

ORDINATEUR PERSONNEL

**CASIO PB-100**

---

**MODE D'EMPLOI**

---

Merci beaucoup pour avoir porté votre choix sur le Casio PB-100. Cet instrument est un ordinateur personnel "de poche" convenant parfaitement à ceux qui veulent s'initier à l'utilisation des ordinateurs.

En utilisant cet ordinateur personnel de poche, vous pouvez entrer dans le monde de l'informatique et exécuter facilement des programmes en langage BASIC.

## TABLE DES MATIERES

<b>Avant l'emploi</b> .....	<b>3</b>
Précautions d'emploi .....	3
Alimentation et changement des piles .....	4
<b>Chapitre 1 Nom et utilisation de chaque section</b> .....	<b>5</b>
1-1 Nom de chaque section .....	5
1-2 Comment lire l'affichage .....	10
<b>Chapitre 2 Avant les calculs</b> .....	<b>11</b>
2-1 Réglage du contraste .....	11
2-2 Module RAM pour extension (option) .....	11
2-3 Extension mémoire .....	12
2-4 Coupure automatique de l'alimentation .....	13
<b>Chapitre 3 Comment calculer</b> .....	<b>14</b>
3-1 Ordre de priorité des calculs .....	14
3-2 Nombre de position d'entrée/de sortie et nombre de positions d'opération .....	14
3-3 Comment exécuter les calculs élémentaires .....	15
3-4 Rappel du résultat du calcul précédent .....	16
3-5 Messages d'erreur .....	17
3-6 Utilisation des touches .....	17
<b>Chapitre 4 Calculs manuels</b> .....	<b>20</b>
4-1 Qu'est-ce qu'un calcul manuel? .....	20
4-2 Méthode d'utilisation pour les calculs manuels .....	20
4-3 Exemples de calculs manuels .....	21
4-3-1 Comment exécuter les calculs élémentaires .....	21
4-3-2 Comment exécuter les calculs de fonction .....	23
4-4 Rangs .....	26

Ce manuel explique l'utilisation élémentaire d'un ordinateur et vous donne les grandes lignes du langage BASIC.

Avant d'utiliser cet ordinateur, veuillez lire ce manuel en totalité afin de bien maîtriser chaque fonction. Afin d'obtenir la longévité maximale de l'appareil, bien faire attention aux précautions d'emploi.

<b>Chapitre 5</b>	<b>Calculs programmés</b>	<b>27</b>
<b>5-1</b>	<b>Description générale d'un programme</b>	<b>27</b>
<b>5-2</b>	<b>Principes de programmation</b>	<b>28</b>
5-2-1	Constantes et variables	28
5-2-2	Instructions de substitution	29
<b>5-3</b>	<b>Écriture et exécution de programme</b>	<b>30</b>
5-3-1	Écriture de programme	30
5-3-2	Exécution de programme	32
<b>5-4</b>	<b>Édition de programme</b>	<b>34</b>
<b>5-5</b>	<b>Mise au point de programme</b>	<b>40</b>
<b>5-6</b>	<b>Commandes de programme</b>	<b>44</b>
5-6-1	Commandes d'entrée (INPUT, KEY)	44
5-6-2	Commandes de sortie (PRINT, CSR)	45
5-6-3	Commande de saut (GOTO)	47
5-6-4	Commande de décision (IF ~ THEN)	48
5-6-5	Commande de boucle (FOR-NEXT)	49
5-6-6	Commande de sous-programme (GOSUB, RETURN)	51
5-6-7	Instruction multiple	54
5-6-8	Commande d'arrêt (STOP)	54
5-6-9	Commande de fin (END)	54
5-6-10	Commande d'exécution (RUN)	54
5-6-11	Commande de listage (LIST)	55
5-6-12	Désignation de mode (MODE)	55
5-6-13	Format de sortie (SET)	56
5-6-14	Fonctions de caractères (LEN, MID, VAL)	56
5-6-15	Effacement de mémoire (VAC)	57
5-6-16	Effacement de programme (CLEAR, CLEAR A)	57
5-6-17	Spécifications d'option	58
	Magnétophone (SAVE, LOAD, SAVE A, LOAD A, PUT, GET, VER)	58
	Imprimante	60
	<b>Liste de messages d'erreur</b>	<b>61</b>
	<b>Liste de commandes de programme</b>	<b>62</b>
	<b>Plage des fonctions</b>	<b>64</b>
	<b>Caractéristiques</b>	<b>65</b>

## Avant l'emploi

Avant son expédition de la fabrique, cet ordinateur, fruit d'une technologie électronique de pointe, a subi un strict processus d'essai et un sévère contrôle de qualité. Afin d'assurer sa longévité maximale, veuillez noter les précautions suivantes.

### ■ Précautions d'emploi

- Cet ordinateur étant construit avec des composants électroniques de précision, ne jamais essayer de le démonter. De plus, ne pas lui faire subir de choc et ne pas le soumettre à des températures extrêmes. Bien prendre garde à éviter les endroits très chauds, très humides ou très poussiéreux. Qui plus est, si la température ambiante est basse, le temps de réponse de l'affichage peut être long ou il peut ne pas y avoir d'affichage du tout. Toutefois, ces problèmes disparaissent dès que la température ambiante atteint un niveau normal.
- Ne jamais essayer de brancher sur le connecteur un appareil autre que ceux fournis en option par CASIO.
- Lorsque l'ordinateur est en marche, un "—" est affiché. Dans ce cas, les appuis sur les touches ne donnent rien, sauf pour une section. Par conséquent, ne jamais oublier d'appuyer sur les touches tout en contrôlant l'affichage.
- Ne pas oublier de changer les piles tous les deux ans, quel que soit leur niveau d'usure. Si des piles usées sont laissées dans leur compartiment, elles fuieront et entraîneront ainsi des pannes.
- Ne pas utiliser de diluant ou de benzine pour nettoyer cet appareil. L'essuyer avec un chiffon doux et sec ou imbibé de détergent neutre.
- En cas de panne, adressez vous à votre revendeur.
- Avant de demander un dépannage, veuillez relire ce manuel et contrôler si l'alimentation est correcte et s'il n'y a pas d'erreur de programme ou d'utilisation.

## ■ Alimentation et changement des piles

L'alimentation est faite par deux piles au lithium (CR2032)

Si le contraste est faible même quand la commande de contraste est réglée pour un contraste maximal (voir page 11), cela signifie que les piles sont usées. Dans ce cas, changer les piles le plus tôt possible en procédant comme suit.

Qui plus est, même si l'instrument fonctionne normalement, ne pas oublier de changer les piles tous les deux ans.

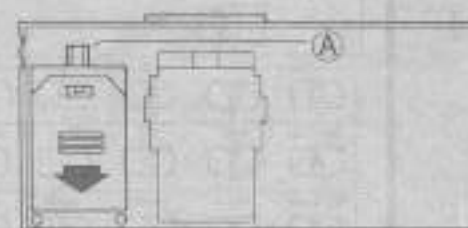
### ● Comment changer les piles

- (1) Après avoir coupé l'alimentation, enlever les deux vis du panneau arrière puis enlever ce panneau.

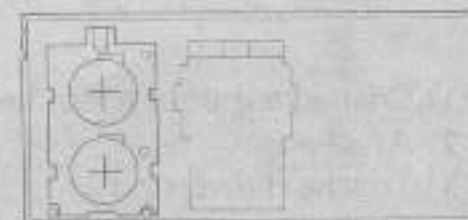


**Bouton de remise à zéro générale**  
(Après avoir changé les piles, appuyer sur ce bouton avec un objet pointu.)

- (2) Tout en appuyant sur **A**, faire coulisser le couvercle du compartiment à piles dans le sens de la flèche pour l'enlever.



- (3) Enlever les vieilles piles.  
(Cette opération est plus facile si vous tapez doucement sur l'instrument avec le compartiment à piles orienté vers le bas.)



- (4) Bien essuyer les piles neuves avec un chiffon sec puis les mettre en place avec le côté **+** en haut.

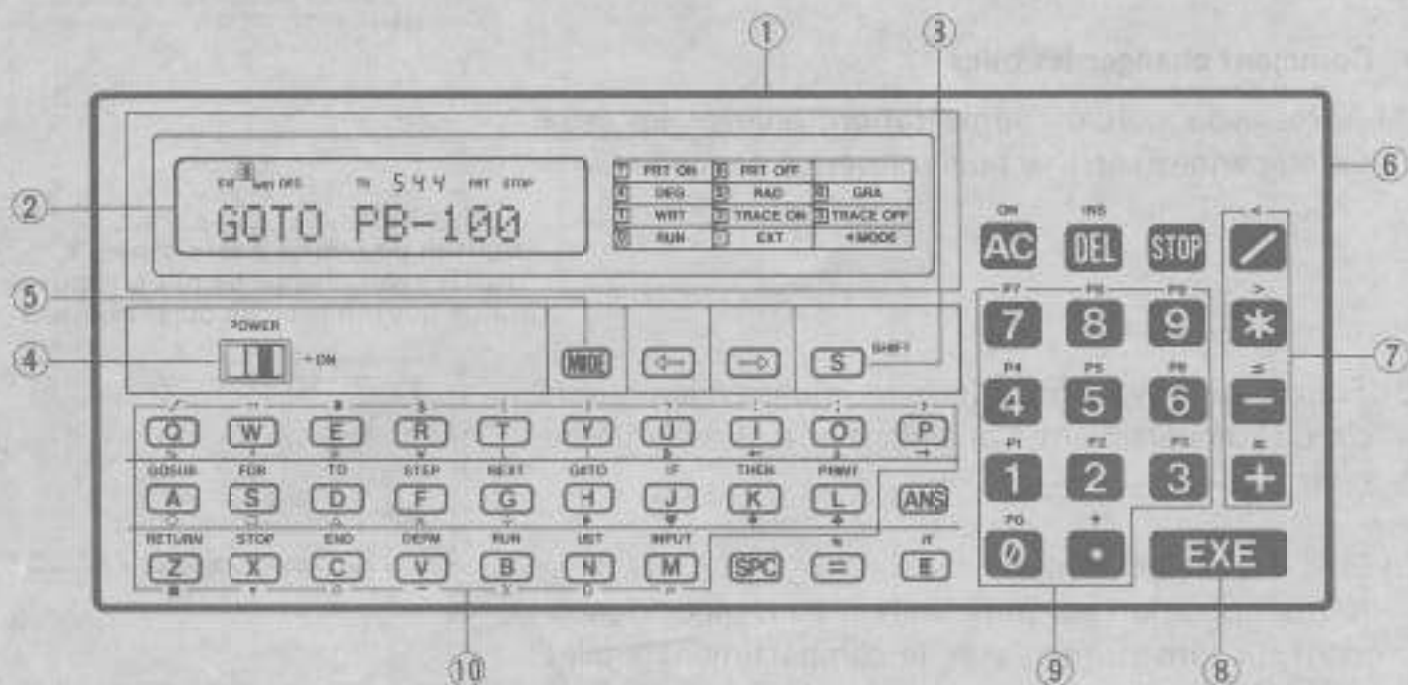
- (5) Remonter le couvercle du compartiment à piles.

- (6) Remonter le panneau arrière et serrer ses vis. Après avoir mis l'appareil sous-tension, appuyer sur le bouton de remise à zéro générale.

- \* Être sûr de changer *les deux* piles.
- \* Il est dangereux de jeter des vieilles piles au feu; elles pourraient exploser.
- \* Être sûr que les polarités des piles ne sont pas inversées.

# Chapitre 1

## Nom et utilisation de chaque section



- ① Connecteur pour adaptateur
- ② Affichage
- ③ Touche d'inversion
- ④ Commutateur d'alimentation
- ⑤ Touche de mode
- ⑥ Commande de contraste d'affichage
- ⑦ Touches de commande d'opérations
- ⑧ Touche d'exécution
- ⑨ Touches numériques et touche de virgule décimale
- ⑩ Touches alphabétiques

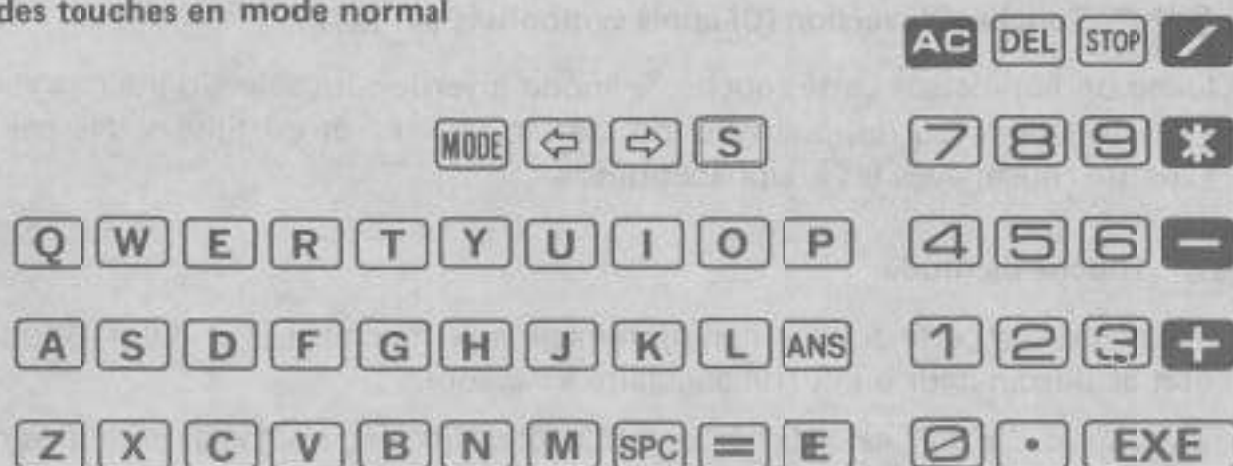
### 1-1 Nom de chaque section

Chaque touche a une ou deux fonctions. Ces fonctions peuvent être sélectionnées en mode "normal", dans lequel on appuie directement sur les touches, et en mode "inversion", dans lequel on appuie sur les touches après avoir appuyé sur la touche **I** (touche d'inversion).

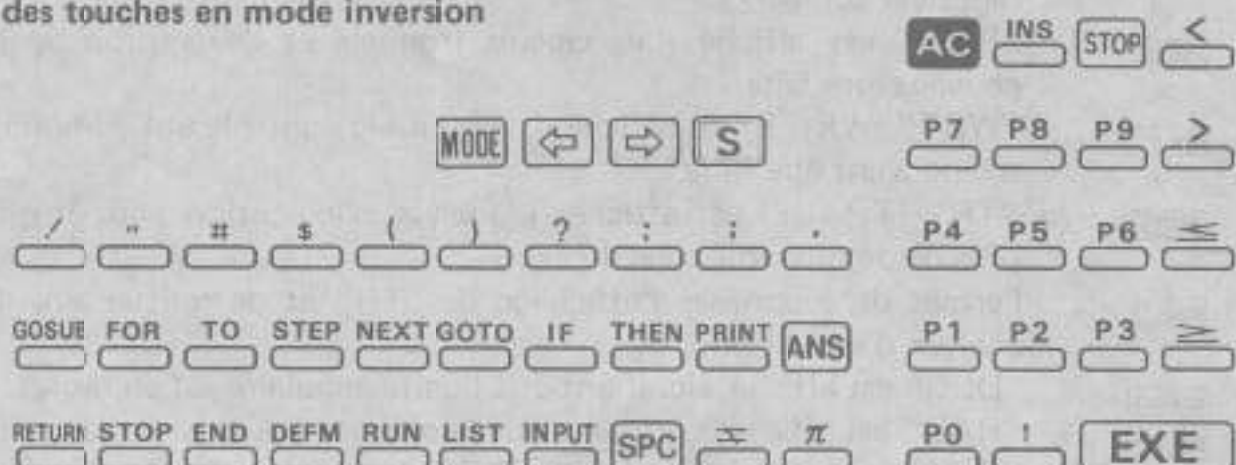
**Exemple:**

GOSUB — Mode inversion  
**A** — Mode normal

## Utilisation des touches en mode normal



## Utilisation des touches en mode inversion

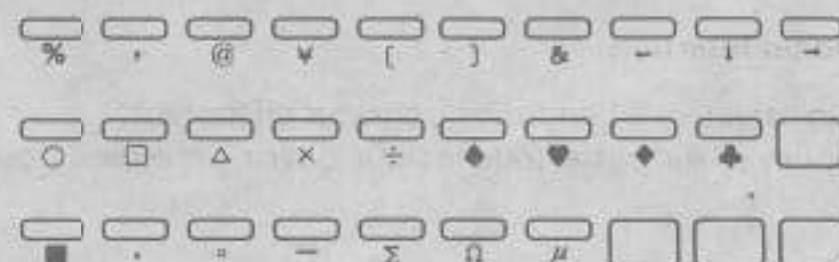


- En mode inversion, les touches alphabétiques deviennent des touches d'instructions et les touches numériques des touches de désignation de zone de programme.
- En plus de ceci, en mode extension (appuyer sur  $\text{MODE} \cdot$ , "EXT" sera affiché) l'appui direct sur les touches alphabétiques permet l'obtention de minuscules, et l'appui sur ces touches après la touche d'inversion permet l'affichage de symboles spéciaux.

