Ordinateur Personnel

FX-750P

MANUEL DE L'UTILISATEUR



CASIO

Ordinateur Personnel

FX-750P

MANUEL DE L'UTILISATEUR

- Le contenu de ce manuel est susceptible d'être modifié sans avis préalable.
- Toute reproduction, même partielle, faite sans autorisation préalable, est illicite. Sachez que l'utilisation de ce manuel à des fins autres que personnelles et sans l'autorisation de CASIO est interdite par la loi sur le copyright.
- CASIO Computer Co., Ltd. ne pourra être tenue pour responsable des dommages ou pertes survenus par suite de l'utilisation de ce manuel.

CASIO®

Avant-propos

Nous vous remercions beaucoup d'avoir acheté un FX-750P Casio.
Tous les ordinateurs portatifs ordinaires sont en général équipés d'une unité de mémoire dans laquelle des données et des programmes peuvent être introduits, retirés ou effacés. Le FX-750P a, toutefois, une caractéristique particulière: c'est une carte RAM qui lui sert d'unité de mémoire. On peut insérer cette carte RAM dans l'unité principale ou l'en retirer quand on veut, et donc remplacer ou sauvegarder des données beaucoup plus facilement qu'avec la bande cassette conventionnelle. La carte RAM est également équipée d'une pile de sécurité afin de préserver des données et des programmes enregistrés. Etant donne que le FX-750P peut recevoir deux cartes RAM, des fichiers d'adresses, des listes de clients et des rapports peuvent être rapidement établis sur place, pourvu que les données et les programmes aient été préparés à l'avance par unités de cartes.
Un autre avantage du FX-750P réside dans ses nombreux dispositifs fonctionnels. Grâce à ses multiples fonctions, comme les fonctions numériques, statistiques et autres, il eput en effet accomplir des calculs scientifiques, techniques et statistiques complèxes en une seule opération.
Ce manuel a été conçue pour vous aider à comprendre facilement les diverses fonctions du FX-750P et à les utiliser pleinement dans vos études commerciales, scientifiques ou techniques.
Utilisez le comme un guide afin que le FX-750P puisse vous servir en toute confiance pendant de nombreuses années.

Chapitre 1	Installation des piles, insertion et enlèvement de la carte RAM et précautions d'utilisation.
Chapitre 2	Utilisation du FX-750P comme calculatrice scientifique sophistiquée.
Chapitre 3	Essentiel de la programmation en BASIC.
Chapitre 4	Explication des commandes de BASIC accompagnée d'exemples concrets.
Chapitre 5	Indication des commandes.
Chapitre 6	Bibliothèque pratique pour un usage immédiat.
Chapitre 7	Liste de fonctions pouvant servir de guide.

Avant utilisation

Ce FX-750P a été soumis à des opérations d'inspection rigoureuses, basées sur la technique électronique de haut niveau de Casio et l'exigence de ses contrôles de qualité. Pour que vous puissiez profiter pendant de nombreuses années de cet ordinateur, nous vous prions de considérer les point suivants.

• Précautions d'utilisation

- L'unité centrale et la carte RAM sont composés d'éléments électroniques de haute précision: ne jamais tenter de les démonter. Préservez cet ordinateur des chocs, évitez de le jeter ou de le laisser tomber; préservez le aussi de l'électricité statique et des brusques écarts de température. Par ailleurs, lorsque la température est basse, il peut arriver que le temps nécessaire à l'affichage soit plus long ou que l'affichage reste vide; mais tout cela redeviendra normal dès que la température aura été rétablie.
- Le raccord de l'ordinateur est exclusivement prévu pour l'imprimante de caractères FA-20 à interface de cassette; aucun autre appareil ne devra donc être branché à ce raccord. Si vous utilisez l'ordinateur de façon indépendante, n'oubliez pas d'obturer cette borne, cela vous évitera un contact accidentel.
- Ne touchez jamais les contacts de la carte RAM. Lorsque la carte RAM n'est pas dans l'ordinateur, recouvrez les contacts avec les languettes situées de part et d'autre de la carte vous pouvez alors glisser la carte dans son boîtier et la ranger.
- Etant donné que cet ordinateur n'est pas équipé d'une RAM incorporée, ne l'utilisez pas sans avoir introduit au préalable une carte RAM dans la fente prévue à cet usage.
- Lorsque vous remplacez les piles, commencez toujours par enlever les cartes RAM afin que leurs contenus ne risquent pas d'être modifiés.
- Même si le commutateur d'alimentation est sur la position marche (ON), l'alimentation électrique ne se fera pas tant que le verrouillage de la carte RAM ne sera pas bloqué. Lorsque vous utilisez le FX-750P, le verrouillage doit toujours être sur la position LOCK (verrouillé).

