un espace à une valeur-chaîne.
- Espace insuffisant pour charger un programme ou un sous-programme en mémoire.
- Espace insuffisant pour ouvrir (OPEN) un fichier ou un périphérique.
- Espace insuffisant pour affecter une chaîne à une touche utilisateur.

03 **Mismatch** (défaut de concordance)

- Utilisation d'un argument-chaîne à la place d'un argument-numérique ou d'un argument-numérique à la place d'un argument-chaîne.
- Affectation d'une valeur-chaîne à une variable numérique ou d'une valeur numérique à une variable-chaîne.
- Variable ou expression numérique introduite en tant que message-guide dans une instruction INPUT ou LINPUT.

12 **Missing statement** (absence d'instruction)

- Une sous-routine de traitement d'erreurs se termine par une instruction SUBEXIT ou SUBEND au lieu d'une instruction RETURN.
- Absence de SUBEND dans un sous-programme.

ANNEXE G

MESSAGES D'ERREUR

Code	Message/Cause
12	Missing statement (absence d'instruction) (<i>Cont.</i>) <ul style="list-style-type: none">• Rencontre d'une instruction SUB à l'intérieur d'un sous-programme ; un sous-programme ne peut en contenir un autre.• Exécution d'une instruction RETURN sans exécution préalable de l'instruction GOSUB correspondante.
20	Name table full (table des noms pleine) <ul style="list-style-type: none">• Définition de plus de 95 noms de variables.
31	No RAM (pas de mémoire vive) <ul style="list-style-type: none">• Appel du sous-programme ADDMEM sans qu'un module soit en place, ou avec un module ne contenant pas de mémoire vive.• Appel de PUT ou de GET sans qu'un module soit en place, ou avec un module ne contenant pas de mémoire vive.
17	Not defined (non défini) <ul style="list-style-type: none">• Tentative de calcul avec une variable qui n'a pas été définie.• Rencontre d'une variable non définie dans un programme ou un sous-programme. Cette erreur peut se produire quand CONTINUE numéro-de-ligne désigne une ligne qui n'est pas dans le même programme ou sous-programme que celui dans lequel l'interruption est intervenue.
13	Not found (non trouvé) <ul style="list-style-type: none">• L'instruction RUN ne trouve pas de programme spécifié.• L'instruction CALL ne trouve pas le sous-programme spécifié.
25	Overflow (dépassement de capacité) <ul style="list-style-type: none">• Introduction d'une valeur numérique ou évaluation d'une expression numérique ayant pour résultat un nombre dont la valeur absolue est supérieure à $9.999999999999999E + 127$; cette valeur est remplacée par $9.999999999999999E + 127$ accompagné du signe algébrique approprié.

ANNEXE G

MESSAGES D'ERREUR

Code	Message/Cause
21	Parenthesis (parenthèse) <ul style="list-style-type: none">• Une instruction ou expression ne contient pas le même nombre de parenthèses ouvertes et fermées.• Les parenthèses ouvertes ou fermées dans une instruction ou une expression ne sont pas appariées. Par exemple SIN(1 +)PI/2(alors qu'on voulait écrire SIN(1 + (PI/2)).
09	Previously defined (déjà défini) <ul style="list-style-type: none">• La variable d'une instruction DIM figure déjà dans le programme ou sous-programme en cours.• Variable référencée en utilisant un nombre de dimensions inexact. Par exemple, une variable est d'abord utilisée en tant que variable simple, puis en tant que tableau dans le même programme ou sous-programme.
15	Protection error (erreur sur un programme protégé) <ul style="list-style-type: none">• Tentative d'insérer, de supprimer ou de corriger une ligne dans un programme protégé.• Tentative d'exécuter LIST, SAVE, NUMBER ou RENUMBER sur un programme protégé.
05	Stack underflow (pile vide) <ul style="list-style-type: none">• Tentative de retirer une valeur de la pile d'exécution alors qu'elle est vide. Cette erreur ne se produit que lorsque le contenu de la mémoire a été modifié par erreur (voir System error).
01	Syntax (syntaxe) <ul style="list-style-type: none">• Parenthèse(s) ou guillemet(s) absent(s) ou manquant(s).• Séparateur d'instructions (:) ou symbole de remarque finale (!) absent.