## Mode normal en mode extension



## Mode inversion en mode extension



## **S** <sup>SHIFT</sup> Touche d'inversion (Ci-après symbolisée par **S**)

Quand on appuie sur cette touche, le mode inversion est sélectionné ("**S**" est affiché) et les fonctions correspondantes du clavier peuvent être utilisées. Ne pas confondre **S** (lettre rouge) avec le **S** alphabétique.

## **MODE** Touche de mode

On appuie sur cette touche conjointement aux touches **+** et **☉** à **☽** pour désigner l'état de l'ordinateur ou l'unité angulaire à l'avance.

- MODE** **+**..... "EXT" est affiché. Le mode extension est désigné et les lettres minuscules/symboles spéciaux peuvent être utilisés. Pour quitter le mode extension, appuyer sur **MODE** **+**.
- MODE** **☉**..... "RUN" est affiché. Les calculs manuels et l'exécution de programme peuvent être faits.
- MODE** **1**..... "WRT" (WRITE) est affiché. L'écriture, le contrôle et l'édition de programme peuvent être faits.
- MODE** **2**..... "TR" (TRACE) est affiché. L'analyse d'exécution peut être faite. (Pour plus de détails, voir page 43.)
- MODE** **3**..... Permet de supprimer l'affichage de "TR" et de quitter ainsi la fonction analyse d'exécution.
- MODE** **4**..... "DEG" est affiché, signifiant que l'unité angulaire est en degrés.
- MODE** **5**..... "RAD" est affiché, signifiant que l'unité angulaire est en radians.
- MODE** **6**..... "GRA" est affiché, signifiant que l'unité angulaire est en grades.
- MODE** **7**..... "PRT" (PRINT) est affiché; si une imprimante est branchée, l'impression est possible.
- MODE** **8**..... Permet de supprimer l'affichage de "PRT" et de quitter ainsi la fonction impression.

## **←** **→** Touches de commande du curseur

Appuyer sur ces touches pour déplacer le curseur vers la gauche ou la droite. Un appui entraîne un déplacement d'un caractère. Si la touche est maintenue enfoncée, le déplacement du curseur se fait en continu.

## **ON** **AC** Touche d'effacement général

- Appuyer sur cette touche pour effacer tout l'affichage.
- Si on appuie sur cette touche pendant l'exécution d'un programme, cette exécution est interrompue.
- Quand un message d'erreur est affiché, appuyer sur cette touche pour l'effacer.
- Quand l'alimentation a été coupée automatiquement (fonction automatique d'économie d'énergie, voir page 13), appuyer sur cette touche pour la rétablir.

## **INS** **DEL** Touche de suppression/d'insertion

- Permet de supprimer un caractère à l'endroit du curseur clignotant.
- En mode inversion, appuyer sur cette touche pour ouvrir un espace pour insertion de caractère.



### **STOP** Touche d'arrêt

Si on appuie sur cette touche pendant l'exécution d'un programme, "STOP" est affiché et l'exécution du programme est arrêtée à la fin de la ligne.

Pendant l'analyse d'exécution avec "STOP" affiché, l'appui sur cette touche entraîne l'affichage du numéro de zone de programme et du numéro de ligne.

### **EXE** Touche d'exécution

- Pour obtenir le résultat d'un calcul manuel, appuyer sur cette touche au lieu de "=".
- En mode "WRT", lorsque l'on entre un programme, appuyer sur cette touche pour entrer (mémoriser) chaque ligne dans l'ordinateur. Si on n'appuie pas sur cette touche, rien ne sera entré.
- En mode "RUN", appuyer sur cette touche pour entrer des données pendant l'exécution du programme, ou appuyer sur cette touche pour reprendre l'exécution du programme quand "STOP" est affiché.

### **ANS** Touche de réponse

Lors de calculs manuel, appuyer sur cette touche pour appeler le résultat (la réponse) du calcul précédent.

### $\pi$ **E** Touche d'exposant/Pi

Pour entrer une valeur exponentielle appuyer sur cette touche après avoir entré la mantisse.

**Exemple:**  $2.56 \times 10^{34} \rightarrow$

\* La partie exposant peut être au maximum de  $\pm 99$ . Si cette valeur est dépassée, une erreur se produit.

### $\equiv$ Touche égal/de comparaison



- Appuyer sur cette touche quand on utilise une instruction de substitution ou pour comparaison quand on utilise une instruction IF (signe égal).
- En mode inversion, appuyer sur cette touche pour comparaison quand on utilise une instruction IF.

P7	P8	P9
P4	P5	P6
P1	P2	P3
P0	.	

### Touches numériques/Touches de numéro de programme






- Appuyer sur ces touches quand on entre des valeurs numériques dans l'ordinateur. Appuyer sur à l'endroit de la virgule décimale.
- En mode inversion, à deviennent des touches de désignation de numéro de programme. Quand un programme a été entré, l'appui sur la touche de numéro de programme correspondante le fait démarrer.
- Appuyer sur la touche en mode inversion pour utiliser la fonction  $(x^y)$ .

## Touches de commande d'opération/de comparaison

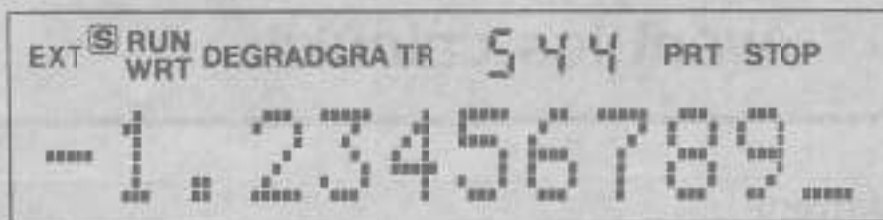
- Pour exécuter des additions, soustractions, multiplications et divisions, appuyer sur ces touches aux endroits respectifs.
  -  est utilisée pour les multiplications (correspond à 'x').
  -  est utilisée pour les divisions (correspond à ÷).
- En mode inversion, appuyer sur ces touches pour comparaison dans une instruction IF.

/	..	#	s	(	)	?	:	:	.
Q	W	E	R	T	Y	U	I	O	P
GOSUB	FOR	TO	STEP	NEXT	GOTO	IF	THEN	PRINT	
A	S	D	F	G	H	J	K	L	
RETURN	STOP	END	DEFM	RUN	LIST	INPUT			
Z	X	C	V	B	N	M	SPC		

## Touches alphabétiques/de commande à une touche/de caractère

- Quand on entre un programme ou une commande/commande de fonction, l'appui sur ces touches entraîne l'affichage des lettres correspondantes. Appuyer sur la touche  quand un espace est nécessaire.
- Touches  ~  : En mode inversion, l'appui sur ces touches entraîne l'affichage des caractères correspondants.
- Touches  ~  : En mode inversion, l'appui sur ces touches entraîne l'affichage des commandes à une touche correspondantes.

## 1-2 Comment lire l'affichage



L'affichage permet de visualiser les valeurs de calcul ou résultats. Chaque position d'affichage est constitué par une matrice de 5 points horizontaux sur 7 points verticaux. Un maximum de 12 positions sont disponibles pour l'affichage de nombres ou caractères. (Zéro est affiché sous la forme 0.) Si une formule ou instruction dépasse 12 positions, les nombres ou caractères se déplacent vers la gauche. Ainsi, un maximum de 62 caractères peuvent être entrés.

Le curseur clignotant est affiché jusqu'à ce que 55 caractères soient entrés. A partir du 56e caractère, un "■" clignotant est affiché à la place.

Un affichage numérique à 4 positions est disponible à la partie supérieure de l'affichage pour indiquer le nombre de pas restants. Qui plus est, pendant la marche, un "-" est affiché à l'extrême droite de cet affichage.

De plus, divers symboles tels que "DEG", "RAD" et "GRA" pour les unités angulaires, "S" (quand la touche  $\square$  est enfoncée), "RUN" (mode RUN), "WRT" (mode WRT), "TR" (mode TR), "PRT" (mode PRT) et "STOP" sont affichés pour indiquer l'état actuel.

### ● Exemple d'affichage alphabétique



### ● Exemple d'affichage de symboles

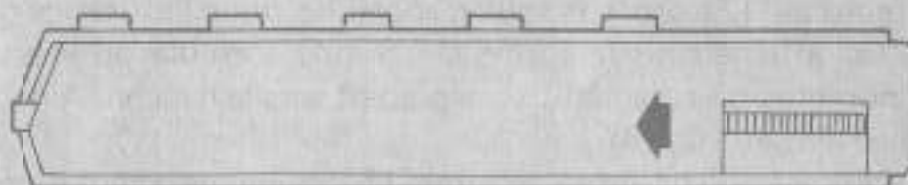


## Chapitre 2

### Avant les calculs

#### 2-1 Réglage du contraste

Le réglage du contraste de l'affichage peut être fait à l'aide de la commande située sur le côté droit de l'appareil.



Tourner dans le sens de la flèche pour augmenter le contraste. Tourner dans le sens opposé pour diminuer le contraste. Cette possibilité de réglage est utilisée pour compenser le contraste de l'affichage en fonction de l'état des piles, de l'éclairage ambiant et de l'angle de vision.

#### 2-2 Module RAM pour extension (option)

La zone (mémoire) RAM standard de cet appareil est de 544 pas/26 mémoires. Toutefois, cette capacité peut être portée à un maximum de 1568 pas/222 mémoires par l'ajout du bloc RAM OR-1 optionnel. Cette zone RAM étendue peut être utilisée de la même manière que la zone standard et permet une augmentation du nombre de pas et de la capacité mémoire (voir page 12).

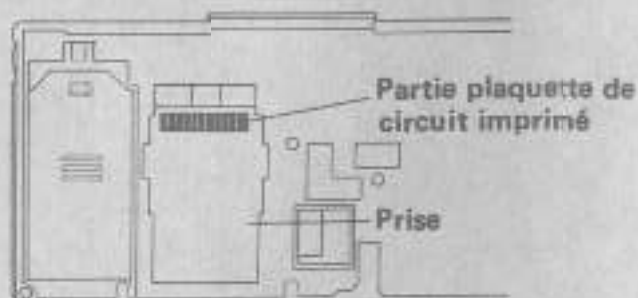
##### • Comment monter le module RAM

###### (Préparation)

Etant donné que les circuits internes peuvent être endommagés par l'électricité statique, avant de manipuler le module, ne pas oublier de toucher un objet métallique tel que poignée de porte afin de décharger toute électricité statique.

###### (Procédure)

- (1) Couper l'alimentation.
- (2) Desserrer les deux vis du panneau arrière puis enlever ce panneau.



(3) Insérer le module dans la prise du corps de l'appareil puis mettre le fermoir en position de blocage.

\* Ne jamais toucher la partie connecteur du module RAM ni la partie plaquette de circuit imprimé du corps de l'appareil.



(4) Remonter le panneau arrière et serrer ses vis.

- Après avoir monté ou enlevé le module RAM, ne pas oublier d'appuyer sur le bouton de remise à zéro générale avec un objet pointu après avoir mis l'appareil sous tension. Si on n'appuie pas sur le bouton de remise à zéro générale, le contenu de la mémoire peut être changé ou un affichage qui n'a pas de sens peut apparaître.
- Prendre garde à ne pas salir ou toucher la partie connecteur du module ou la partie plaquette de circuit imprimé de l'appareil, cela entraînerait un mauvais contact.
- Quand le module est enlevé, ne pas oublier de le remettre dans son boîtier, et le ranger à l'abri de la poussière.

## 2-3 Extension mémoire

Normalement, il y a 26 mémoires (variables). Dans ce cas, le nombre de pas est de 544. Le nombre maximal de mémoires standard est de 94. Le montage d'un module RAM permet de porter ce maximum à 222. Pour l'extension mémoire, les pas de programme sont convertis en mémoires à raison de 8 pas par mémoire.

Nombre de mémoires	Nombre de pas de programme	
	Standard	Avec extension
26	544	1568
27	536	1560
28	528	1552
⋮	⋮	⋮
46	384	1408
⋮	⋮	⋮
94	0	1024
⋮	⋮	⋮
200	—	176
⋮	⋮	⋮
222	—	0

L'extension mémoire est faite par paliers d'une unité à l'aide d'une commande DEFM.

**Exemple:**

Etendre de 30 pour porter à 56.

**Opération:**

Sélectionner le mode RUN (appuyer sur **MODE** (2)) ou le mode WRT (appuyer sur **MODE** (1)).

DEFM 30 **EXE**

\*\*\*VAR: 56

- \* DEFM peut être entré en appuyant sur **D E F M** ou en appuyant sur **DEFM**.

Une commande DEFM est aussi utilisée pour contrôler le nombre de mémoires désignées.

**Exemple:**

56 mémoires sont désignées.

DEFM **EXE**

\*\*\*VAR: 56

- Quand un grand nombre de pas de programme est déjà utilisé, afin de protéger le programme existant, si une désignation pouvant entraîner un insuffisant nombre de pas est tentée, une erreur se produit (ERR 1 ..... nombre de pas insuffisant)
- La variable à caractères exclusive (\$) étant une mémoire spéciale, elle n'est pas comptée lors de la désignation.

## 2-4 Coupure automatique de l'alimentation

Cette fonction empêche tout gaspillage d'énergie quand on oublie de couper l'alimentation de l'appareil. L'alimentation est coupée automatiquement environ 7 minutes après le dernier appui sur une touche (sauf pendant l'exécution de programme). Dans ce cas, l'alimentation peut être rétablie en appuyant sur la touche **AC** ou en ouvrant puis refermant le commutateur d'alimentation.

- \* Même si l'alimentation est coupée, les contenus des mémoires et les programmes ne sont pas effacés. Toutefois, la désignation d'unité angulaire et la désignation de mode ("WRT", "TF", "PRT", etc.) sont effacées.

# Chapitre 3

## Comment calculer

Les calculs manuels et les calculs programmés sont exécutés en mode "RUN". (Appuyer sur  $\square$   $\square$  et RUN sera affiché.)

Qui plus est, en ce qui concerne "DEG", "RAD" et "GRA", étant donné que ces signes ne s'appliquent qu'aux unités angulaires, leur affichage n'a pas d'effet sur un calcul qui n'a rien à voir avec des unités angulaires.

### 3-1 Ordre de priorité des calculs

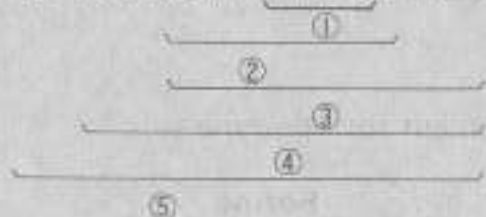
Cet appareil détermine l'ordre de priorité des calculs intérieurement et exécute les calculs suivant cet ordre. L'ordre de priorité des calculs est déterminé comme suit.

- ① Fonctions (SIN, COS, TAN, etc.)
- ② Puissance
- ③ Multiplication et division (\* et /)
- ④ Addition et soustraction (+ et -)

Quand la priorité est identique, le calcul commence par la gauche.

Qui plus est, quand des parenthèses sont utilisées, elles ont la plus haute priorité.

**Exemple:**

$$2 + 3 * \text{SIN} (17 + 13) \uparrow 2 = 2.75$$


### 3-2 Nombre de position d'entrée/de sortie et nombre de positions d'opération

Sur cet appareil, les nombres de positions sont de 12 positions pour la mantisse et 2 positions pour l'exposant. Les opérations internes sont aussi exécutées en utilisant 12 positions pour la mantisse et 2 positions pour l'exposant.