- Les contenus de la mémoire peuvent être modifiés ou le fonctionnement des touches peut être empêché si l'ordinateur ou la carte RAM sont soumis à une forte électricité statique. Si cela se produisait, enlevez momentanément les piles et remettez les ensuite.
- Lorsque vous raccordez l'ordinateur à un dispositif optionnel, n'oubliez jamais de mettre le commutateur d'alimentation de l'ordinateur sur arrêt (OFF).
- Qu'on ait ou non utilisé l'ordinateur, les piles doivent être changées tous les deux ans. Ne laissez jamais des piles usagées dans l'ordinateur, car il pourrait se produire une fuite du liquide de ces piles et cela pourrait l'endommager.
- Evitez de nettoyer l'ordinateur avec des liquides volatiles comme les solvants ou le benzène. Essuyez le avec un chiffon sec et doux ou avec un chiffon imbibé d'un détergent neutre.
- Ne mettez pas le commutateur d'alimentation sur arrêt au cours de l'exécution d'un programme ou d'un calcul.
- L'ordinateur est constitué de pièces électroniques de précision. Evitez donc les chocs pendant l'exécution d'un programme car ils peuvent arrêter le programme ou modifier le contenu de la mémoire.
- En cas de mauvais fonctionnement, prendre contact avec le magasin qui vous a vendu l'ordinateur ou avec le concessionnaire le plus proche.
- Avant de faire appel au service après-vente, relisez attentivement ce manuel et vérifiez les erreurs d'alimentation ou de programme.

TABLE DES MATIERES

CHAPITRE 1 UNITE PRINCIPALE + CARTE RAM = FX-750P 1

1-1 Alimentation électrique et remplacement des piles 2
1-2 RAM CARD (Carte RAM) 4
1. Caractéristiques de la carte RAM 4
2. Méthode d'utilisation de la carte RAM et précautions à prendre 5
3. Les contenus d'une carte RAM (affectation de mémoire interne) 12
4. Types de cartes RAM et nombre d'octets 14
5. Méthode d'utilisation selon les combinaisons de cartes RAM 15
6. Zone de système et zone de l'utilisateur lorsqu'on utilise deux cartes RAM 18
7. Messages d'erreur consécutifs à l'insertion ou à l'enlèvement d'une carte RAM 21
1-3 Utilisation de votre FX-750P 23
L'affichage 24
Fonction des touches 26
1-4 Option 31
La section imprimante 32
Section liaison cassette 32
Stockage et chargement de données (PUT, GET) 35

CHAPITRE 2 FONCTIONS DE CALCUL 37

2-1 Avant d'effectuer les calculs 38
2-2 Calculs Manuels et corrections d'entrées 42
2-3 Fonctions numériques 46
2-4 Fonctions de caractères 53

2-5 Calculs statistiques 55
2-6 Constantes scientifiques 61

CHAPITRE 3 REFERENCE DE "BASIQUE" 65

3-1 A propos des programmes 66
3-2 L'organigramme 67
3-3 La programmation 71
3-4 Les variables 74
3-5 Introduction et exécution d'un programme 79
3-6 Modification de programme 83
3-7 Précautions à respecter pour les programmes pratiques du chapitre 4 87

CHAPITRE 4 APPLICATIONS 91

4-1 Conversion des devises étrangères 92
4-2 Calculs de circuits électriques (Calcul de l'impédance) 106
4-3 Analyse comparative 120
4-4 Analyse de régression (régression linéaire) 129
4-5 Gestion du budget familial 136
4-6 Emploi du temps hebdomadaire 153
4-7 Contrôle $\overline{X}$-R 171
4-8 Analyse ABC 175
4-9 Calculs chimiques (calcul de l'atténuation de la radiation) 183
4-10 Jeu: devinez les chiffres 197