ANNEXE G

MESSAGES D'ERREUR

Code Message/Cause

01 Syntax (syntaxe) (Cont.)

- Virgule(s) absente(s) ou excédentaire(s).
Par exemple :
 - entre des arguments dans une liste-d'arguments
 - entre des numéros de ligne dans une liste-de-lignes
 - entre des variables dans une liste-de-variables
 - après le numéro-de-fichier dans des instructions d'entrée/sortie.
- Absence de trait d'union entre deux numéros de lignes.
- Caractère non autorisé dans une instruction. Par exemple, "%", "?", etc., ne sont autorisés qu'à l'intérieur de guillemets ou dans une instruction IMAGE ou REM.
- Caractère non autorisé à l'intérieur d'une constante numérique.
- Mot-clé mal placé. Par exemple :
 - utilisation de DIM ou de SUBEND après une instruction DIM dans une ligne à instructions multiples
 - instruction ne commençant pas par un mot-clé "instruction" comme TO, ERROR, VARIABLE, SIZE
 - faute d'orthographe entraînant la confusion entre un mot-clé et une variable
 - mot-clé utilisé comme variable, comme dans ON VAL GOTO ou IF STOP = 1 THEN
- Absence de mot-clé. Par exemple :
 - pas de TO après FOR
 - pas de THEN après IF
 - pas de GOTO ni de GOSUB après ON expression-numérique
 - pas de STOP, NEXT ni d'ERROR après ON BREAK
 - pas de PRINT, NEXT ni d'ERROR après ON WARNING

ANNEXE G

MESSAGES D'ERREUR

Code Message/Cause

01 Syntax (syntaxe) (Cont.)

- Utilisation de l'instruction SUB après la première instruction dans une ligne à instructions multiples
- Instruction autre que REM, !, END ou SUB rencontrée après une instruction SUBEND
- Nom-de-fichier absent ou incorrect dans OLD, SAVE, VERIFY, ou dans la commande DELETE.
- Répétition d'une option dans une instruction d'entrée/sortie. Par exemple :
 - plus d'un AT, SIZE, ERASE ALL dans un ACCEPT ou DISPLAY
 - plus d'une expression-chaîne dans VALIDATE
 - plus d'un mode-d'ouverture, d'un type-de-fichier, ou d'une organisation-de-fichier dans OPEN
- Absence d'argument ou de clause. Par exemple :
 - pas de valeur limite après TO ou de pas après STEP
 - pas de numéro de ligne ou d'instruction après THEN ou ELSE
 - pas de constante-chaîne après IMAGE
 - pas de numéro-de-ligne ou d'expression-chaîne après USING
 - pas de valeur avant ni après un opérateur binaire tel que *, /, ^ ou &
 - pas de variable d'entrée après INPUT, LINPUT, ACCEPT ou READ
- Argument ou clause non autorisés. Par exemple :
 - une variable-chaîne est utilisée en tant que variable-de-commande dans FOR
 - une variable numérique est utilisée en tant que variable d'entrée dans LINPUT
 - VALIDATE ou NULL est utilisé dans une instruction DISPLAY
 - USING ou TAB est utilisé avec un fichier de type INTERNAL
 - la dimension d'un élément à écrire dépasse celle de l'enregistrement pour un fichier de type INTERNAL.

ANNEXE G MESSAGES D'ERREUR

Code Message/Cause

- 01 Syntax (syntaxe) (*Cont.*)
- Les instructions ACCEPT, CALL avec des sous-programmes en langage BASIC, GOSUB, GOTO, INPUT, LINPUT, ON ERROR numéro-de-ligne, ON GOSUB, ON GOTO, READ, RESTORE numéro-de-ligne, SUB, SUBEXIT et SUBEND ne peuvent être exécutées que dans un programme.
 - Instructions SUBEXIT ou SUBEND rencontrées dans un programme principal.
 - Utilisation de CALL ADDMEM, CONTINUE, DELETE groupe-de-lignes, LIST, NEW, NUMBER, OLD, RENUMBER, SAVE ou VERIFY dans un programme.
- 126 System error (erreur système)
- Ce message apparaît généralement quand le contenu de la mémoire a été altéré ou modifié par erreur. Par exemple, la mémoire peut avoir été modifiée par défaut d'alimentation.
- 22 Too Long (trop long)
- La représentation interne d'une ligne de programme ou d'une ou plusieurs instructions immédiates est trop longue.
 - La représentation externe d'une ligne de programme dépasse les 80 caractères autorisés.
 - Plus de 15 caractères dans un nom de variable ou de sous-programme.
- 28 Truncation (troncature)
- Une opération sur une chaîne (concaténation ou RPT\$) a produit une chaîne comportant plus de 255 caractères ; les caractères excédentaires sont supprimés.