La plage est  $1 \times 10^{-99} \sim \pm 9.99999999999 \times 10^{99}$  et 0.

Les nombres de positions de sortie sont de 10 positions pour la mantisse et 2 positions pour l'exposant. Toutefois, si un exposant est rattaché, la mantisse sera de 8 positions.

\* Pour les résultats de fonction, etc., quand le nombre de positions d'affichage (12 positions) est dépassé, 12 positions sont affichées, 0 et virgule décimale compris.

**Exemple:**

$$(1 \times 10^5) \div 7 = 14285.71429$$

1 [E] 5 [÷] 7 [EX]

14285.71429

$$(1 \times 10^5) \div 7 - 14285 = 0.7142857$$

1 [E] 5 [÷] 7 [=] 14285 [EX]

0.7142857

Quand le résultat d'un calcul est supérieur à  $10^{10}$  (10.000.000.000) ou inférieur à  $10^{-3}$  (0,001), il apparaît automatiquement sous forme d'affichage exponentiel.

**Exemple:**

$$1234567890 \times 10 = 12345678900$$

1234567890 [×] 10 [EX]

1.2345678E10

$$(= 1.23456789 \times 10^{10})$$

Signe d'exposant

\* L'exposant est affiché avec un signe d'exposant à la suite de la mantisse.

$$1.234 \div 10000 = 0.0001234$$

1.234 [÷] 10000 [EX]

1.234E-04

$$(= 1.234 \times 10^{-4})$$

**3-3 Comment exécuter les calculs élémentaires****(1) Symboles de calcul et commandes de fonction**

En ce qui concerne les symboles de calcul utilisés en BASIC, le "+" et le "-" sont utilisés pour les additions et soustractions. Toutefois, pour les multiplications et divisions, "\*" et "/" sont utilisés à la place de "x" et "÷".

**Exemple:**

$$2 + 3 - 4 \times 5 \div 6 \text{ devient } 2 + 3 - 4 * 5 / 6$$

De plus, les fonctions utilisables avec cet instrument sont comme suit.

Fonction		Forme
Fonctions trigonométriques	$\sin x$	SIN $x$
	$\cos x$	COS $x$
	$\tan x$	TAN $x$
Fonctions trigonométriques inverses	$\sin^{-1} x$	ASN $x$
	$\cos^{-1} x$	ACS $x$
	$\tan^{-1} x$	ATN $x$
Racine carrée	$\sqrt{x}$	SQR $x$
Exponentielle	$e$	EXP $1^*$
Logarithme naturel	$\ln x$	LN $x$
Logarithme décimal	$\log x$	LOG $x$
Changement en un entier	INT $x$	INT $x$



Suppression de la partie décimale de la partie entière	FRAC $x$	FRAC $x$
Obtention de la valeur absolue	$ x $	ABS $x$
Symboliser	nombre positif $\rightarrow 1$ $0 \rightarrow 0$ nombre négatif $\rightarrow -1$	SGN $x$
Arrondis	(arrondir $x$ à $10^y$ )	RND ( $x, y$ ) <sup>*</sup>
Nombre aléatoire		RAN #

\* Dans le cas de la fonction RND, l'argument doit être entre parenthèses.

★ EXP est une commande pour rappeler une valeur numérique de la table exponentielle.

### 3-4 Rappel du résultat du calcul précédent

Le résultat obtenu en exécutant un calcul manuel ou un calcul programmé est mémorisé jusqu'à ce que le calcul suivant soit exécuté. Ce résultat peut être affiché en appuyant sur la touche **ANS**.

**Exemple:**

$$741 + 852 = 1593$$

$$2431 - 1593 = 838$$

**Opération:**

7 4 1 + 8 5 2

**EXE**

2 4 3 1 - **ANS**

**EXE**

741+852

1593

2431-1593

838

De plus, la valeur numérique qui est affichée à la suite d'un calcul peut être utilisée telle quelle dans le calcul suivant.

**Exemple:**

$$25.3 + 13.9 = 39.2$$

$$39.2 \times 7.6 = 297.92$$

**Opération:**

2 5 . 3 + 1 3 . 9 **EXE**

\* 7 . 6

**EXE**

39.2

39.2\*7.6

297.92

### 3-5 Messages d'erreur

Si une formule ou instruction de substitution ne se conforme pas à la syntaxe du BASIC ou si la plage de calcul de l'appareil est dépassée, une erreur se produit pendant l'exécution et un message d'erreur est affiché. En ce qui concerne les puissances ( $x^y$ ), toutefois, quand  $y$  est un nombre naturel, aucun message d'erreur ne sera affiché même si  $x$  est inférieur à 0 (zéro). Les messages d'erreur suivants peuvent apparaître lors de calculs manuels

ERR2	(Erreur de syntaxe)
------	---------------------

ERR3	(Erreur mathématique)
------	-----------------------

Les messages d'erreur suivants peuvent apparaître lors de calculs programmés.

ERR2 P0-10	(Erreur de syntaxe à la ligne 10 du programme P0)
------------	---

ERR3 P2-30	(Erreur mathématique à la ligne 30 du programme P2)
------------	---

(Pour une explication des messages d'erreur, voir page 61).

- \* Si le résultat d'un calcul dépasse  $\pm 9,99999999999 \times 10^{99}$ , il se produit un débordement et un message d'erreur ERR3 est affiché. De plus, si le résultat d'un calcul est inférieur à  $1,0 \times 10^{-99}$ , un dépassement inférieur de capacité se produit et le résultat du calcul devient 0.

### 3-6 Utilisation des touches

Pour les calculs manuels aussi bien que pour les calculs programmés et l'entrée de programme, l'utilisation des touches se fait comme suit.

#### (1) Entrée

##### • Entrée de lettres

Exemple: Entrer ABC

Opération: 

A	B	C
---	---	---

ABC
-----

Exemple: Entrer SIN

Opération: 

S	I	N
---	---	---

SIN
-----

##### • Entrée numérique

Exemple: Entrer 123

Opération: 

1	2	3
---	---	---

123
-----

Exemple: Entrer 96,3

Opération: 

9	6	.	3
---	---	---	---

96.3
------

● **Entrée de symboles**

**Exemple:** Entrer \$#?

**Opération:**

\$#?

**Exemple:** Entrer @¥Ω

**Opération:** ..... (Désignation du mode extensior)

Mode extension

EXT

EXT  
@¥Ω

..... (Pour quitter le mode extensior)

@¥Ω

● **Entrée de valeur numérique avec exposant**

**Exemple:** Entrer 7,896 x 10<sup>15</sup>

**Opération:**

7.896E15

**Exemple:** Entrer -2,369 x 10<sup>-45</sup>

**Opération:**

-2.369E-45

**(2) Changement d'entrée (correction, suppression, insertion)**

● **Correction**

Pour effectuer une correction, amener le curseur à l'endroit à corriger (à l'aide des touches et ) puis appuyer sur la touche de caractère, numéro ou symbole correcte.

**Exemple:** Remplacer "A\$" to "B\$".

A\$ \_

**Opération:** Déplacer le curseur de 2 caractères à gauche.

A\$

Appuyer sur la touche .

B\$

**Exemple:** Remplacer "LIST" par "RUN"

LIST \_

**Opération:** Déplacer le curseur de 4 caractères à gauche.

LIST

Appuyer sur ou .

RUN \_

## ● Suppression

Pour effectuer une suppression, amener le curseur à l'endroit à supprimer puis appuyer sur la touche **DEL**. A chaque appui sur cette touche, un caractère est supprimé et les caractères situés à la droite du caractère supprimé sont décalés d'une position à gauche.

**Exemple:** Supprimer un des "I" de "SIIN".

SIIN\_

**Opération:** Déplacer le curseur de 2 caractères à gauche.

←←

Appuyer sur **DEL**.

SIIN

SIN

**Exemple:** Supprimer "X," de "INPUT X, Y".

INPUT X,Y\_

**Opération:** Déplacer le curseur de 3 positions à gauche.

←←←

Appuyer sur **DEL DEL**.

INPUT X,Y

INPUT Y

## ● Insertion

Pour effectuer une insertion, amener le curseur à droite du caractère après lequel on veut faire l'insertion. Appuyer sur **SHIFT INS**; un espace d'un caractère est ouvert. Ensuite, appuyer sur la touche du caractère, numéro ou symbole désiré.

**Exemple:** Remplacer "T=A\$" par "T\$=A\$".

T=A\$\_

**Opération:** Déplacer le curseur de caractères à gauche.

←←←

Appuyer sur **SHIFT INS** pour ouvrir l'espace d'un caractère.

Appuyer sur **SHIFT =**.

T=A\$

T\_ =A\$

T\$=A\$

**Exemple:** Remplacer "PRINT X" par "PRINT SIN X".

PRINT X\_

**Opération:** Déplacer le curseur d'une position à gauche.

←

Appuyer sur **SHIFT INS SHIFT INS SHIFT INS**.

Appuyer sur **S I N**.

PRINT X

PRINT \_ X

PRINT SINX

# Chapitre 4

## Calculs manuels

### 4-1 Qu'est-ce qu'un calcul manuel?

Dans ce cas, les calculs ne sont pas faits automatiquement en mémorisant des formules de calcul comme un programme.

Les calculs sont effectués manuellement en remplaçant le côté gauche de la formule numérique par le côté droit ou en rappelant le contenu de la variable.

### 4-2 Méthode d'utilisation pour les calculs manuels

- Additions, soustractions, multiplications et divisions sont exécutées par une vraie logique algébrique. Les touches  $+$ ,  $-$ ,  $\times$  ( $\times$ ),  $\div$  ( $\div$ ) et  $\text{EXE}$  ( $=$ ) sont utilisées. La touche  $\text{EXE}$  est utilisée pour obtenir le résultat d'un calcul.

**Exemple:**  $12+36-9\times 5\div 4=36.75$

**Opération:**

1 2 + 3 6 - 9 \* 5 / 4

$\text{EXE}$

12+36-9\*5/4

36.75

- Un calcul de fonction est exécuté de la même façon qu'une formule normale, comprenant addition, soustraction, multiplication et division, et la donnée est écrite après la commande de fonction.

**Exemple:**  $\log 1.23=0.0899051114$

**Opération:**

LOG 1.23

$\text{EXE}$

LOG 1.23

0.0899051114

- \* Dans ce manuel, les cadres autour des lettres et chiffres sont omis.

**Exemple:**  $\text{SIN}(15+8)$

- Pour les calculs avec mémoire, etc., quand on mémorise des valeurs numériques ou des résultats de calcul ou quand on totalise, des variables sont employées. Des lettres (A à Z) ou une combinaison de lettres et chiffres (dans le cas de variables de rang) sont utilisées pour les variables; et ces variables fonctionnent comme des mémoires. Une formule de substitution est utilisée pour entrer une valeur numérique ou le résultat d'un calcul dans une variable.

**Exemple:** Entrer 1234 dans la variable A.

**Opération:** A  $\equiv$  1 2 3 4

EXE

A = 1 2 3 4 \_

\_

**Exemple:** Ajouter le résultat de  $23 \times 56$  dans la variable K.

**Opération:** K  $\equiv$  K  $\div$  23  $\times$  56

EXE

K = K + 23 \* 56 \_

\_

Cette opération est exécutée manuellement comme une instruction de substitutions dans un programme.

- \* Avant d'appuyer sur la touche **EXE**, toute correction peut être faite en amenant le curseur à l'endroit à corriger puis en appuyant sur la touche correcte. (Voir page 18.)
- \* Pour effacer tout l'affichage, appuyer sur **AC**.

## 4-3 Exemples de calculs manuel

### 4-3-1 Comment exécuter les calculs élémentaires

#### • Additions, soustractions, multiplications et divisions

**Exemple:**  $23 + 4.5 - 53 = -25.5$

**Opération:** 23  $\div$  4.5  $\ominus$  53 **EXE**

-25.5

**Exemple:**  $56 \times (-12) \div (-2.5) = 268.8$

**Opération:** 56  $\times$  **SHIFT**  $\div$  12 **SHIFT**  $\div$  2.5 **SHIFT**  $\div$  **EXE**

268.8

**Exemple:**  $12369 \times 7532 \times 74103 = 6.9036806 \times 10^{12}$  (=6903680600000)

**Opération:** 12369  $\times$  7532  $\times$  74103 **EXE**

6.9036806 E 12

**Exemple:**  $1.23 \div 90 \div 45.6 = 2.9970760 \times 10^{-4}$  (=0.00029970760)

**Opération:** 1.23  $\div$  90  $\div$  45.6 **EXE**

2.9970760 E - 04

- \* Quand le résultat est supérieur à  $10^{10}$  (10.000.000.000) ou inférieur à  $10^{-3}$  (0,001), il apparaît sous forme d'affichage exponentiel.

**Exemple:**  $7 \times 3 + 4 \times 5 = 76$

**Opération:** 7  $\times$  3  $\div$  4  $\times$  5 **EXE**

76

**Exemple:**  $12 + (2.4 \times 10^5) \div 42.6 - 78 \times 36.9 = 2767.602817$

**Opération:** 12  $\div$  2.4 **E** 5  $\div$  42.6  $\ominus$  78  $\times$  36.9 **EXE**

2767.602817

• Calculs avec mémoire

Exemple:  $12 \times 45 = 540$

$12 \times 31 = 372$

$75 \div 12 = 6.25$

Opération:  $A \equiv 12$  **EXE**

$A \times 45$  **EXE**

$A \times 31$  **EXE**

$75 \div A$  **EXE**

—
540
372
6.25

Exemple:  $23 + 9 = 32$

$53 - 6 = 47$

$45 \times 2 = 90$

$99 \div 3 = 33$

Total 22

Opération:  $M \equiv 23 + 9$  **EXE**

$M \equiv M + 53 - 6$  **EXE**

$M \equiv M - 45 \times 2$  **EXE**

$M \equiv M + 99 \div 3$  **EXE**

$M$  **EXE**

22
----

\* Pour cette méthode de calcul, étant donné que les résultats des calculs respectifs ne sont pas connus, quand on veut les voir, procéder comme suit.

$23 + 9$  **EXE**

32
----

$M \equiv$  **ANS** **EXE**

$53 - 6$  **EXE**

47
----

$M \equiv M +$  **ANS** **EXE**

$45 \times 2$  **EXE**

90
----

$M \equiv M -$  **ANS** **EXE**

$99 \div 3$  **EXE**

33
----

$M \equiv M +$  **ANS** **EXE**

$M$  **EXE**

22
----

### 4-3-2 Comment exécuter les calculs de fonction

#### • Fonctions trigonométriques (sin, cos, tan) et fonctions trigonométriques inverses ( $\sin^{-1}$ , $\cos^{-1}$ , $\tan^{-1}$ )

Lorsque l'on utilise les fonctions trigonométriques/trigonométriques inverses, ne pas oublier de désigner l'unité angulaire.

Exemple:  $\sin 12.3456^\circ = 0.2138079201$

Opération:  $\text{MODE} \rightarrow \text{°DEG}^\circ$

SIN 12.3456  $\text{EXE}$

0.2138079201

Exemple:  $2 \cdot \sin 45^\circ \times \cos 65.1^\circ = 0.5954345575$

Opération: 2  $\text{SIN}$  45  $\text{COS}$  65.1  $\text{EXE}$

0.5954345575

Exemple:  $\sin^{-1} 0.5 = 30^\circ$

Opération: ASN 0.5  $\text{EXE}$

30

Exemple:  $\cos\left(\frac{\pi}{3}\text{rad}\right) = 0.5$

Opération:  $\text{MODE} \rightarrow \text{°RAD}^\circ$

COS  $\text{SHIFT}$   $\text{SHIFT}$   $\pi$  3  $\text{SHIFT}$   $\text{EXE}$

0.5

Exemple:  $\cos^{-1} \frac{\sqrt{2}}{2} = 0.7853981634\text{rad}$

Opération: ACS  $\text{SHIFT}$   $\text{SHIFT}$  SQR 2 2  $\text{SHIFT}$   $\text{EXE}$

0.7853981634

Exemple:  $\tan(-35\text{gra}) = -0.612800788$

Opération:  $\text{MODE} \rightarrow \text{°GRA}^\circ$

TAN 35  $\text{EXE}$

-0.612800788

#### • Fonctions logarithmiques (log, ln) et fonctions exponentielles ( $e$ , $x^y$ )

EXP ( $e$ ) ne peut pas être utilisée dans un calcul continu. De plus, elle ne peut pas être utilisée dans une instruction multiple.