CHAPITRE 5 DETAIL DES COMMANDES ET FONCTIONS 205

5-1 Commandes manuelles 207
CONT 207
DELETE 209
EDIT 210
LIST, LLIST 211
LOAD [ALL], LOAD, A [, M] 212
NEW [ALL] 215
PASS 216
PROG 217
RUN 217
SAVE [ALL], SAVE, A 218
STAT LIST, STAT LLIST 219
SYSTEM 220
VERIFY 221
5-2 Commandes programées 222
ANGLE 222
BEEP 223
CHAIN 224
CLEAR 225
CLS 226
DIM 227
END 229
ERASE 229
FOR ~ TO ~ STEP/NEXT 230
GET 232
GOSUB/RETURN 233
GOTO 234

IF ~ THEN ~ ELSE 235
INPUT 236
LET 237
LOCATE 238
ON ~ GOSUB 239
ON ~ GOTO 240
PRINT, LPRINT 241
PRINT ON, PRINT OFF 243
PUT 244
READ, RESTORE, DATA 245
REM 247
STAT 248
STAT CLEAR 249
STOP 249
TRON, TROFF 251
WAIT 252
5-3 Fonctions de caractères 253
ASC 253
CHR$ 253
VAL 254
STR$ 255
DMS$ 256
HEX$ 257
LEFT$ 258
RIGHT$ 259
MID$ 260
LEN 261
INKEY$ 261
5-4 Fonctions de contrôle 262

TAB 262
USING 263
5-5 Fonctions numériques 265
SIN, COS, TAN 265
ASN, ACS, ATN 266
HYPSIN, HYPCOS, HYPTAN 267
HYPASN, HYPACS, HYPATN 268
SQR 268
LGT, LOG 269
EXP 269
ABS 270
INT 270
FRAC 271
SGN 271
ROUND 272
DEG 273
PI 274
RND 274
5-6 Fonctions statistiques 275
CNT 275
SUMX, SUMY 275
SUMXY 276
SUMX2, SUMY2 276
SDX, SDY 277
SDXN, SDYN 277
LRA 278
LRB 278
COR 279
EOX 279

EOY 280
5-7 Commandes diverses 281
&H 281

CHAPITRE 6 BIBLIOTHEQUE DE PROGRAMMES 283

6-1 Traitement d'exécution 284
6-2 Programme de tabulation 293
6-3 Aide memoire (memo card) 301
6-4 Contrôle de la méthode d'appel et de l'efficacité d'un démarchage par correspondance 306
6-5 Jeu: le monstre de la route 310

CHAPITRE 7 LISTE DES FONCTIONS ET NORMES 313

7-1 FONCTIONS NUMERIQUES 314
7-2 FONCTIONS STATISTIQUES 316
7-3 FONCTIONS DE CARACTERES 317
7-4 CONSTANTES SCIENTIFIQUES 318
7-5 COMMANDES MANUELLES 319
7-6 COMMANDES DE PROGRAMME 321
7-7 FONCTIONS DE CONTRÔLE 323
7-8 MESSAGES D'ERREUR 324
7-9 TABLEAU DES CODES DE CARACTERES 328
7-10 LISTE DES MOTS RESERVES 329
7-11 LISTE DES SYMBOLES DES ORGANIGRAMMES 330
CARACTERISTIQUES 332
INDEX 335

CHAPITRE 1

UNITE PRINCIPALE + CARTE RAM = FX-750P

On trouvera dans ce chapitre des instructions pour le changement des piles de l'unité centrale et de la carte RAM (mémoire vive), pour l'insertion et le retrait de la carte RAM ainsi que les précautions à respecter pendant l'utilisation, la configuration de l'unité principale, etc.
Pour utiliser le FX-750P, il est important de bien comprendre toutes ces instructions.