ANNEXE G MESSAGES D'ERREUR

Liste des codes d'erreur par ordre croissant

Code	Message	
00	I/O error	(erreur d'entrée/sortie)
01	Syntax	(syntaxe)
02	Complex	(complexe)
03	Mismatch	(défaut de concordance)
04	Bad value	(valeur incorrecte)
05	Stack underflow	(pile vide)
06	FOR/NEXT error	(erreur FOR/NEXT)
07	Bad data	(données incorrectes)
08	Bad dimension	(dimension incorrecte)
09	Previously defined	(déjà défini)
10	Can't do that	(ne peut se faire)
11	Line number error	(erreur de numérotation de ligne)
12	Missing statement	(absence d'instruction)
13	Not found	(non trouvé)
14	Bad program type	(type de programme incorrect)
15	Protection error	(erreur : programme protégé)
16	In use	(actif)
17	Not defined	(non défini)
18	Image error	(erreur format)
19	File error	(erreur fichier)
20	Name table full	(table des noms pleine)
21	Parenthesis	(parenthèse)
22	Too long	(trop long)
23	Bad argument	(argument incorrect)
24	Extension missing	(extension absente)
25	Overflow	(dépassement de capacité)
26	Division by zero	(division par zéro)
27	Contents may be lost	(le contenu peut être altéré)
28	Truncation	(troncature)
29	Break	(interruption)
30	Initialized	(initialisé)
31	No RAM	(pas de mémoire vive)
32	DATA error	(erreur données)
126	System error	(erreur système)
127	Memory full	(mémoire pleine)

ANNEXE G

MESSAGES D'ERREUR

Codes d'erreurs d'entrée/sortie

La liste suivante explique les codes d'erreurs d'entrée/sortie (I/O) standard. Certains périphériques peuvent avoir des codes d'erreurs supplémentaires ; si c'est le cas, ceux-ci sont expliqués dans le manuel du périphérique.

Les codes d'erreurs d'I/O sont affichés sous l'une des formes suivantes :

- I/O error ccc # fff
- I/O error ccc "ddd"

Où ccc est le code d'erreur I/O expliqué ci-dessous ou dans le manuel du périphérique, fff est le numéro du fichier affecté dans une instruction OPEN, et ddd est le code du périphérique correspondant à l'unité utilisée.

Code	Définition
------	------------

- | | |
|---|---|
| 1 | ERREUR OPTIONS PERIPHERIQUE/FICHER <ul style="list-style-type: none">• Option incorrecte ou non autorisée dans "<i>périphérique.nom-de-fichier</i>".• <i>Nom-de-fichier</i> trop long ou absent dans "<i>périphérique.nom-de-fichier</i>". |
| 2 | ERREUR ATTRIBUTS <ul style="list-style-type: none">• Des attributs incorrects (<i>type-de-fichier</i>, <i>organisation-de-fichier</i>, <i>mode-d'ouverture</i>, <i>longueur-d'enregistrement</i>) sont spécifiés à l'ouverture d'un fichier existant. |
| 3 | FICHER NON TROUVE <ul style="list-style-type: none">• Le fichier spécifié dans l'une des opérations suivantes n'existe pas :<ul style="list-style-type: none">— Instruction OPEN en mode INPUT— OLD "<i>périphérique.nom-de-fichier</i>"— RUN "<i>périphérique.nom-de-fichier</i>"— DELETE "<i>périphérique.nom-de-fichier</i>" |
| 4 | PERIPHERIQUE/FICHER NON OUVERT <ul style="list-style-type: none">• Tentative d'accéder à un fichier fermé, dans une opération INPUT, LINPUT, PRINT ou CLOSE.• Le fichier spécifié dans la fonction EOF est fermé. |

ANNEXE G

MESSAGES D'ERREUR

Code	Définition
------	------------

- | | |
|----|--|
| 5 | PERIPHERIQUE/FICHER DEJA OUVERT <ul style="list-style-type: none">• Tentative d'ouverture (OPEN) ou de suppression (DELETE) d'un fichier ouvert.• Tentative de "formater" un support physique sur un périphérique qui a un fichier ouvert. |
| 6 | ERREUR PERIPHERIQUE <ul style="list-style-type: none">• Une panne s'est produite dans un périphérique. Cette erreur peut intervenir quand le répertoire d'une bande magnétique est altéré, quand le périphérique détecte une erreur de transmission, quand le support est abîmé etc. |
| 7 | FIN DE FICHER <ul style="list-style-type: none">• Tentative de lecture après la fin d'un fichier. |
| 8 | TROP DE DONNEES <ul style="list-style-type: none">• Tentative d'écriture d'un enregistrement trop long pour la capacité du périphérique.• Un fichier dépasse la longueur maximale sur le périphérique concerné. |
| 9 | ERREUR PROTECTION ECRITURE <ul style="list-style-type: none">• Tentative de "formatage" d'un support physique protégé en écriture.• Tentative d'ouverture d'un fichier protégé en écriture, en mode OUTPUT ou UPDATE.• Tentative de suppression d'un fichier sur un support physique protégé en écriture. |
| 10 | PAS DE SERVICE DEMANDE <ul style="list-style-type: none">• Réponse à une demande de service quand le périphérique spécifié n'a pas fait de demande. (Ce code est utilisé dans des applications spéciales et ne devrait pas apparaître durant l'exécution normale de programmes BASIC.) |
| 11 | REPertoire PLEIN <ul style="list-style-type: none">• Tentative d'ouverture (OPEN) d'un nouveau fichier sur un périphérique dont le répertoire est complet. |