Exemple:  $\log 1.23 (= \log_{10} 1.23) = 0.0899051114$

Opération: LOG 1.23  $\text{EXE}$

0.0899051114

Exemple:  $\ln 90 (= \log_e 90) = 4.49980967$

Opération: LN 90  $\text{EXE}$

4.49980967



**Exemple:**  $e = 2.718281828$

(Cette commande permet rappeler une valeur numérique de la table exponentielle)

**Opération:** EXP 1 **EXE**

2.718281828

**Exemple:**  $10^{1.23} = 16.98243652$

(Obtenir le cologarithme du logarithme décimal 1,23)

**Opération:** 10 **SFT** **1/x** 1.23 **EXE**

16.98243652

**Exemple:**  $5.6^{2.3} = 52.58143837$

**Opération:** 5.6 **SFT** **1/x** 2.3 **EXE**

52.58143837

**Exemple:**  $123^{\frac{1}{2}} (= \sqrt{123}) = 1.988647795$

**Opération:** 123 **SFT** **1/x** **SFT** **1/x** 1 **7** **SFT** **1/x**

1.988647795

**Exemple:**  $\log \sin 40^\circ + \log \cos 35^\circ = -0.278567983$

Le cologarithme est 0,5265407845 (calcul logarithmique de  $\sin 40^\circ \times \cos 35^\circ$ )

**Opération:** **MODE** **4** **→** "DEG"

LOG SIN 40 **+** LOG COS 35 **EXE**

-0.278567983

10 **SFT** **1/x** **AVG** **EXE**

0.5265407845

\* La plage d'entrée pour les puissances ( $x \uparrow y$ ) est  $x > 0$ .

• Autres fonctions ( $\sqrt{\quad}$ , SGN, RAN#, RND, ABS, INT, FRAC)

**Exemple:**  $\sqrt{2} + \sqrt{5} = 3.65028154$

**Opération:** SQR 2 **+** SQR 5 **EXE**

3.65028154

**Exemple:** Donne "1" à un nombre positif, "-1" à un nombre négatif et "0" à zéro.

**Opération:** SGN 6 **EXE**

1

SGN 0 **EXE**

0

SGN **-** 2 **EXE**

-1

**Exemple:** Génération de nombre aléatoire (pseudo nombre aléatoire de  $0 < \text{RAN\#} < 1$ )

**Opération:** RAN **SFT** **1/x** **EXE**

0.790373907

**Exemple:** Le résultat de  $12,3 \times 4,56$  est arrondi à  $10^{-2}$ .

$12.3 \times 4.56 = 56.088$

**Opération:** RND **SFT** **1/x** 12.3 **x** 4.56 **SFT** **1/x** **-** 2 **SFT** **1/x** **EXE**

Pour RND ( $x, y$ ),  $y$  est tel que  $|y| < 100$ .

56.1

**Exemple:**  $|-78.9 \div 5.6| = 14.08928571$

**Opération:** ABS  $\square$   $\square$  78.9  $\square$  5.6  $\square$   $\square$   $\square$

14.08928571

**Exemple:** La partie entière de  $\frac{7800}{96}$  est 81.

**Opération:** INT  $\square$  7800  $\square$  96  $\square$   $\square$

81

\* Cette fonction obtient l'entier maximal qui ne dépasse pas la valeur numérique d'origine.

**Exemple:** La partie décimale de  $\frac{7800}{96}$  est 0,25.

**Opération:** FRAC  $\square$  7800  $\square$  96  $\square$   $\square$

0.25

### • Désignation des nombres de chiffres pour les parties entière et décimale

La désignation des nombres de chiffres pour les parties entière et décimale se fait à l'aide d'une commande "SET".

Désignation du nombre de chiffres pour la partie entière ..... SET E  $n$  ( $n = 0$  à 9)

Désignation du nombre de chiffres pour la partie décimale ..... SET F  $n$

Annulation de désignation ..... SET N

\* Quand la désignation de nombre de chiffres pour la partie entière est "SET E 0", le nombre de chiffres est 8.

\* Le dernier chiffre est arrondi.

Qui plus est, les valeurs numériques d'origine restent dans l'ordinateur et dans la mémoire.

**Exemple:**  $100 \div 6 = 16.66666666 \dots$

**Opération:** SET E 4  $\square$  (Désigne 4 chiffres pour la partie entière.)

100  $\square$  6  $\square$

1.667E01

**Exemple:**  $123 \div 7 = 17.57142857 \dots$

**Opération:** SET F 2  $\square$  (Désigne 2 chiffres pour la partie décimale.)

123  $\square$  7  $\square$

17.57

**Exemple:**  $1 \div 3 = 0.333333333 \dots$

**Opération:** SET N  $\square$  (Annule la désignation)

1  $\square$  3  $\square$

0.333333333

## 4-4 Rangs

En ce qui concerne les rangs, des rangs unidimensionnels sont utilisés avec des lettres rattachées comme A (i), B (j), etc. Etant donné que ces rangs sont utilisés et avec les 26 mémoires normales et avec les mémoires étendues, bien faire attention à la disposition de rang suivante.

$$A=A(0)$$

$$B=A(1)=B(0)$$

$$C=A(2)=B(1)=C(0)$$

$$D=A(3)=B(2)=C(1)=D(0)$$

$$E=A(4)=B(3)=C(2)=D(1)=E(0)$$

$$Y=A(24)=B(23)=C(22)\dots\dots\dots=Y(0)$$

$$Z=A(25)=B(24)=\dots\dots\dots=Y(1)=Z(0)$$

$$A(26)=B(25)=\dots\dots\dots=Y(2)=Z(1)$$

$$A(27)=B(26)=\dots\dots\dots=Y(3)=Z(2)$$

Mémoires étendues

$$A(93)=B(92)=\dots\dots\dots=Y(69)=Z(68)$$

Quand des rangs sont utilisés de cette façon, étant donné que la même mémoire peut être utilisée selon l'argument de rang, éviter d'utiliser la même mémoire dans le même programme.

### Exemple:

Peuvent être utilisées en même temps ..... A, B, C, F (0), F (9)

Ne peuvent pas être utilisées en même temps ..... F, G, A (5), A (6)

Effectuer l'extension mémoire correctement en fonction de la taille du rang.

# Chapitre 5

## Calculs programmés

### 5-1 Description générale d'un programme

Le calcul programmé est une méthode pour :

- ① Programmer le contenu des calculs à exécuter.
- ② Stocker le programme dans l'ordinateur.
- ③ A l'aide de ce programme, obtenir les résultats automatiquement après simple entrée des données.

Examinons le concept et la procédure de programmation nécessaires pour traiter un problème donné en utilisant l'ordinateur.

#### ● Programmes et programmation

Quand des utilisateurs d'ordinateur traitent un problème avec un ordinateur, ils composent des instructions qui sont écrites dans un langage que l'ordinateur peut comprendre. Ces instructions forment un programme, et la composition de ces instructions est appelée programmation.

#### ● Qu'est-ce qu'un programme?

Pour créer un programme, il y a diverses règles grammaticales, dont le détail sera expliqué plus loin. Premièrement, voyons un exemple de programme simple et élémentaire.

	Commande	Opérande	
10	INPUT	A,B	Instruction d'entrée
20	C=A+B		Instruction d'opération
30	PRINT	C	Instruction de sortie

Le programme ci-dessus représente un programme élémentaire et se compose d'une instruction d'entrée, d'une instruction d'opération, d'une instruction de sortie et de numéros de ligne. Une instruction d'entrée est utilisée pour entrer les données. Une instruction d'opération est utilisée pour traiter ces données. Une instruction de sortie est utilisée pour sortir le résultat de l'exécution. Les numéros de ligne sont utilisés au début de chaque ligne.

Sur une ligne, à la suite de son numéro, on trouve un mot constitué de lettres, qui a une signification particulière, et qui dit à l'ordinateur ce qu'il doit faire ensuite. Ce mot est appelé une "commande". Cette commande est suivie d'une chaîne de caractères qui contient les informations nécessaires à son exécution. Ceci est appelé "opérande".

● **Comment compter le nombre de pas**

Une commande ou commande de fonction utilisée dans un programme utilise 1 pas.  
Un numéro de ligne (valeur numérique de 1 à 9999) utilise 2 pas.

**Exemple:**

1	INPUT	A	EXE	-----	5 pas
2					
10	B = SIN	A	EXE	-----	7 pas
2					
100	PRINT "B="	;	B	EXE	-----
2		4			-----
					Total 22 pas

## 5-2 Principes de programmation

### 5-2-1 Constantes et variables

Les caractères qui peuvent être utilisés en BASIC sont les lettres majuscules de l'alphabet (A à Z) et les chiffres (0 à 9) ainsi que certains caractères spéciaux (tels que les symboles).

● **Constantes**

Une constante est une valeur numérique fixe qui peut être directement incorporée à un programme.

**Exemple:** L'expression  $S = \pi r^2$  devient  $S = \pi * R \uparrow 2$ , 2 est la constante.

● **Variables**

Une variable est une valeur numérique qui est représentée dans le programme mais qui est introduite au clavier pendant son exécution. Elle est également utilisée pour remplacer le résultat d'un calcul qui était initialement inconnu.

Une variable est représentée par une lettre majuscule (A à Z) ou une lettre majuscule avec un symbole "\$" rattaché (variable à caractères). Les variables peuvent être choisies dans cette gamme.

**Exemple:** L'expression  $S = \pi r^2$  devient  $S = \pi * R \uparrow 2$ , R est la variable.

**Exemple:**  $Y = 2 * X \uparrow 2 + 3 * X + 4$

Variable		Variable		Constante		Constante
Constante		Constante		Variable		

Autrement dit, les termes algébriques sont des "variables" et les valeurs numériques constantes sont des "constantes".

De plus, il existe des constantes à caractères et des variables à caractères.

Une constante à caractères est une chaîne de caractères entre guillemets, comme "ABC" et "END" qui est écrite directement. Une variable à caractères n'est pas une valeur numérique mais une variable qui contient une chaîne de caractères. Diverses variables à caractères peuvent être obtenues en changeant la chaîne de caractères.

De plus, une chaîne de caractères est un groupe de caractères entre guillemets, comme "123", et ne possède pas de valeur numérique. Ainsi, "123" n'est que la suite des chiffres 1, 2 et 3, et est considéré comme par exemple "ABC". Une variable à caractères est faite en rattachant un "\$" à une variable habituelle (A, B, X, Y, etc.). Les variables à caractères peuvent être choisies dans cette gamme.

**Exemple: A\$, B\$, C\$, X\$, Y\$**

Les variables à caractères peuvent être comparées ou ajoutées, mais les autres opérations telles que soustraction, multiplication et division ne peuvent pas être exécutées.

**Exemple:** Si A\$ = "123" et B\$ = "456"

par suite C\$ = A\$ + B\$, C\$ devient "123456".

(Pour C\$ = B\$ + A\$, C\$ devient "456123".)

Une variable à caractères peut contenir jusqu'à 7 caractères. En plus de ces variables à caractères, il existe aussi une variable à caractères exclusive. Cette variable à caractères exclusive est "\$" et peut contenir jusqu'à 30 caractères.

**Exemple: \$ = "1234567890ABCDEFG"**

Cette variable à caractères exclusive peut utiliser la fonction de caractères (fonction MID) qui sera décrite plus loin.

\* Les variables numériques et les variables à caractères désignées par la même lettre ne peuvent pas être utilisées en même temps.

Variable numérique A  Ne peuvent pas être utilisées en même temps  
Variable à caractères A\$

## 5-2-2 Instructions de substitution

En BASIC, les instructions de substitution ont la forme suivante.

Variable = formule numérique

Dans une instruction de substitution, le côté droit qui contient des additions, soustractions, multiplications, divisions ou autres fonctions est appelé une "formule numérique".

**Exemple: Y = 2 \* X + 3**

Dans Y = 2 \* X + 3, le terme "2 \* X + 3" du côté droit est une formule numérique. Le signe "=" ne signifie pas "égale", mais "substituer".

Dans Y = 2 \* X + 3, le côté gauche est la variable et le côté droit est la formule numérique. Ainsi, cela ne signifie pas, comme d'habitude en mathématiques, que le terme gauche Y est égal au terme droit 2 \* X + 3. Cela signifie que le résultat de 2 \* X + 3 est affecté à Y. Il peut être plus facile de comprendre Y = 2 \* X + 3 si on l'écrit Y ← 2 \* X + 3.

### 5-3-1 Ecriture de programme

Le stockage d'un programme dans la mémoire de l'ordinateur est appelé écriture de programme.

Cette opération est exécutée en frappant les entrées au clavier comme suit.

1. Désignation du mode WRT
2. Désignation de la zone de programme
3. Entrée du programme ligne par ligne (écriture)

Il y a 10 zones de programme, à savoir P0 à P9. Les programmes peuvent être écrits dans toute zone de programme.

#### (1) Désignation du mode WRT

L'écriture de programme est effectuée en mode WRT; appuyer sur **MODE** **1** et "WRT" sera affiché.

#### (2) Désignation de la zone de programme

Pour désigner la zone de programme, appuyer sur la touche **WRT** puis sur une touche numérique de **0** à **9**.

<b>WRT</b> <b>0</b> → P 0	<b>WRT</b> <b>5</b> → P 5
<b>WRT</b> <b>1</b> → P 1	<b>WRT</b> <b>6</b> → P 6
<b>WRT</b> <b>2</b> → P 2	<b>WRT</b> <b>7</b> → P 7
<b>WRT</b> <b>3</b> → P 3	<b>WRT</b> <b>8</b> → P 8
<b>WRT</b> <b>4</b> → P 4	<b>WRT</b> <b>9</b> → P 9

#### (3) Entrée du programme (écriture)

L'écriture de programme s'effectue ligne par ligne. Une ligne peut comprendre jusqu'à 62 caractères, y-compris le numéro de ligne. Appuyer sur **EXE** à la fin de la ligne.

##### ● Rôle de la touche **EXE**

La touche **EXE** est utilisée pour écrire les programmes, entrer des données et obtenir les résultats en calcul manuel. Lors de l'écriture de programme, appuyer sur la touche **EXE** une fois que la ligne a été frappée au clavier pour la stocker dans l'ordinateur. Les changements, ajouts et suppressions sont aussi confirmés par appui sur la touche **EXE**. Même si le contenu de l'affichage est changé, le contenu mémorisé ne sera pas changé si on n'appuie pas sur la touche **EXE**.

**Exemple:** Entrer le programme suivant en P0.

```

10 INPUT A,B
20 V=A+B
30 W=A-B
40 PRINT V,W
50 END

```

### Opération:

- ① Désigner le mode WRT.

**MODE** **1**

Nombre de pas restant:

**P** 0 123456789  
↑  
Zone de programme présente-  
ment désignée (clignote)

Zones de programme non écrites

- \* Cet affichage varie suivant le nombre de mémoires et la taille du programme écrit.
- \* Les numéros des zones dans lesquelles des programmes sont déjà écrits ne sont pas affichés.

- ② Désigner la zone de programme P0.

**SHIFT** **P0**

**P** 0 123456789

- ③ S'il y reste un programme antérieur, l'effacer.

**CLEAR** **EXE**

**P** 0 123456789

- \* Pour effacer toutes les zones de programme (P0 à P9), appuyer sur **C** **L** **E** **A** **R** **A** **EXE**

- ④ Ecrire la ligne numéro 10.

10 INPUT A,B **EXE**

10 INPUT A,B

Signifie un espace d'un caractère (peut être omis)

Ne pas oublier d'appuyer sur **EXE** à la fin de la ligne.

- ⑤ Ecrire la ligne numéro 20.

20 V = A + B **EXE**

20 V = A + B

- ⑥ Ecrire la ligne numéro 30.

30 W = A - B **EXE**

30 W = A - B

- ⑦ Ecrire la ligne numéro 40.

40 PRINT V,W **EXE**

40 PRINT V,W

- ⑧ Ecrire la ligne numéro 50.

50 END **EXE**

50 END

- Ecrire une commande "END" pour terminer un programme. Cette commande peut être omise dans le cas d'un programme comme ci-dessus, mais elle doit être écrite pour désigner clairement la fin d'un programme contenant des instructions GOTO ou GOSUB.