1-1 Alimentation électrique et remplacement des piles

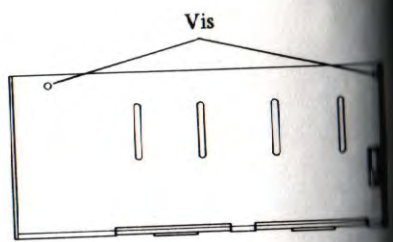
Deux piles au lithium (CR2032) assurent l'alimentation électrique du FX-750P. La durée de vie de ces piles sera de 120 heures si on utilise uniquement l'unité centrale, mais elle sera raccourcie si on se sert beaucoup de l'avertisseur sonore. Si la luminosité de l'affichage ne peut plus être améliorée à l'aide de la commande de contraste (voir page 25), cela veut dire que les piles sont usées et qu'il convient de les changer dans les meilleurs délais. Remplacez toujours les deux piles en même temps.

* Remplacez les piles une fois tous les deux ans, même si l'ordinateur n'a pas été utilisé, ceci afin d'éviter tout risque de fuite du liquide de pile.

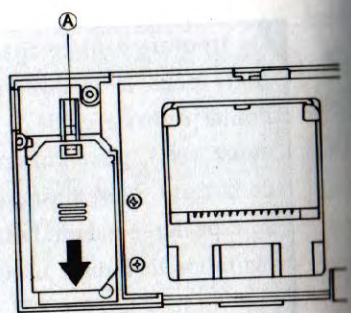
■ Remplacement des piles

Si les cartes RAM sont introduites dans l'ordinateur, enlevez-les avant de changer les piles. Une fois les piles changées, remettre chaque carte RAM dans la fente qu'elle occupait auparavant (voir page 7).

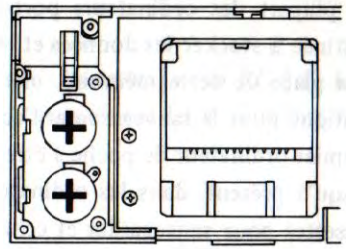
1) Après avoir mis le commutateur d'alimentation sur OFF, dévissez les deux vis de l'arrière et enlevez le couvercle arrière (utilisez un tournevis de bonne qualité et muni d'une pointe fine).



2) Faire glisser le couvercle de maintien de la pile dans le sens de la flèche tout en appuyant sur l'attache A comme indiqué sur le schéma.



3) Retirez les deux piles usagées (il suffit pour cela de tourner le compartiment des piles vers le bas et de tapoter légèrement).



4) Essuyez les nouvelles piles avec un chiffon sec et les introduire avec le côté positif (+) vers le haut. Assurez vous que le côté (+) de chacune des piles est bien tourné vers le haut.

5) Remettez en place et fermer la plaque de maintien des piles.

6) Remettez en place le couvercle arrière et serrez les deux vis.

7. **Ne jamais jeter des piles usagées dans le feu, car elles pourraient exploser.**

8. **Gardez les piles hors de portée des enfants. Si un enfant en avalait une, contactez immédiatement votre médecin.**

1-2 RAM CARD (carte RAM)

1. Caractéristiques de la carte RAM

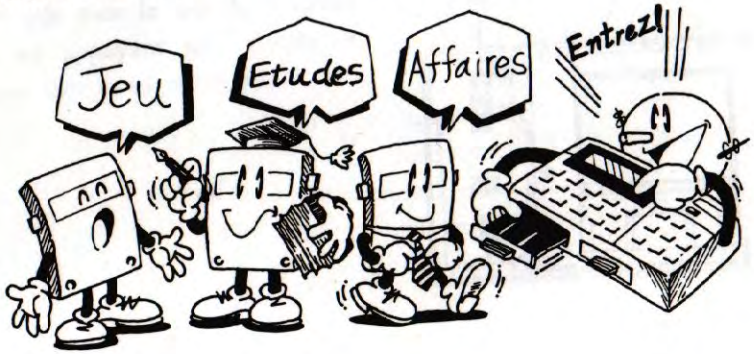
La plupart des ordinateurs portatifs sont équipés d'une mémoire incorporée destinée à stocker les données et les programmes; le FX-750P, quant à lui, utilise à la place de cette mémoire, une "carte RAM" facile à enlever. Extrêmement pratique pour la sauvegarde et l'échange des données et des programmes, il est le premier ordinateur de poche à être conçu de cette façon.