ANNEXE G

MESSAGES D'ERREUR

Code	Définition
------	------------

12	ERREUR TAILLE TAMPON
----	----------------------

- Longueur d'enregistrement spécifiée (VARIABLE XXX) inférieure à celle du plus grand enregistrement existant dans un fichier ouvert en entrée ou mise à jour.
- La commande VERIFY constate que le programme en mémoire est plus court que celui sur le support physique.

13	COMMANDE NON PREVUE
----	---------------------

- Tentative d'opération non acceptée par le périphérique.

14	PERIPHERIQUE NON OUVERT EN SORTIE (OUTPUT)
----	--

- Tentative d'écriture sur un fichier ou un périphérique ouvert en lecture.

15	PERIPHERIQUE NON OUVERT EN ENTREE (INPUT)
----	---

- Tentative de lecture d'un fichier ou d'un périphérique ouvert en écriture ou en APPEND.

16	ERREUR TOTAL DE CONTROLE
----	--------------------------

- Le total de contrôle calculé sur un enregistrement lu est incorrect.

17	FICHIERS A ACCES DIRECT NON PREVU
----	-----------------------------------

- Le périphérique spécifié dans OPEN n'accepte pas une organisation de fichier de type RELATIVE.

19	MODE APPEND NON PREVU
----	-----------------------

- Le périphérique spécifié dans l'instruction OPEN n'accepte pas le mode APPEND.

20	MODE SORTIE (OUTPUT) NON PREVU
----	--------------------------------

- Le périphérique spécifié dans l'instruction OPEN n'accepte pas le mode écriture.

21	MODE ENTREE (INPUT) NON PREVU
----	-------------------------------

- Le périphérique spécifié dans l'instruction OPEN n'accepte pas le mode lecture.

ANNEXE G

MESSAGES D'ERREUR

Code	Définition
------	------------

22	MODE MISE A JOUR (UPDATE) NON PREVU
----	-------------------------------------

- Le périphérique spécifié dans l'instruction OPEN n'accepte pas le mode mise à jour (lecture ou écriture).

23	ERREUR TYPE FICHIER
----	---------------------

- Le type de fichier dans l'instruction OPEN n'est pas accepté par le périphérique désigné.
- Le type de fichier spécifié dans l'instruction OPEN ne coïncide pas avec celui du fichier ou du périphérique existant.

24	ERREUR A LA VERIFICATION
----	--------------------------

- Le programme ou les données en mémoire ne coïncident pas avec le programme ou le support physique spécifié.

25	PILES PERIPHERIQUES FAIBLES
----	-----------------------------

- Tentative d'une opération I/O avec un périphérique dont les piles sont faibles.

26	SUPPORT NON "FORMATE"
----	-----------------------

- Tentative d'ouverture d'un fichier sur un support non initialisé.
- Tentative d'ouverture d'un fichier sur un support qui a été accidentellement effacé ou détérioré.

32	SUPPORT PLEIN
----	---------------

- Plus d'espace disponible sur le support physique.

255	ERREUR TEMPS DEPASSE
-----	----------------------

- Communication interrompue avec le périphérique spécifié.
- Le périphérique spécifié n'est pas raccordé au bus I/O.

ANNEXE H
PRECISION DES NOMBRES

Précision des nombres

Précision des calculs

Le TI-74, comme tous les ordinateurs, fonctionne selon des règles déterminées à l'intérieur de limites pré-établies. Sa précision mathématique dépend du nombre de chiffres utilisés lors des calculs.

L'ordinateur utilise un minimum de 13 chiffres pour effectuer les calculs.

En l'absence d'autres indications, les résultats sont arrondis à 10 chiffres. La technique d'arrondi utilisée consiste à ajouter 1 au chiffre le moins significatif affiché si le premier chiffre non affiché est égal ou supérieur à 5. Sans ces chiffres supplémentaires, des résultats imprécis tels que le suivant seraient fréquemment affichés :

$1/3 \times 3 = .9999999999$

Ceci vient du fait que la représentation interne finie de 1/3 est 0,3333333333. Ici grâce au fait que 1/3x3 est arrondi à 10 chiffres, la réponse 1 . est affichée.