- L'espacement entre les numéros de lignes et les commandes et entre les commandes et les opérandes permet de faciliter la lecture de l'affichage. En langage BASIC, cet espacement n'a pas de signification particulière, sauf pour les messages tels que PRINT.
- Dans ce programme, les numéros de ligne ont été choisis par incréments de 10 mais ils peuvent être librement choisis entre 1 et 9999. Toutefois, cet incrément de 10 est plus pratique pour effectuer des ajouts et insertions par la suite. Etant donné que l'exécution du programme est faite dans l'ordre croissant des numéros de ligne, utiliser des numéros de ligne dans l'ordre d'exécution désiré.
- Pour effacer le programme d'une zone de programme, utiliser une commande CLEAR. Pour effacer tous les programmes des zones P0 à P9, utiliser une commande CLEAR A.

### 5-3-2 Exécution de programme

L'exécution de programme se fait en mode RUN. (Appuyer sur **MODE** **□** et "RUN" sera affiché.) Il existe deux méthodes d'exécution d'un programme écrit.

#### 1. Méthodes d'exécution de programme

##### ① Exécution par désignation de zone de programme

Dans cette méthode, l'exécution commence dès que la zone de programme est désignée.

**□**  $\left\{ \begin{array}{l} \text{P0} \\ \text{□} \\ \{ \\ \text{P9} \\ \text{□} \end{array} \right\}$  (Appuyer sur **□** puis sur la touche de zone de programme désirée.)

**Exemple:** Démarrer le programme de l'exemple précédent.

**Opération:** **□** **□**

Mode RUN (omis ci-après)

?

\* Ce signe "?" est affiché parce qu'il y a une instruction INPUT au début du programme.

##### ② Exécution par commande RUN

RUN **□** ("RUN" peut être entré en appuyant sur **R** **U** **N** **□** ou **□** **□** **□** **□**.)

?

- \* En reprenant l'exemple précédent, "?" est affiché. Quand le programme est en condition d'attente d'entrée, ce signe ne peut pas être effacé par appui sur **□**. Par conséquent, appuyer sur **MODE** **□** puis exécuter l'opération ② pour entrer de nouveau les données.

Pour obtenir une exécution à mi-programme, entrer le numéro de ligne après la commande RUN puis appuyer sur la touche **□**.

**Exemple:** Démarrer à la ligne 20.

**Opération:** RUN 20 **EXE**

\* Avec la méthode ①, il est inutile de désigner la zone du programme à exécuter. Toutefois, pour la méthode ②, cette désignation est nécessaire. (Si la zone de programme était différente, le programme écrit dans cette zone serait exécuté.)

## 2. Entrée au clavier pendant l'exécution de programme

L'entrée au clavier pendant l'exécution de programme peut être faite à l'aide d'une instruction INPUT et d'une fonction KEY. L'entrée au clavier par la fonction KEY consiste en une seule touche, mais si aucune touche n'est frappée, l'exécution continue. Pour l'entrée au clavier utilisant une instruction INPUT, un "?" est affiché et le programme se met en condition d'attente d'entrée. L'exécution reprend par appui sur la touche **EXE** une fois que les données sont entrées.

**Exemple:** Exécuter le programme écrit en P0 dans l'exemple précédent.

**Opération:**

- Exécuter le programme

**SHIFT** **PROG**

?

- Dans ce programme, vu que deux variables sont entrées, entrer d'abord la valeur de la variable A.

47 **EXE**

?

- Ensuite, entrer la valeur de la variable B.

69 **EXE**

116

**EXE**

-22

Ainsi, une donnée est entrée pendant l'exécution comprenant une instruction INPUT comme suit.

donnée **EXE**

Quand le programme est en état d'attente d'entrée par suite d'une instruction INPUT, d'autres opérations, telles que calcul manuel, peuvent être effectuées.

Toutefois, dès qu'on appuie sur la touche **EXE**, le résultat affiché sera rentré comme donnée. Quand on veut arrêter l'exécution du programme lorsqu'il est en état d'attente d'entrée, appuyer sur **SHIFT** **PROG**.

## 5-4 Edition de programme

- L'édition de programme sert à vérifier, un programme est logiquement correct. Cette opération permet d'effectuer des changements/ajouts/suppressions de lignes complètes et de changer les numéros de ligne.
- L'édition de programme est effectuée en appelant chaque ligne à l'aide d'une commande LIST.
- La commande LIST peut être utilisée et en mode RUN et en mode WRT. Quand elle est utilisée en mode RUN, le contenu du programme est affiché. Quand elle est utilisée en mode WRT, le programme peut être modifié.

### 1. Affichage de listage de programme en mode RUN

Opération:

LIST **EXE**

(LIST peut être entré en appuyant sur

**L I S T** **EXE** ou **SHIFT** **LIST** **EXE**.)

10 INPUT A,B	Affiché pendant environ 2 secondes (idem pour la suite)
20 V=A+B	
30 W=A-B	
40 PRINT V,W	
50 END	
READY P0	

Si le listage n'est pas nécessaire à partir du début, désigner le numéro de ligne. Pour lister à partir de la ligne numéro 30.

Opération:

LIST 30 **EXE**

30 W=A-B
40 PRINT V,W
50 END
READY P0

- \* Pendant l'exécution de la commande LIST, chaque ligne est séquentiellement affichée. Pour interrompre le déroulement de cet affichage, appuyer sur la touche **STOP**. Pour reprendre l'exécution de la commande LIST, appuyer sur la touche **EXE**.

### 2. Changement/ajout/suppression en mode WRT

Désigner le mode WRT en appuyant sur **MODE** **1**.

#### Changement

① chaque appui sur la touche **EXE**, une ligne est affichée. Cet affichage commence par la liste qui est désignée dans la commande LIST. Si la désignation de numéro de ligne est omise, l'affichage commence à la première ligne.

### a. Changement partiel

**Exemple:** Remplacer le "+" de la ligne 20 de l'exemple précédent par "\*".

**Opération:**

- Si la zone de programme P0 n'est pas désignée, désigner P0.

**INVT** **P0**

P 123456789

Clignote signifie qu'un programme est écrit et indique la zone de programme actuellement désignée.

- Appeler la ligne 20 à l'aide d'une commande LIST.

LIST 20 **EXE**

20 V=A+B

- Amener le curseur sous le "+".

**←** **→**

20 V=A±B

- \* Un déplacement rapide et continu du curseur peut être obtenu en appuyant sur la touche de déplacement de curseur (**←**, **→**) pendant plus d'une seconde.

- Faire le changement.

**3** **EXE**

30 W=A-B

- \* Ne pas oublier d'appuyer sur la touche **EXE** sinon seul l'affichage sera changé et le programme écrit ne le sera pas.

- A présent, la ligne 30 est changée. Appuyer sur **AC** pour terminer le changement.

**AC**

—

- \* L'appui sur des touches quand on se trouve sur une ligne ne nécessitant pas de changement entraînera un changement inutile; n'appuyer sur aucune touche autre que **EXE** et **AC**.

- En listant le programme, contrôler si le changement a été fait.

**MEM** **☐**

LIST **EXE**

READY P0

10 INPUT A,B

20 V=A\*B

30 W=A-B

40 PRINT V,W

50 END

READY P0

## b. Changement de toute une ligne

**Exemple:** Remplacer "W = A - B" de la ligne 30 par "W = V/2".

**Opération:**

```
P 123456789
```

- Ecrire la nouvelle ligne 30.

30 W V 2

```
30 W=V/2
```

- Contrôler le changement en listant le programme.

LIST

```
READY P0
```

```
10 INPUT A,B
```

```
20 V=A*B
```

```
30 W=V/2
```

```
40 PRINT V,W
```

```
50 END
```

```
READY P0
```

## ② Ajout

Il est possible d'ajouter une ou plusieurs lignes entre les lignes existantes.

**Exemple:** Ajouter "U = V \* 2" entre les lignes 30 et 40 de l'exemple précédent et remplacer la ligne 40 par "PRINT V, W, U".

**Opération:**

```
P 123456789
```

- Entrer la ligne numéro 35 pour faire l'ajout entre les lignes 30 et 40.

35 U V 2

```
35 U=V*2
```

\* Pour faire l'ajout entre les lignes 30 et 40, le numéro de ligne peut être librement choisi entre 31 à 39.

- Pour changer la ligne 40, l'appeler à l'aide d'une instruction LIST puis ajouter "U".

LIST 40

U

```
40 PRINT V,W
```

```
50 END
```

- Lister le programme pour contrôler les ajouts.

**NRX** **☐**  
LIST **EXE**

```
READY P0
10 INPUT A,B
20 V=A*B
30 W=V/2
35 U=V*2
40 PRINT V,W
PRINT V,W,U
50 END
READY P0
```

### ③ Suppression

#### a. Suppression partielle

**Exemple:** Supprimer "V," de la ligne 40 de l'exemple précédent.

**NRX** **1**

```
P 123456789
```

- Appeler la ligne 40 à l'aide d'une instruction LIST puis amener le curseur sous le "V".

LIST 40 **EXE**

```
40 PRINT V,W
```

**☐** **☐**

```
40 PRINT V,W
```

- Supprimer "V," en appuyant sur la touche **DEL**.

**DEL** **DEL**

**EXE**

```
40 PRINT W,U
50 END_
```

- \* Si on n'appuie pas sur la touche **EXE**, le contenu du programme ne sera pas changé.

**AC**

```
502
```

- \* Ne pas oublier d'appuyer sur **AC** pour annuler l'état de changement pour la ligne 50.

- Lister le programme pour contrôler la suppression.

**MEM** **☐**

LIST **EXE**

```

READY P0
10 INPUT A,B
20 V=A*B
30 W=V/2
35 U=V*2
40 PRINT W,U
50 END
READY P0
  
```

**b. Suppression de toute une ligne**

Si on entre le numéro de la ligne à supprimer, toute cette ligne sera supprimée.

**Exemple:** Supprimer la ligne 30.

Opération: **MEM** **1**

```

P 502 123456789
  
```

- Entrer le numéro de la ligne 30.

30 **EXE**

```

510
  
```

- Contrôler la suppression.

**MEM** **☐**

LIST **EXE**

```


READY P0
10 INPUT A,B
20 V=A*B
35 U=V*2
40 PRINT W,U
50 END
READY P0
  
```

#### ④ Changement de numéro de ligne

**Exemple:** Ecrire le programme suivant dans P2.

```
10 INPUT N
20 M=N*N
30 L=SQR N
40 PRINT M,L
50 END
```

Amener la ligne 20 entre les lignes 30 et 40.

**Opération:**  **F1**









```
P 1 3456789
```

- Appeler la ligne 20 à l'aide d'une commande LIST.

LIST 20  **F2**


```
20 M=N*N
```

- Amener le curseur sous le "2" du numéro de ligne "20".


       

```
20 M=N*N
```

- Remplacer 20 par 35 et entrer.

35  **F2**

```
30 L=SQR N
```

- Pour terminer le changement, appuyer sur  **F3** afin de quitter l'état de changement.

 **F3**

- Lister le programme pour voir comment le contenu a été changé.

  **F2 F1**

LIST  **F2**

```
READY P2
```

```
10 INPUT N
```

```
20 M=N*N
```

```
30 L=SQR N
```

```
35 M=N*N
```

```
40 PRINT M,L
```

```
50 END
```

```
READY P2
```



- Maintenant, le contenu de la ligne 20 a été déplacé entre les lignes 30 et 40 mais la ligne 20 est toujours là; il faut donc la supprimer.

MOR 1

20 EXE

P	1	3456789
---	---	---------

- Ceci termine le changement de numéro de ligne. Contrôler en listant le programme.

MOR 2

LIST EXE

READY	P2
10	INPUT N
30	L=SQR N
35	M=N*N
40	PRINT M,L
50	END
READY	P2

## 5-5 Mise au point de programme

### (1) Système de mise au point de programme

Le système de mise au point de cet ordinateur est divisé en mise au point structurale et mise au point par conversation avec l'affichage.

#### Système de mise au point

I. Mise au point structurale

a. Mise au point complète  
Contrôle de la structure logique du programme.

b. Mise au point partielle  
Contrôle par ligne de programme.

II. Mise au point par conversation avec l'affichage  
Emploi de la fonction contrôle automatique de l'ordinateur pour contrôler le déroulement du programme et identifier les erreurs de syntaxe de langage BASIC.

La mise au point structurale est exécutée pendant la programmation; nous décrivons ici la mise au point par conversation via l'affichage.

## (2) Mise au point conversationnelle

Si une erreur se produit pendant l'exécution du programme, un message d'erreur est affiché. Ces messages indiquent le numéro de ligne et le type de l'erreur. Sur la base du message d'erreur affiché, la mise au point est faite manuellement par conversation avec l'affichage. Pour la signification des messages d'erreur, voir la Liste de messages d'erreur, page 61.

### Exemple:

```
10 INPUT X
20 IF X ≤ 0; PRINT "X ≤ 0"; GOTO 10
30 Y = X↑2 + 3 * X + 15
40 PRINT Y
50 END
```

La ligne 20 du programme ci-dessus est un jugement de la plage d'entrée de la fonction puissance ( $x \uparrow y$ ). Quand  $x < 0$ , l'exécution du programme entraîne un retour à l'instruction INPUT de la ligne 10.

Supposer que la ligne 30 du programme ci-dessus est mal entrée comme suit.

```
30 Y = X↑2 + 3X + 15
```

### Opération:

- Si l'exécution de ce programme est commandée, un "?" est affiché par suite de l'instruction INPUT de la ligne 10.

**MODE** **☐** **RUN** **EXE**

?

- Maintenant, entrer 45.

45 **EXE**

ERR2 P0-30

- Le message d'erreur indique qu'il y a une erreur de syntaxe à la ligne 30. Le contenu du programme doit donc être contrôlé.

**AC** **MODE** **T**

LIST 30 **EXE**

P \_ 123456789

30 Y = X↑2 + 3X ±

- Le "\*" a été oublié entre "3" et "X" à la ligne 30. Corriger cette ligne en suivant la procédure d'édition de programme.

**☐** **DEL** **MS**

**X** **EXE**

30 Y = X↑2 + 3\_ X

40 PRINT Y\_

### (3) Mise au point d'un programme pendant son exécution

La mise au point conversationnelle est effectuée d'après les informations délivrées par l'ordinateur sous la forme de messages d'erreur. Toutefois, quand aucun message d'erreur n'est affiché alors que le résultat des calculs n'est pas celui escompté, répéter l'exécution du programme et contrôler les résultats de calculs pas à pas.

Il existe deux façons d'effectuer ce contrôle: (1) l'exécution est interrompue à l'aide d'une commande STOP; (2) l'exécution est faite ligne par ligne en mode TR (analyse).

#### ■ Mise au point à l'aide d'une commande STOP

**Exemple:** Ecrire le programme suivant.

```
10 Y=0
20 INPUT N,X
30 FOR I=1 TO N
40 Y=Y+X*X
50 NEXT I
60 PRINT Y
70 END
```

Afin d'obtenir la valeur de Y dans cette boucle FOR-NEXT, le résultat de chaque boucle est visualisé à l'aide d'une instruction STOP.

#### Opération:

- L'instruction STOP doit être placée juste après la formule de calcul, c'est à dire entre les lignes 40 et 50.

**MOD** **1**

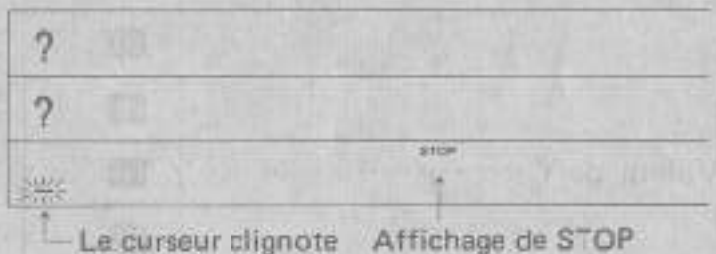
45 STOP **EXE**

- Par cette instruction, une fois que le calcul de la ligne 40 est terminé, le déroulement du programme est arrêté, et un contrôle peut être fait.

**MOD** **1** **RUN** **EXE**

4 **EXE**

87 **EXE**



- Quelle est la valeur de Y au moment de l'arrêt?

Y **EXE**

7569 **STOP**

- Quand le programme est redémarré, il s'arrête à l'instruction STOP suivante et la valeur de Y peut de nouveau être obtenue.