Jusqu'à présent, dans les ordinateurs portatifs conventionnels, on utilisait des cassettes pour sauvegarder et échanger des données et des programmes, mais la "carte RAM" a rendu caduque le travail avec des cassettes et a grandement contribué à la prolifération de l'usage des ordinateurs portatifs. Ce système est extrêmement commode, puisque les données et les programmes peuvent être facilement échangés et traités avec une simple unité de carte. De plus, une pile incorporée dans la carte RAM soutient les contenus de cette carte, si bien qu'ils ne seront pas perdus, même si la carte est enlevée de l'unité centrale.

Il existe deux types de cartes RAM qu'on peut se procurer: la RC-4 (4 kilo-octets) et la RC-2 (2 kilo-octets). Le FX-750P peut recevoir jusqu'à deux de ces cartes quelque soit leur type.

La gamme des possibilités peut être considérablement étendue si les données et programmes sont introduits dans les cartes RAM qui leur correspondent, car ces cartes pourront ensuite être chargées et traitées là et où ce sera nécessaire.

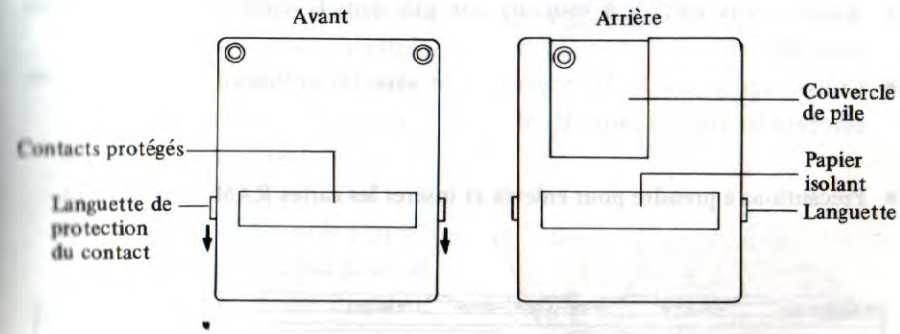
* Etant donné que cet ordinateur n'a pas de zone RAM incorporée, il ne peut pas être utilisé sans qu'une carte RAM ait été chargée.



2. Méthode d'utilisation de la carte RAM et précautions à prendre

■ Précautions d'utilisation

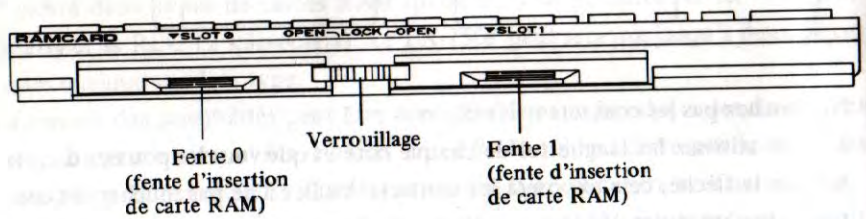
Il y a deux types de cartes RAM: la RC-4 (4 kilo-octets) et la RC-2 (2 kilo-octets) qui servent exclusivement au FX-750P, mais toutes deux s'utilisent de la même façon.



- Ne touchez pas les contacts
  Si vous saisissez les languettes de chaque côté et que vous les poussez dans le sens de la flèche, cela exposera les contacts. Veillez à ne pas toucher ces contacts avec un doigt ou avec un objet métallique, car cela pourrait rendre la carte RAM inutilisable. Quand la carte RAM n'est pas dans l'unité principale, les contacts doivent toujours être protégés.
- Ne soumettez pas la carte RAM à de fortes charges électrostatiques, car cela pourrait modifier des données stockées ou rendre impossible les opérations de touche. Si cela se produisait, enlevez et réinsérez la pile de la carte RAM (toutes les données enregistrées sont alors effacées).
- N'essayez jamais de tordre, de plier ou de démonter une carte RAM.
- Lorsque vous enlevez et rangez une carte RAM, placez la toujours dans le boîtier fourni pour la protéger de la poussière et rangez-la à l'abri des rayons directs du soleil.