Les fonctions mathématiques les plus complexes sont calculées selon des méthodes itératives et polynômiales. La propagation des erreurs d'arrondi est normalement limitée aux chiffres qui suivent le dixième, de façon à ce que les valeurs affichées soient exactes. Il est alors inutile de prendre en considération les chiffres non affichés. Cependant, certains calculs peuvent amener ces chiffres supplémentaires à être significatifs, comme ci-dessous :

$2/3 = 0,66666666666667$ et $1/3 = 0,33333333333333$
 $2/3 - 1/3 - 1/3 = 0,00000000000001$ (affiché 1.E-14)

De tels écarts dans les chiffres les moins significatifs du résultat d'un calcul sont importants lors de la comparaison de ce résultat avec une autre valeur, à la recherche d'une égalité. Des précautions doivent alors être prises pour éviter des interprétations incorrectes.

Une technique utile consiste à déterminer si les deux valeurs sont suffisamment proches l'une de l'autre plutôt que strictement égales. Par exemple :

Au lieu d'utiliser

IF X = Y THEN ...

ANNEXE H
PRECISION DES NOMBRES

Ecrivez

IF ABS(X-Y) < 1E-11 THEN ...

Représentation interne des nombres

L'ordinateur utilise la numérotation à base 100 pour les calculs internes. Un chiffre unique en base 100 s'échelonne entre 0 et 99 en base 10. L'ordinateur utilise une mantisse à 7 chiffres, soit une précision décimale de 13 à 14 chiffres. Un exposant en base 100 a pour valeurs extrêmes -64 et +63, qui donnent des puissances décimales entre 10^{-128} et 10^{+126} . L'exposant et la mantisse à 7 chiffres combinés permettent de représenter les valeurs décimales situées entre -9,999999999999E + 127 et -1,E - 128 ; zéro ; et entre +1,E - 128 et +9,999999999999E + 127.

Huit octets sont nécessaires à la représentation interne d'un nombre en base 100. Le premier octet contient l'exposant et le signe algébrique de tout le nombre "en virgule flottante". L'exposant est une valeur hexadécimale de 7 bits, décalée de 40_{16} (dans la présente annexe, l'indice 16 indique les valeurs hexadécimales). La correspondance entre les valeurs d'exposant est la suivante :

Valeur hexadécimale décalée	00_{16}	à	40_{16}	à	$7F_{16}$
Valeur en base 100	-64	à	0	à	+63
Valeur décimale	-128	à	0	à	+126

Si le nombre est négatif, le premier octet (la valeur de l'exposant) est complété à 1. Chaque octet de la mantisse contient un chiffre en base 100 en décimal codé binaire (BCD). En d'autres termes, les quatre bits les plus significatifs de chaque octet représentent un chiffre décimal de 0 à 9 et les quatre bits les moins significatifs représentent un chiffre décimal de 0 à 9. Le premier octet de la mantisse contient le chiffre le plus significatif du nombre en base 100. Enfin, le nombre est normalisé, c'est-à-dire que la virgule, si elle était représentée, viendrait immédiatement après le chiffre le plus significatif en base 100.

ANNEXE H

PRECISION DES NOMBRES

Les exemples suivants présentent quelques valeurs décimales et leur représentation interne.

Nombre décimal	Valeur interne							
1	40 ₁₆	01 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆
10	40 ₁₆	10 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆
100	41 ₁₆	01 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆
1234	41 ₁₆	12 ₁₆	34 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆	00 ₁₆
PI	40 ₁₆	03 ₁₆	14 ₁₆	15 ₁₆	92 ₁₆	65 ₁₆	35 ₁₆	00 ₁₆
- PI	BF ₁₆	03 ₁₆	14 ₁₆	15 ₁₆	92 ₁₆	65 ₁₆	35 ₁₆	00 ₁₆

ANNEXE I

EN CAS DE DIFFICULTES

En cas de difficultés

En cas de problèmes avec votre calculatrice, reportez-vous aux instructions ci-après qui vous aideront à les résoudre.

Procédure en cas de difficultés

1. Assurez-vous que les piles sont correctement placées.
2. Tournez le bouton de réglage du contraste situé à droite de la calculatrice pour rendre l'affichage visible.
3. Si la difficulté rencontrée entraîne des erreurs de calcul ou si la calculatrice ne réagit pas à des entrées par le clavier :
 - Pressez **[CE/C]** s'il y a un message d'erreur, ou **[ON]** si le dispositif d'arrêt automatique APD™ a éteint la calculatrice.
 - Pressez **[RESET]** **[CE/C]** et vérifiez que la machine est en mode calculatrice. Vérifiez la valeur dans chaque mémoire et refaites les calculs.
 - Vérifiez les instructions de fonctionnement pour vous assurer que les calculs ont été effectués comme décrits dans la présente notice. L'emploi impropre de séquences de touches peut donner des résultats inattendus.
4. Si la difficulté rencontrée entraîne des erreurs au niveau de l'ordinateur ou si celui-ci ne réagit pas à des entrées par le clavier :
 - Si l'indicateur d'entrée/sortie (I/O) est allumé, **[BREAK]** est sans effet. Si cet indicateur n'est pas allumé, appuyez sur la touche **[BREAK]** pour interrompre l'exécution. Si le mot BREAK apparaît à l'affichage, pressez **[CLR]** et faites CON pour poursuivre l'exécution du programme en mémoire.
 - Si la touche **[BREAK]** est sans effet, appuyez sur le bouton de réinitialisation. Le message W27 contents may be lost est affiché. Appuyez sur **[CLR]** pour effacer l'affichage. Vérifiez ensuite si le programme est toujours en mémoire.