EXE  
Y EXE

—	STOP
15138	STOP

- En répétant cette opération, le processus de calcul peut être vu.

Cet exemple montre un programme simple; lorsqu'il s'agit d'un programme complexe, il devient très difficile de contrôler le processus de calcul à l'aide de la mise au point structurelle. Si les variables sont contrôlées comme expliqué ci-dessus à l'aide d'instructions STOP, les erreurs de programmation peuvent être trouvées et corrigées plus facilement.

### ■ Mise au point en mode TR (analyse)

Si l'exécution du programme est faite en mode TR (appuyer sur **MODE** **2**), le déroulement s'arrête à chaque ligne pour faciliter la mise au point.

Utilisons le mode TR pour mettre au point l'exemple qui vient d'être mis au point à l'aide d'une commande STOP.

#### Opération:

Désigner le mode RUN..... **MODE** **0**

Désigner le mode TR..... **MODE** **2**

RUN **EXE**

**EXE**

Contrôler le processus d'exécution..... **STOP**

**STOP**

Continuer l'exécution du programme..... **EXE**

4 **EXE**

87 **EXE**

**STOP**

**EXE**

**EXE**

**EXE**

Valeur de Y..... Y **EXE**

**EXE**

Répété par la suite

READY P0		
READY P0	TR	
P0-10	TR	STOP
?	TR	
?	TR	STOP
P0-20	TR	STOP
?		"TR" et "STOP" sont omis par la suite.
?		
—		
P0-20		
P0-30		
P0-40		
7569		
P0-45		

La mise au point en mode TR est idéale pour l'examen du déroulement global du programme et est pratique pour découvrir l'endroit où les erreurs se sont produites.

### 5-6-1 Commandes d'entrée

#### ● Instruction d'entrée

Une commande d'entrée est utilisée pour entrer les données pendant le calcul programmé.

Une instruction d'entrée est utilisée pour entrer des données dans une variable pendant l'exécution du programme. L'exécution du programme s'arrête après l'affichage d'un "?".

**Format:** INPUT ["chaîne de caractères",] variable ["chaîne de caractères", variable]

(Les articles entre crochets peuvent être omis.)

La "chaîne de caractères" peut être omise. Toutefois, quand elle est écrite, étant donné que les caractères entre guillemets seront affichés avant le point d'interrogation, elle peut être utilisée comme message pendant l'entrée.

Les variables suivant l'instruction INPUT peuvent être des variables numériques (A, B, etc.), des variables à caractères (XS, Y\$, etc.) et la variable à caractères exclusive (\$). Elles peuvent être écrites à la suite en utilisant des virgules (",").

#### Exemple:

INPUT A

?
DATA=?

INPUT "DATA=" ,A

Par suite d'une instruction INPUT, un "?" est affiché et l'appareil se met en état d'attente d'entrée. Dans ce cas, une fois que les données sont entrées et qu'on a appuyé sur la touche **EXE**, l'exécution du programme est reprise.

Quand l'appareil est en état d'attente d'entrée, l'appui sur la touche **AC** n'entraîne pas l'annulation de cet état. Quand on veut interrompre le programme, appuyer sur **MEM** **ON/OFF**.

- Les données qui peuvent être entrées à l'aide d'une instruction INPUT sont: valeurs numériques, résultats (réponses) d'expression numérique (pour les variables numériques) et chaînes de caractères (pour les variables à caractères).

Cas de INPUT A:

Valeur numérique ..... 123 **EXE** → A=123

Résultat d'une expression numérique ..... 14 **×** 25 **EXE** → A=350

Cas de INPUT B\$:

Chaîne de caractères: ..... ABC **EXE** → B\$=ABC

789 **EXE** ← B\$=789

D'autres variables numériques peuvent aussi être utilisées comme entrée pour des variables numériques.

Cas de INPUT A (X = 987654)

Variable ..... X **EXE** → A=X

=987654

### ● Fonction KEY

Cette fonction est utilisée pour entrer un caractère dans une variable à caractères en appuyant sur une touche pendant l'exécution du programme. Cette fonction est différente d'une instruction INPUT et n'entraîne pas le passage en état d'attente d'entrée (affichage de "?"). L'exécution du programme se poursuit même s'il n'y a pas d'entrée au clavier.

**Format:** variable à caractères = KEY

A\$, \$, etc. sont utilisées pour la variable à caractères.

#### Exemple:

```
10 A$=KEY
20 IF A$="A" THEN 100
30 IF A$="B" THEN 200
40 IF A$="C" THEN 300
50 GOTO 10
  .
  .
  .
```

Ce programme présente une entrée de données en utilisant la fonction KEY et une partie de la distribution. Toutefois, un test est fait à l'aide d'instructions IF pour voir si le caractère lu par la fonction KEY de la ligne 10 a été entré.

Avec la fonction KEY, même si on n'appuie pas sur la touche **EXE**, seule la première entrée au clavier est lue. Toutefois, étant donné que l'exécution du programme ne s'arrête pas comme lorsque l'on utilise une instruction INPUT, un état d'attente d'entrée est obtenu en incorporant les instructions IF suivantes.

Les instructions IF des lignes 20 à 40 sont des commandes décision et effectuent une distribution en utilisant les variables à caractères qui ont été entrées en utilisant la fonction KEY. Pour plus de détails concernant les instructions IF, voir page 48.

## 5-6-2 Commandes de sortie

### ● Instruction PRINT

Une instruction PRINT est utilisée pour afficher le résultat d'un calcul ou une donnée. Elle affiche la chaîne de caractères, le contenu de la variable et le résultat de calcul suivant la commande.

**Format:** PRINT [ CSR ] [ ( expression-numérique ) ( ( ; ) ..... ) ]

Chacun des articles entre || peut être utilisé.

Les articles entre [ ] peuvent être omis.

La fonction de commande de sortie suivant l'instruction PRINT est une fonction CSR, qui désigne la position où la donnée suivante doit être affichée.

Pour l'expression numérique, une variable ou formule ce calcul est écrite. Dans le cas d'une variable, son contenu sera affiché. Dans le cas d'une formule de calcul, le résultat du calcul sera affiché.

**Exemple:**

PRINT A (A = 12345)	12345
PRINT 789	789
PRINT A*2 (A = 147)	294
PRINT B\$ (B\$ = "PB-100")	PB-100

Dans le cas d'une expression de caractères, les caractères entre guillemets seront affichés.

**Exemple:**

PRINT "ABC"	ABC
PRINT "XYZ" + "123"	XYZ123

Les expressions numériques ou les expression de caractères peuvent être écrites a la suite en utilisant un ";" ou un ",". Toutefois, le nombre de caractères qui peuvent être écrits sur une ligne doit être au maximum de 62, y compris le numéro de ligne. Le nombre de caractères entre guillemets dans la chaîne de caractères doit être au maximum de 30.

La différence entre le ";" et la "," est qu'avec le ";", l'expression numérique ou l'expression de caractères sera affichée à la suite de l'expression précédente, alors qu'avec la ",", l'affichage d'arrêtera a la première chaîne, un STOP sera affiché, pour obtenir l'affichage de la deuxième chaîne (après la ",",) il faudra appuyer sur **EXE**.

Quand un ";" n'est pas écrit après la donnée, "STOP" est affiché une fois que la donnée a été affichée, et l'exécution du programme est interrompue. Par conséquent, quand on veut afficher la donnée suivante ou reprendre l'exécution du programme, appuyer sur la touche **EXE**.

● **Fonction CSR**

La fonction CSR est une fonction de commande de sortie qui désigne la position où la donnée doit être affichée.

**Format:** PRINT CSR expression numérique\* { { \* } { expression numérique } { ; } { expression de caractères } ..... }

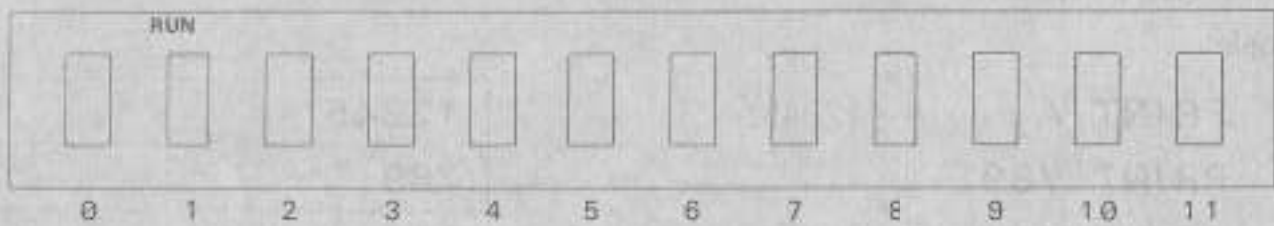
Chacun des articles entre | | peut être utilisé.

Les articles entre { } peuvent être omis.

\* La valeur de l'expression numérique est 0 à 11 en ignorant la partie décimale.

En utilisant la valeur de cette expression numérique, on désigne à quelle position à partir de la gauche l'affichage de la donnée de sortie commencera.

La méthode de comptage des positions d'affichage est donnée ci-dessous.



**Exemple:**

```

PRINT A      (A = 12345)
PRINT CSR 1;A
PRINT CSR 5;A
PRINT B$     (B$ = ABCDE)
PRINT CSR 2;B$
PRINT CSR 10;B$
    
```

12345
12345
12345
ABCDE
ABCDE
AB
ABCDE

\* Si une ", " est utilisée au lieu d'un ";" à la suite de la fonction CSR, l'affichage est effacé une fois puis l'affichage est fait à partir de la gauche lorsqu'on appuie sur la touche **ESC**.

### 5-6-3 Commande de saut

#### ● Instruction GOTO

Une instruction GOTO est aussi appelée "saut inconditionnel" et est une commande qui fait passer inconditionnellement l'exécution du programme à une endroit désigné (numéro de ligne).

**Format:** GOTO { expression numérique ..... numéro de ligne (1 à 9999)  
 # expression numérique ..... numéro de programme (0 à 9)

Quand une expression numérique est écrite juste après l'instruction GOTO, l'exécution du programme passe à un numéro de ligne. Quand un "#" est écrit juste après l'instruction GOTO, l'exécution du programme passe à une zone de programme.

**Exemple:**

```

GOTO 10      ..... saut à la ligne 10
GOTO N      ..... saut à la ligne dont le numéro est la valeur de la variable N
                (saut à la ligne 30 si N contient 30)
    
```



- GOTO  $A*100$  ..... saut à la ligne dont le numéro est le résultat de  $A * 100$   
(saut à la ligne 200 si A contient 2)
- GOTO #2 ..... saut à la zone de programme P2
- GOTO #X ..... saut à la zone de programme dont le numéro est la valeur de  
la variable X (saut à la zone de programme P8 si X contient  
8)
- GOTO #P+1 ..... saut à la zone de programme dont le numéro est le résultat  
de P+1 (saut à la zone de programme P5 si P vaut 4)

Une instruction GOTO est utilisée pour répéter l'exécution du programme à partir du début ou pour sauter à un autre programme pour exécuter un calcul particulier.

#### 5-6-4 Commande de décision

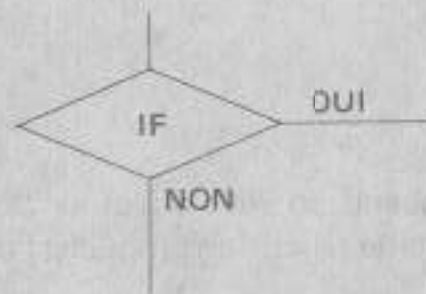
##### • Instruction IF

Une instruction IF est aussi appelée "saut conditionnel". Cette commande est utilisée pour effectuer une opération ou pour sauter à un endroit désigné uniquement quand une certaine condition est satisfaite.

**Format:** IF expression de comparaison { THEN numéro de ligne ou #n (n=0 à 9)  
; commande ou instruction de substitution }

L'expression de comparaison suivant le "IF" compare le côté droit et le côté gauche de "=" ou "<". Si la réponse à cette comparaison est OUI, l'exécution du programme passe à l'endroit désigné après "THEN" ou ";" . Si la réponse à la comparaison est NON, l'exécution du programme passe à la ligne suivante.

Cette opération est indiquée dans l'organigramme suivant.



Ceci signifie que si la condition de l'instruction IF est remplie, le déroulement va dans la direction "OUI", et si la condition n'est pas remplie, le déroulement va dans la direction "NON".

Autrement dit, une instruction IF indique un branchement et choisit l'opération suivante à la suite d'un jugement. Une instruction IF peut être utilisée pour terminer une boucle (répétition) quand le nombre de données est inconnu ou pour choisir l'opération suivante en fonction d'un résultat de calcul, etc.

Constantes, variables, expressions numériques, constantes à caractères et variables à caractères peuvent être utilisées pour cette comparaison.

$A > 10$	variable et constante (si A est supérieure à 10 → OUI)
$X \geq Y$	variable et variable (si X est égale ou supérieure à Y → OUI)
$N = L + 3$	variable et expression numérique (si N est égale à la somme de L et 3 → OUI)
$A\$ = "XYZ"$	variable à caractères et constante à caractères (si la chaîne de caractères contenue dans A\$ est égale à "XYZ" → OUI)
$P\$ \neq Q\$$	variable à caractères et variable à caractères (si la chaîne de caractères de P\$ et la chaîne de caractères de Q\$ sont différentes → OUI)

- \* La comparaison de variables numériques et de variables à caractères ne peut pas être faite.
- \* La comparaison de chaînes de caractères est basée sur le code ASCII.

"THEN" ou ";" sont utilisées séparément suivant ce qui suit.

```

THEN 150 (numéro de ligne)
THEN #5 (zone de programme)

; PRINT A
; Z = X + Y

```

### 5-6-5 Commande de boucle

#### ● Instruction FOR-NEXT

Une instruction FOR-NEXT est utilisée quand on veut effectuer des opérations similaires de manière répétée et quand le nombre de répétitions (boucles) est connu.

**Format:** FOR variable =  $\underbrace{n}_{\substack{\text{valeur} \\ \text{initiale}}}$  TO  $\underbrace{m}_{\substack{\text{valeur} \\ \text{finale}}}$  [STEP  $\underbrace{\ell}_{\substack{\text{incrément}}}$ ]  
 }  
 NEXT variable

(L'article entre crochets peut être omis)  
 ( $n$ ,  $m$  et  $\ell$  sont des expressions numériques)

Autrement dit, ceci est une commande pour exécuter de manière répétée la commande entre "FOR" et "NEXT" tandis qu'une variable change de  $n$  à  $m$  par incréments de  $\ell$ . Quand l'exécution atteint  $m$ , elle passe à la commande suivant "NEXT".

**Exemple:**

Augmenter la variable I par incréments de 2 entre 1 et 10.

```
FOR I=1 TO 10 STEP 2
  {
NEXT I
```

Diminuer la variable A par décréments de 0,5 entre 50 et 1.

```
FOR A=50 TO 1 STEP -0.5
  {
NEXT A
```

Augmenter la variable P par incréments de 1 entre Q et R.

```
FOR P=Q TO R
  {
NEXT P
```

\* Quand l'augmentation est faite par incréments de 1, "STEP" peut être omis.

**\* Emboîtement**

Jusqu'à 4 boucles FOR-NEXT pouvant être groupées. Ce groupement est appelé "emboîtement".

```
FOR A = .....
  FOR B = .....
    FOR C = .....
      FOR D = .....
        {
      NEXT D
    NEXT C
  NEXT B
NEXT A
```

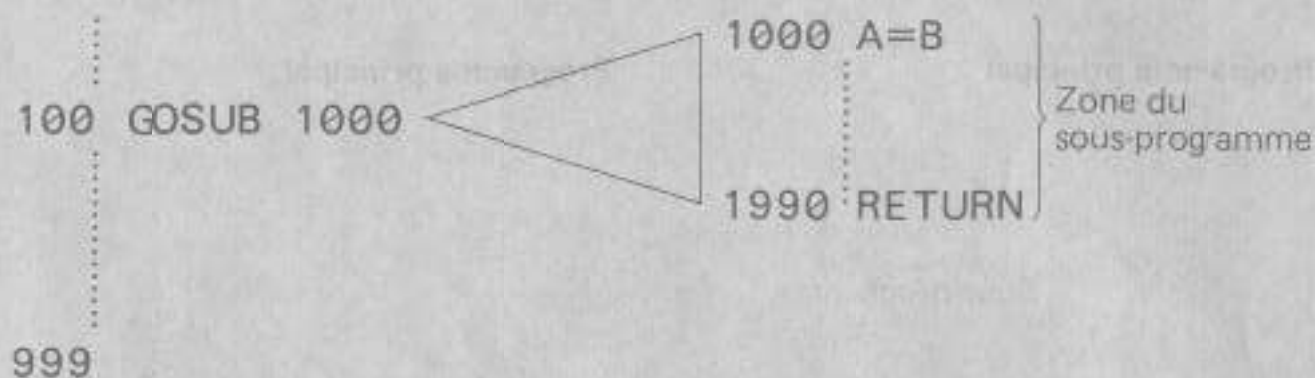
Ceci montre un emboîtement à 4 niveaux.