- Les cartes RAM sont équipées d'une pile au lithium pour le soutien de la mémoire. Les données enregistrées seront perdues si la pile est enlevée, lorsqu'il devient nécessaire de remplacer la pile, il faut donc d'abord transférer les contenus stockés sur une cassette. Après avoir changé la pile, on pourra charger à nouveau ces contenus sur la carte RAM (voir page 33).
- N'enlevez jamais le papier isolant, car il sert à protéger les contacts dans le cas où la carte RAM serait introduite à l'envers.
- Assurez vous qu'il y a toujours une pile dans la carte RAM lorsque vous l'utilisez.
- Utilisez ces cartes RAM exclusivement avec les ordinateurs portatifs Casio compatibles avec les cartes RAM.

■ **Précautions à prendre pour enlever et insérer les cartes RAM**

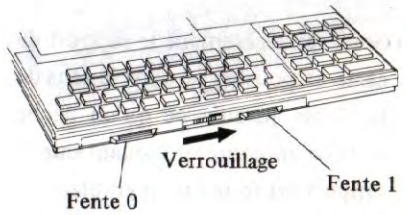


- On peut charger deux cartes RAM dans cet ordinateur, mais lorsqu'on utilise une seule carte RAM, il faut toujours l'insérer dans la fente 0. Faute de cela les opérations de touche ne seront pas possibles.

■ **Mise en place de la carte RAM dans l'ordinateur**

L'ordinateur ne peut pas recevoir plus de deux cartes quel que soit leur type. Cependant, si vous utilisez une seule carte RAM, elle devra toujours être mise dans la fente 0. La méthode d'insertion est la même pour les deux fentes.

1) Mettre la commutateur d'alimentation de l'ordinateur sur OFF.



2) Ecarter le verrouillage de la fente dans laquelle vous allez mettre la carte RAM (le commutateur d'alimentation de l'ordinateur sera toujours sur OFF).

Attention:

N'essayez pas de forcer le support de la carte sans avoir fait glisser le verrouillage, cela l'endommagerait.



3) Tirer la carte par son support tout en appuyant sur le bord du support.

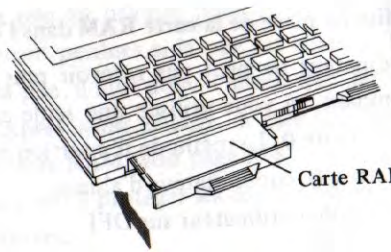
Attention:

Le support ne peut être retiré qu'en partie. N'essayez pas de le forcer, car cela l'endommagerait.

4) Tournez la carte RAM vers le haut (contacts vers le haut) et insérez la dans le support de carte, la protection de contact étant mise.

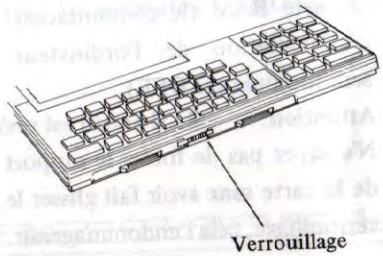


5) Appuyez doucement sur la lèvre du support de carte, de façon à ce que la carte RAM soit de niveau et enfoncez la complètement.



6) Soulevez légèrement le support de carte et poussez le dans le sens de la flèche jusqu'à ce qu'un déclic se fasse entendre, indiquant que le support est fermement en place.

7) Faire glisser le verrouillage sur la position centrale.



Attention:
Même si le commutateur d'alimentation est sur la position ON, l'alimentation ne pourra pas se faire tant que le verrouillage ne sera pas en position de bloquage. N'oubliez donc pas de refermer le verrouillage après avoir changé de carte RAM.

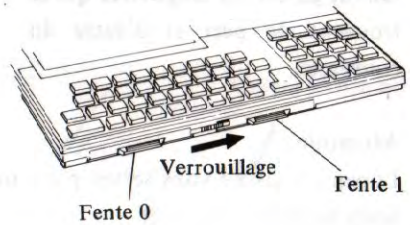
8) Pour pouvoir utiliser l'ordinateur, il suffit maintenant de mettre le commutateur d'alimentation sur ON.