ANNEXE I EN CAS DE DIFFICULTES

- Si une pression sur le bouton de réinitialisation n'entraîne pas la réapparition du curseur, retirez les piles pendant quelques minutes. Normalement, le système est alors initialisé et tout programme se trouvant en mémoire est effacé.
5. En cas de problème pendant l'utilisation d'un périphérique :
- Vérifiez l'indicateur I/O. S'il est allumé et donc que des opérations d'entrée/sortie sont en cours, attendez la fin de toute activité périphérique.
 - Vérifiez que tous les périphériques sont correctement reliés à l'ordinateur et alimentés. Les câbles utilisés doivent être les mêmes que ceux destinés aux périphériques de la série TI-70. Vérifiez que les câbles sont correctement enfichés et qu'il n'y a pas de fils coupés ou de prises mal serrées. Ces vérifications peuvent éventuellement faire apparaître un problème évident et facile à résoudre.
 - Appuyez sur **[RESET]** puis éteignez le système. Isolez l'ordinateur en débranchant le câble du périphérique. Mettez l'ordinateur en marche.
 - Si le curseur clignotant apparaît à l'affichage, éteignez l'ordinateur et raccordez un périphérique. Mettez d'abord le périphérique en marche puis l'ordinateur.
 - Si le curseur clignotant apparaît, éteignez l'ordinateur et raccordez les périphériques les uns après les autres. Si le curseur clignotant ne réapparaît pas une fois que l'un des périphériques est raccordé, c'est soit le câble soit le périphérique qui peut être à l'origine du problème. Débranchez le périphérique et raccordez-le avec un câble réputé en bon état. Mettez l'ordinateur en marche, vous êtes alors en mesure d'établir si le problème réside dans le câble ou dans l'unité externe.

Si le problème persiste

Si ces procédures ne résolvent pas le problème rencontré, veuillez alors rapporter la calculatrice, ainsi que les extensions périphériques s'il y a lieu, à votre revendeur. Reportez-vous à l'annexe K concernant l'application de la garantie.

ANNEXE J INDEX

Index

Utilisez cette liste pour trouver un sujet de référence. Reportez-vous également à l'index des touches.

A

ABS — 81
ACOS — 86
ACOSH — 87
ACCEPT — 82
Addition en mémoire — 25
ADDMEM — 88
Affectation de valeurs — 63
Affectation d'un texte — 59
Affichage — 15, 49, 64
Affichage en attente — 168
AND — 71, 253
Angles
 conversions — 28
 unités — 28
APPEND — 161
Arc-cosinus — 30, 86
Arc-sinus — 30, 90
Arc-tangente — 30, 92
Argument,
 sous-programme — 73, 199
Arrêt — 5
Arrêt automatique — 5
ASC — 89
ASIN — 90
ASINH — 91
AT — 82, 107
ATANH — 92
ATN — 93

B

BASIC

 Calculs — 57
 Touches de commande — 56
 Mode — 7
 mots-clés — 80, 242
Boucle — 72
Branchement d'un magnétophone — 216
BREAK — 94

ANNEXE J

INDEX

C

Calculs arithmétiques — 19
CALL — 73, 95
 ADDMEM — 88
 ERR — 112
 GET — 119
 IO — 137
 KEY — 138
 PUT — 176
Caractères — 244
Carré — 24
Chaîne vide — 68
Chaîne
 constante — 67
 fonctions — 68
CHR\$ — 96
Clavier de commande — 56
Clavier de fonction — 52
Clavier en position majuscules — 54
Clavier primaire — 50
CLOSE — 97
Code d'erreurs d'entrée/sortie (I/O) — 268, 269, 270, 271
Codes caractères ASCII — 244
Coefficient de corrélation — 40, 43
Combinaisons — 37
Commandes — 56
Comparaisons — 70
Concaténation — 68
CONTINUE — 98
Contrôle (instructions de) — 72
Coordonnées polaires/cartésiennes — 35
Copie d'un programme en mémoire — 176
COS — 99
Cosinus — 29
COSH — 100
Curseur — 49