Lors d'un tel emboîtement, il faut prendre garde à l'instruction NEXT correspondant à l'instruction FOR, et aux variables concernées.



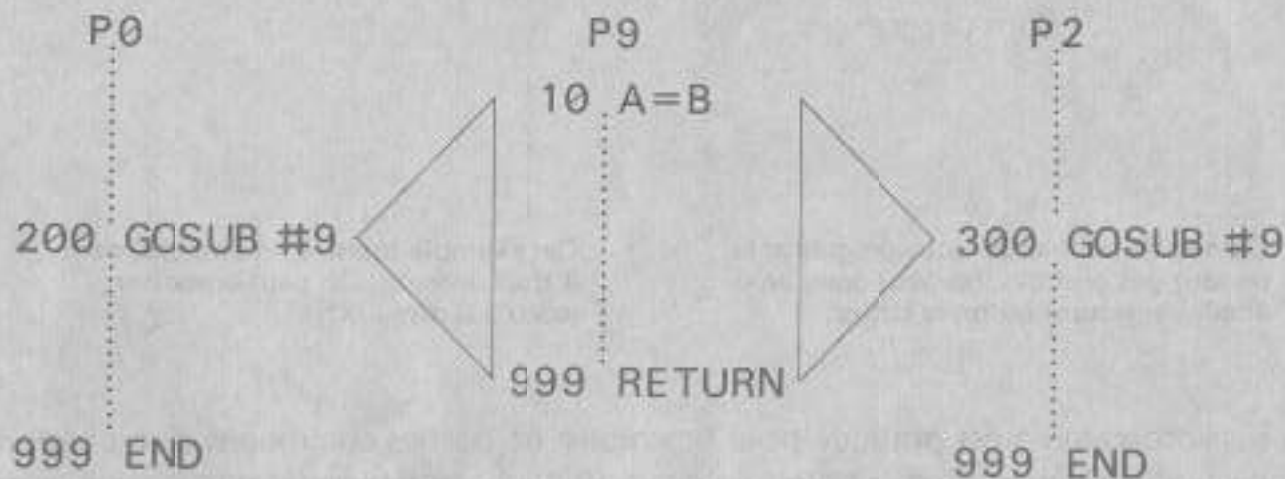
L'expression numérique suivant l'instruction GOSUB indique le numéro de la première ligne de la zone du sous-programme. S'il n'y a pas d'instruction RETURN à la fin de la zone du sous-programme, l'exécution ne peut pas revenir au programme principal.

**Exemple:**



L'expression numérique suivant une instruction GOSUB peut être soit une variable numérique soit une formule de calcul. Dans le cas d'une variable ou expression numérique, le sous-programme qui est appelé sera différent suivant la valeur contenue dans la variable ou le résultat de l'expression numérique. Quand un '#' précède l'expression numérique, une autre zone de programme (P0 à P9) est utilisée comme sous-programme. Cette méthode est très pratique car elle permet d'utiliser le même sous-programme pour des programmes différents.

**Exemple:**

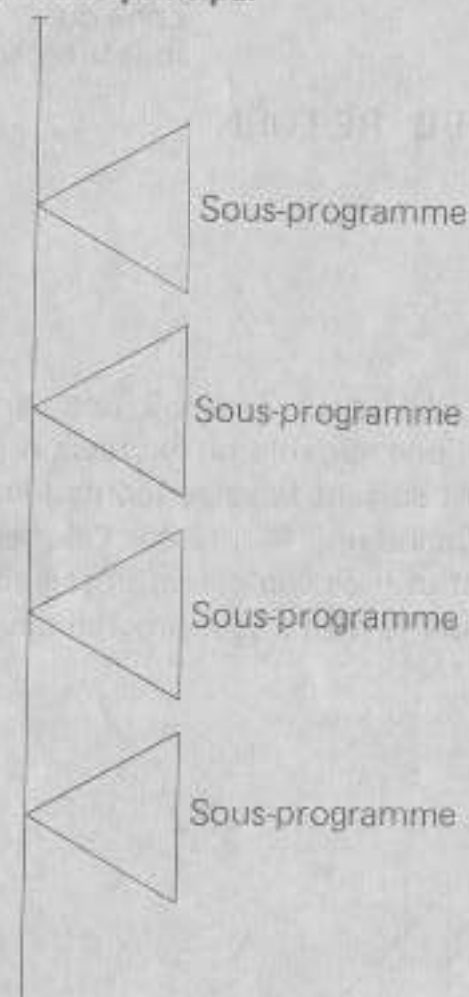


## • Emboîtement

Tout comme pour l'instruction FOR•NEXT, des instructions GOSUB peuvent aussi être groupées. Le nombre de fois que le sous-programme est appelé est fixe.

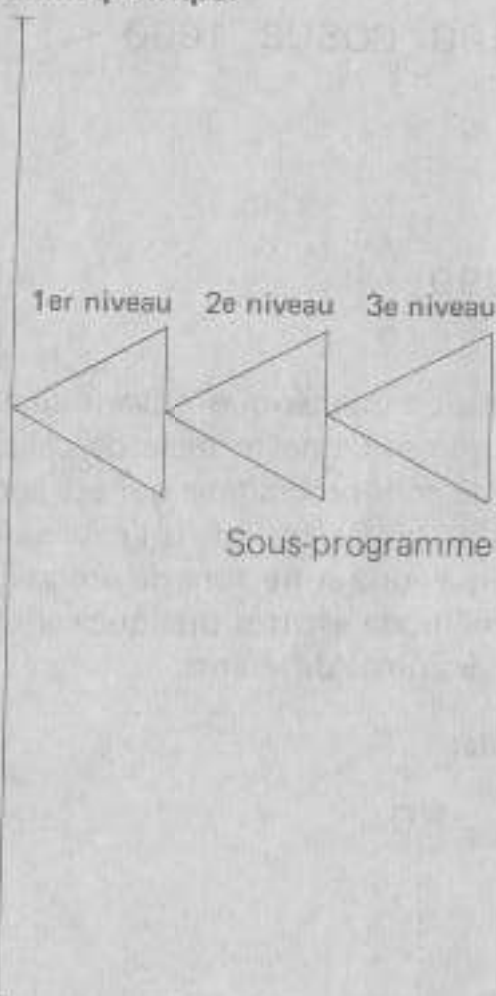
Cet emboîtement peut être effectué jusqu'à 8 niveaux. Un sous-programme peut être appelé à partir d'un autre sous-programme.

Programme principal



Dans cet exemple, les sous-programmes ne sont pas groupés; on peut donc en employer autant qu'on le désire.

Programme principal



Cet exemple montre un emboîtement à trois niveaux. On peut emboîter jusqu'à 8 niveaux.

Un sous-programme est pratique pour assembler les parties communes d'un programme principal afin d'économiser le nombre de pas ou pour assembler des portions séparément comme sous-programmes quand on assemble un programme complexe.

### 5-6-7 Instruction multiple

Une instruction multiple est utilisée pour réunir plusieurs commandes à l'aide de " : ".


Exemple:

```
10 A=2  
20 B=10  
30 C=50  
10 A=2:B=10:C=50  
  
10 PRINT "NO." :N:  
20 INPUT A  
10 PRINT "NO." :N: : INPUT A
```

**Note:** Une commande VAC (effacement de mémoire) ne peut pas être utilisée dans une instruction multiple.

### 5-6-8 Commande d'arrêt

#### • Instruction STOP

Une instruction STOP est une commande pour arrêter temporairement l'exécution du programme. Quand un programme est arrêté avec une telle commande, appuyer sur la touche  pour reprendre son exécution.

Tout nombre de commandes STOP peut être utilisé dans un programme.

### 5-6-9 Commande de fin

#### • Instruction END

Une instruction END est une commande pour terminer l'exécution du programme. L'exécution du programme ne peut pas être reprise comme dans le cas d'une instruction STOP.

Cette instruction END est écrite à la fin d'un programme. Quand un sous-programme suit le programme principal, ne pas oublier d'écrire une instruction END à la fin du programme principal.

### 5-6-10 Commande d'exécution




#### • Commande RUN

Une commande RUN est utilisée pour exécuter un programme. Elle ne peut pas être utilisée par écriture dans un programme.

**Format:** RUN [numéro de ligne] (L'article entre crochets peut être omis)

Quand cette commande est suivie d'un numéro de ligne, le programme démarra à la ligne correspondante. Quand le numéro de ligne est omis, le programme démarre à la première ligne.

Exemple:

```
RUN  ..... démarrage à la première ligne  
RUN 20  ..... démarrage à la ligne 20  
RUN 55  ..... démarrage à la ligne 55
```

Quand le numéro de ligne désigné n'existe pas, l'exécution démarre à la ligne portant le numéro le plus proche.

## 5-6-11 Commande de listage

### ● Commande LIST

Une commande LIST est utilisée pour afficher le contenu du programme. Elle peut être utilisée en mode "RUN" et en mode "WRT".

**Format:** LIST [numéro de ligne] (L'article entre crochets peut être omis)  
LIST A

Quand cette commande est suivie d'un numéro de ligne, l'affichage est fait dans l'ordre à partir de la ligne correspondante de la zone de programme actuellement désignée. Quand aucun numéro de ligne n'est désigné, l'affichage est fait à partir du début du programme.

La commande "LIST A" permet d'afficher les programmes de toutes les zones de programme. Elle affiche les contenus des programmes dans l'ordre de P0 à P9. Cette commande ne peut pas être utilisée par écriture dans un programme.

Quand le listage est effectué en mode "RUN", le contenu du programme est affiché dans l'ordre à partir de la ligne désignée. Quand le listage est effectué en mode "WRT", une ligne est affichée à chaque appui sur la touche **EXE**.

### Exemple:

(Mode RUN)

LIST <b>EXE</b>	10 A=0
	20 INPUT B
	30 A=A+B
	40 GOTO 20

(Mode WRT)

LIST 20 <b>EXE</b>	20 INPUT B_
<b>EXE</b>	30 A=A+B_
<b>EXE</b>	40 GOTO 20_

De plus, lors du listage en mode "WRT", la modification de programme (voir page 34) est possible, faire attention.

## 5-6-12 Désignation de mode

### ● Commande MODE

La commande MODE est utilisée pour désigner l'unité angulaire ou l'état de sortie d'imprimante dans un programme.

**Format:** MODE n (n = 4 à 8)

MODE 4	Désignation de *DEG*	} Désignation d'unité angulaire
MODE 5	Désignation de *RAD*	
MODE 6	Désignation de *GRA*	
MODE 7	Désignation du mode PRINT	
MODE 8	Libération du mode PRINT	

Cette commande MODE, faite par écriture lettre par lettre dans un programme, a une fonction identique à celle de la touche **MODE** lors de l'utilisation manuelle.

Mais en programmation on ne peut pas utiliser la touche **MODE** il faut composer le mot MODE au clavier.



## 5-6-13 Format de sortie

### • Commande SET

Une commande SET est utilisée pour désigner le format de l'affichage de sortie. Elle désigne les nombres de chiffres pour les parties entière et décimale.

**Format:** SET E *n* ..... désignation du nombre de chiffres pour la partie entière  
SET F *n* ..... désignation du nombre de chiffres pour la partie décimale  
SET N ..... annulation de désignation  
(*n* = 0 à 9)

- Si SET E 0 est utilisé comme désignation du nombre de chiffres pour la partie entière, le nombre de chiffres est 8.  
Cette commande peut être faite manuellement ou par écriture dans un programme.  
Pour le contenu de l'affichage, voir page 25.

## 5-6-14 Fonctions de caractères

### • LEN

La fonction LEN est utilisée pour compter le nombre de caractères dans une variable à caractères. Elle permet de connaître la taille d'une variable à caractères.

**Format:** LEN (variable à caractères)

#### Exemple:

Si A\$ = "ABCDE", LEN (A\$) = 5.

### • MID

La fonction MID n'est utilisée qu'avec la variable à caractères exclusive (\$). Elle extrait un certain nombre de caractères de la chaîne de caractères de cette variable.

**Format:** MID (*m* [, *n*]) *m* et *n* sont des expressions numériques et doivent être entre 1 et 30.  
(Les articles entre [ ] peuvent être omis.)

Cette commande signifie extraire *n* caractères à partir du *m*<sup>ième</sup> caractère de la chaîne de caractères stockée dans l'exclusive variable à caractères (\$).

L'expression numérique *m* ne doit pas être supérieure au nombre de caractères stockés. De plus, *m* + *n* ne doit pas être supérieur au nombre de caractères stockés + 1.

Qui plus est, quand l'expression numérique *n* est omise, tous les caractères à partir du *m*<sup>ième</sup> caractères sont extraits.







#### Exemple:

Si \$ = "PB-100"

MID (2,3) = "B-1" et MID (4) = "100"



### Exemple:

-   CLEAR     ..... effacement d'un seul programme
-   CLEAR A     ..... effacement de tous les programmes

## 5-6-17 Spécifications d'option

### ■ Magnétophone

Pour enregistrer des programmes ou données sur une cassette en utilisant cet appareil, utiliser l'interface FA-3 pour cassette et enregistrer avec un magnétophone ordinaire. De plus, si le magnétophone est muni d'une borne de télécommande, la commande à distance peut être commodément exécutée à partir du PB-100 via le FA-3.

Pour les procédures de branchement du magnétophone et les procédures d'utilisation détaillées, voir le mode d'emploi du FA-3.

### ● Enregistrement de programme

**Format:** SAVE ["nom de fichier"] (L'article entre crochets peut être omis.)

Le nom de fichier peut être composé d'un maximum de 8 caractères entre guillemets et peut contenir des lettres, chiffres et symboles.

### Exemple:

"ABC"

"NO.1"

Cette commande démarre le magnétophone en position d'enregistrement.

**Opération:** SAVE ["nom de fichier"] 

Une commande SAVE ne peut être utilisée que manuellement.

### ● Appel de programme

**Format:** LOAD ["nom de fichier"] (L'article entre crochets peut être omis.)

Cette commande démarre le magnétophone en position de lecture.

**Opération:** LOAD ["nom de fichier"] 

Affichage pendant le chargement:

PF:ABC

Nom de fichier    Fichier-programme

Même si un programme a déjà été écrit dans la zone de programme désignée avant l'appel, l'ancien programme est effacé à partir de la première ligne du programme à charger et le nouveau chargement se fait correctement.

### ● Enregistrement de tous les programmes

**Format:** SAVE A ["nom de fichier"] (L'article entre crochets peut être omis.)

Cette commande enregistre simultanément tous les programmes qui sont écrits dans les zones de programme P0 à P9.

Le mode opératoire est identique à celui de la commande SAVE et le magnétophone est démarré en position d'enregistrement.

#### Opération:


SAVE A ["nom de fichier"] 

### ● Appel de tous les programmes

**Format:** LOAD A ["nom de fichier"] (L'article entre crochets peut être omis.)

Cette commande appelle simultanément les programmes de toutes les zones de programmes qui ont été enregistrés en utilisant la commande SAVE A. Le mode opératoire est identique à celui de la commande LOAD et le magnétophone est démarré en position de lecture.

#### Opération:

LOAD A ["nom de fichier"] 

Affichage pendant le chargement de programme

AF:PB-100

Nom de fichier    Tous les fichiers

Même si des programmes sont déjà écrits dans les zones de programme avant l'appel, les anciens programmes sont effacés puis les nouveaux programmes sont appelés.

De plus, les commandes SAVE A et LOAD A ne peuvent être utilisées que manuellement.

### ● Enregistrement de données

**Format:** PUT ["nom de fichier"] variable 1, variable 2  
(L'article entre crochets peut être omis.)

Les données qui sont enregistrées sur la bande magnétique sont les données des variables 1 à 2.