* Si un message autre que "READY P0" est affiché, cela veut dire que la carte RAM n'est pas mise correctement. Consultez la page 21 et remettez la carte correctement.

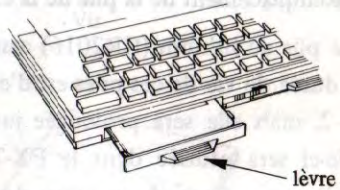
■ **Enlèvement de la carte RAM**
(Même méthode pour les fentes 0 et 1.)

1) Mettre le commutateur d'alimentation de l'ordinateur sur OFF.

2) Ecartez le verrouillage de la fente dont vous allez extraire la carte RAM afin de libérer le verrouillage.



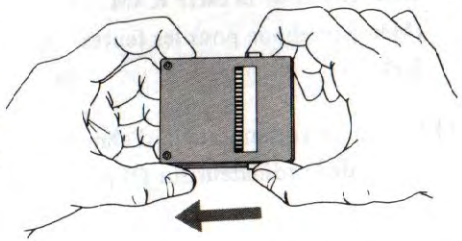
3) Retirer le support de carte tout en appuyant sur sa lèvre.



4) Tirez la carte RAM en la tenant par les deux côtés tout en appuyant légèrement sur la lèvre du support de carte. Veillez à ne pas toucher les contacts.



5) Les contacts de la carte RAM seront à ce moment là exposés, couvrez les immédiatement en faisant glisser les languettes qui se trouvent de part et d'autre du couvercle.



Attention:
Lorsque vous ne vous servez pas d'une carte RAM, rangez la toujours dans la boîte fournie à cet effet.

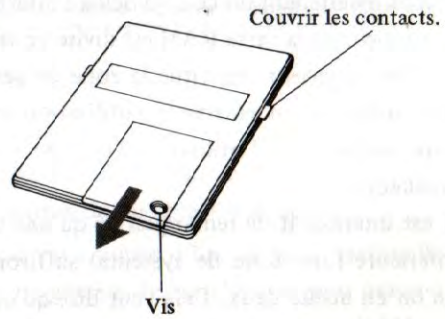
■ **Remplacement de la pile de la carte RAM (pile de soutien de la mémoire)**

Une pile au lithium (CR2016) assure l'alimentation électrique des cartes RAM. La durée de vie de ces piles est d'environ un an pour la RC-4 et deux ans pour la RC-2, mais elle sera prolongée jusqu'à environ deux ans pour la RC-4 lorsque celle-ci sera utilisée dans le FX-750P car elle sera soutenue par les piles de l'ordinateur. Cependant, étant donné qu'il y a un risque de fuite de liquide de pile au bout de deux années d'utilisation, ne manquez pas de changez les piles tous les deux ans au moins.

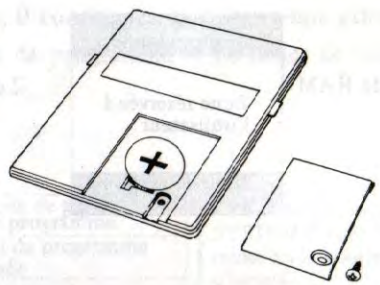
• **Remplacement de la pile de la carte RAM**

Si l'on touchait les contacts, cela provoquerait des problèmes; veiller donc à les garder couverts lors du remplacement d'une carte RAM.

1. Enlever le couvercle de pile en retirant la vis et en poussant le couvercle dans le sens de la flèche. Retirer ensuite la pile usée.



2. Essuyer la pile neuve avec un chiffon sec, l'insérer avec le côté plus (+) tourné vers le haut et remettre le couvercle. Lors de l'insertion d'une pile, vérifier toujours que la polarité est bien respectée.



• Ne jamais jeter une pile usée dans le feu. Ce serait extrèmement dangereux car elle pourrait exploser.

• **Garder les piles hors de portée des enfants.**
Si un enfant en avalait une, contactez immédiatement votre médecin.