ANNEXE J

INDEX

D

DATA — 101
DEG — 103
Degrés — 103
 décimaux — 30
 minutes/secondes — 30
 mesure d'angle — 29
DELETE — 104
Difficultés, en cas de — 275
Difficultés, magnétophone — 215
DIM — 106
Dimensions — 69, 106
DISPLAY — 107
Domaine des valeurs de sortie — 241
Domaine des valeurs d'entrée — 240
Données de type Display — 161
Duplication de ligne — 62

E

Ecart de pourcentage — 33
Ecart type — 41, 43
Echange des valeurs d'un couple — 34
Echange mémoire — 25
Effacement de la calculatrice — 16
Effacement de la mémoire — 25
Effacement du programme — 147
Effacement global — 147
END — 110
Entrées — 17
Entrée
 de couples — 34
 domaine des valeurs — 240
EOF — 111
ERASE ALL — 83, 107
ERR — 112
Erreur de numérotation de ligne — 260
Erreur
 correction — 22
 causes — 239
Exemple de force d'un ressort — 42
EXP — 114
Exponentielle Séquences de touches — 26
Exponentielles — 26

ANNEXE J

INDEX

Extension mémoire — 230

F

Factorielle — 37

Fichier de données — 75

Fichier relatif — 161

Fichier

 cassette de — 162

 nom de — 75

 organisation de — 161

Fichiers à accès

 direct — 161

 séquentiel — 162

Fichiers de type

 INTERNAL — 161

Flèche en haut [↑] — 51, 53, 54, 55

Fonctions hyperboliques — 26, 67

Fonctions hyperboliques inverses — 27, 67

Fonctions inverses — 14

Fonctions secondes — 12

Fonctions trig inverses — 30, 66

Fonctions trig. — 29, 66

FORMAT — 117

FOR TO STEP — 115, 116

FRE — 118

G

Garantie — 287

GET — 119

GOSUB — 120

GOTO — 121

GRAD — 122

H

Hiérarchie — 19

Hiérarchie algébrique — 19

I

IF THEN ELSE — 123, 124

IMAGE — 125, 126

Impression — 45, 167, 228, 229

ANNEXE J

INDEX

Imprimante — 45, 167, 228, 229

Indicateurs — 15, 50

INPUT (avec fichiers) — 133

 (avec clavier) — 130

Instructions — 62

Instructions multiples — 62

INT — 136

INTERNAL — 161

Inverse — 14

K

KEY — 138

KEY\$ — 139

L

LEN — 140

LET — 141

Limites, trigonométriques — 240

LINPUT — 142

LIST — 144

LN — 26, 145

LOG — 26, 146

Logarithme — 26, 145, 146

Logarithme décimal — 26, 146

Logarithme népérien — 26, 145

Longueur d'enregistrement — 161

M

Magnétophone à cassette — 215

 branchements — 216

 réglages — 218

Majuscules — 54

Mantisse — 19

Messages — 235, 255

Messages cassette — 220

Messages d'erreur — 255

Mémoire — 25

Mémoire disponible — 48

Mémoire module — 299

Mode calculatrice — 11

Mode OUTPUT — 161

Mode ouvert — 161

ANNEXE J

INDEX

Mode UPDATE — 161
Modèle de nombre entier — 27
 fonction (INT) — 136
Modèle d'un nombre décimal — 28
Modification des informations affichées — 55
Mots réservés — 240
Mots-clés par ordre alphabétique — 242
Moyenne — 40, 41, 44

N

NEW — 147
NEXT — 148
Nombre aléatoire — 178, 187
Nombres négatifs — 18
NOT — 71, 253
Notation algébrique directe — 19
Notation binaire — 251
Notation scientifique
 définition — 18
 touches — 19
 utilisation — 19
NULL — 84
NUMBER — 149
NUMERIC — 150
Numérique
 type-de-données — 65
 opérations — 65
 variable — 65
Numérotation de ligne — 62

O

OLD — 75, 151
ON BREAK — 152
ON ERROR — 154
ON GOSUB — 156
ON GOTO — 157
ON WARNING — 158
OPEN — 76, 160

ANNEXE J

INDEX

Opérateurs

 numériques — 65
 logiques — 70, 252
 de relation — 70
 chaîne — 70
Opérations en attente — 20
OR — 71, 251
Ordre des opérations — 20, 66
Ordre d'exécution — 20, 66

P

Parenthèses — 21
PAUSE — 163
Permutations — 37
Périphériques — 45, 213
Pi — 165
Pi, valeur — de 18
Point décimal — 18
Point d'exclamation — 63
Points d'arrêt — 94
Polaires/cartésiennes conversions — 35
POS — 166
Pourcentage — 31
Précision des calculs — 272
PRINT (avec affichage) — 167
PRINT (avec fichiers) — 171
Programme
 exécution de — 58
 lignes de — 62
 stockage de — 74, 191
 fin de — 110
PROTECTED — 191
Puissances et racines — 24, 65
PUT — 176