**Exemple:** PUT "PB" A ..... données de la variable A


PUT "1-2" A, Z ..... données des variables A à Z

PUT "DT" \$, A, Z (10) ..... données de la variable à caractères \$ et des variables A à Z (10)

Quand on enregistre les données de la variable à caractères exclusive \$, écrire \$ en premier. Cette commande peut être utilisée soit manuellement soit par écriture dans un programme.

Pour l'utilisation manuelle, démarrer le magnétophone en position d'enregistrement.

#### Opération:

PUT ["nom de fichier"] variable 1, variable 2 

Pour utilisation par écriture dans un programme, écrire la commande PUT ainsi que le numéro de ligne puis démarrer le programme écrit.

## ● Appel de données

**Format:** GET ["nom de fichier"] variable 1 [, variable 2 ]  
(L'article entre crochets peut être omis.)

Cette commande peut être utilisée soit manuellement soit par écriture dans un programme.

Pour l'utilisation manuelle, mettre le magnétophone en position de lecture puis procéder comme suit.

GET ["nom de fichier"] variable 1 [, variable 2] **EXE**

Pour utilisation dans un programme, écrire la commande sur une ligne puis démarrer le programme.

## ● Contrôle du fichier qui a été enregistré sur la bande magnétique

Une commande VER est utilisée pour contrôler si les programmes ou données ont été enregistrés correctement.

**Format:** VER ["nom de fichier"] (L'article entre crochets peut être omis.)

Le mode opératoire est identique à celui du chargement de programme.

## ■ Imprimante

Une mini-imprimante spéciale peut être branchée au PB-100. En branchant cette imprimante, des listages de programme ou de données peuvent être obtenus. De plus, les résultats de calcul pendant l'exécution peuvent être imprimés.

Pour les procédures de branchement et d'utilisation, voir le mode d'emploi de la mini-imprimante.

Pour imprimer un listage de programme, appuyer sur **MODE** **F7** pour désigner le mode PRT.

### Listage de programme

**MODE** **F7** **MODE** **F7**

LIST **EXE** ou LIST A **EXE**

**MODE** **F8** (Pour quitter le mode PRT)

Une fois que l'impression est terminée, ne pas oublier d'appuyer sur **MODE** **F8** pour quitter le mode PRT.

De plus, pour imprimer des résultats de calcul ou des contenus d'opération, l'impression peut être faite automatiquement en écrivant "MODE 7" et "MODE 8" dans le programme. (Composer le mot MODE au clavier)

### Exemple:

```
{  
100 MODE 7  
110 PRINT A  
120 MODE 8
```

Quand "MODE 7" est écrit dans le programme, ne pas oublier d'écrire "MODE 8" avant la fin du programme pour pouvoir quitter le mode PRT.

## Liste de messages d'erreur

Code d'erreur	Signification	Cause	Mesure corrective
1	Débordement mémoire ou débordement d'emboîtement système.	<ul style="list-style-type: none"> <li>Le programme ne peut pas être écrit du fait d'un insuffisant nombre de pas ou du fait que la mémoire ne peut pas être étendue.</li> <li>Débordement d'emboîtement dû à une formule de calcul compliquée.</li> </ul>	<ul style="list-style-type: none"> <li>Effacer les programmes inutiles ou diminuer le nombre de mémoires.</li> <li>Diviser et simplifier l'expression numérique.</li> </ul>
2	Erreur de syntaxe	<ul style="list-style-type: none"> <li>Une erreur a été faite dans l'écriture du programme, etc.</li> <li>Le format du côté gauche est différent de celui du côté droit dans une instruction de substitution, etc.</li> </ul>	<ul style="list-style-type: none"> <li>Corriger l'erreur dans le programme entré, etc.</li> </ul>
3	Erreur mathématique	<ul style="list-style-type: none"> <li>Le résultat du calcul d'une expression numérique est de <math>10^{100}</math> ou plus.</li> <li>Hors de la plage d'entrée d'une fonction numérique.</li> <li>Le résultat est indéfini ou impossible.</li> </ul>	<ul style="list-style-type: none"> <li>Corriger la formule ou les données du calcul.</li> <li>Vérifier les données.</li> </ul>
4	Numéro de ligne non défini	<ul style="list-style-type: none"> <li>Numéro de ligne non désigné pour une instruction GOTO ou une instruction GOSUB.</li> </ul>	<ul style="list-style-type: none"> <li>Corriger en désignant le numéro de ligne.</li> </ul>
5	Erreur d'argument	<ul style="list-style-type: none"> <li>Pour une commande ou fonction nécessitant un argument, l'argument est hors de la plage d'entrée.</li> </ul>	<ul style="list-style-type: none"> <li>Corriger l'erreur d'argument.</li> </ul>
6	Erreur de variable	<ul style="list-style-type: none"> <li>On a essayé d'utiliser une mémoire qui n'a pas été étendue.</li> <li>On a essayé d'utiliser la même mémoire pour une variable numérique et une variable à caractères en même temps.</li> </ul>	<ul style="list-style-type: none"> <li>Étendre la mémoire correctement.</li> <li>Ne pas utiliser la même mémoire pour une variable numérique et une variable à caractères en même temps.</li> </ul>
7	Erreur d'emboîtement	<ul style="list-style-type: none"> <li>Une instruction RETURN apparaît ailleurs que dans un sous-programme.</li> <li>Une instruction NEXT apparaît ailleurs que dans une boucle FOR ou la variable de l'instruction NEXT est différente de celle de l'instruction FOR.</li> <li>L'emboîtement de sous-programmes dépasse 8 niveaux.</li> <li>L'emboîtement de boucle FOR dépasse 4 niveaux.</li> </ul>	<ul style="list-style-type: none"> <li>Enlever l'instruction RETURN ou l'instruction NEXT inutile.</li> <li>Réduire les sous-programmes ou les boucles FOR-NEXT au nombre maximal de niveaux.</li> </ul>
9	Erreur d'option	<ul style="list-style-type: none"> <li>L'exécution est faite en mode PRT ou une commande d'option telle que SAVE est exécutée alors qu'il n'y a pas d'imprimante ou de magnétophone branché.</li> </ul>	<ul style="list-style-type: none"> <li>Brancher une imprimante ou un magnétophone.</li> <li>Quitter le mode PRT.</li> </ul>

Liste de commandes de programme

Classification	Nom de la commande	Format	Fonction
Instructions d'entrée	INPUT	INPUT chaîne de variables	Entraîne l'entrée des données à partir du clavier pendant l'exécution d'un programme. L'exécution du programme est arrêtée jusqu'à la fin de l'entrée. <b>P.44</b>
	KEY	Variable à caractères = KEY	Lit un caractère entré pendant l'exécution d'un programme et l'affecte à une variable à caractères. Etant donné que le programme n'est pas arrêté par cette commande, rien n'est affecté à la variable à caractères si aucune entrée n'est faite par le clavier. <b>P.45</b>
Instructions de sortie	PRINT	PRINT fonction de contrôle de sortie $\left\{ \begin{array}{l} ; \\ ; \end{array} \right\}$ élément de sortie $\left[ \left\{ \begin{array}{l} ; \\ ; \end{array} \right\} \right]$	Sort l'élément de sortie désigné. <b>P.45</b>
	CSR	CSR $n \left\{ \begin{array}{l} ; \\ ; \end{array} \right\}$ $0 \leq n \leq 11$	Affiche à partir de la $n$ ème position désignée. <b>P.46</b>
Branchements	GOTO	GOTO $\left\{ \begin{array}{l} \text{numéro de ligne} \\ \text{variable} \end{array} \right\}$	Entraîne le saut au numéro de ligne désigné. <b>P.47</b>
	IF...{ THEN }	IF expression de comparaison { THEN numéro de ligne } ; commande	Entraîne le saut au numéro de ligne suivant THEN, ou exécute la commande suivante " ; ", si le résultat de la comparaison est vrai. Entraîne le passage au numéro de ligne suivant si le résultat de la comparaison est faux. <b>P.48</b>
	GOSUB	GOSUB $\left\{ \begin{array}{l} \text{numéro de ligne} \\ \text{variable} \end{array} \right\}$	Appelle le sous-programme du numéro de ligne désigné pour exécution. Une fois que le sous-programme est exécuté, la commande revient à l'instruction GOSUB par l'instruction RETURN pour passer à la commande suivant cette instruction. <b>P.51</b>
	RETURN	RETURN	Signifie la fin d'un sous-programme; renvoie la commande à la ligne suivant l'instruction GOSUB. <b>P.51</b>
Boucle	FOR	FOR $v=e_1$ TO $e_2$ [STEP $e_3$ ]  * $v$ signifie une variable numérique, et $e_1$ , $e_2$ et $e_3$ représentent respectivement une expression numérique.	Déclare le début d'une boucle dans laquelle la valeur numérique $v$ passe de la valeur initiale $e_1$ à la valeur finale $e_2$ par incréments de $e_3$ . La boucle est répétée $\left[ \frac{e_2 - e_1}{e_3} + 1 \right]$ fois entre les instructions FOR et NEXT. Si l'incrément $e_3$ est omis, il est considéré égal à "1". <b>P.49</b>

Classification	Nom de la commande	Format	Fonction
Boucle	NEXT	NEXT V	Signifie la fin d'une boucle FOR. Si le résultat de $v$ plus $e_2$ est égal ou inférieur à $e_1$ , la boucle est encore répétée. Si ce résultat est supérieur à $e_1$ , la commande passe à la ligne suivant l'instruction NEXT. <b>P.49</b>
Arrêt d'exécution	STOP	STOP	Arrête temporairement l'exécution d'un programme pour amener le système dans un état d'attente d'entrée par le clavier. L'exécution peut être reprise par appui sur la touche <b>EXE</b> . <b>P.54</b>
Fin d'exécution	END	END	Signifie la fin d'un programme, le système revenant à son état de pré-exécution. L'exécution d'un programme, une fois terminée, ne peut pas être reprise par appui sur la touche <b>EXE</b> . <b>P.54</b>
Effacement de données	VAC	VAC	Efface toutes les variables (mémoires). <b>P.57</b>
Listage de programme	LIST	LIST [numéro de ligne]	Affiche en descendant toutes les instructions d'un programme à partir du numéro de ligne désigné. <b>P.55</b>
Listage de tous les programmes	LIST A	LIST A	Affiche un listage de tous les programmes. <b>P.55</b>
Exécution de programme	RUN	RUN [numéro de ligne]	Entraîne le démarrage d'un programme à partir du numéro de ligne désigné. <b>P.54</b>
Effacement de programme	CLEAR	CLEAR	Efface la zone de programme désignée d'un programme. <b>P.57</b>
	CLEAR A	CLEAR A	Efface tous les programmes. <b>P.57</b>
Désignation d'unité angulaire	MODE	MODE $\begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}$	Fixe l'unité angulaire en degrés (4), radians (5) ou grades (6). <b>P.55</b>
Désignation de format	SET	SET $\begin{pmatrix} E# \\ F# \\ N \end{pmatrix}$ $(0 \leq n \leq 9)$	Désigne le nombre de positions pour la partie entière et le nombre de positions pour la partie décimale pour la valeur numérique affichée. <b>P.56</b>
Fonctions de caractères	LEN	LEN (variable à caractères)	Calcule la taille de la variable à caractères. <b>P.56</b>
	MID	MID (m [, n])	Extrait n caractères à partir du même caractère de la variable à caractères exclusive (\$). <b>P.56</b>
	VAL	VAL (variable à caractères)	Convertit les chiffres d'une variable à caractères en une valeur numérique. <b>P.57</b>



Classification	Nom de la commande	Format	Fonction
Utilisation d'options	SAVE	SAVE ["nom de fichier"]	N'enregistre, sur une bande magnétique, que le programme de la zone de programme désignée. <b>P.58</b>
	LOAD	LOAD ["nom de fichier"]	Rappelle le programme de la bande magnétique et le charge dans la zone de programme désignée. <b>P.58</b>
	SAVE A	SAVE A ["nom de fichier"]	Enregistre, sur une bande magnétique, les programmes de toutes les zones de programme en même temps. <b>P.59</b>
	LOAD A	LOAD A ["nom de fichier"]	Rappelle tous les programmes de la bande magnétique et les charge dans les zones de programme respectives. <b>P.59</b>
	PUT	PUT ["nom de fichier"] variable	Enregistre les données de la variable sur une bande magnétique. <b>P.59</b>
	GET	GET ["nom de fichier"] variable	Appelle les données de la bande magnétique et les charge dans la variable. <b>P.60</b>
	VER	VER ["nom de fichier"]	Contrôle si les programmes ou données ont été correctement enregistrés sur la bande magnétique. <b>P.60</b>

\* Les articles entre [ ] peuvent être omis.  
Chacun des articles entre { | } peut être utilisé.

### Plage des fonctions

	Plage d'entrée	Précision du résultat
$\sin x, \cos x, \tan x$	$ x  < 1440^\circ$ (8 $\pi$ Bogenmaß, 1600 Neugrad)	$\pm 1$ au 10e chiffre
$\sin^{-1} x, \cos^{-1} x$	$ x  \leq 1$	"
$\tan^{-1} x$	"	"
$\log x, \ln x$	$x > 0$	"
$e^x$	$x = 1$	"
$\sqrt{x}$	$x \geq 0$	"
$x^y (x \neq y)$	$x > 0$	"

# Caractéristiques

## ■ Type

PB-100

## ■ Fonctions de calcul élémentaires

Nombres négatifs, exposants, opérations arithmétiques y-compris parenthèses (addition, soustraction, multiplication et division), avec fonction de jugement d'ordre de priorité (vraie logique algébrique)

## ■ Fonctions incorporées

Fonctions trigonométriques/trigonométriques inverses (mesures angulaires en degrés, radians ou grades), fonctions logarithmiques et exponentielles, racine carrée, puissances, réduction à l'entier, suppression de l'entier, valeur absolue, symbolisation, désignation du nombre de chiffres significatifs, désignation du nombre de décimales, nombres aléatoires,  $\pi$ .

## ■ Commandes

INPUT, PRINT, GOTO, FOR-NEXT, IF-THEN, GOSUB, RETURN, STOP, END, RUN, LIST, LIST A, MODE, SET, VAC, CLEAR, CLEAR A, DEFM, SAVE, SAVE A, LOAD, LOAD A, PUT, GET, VER.

## ■ Fonctions de programme

KEY, CSR, LEN, MID, VAL.

## ■ Plage de calcul

$\pm 1 \times 10^{-99}$  à  $\pm 9,999999999 \times 10^{99}$  et 0 (Calculs internes à mantisse à 12 chiffres)

## ■ Système de programmation

Système à stockage

## ■ Langage de programmation

BASIC

## ■ Nombre de pas

Maximum 544 (maximum 1.568 pas quand le module RAM optionnel est utilisé)

## ■ Capacité de programmation

Maximum 10 programmes (P0 à P9)

## ■ Nombre de mémoires

26 en standard, étendable à 94 (maximum 222 mémoires quand le module RAM optionnel est utilisé) et variable à caractères exclusive (\$).

## ■ Nombre de piles

Sous-programmes — 8 niveaux  
Boucle FOR-NEXT — 4 niveaux  
Valeur numérique — 6 niveaux  
Éléments de calcul — 12 niveaux

## ■ Système et méthode d'affichage

Nombre à 10 chiffres (y-compris le signe moins) ou mantisse à 8 chiffres (7 chiffres dans le cas d'un nombre négatif) et 2 chiffres pour l'exposant. De plus, affichage d'état EXT, [5], RUN, WRT, DEG, RAD, GRA, TR, PRT, STOP.

■ **Éléments d'affichage**

Affichage à matrice par points sur 12 positions (cristaux liquides)

■ **Principaux composants**

C-MOS VLSI et autres

■ **Alimentation**

2 piles au lithium (CR2032)

■ **Consommation**

Maximum 0,02 W

■ **Longévité des piles**

Unité centrale seulement – environ 360 heures (utilisation continue)

■ **Coupure automatique de l'alimentation**

L'alimentation est coupée automatiquement environ 7 minutes après la dernière opération.

■ **Plage de température ambiante**

0°C à 40°C (32°F à 104°F)

■ **Dimensions**

9,8 mmH x 165 mmL x 71 mmP

■ **Poids**

116g, piles comprises

**CASIO®**