Q

Quadrant — 30

R

Racine carrée — 14, 67
RAD — 28, 177

ANNEXE J

INDEX

Radians — 29, 177
Radix-100 — 273
RAM — 48, 230
RAM système — 48
RANDOMIZE — 178
Rappel mémoire — 25
Rappel [PB] — 52
READ — 64, 179
REC — 133, 171
Récupération de programmes — 75, 151, 223
Régression — 42
Régression linéaire — 42
REM — 180
Remarques — 63, 180
Remplacement des piles — 3
RENUMBER — 181
RESTORE — 182
RETURN avec GOSUB — 184
RETURN avec ON ERROR — 185
RND — 187
RUN — 189, 225
RPT\$ — 188

S

SAVE — 74, 191, 222
SEG\$ — 69, 192
SGN — 193
SIN — 194
SINH — 195
Séparateurs — 167
SIZE — 82, 107
Signe — 18
Signe, Fonction — 193
Sinus — 29, 66, 194
Sommes statistiques — 43
Sortie domaine des valeurs — 241
Sous-chaîne — 192
Sous-programmes — 73, 200
Sous-routine — 72
Sous-routine de traitement des erreurs — 154
SQR — 196

ANNEXE J

INDEX

Statistiques

applications — 42
calculs — 41
définitions — 40
touches — 38
procédure — 39
STEP — 115
Stockage de programmes — 74, 191, 222
STOP — 197
STR\$ — 198
SUB — 199
SUBEND — 202
SUBEXIT — 203
Symbole de remarque finale (!) — 63
Symbole d'élévation à la puissance — 66
Symboles audit — 233
Symboles de rappel de fonction — 235

T

TAB — 204
Tableaux — 69
TAN — 206
TANH — 207
Tangente — 29, 206
Touche d'effacement de chiffre — 16
Touche [SHIFT] [PB] — 52
Touches de fonctions trigonométriques — 29
Touches de puissance et racine quelconques — 24
Touches d'introduction de données — 17
Touches d'opérations arithmétiques — 23
Touches d'opérations logarithmiques — 26
Touches préfixe — 13, 51
Traitement des erreurs — 73
Transfert du contrôle — 71

U

UNBREAK — 208
Unités externes — 212
USING — 209

ANNEXE J INDEX

V

VAL — 210
Valeur absolue — 81
Valeurs négatives 18
Valeurs positives 18
VALIDATE 85
VARIABLE 161
VERIFY 211
Vérification d'un magnétophone 216

X

XOR 71, 253

ANNEXE K GARANTIE

Garantie

Veillez vous adresser à votre revendeur Texas Instruments pour toute mise en œuvre de la garantie.

La garantie légale des vices cachés ou défauts des marchandises vendues, s'applique en tout état de cause aux termes des articles 1641 et suivants du Code Civil.

Cette calculatrice Texas Instruments est garantie pièces et main-d'œuvre au premier acheteur pour une durée de un an à partir de la date d'achat pour des conditions d'utilisation normales.

Si un jeu de piles a été placé dans votre calculatrice à titre de démonstration, Texas Instruments ne garantit pas la qualité, ni la durée de vie des piles ni les dommages susceptibles d'être causés à l'appareil par suite d'une fuite de piles en général.

La garantie est nulle si :

1. La calculatrice a été endommagée par accident ou utilisation abusive, par négligence, par réparation impropre, ou tout autre état de cause ne trouvant pas son origine dans les pièces détachées ou leur assemblage;
2. Le numéro de série a été modifié ou effacé.

TEXAS INSTRUMENTS NE SAURAIT ETRE TENUE POUR RESPONSABLE DES PERTES DE JOUISSANCE CONSECUTIVES A UNE PANNE DE LA CALCULATRICE ET/OU TOUT AUTRE DOMMAGE INDIRECT SUBI PAR L'ACHETEUR.

Pendant la période de garantie, la calculatrice ou ses pièces défectueuses seront gratuitement réparées, ajustées et/ou remplacées au choix du fabricant, lorsque la calculatrice aura été retournée au détaillant, accompagnée du justificatif d'achat.

TOUTE MACHINE RETOURNEE SANS JUSTIFICATION DE LA DATE D'ACHAT SERA REPAREE AU COÛT DE LA REPARATION EN VIGUEUR AU MOMENT DU RETOUR.

En cas de remplacement par une nouvelle calculatrice, cette dernière bénéficiera de la poursuite de la garantie contractuelle initialement accordée au modèle acheté. Cette garantie contractuelle ne sera en aucun cas inférieure à 90 jours.