3. Les contenus d'une carte RAM (affectation de mémoire interne)

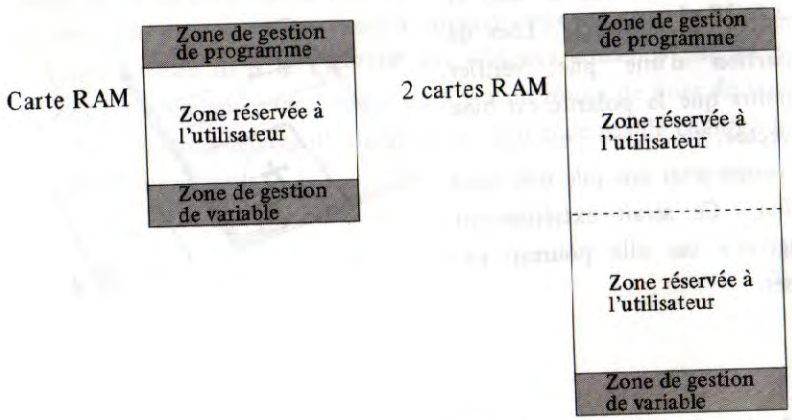
Etant donné que la configuration de la mémoire d'une carte influencera grandement sa capacité de stockage, il sera important, dès la conception d'un programme ou d'un format de données, de commencer par bien se rendre compte de la taille générale du programme et de la quantité des données.

Jetons maintenant un coup d'oeil à l'intérieur d'une carte RAM.

L'intérieur de la carte RAM est divisé en trois grands blocs.

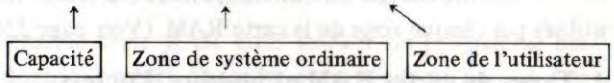
Le bloc supérieur constitue la zone de gestion de programme pour le BASIC, le bloc inférieur est réservé à l'utilisation avec variables et le bloc central est la zone consacrée à l'utilisateur (la zone où les programmes et les données sont stockées).

Il est intéressant de remarquer ici qu'une seule zone supérieure et une seule zone inférieure (une zone de système) suffiront, que l'on utilise une carte RAM ou qu'on en utilise deux. Cela veut dire qu'on obtiendra une zone plus grande pour l'utilisateur, du point de vue du nombre d'octets, en utilisant deux cartes.

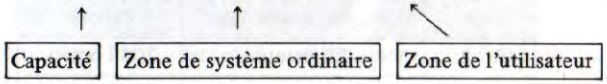


Le nombre d'octets disponibles dans la zone de l'utilisateur variera selon la manière dont on aura combiné les cartes RAM. Pour expliquer le nombre d'octets, c'est une carte RAM de 4 kilo-octets qui est utilisée dans l'exemple suivant.

Une carte de 4 kilo-octets: 4096 octets − 1296 octets = 2800 octets



Deux cartes de 4 kilo-octets: 8192 octets − 1296 octets = 6896 octets

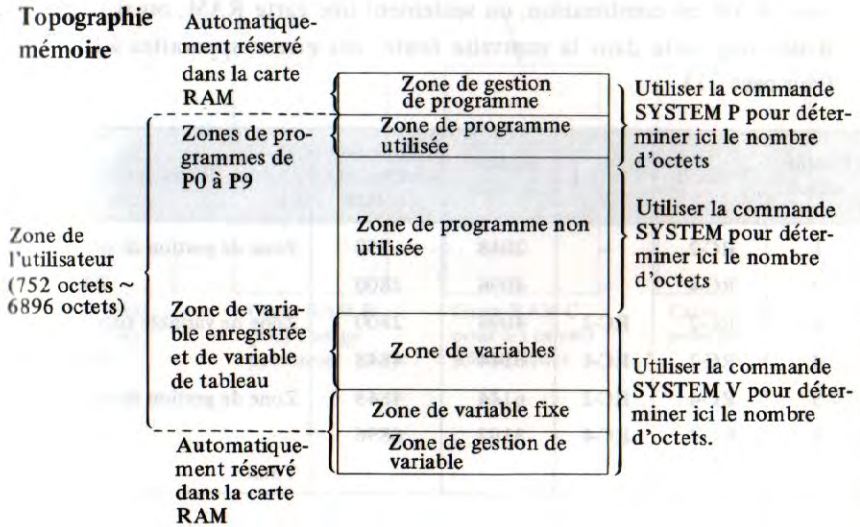


Zone de système ordinaire:

- Zone de variable fixe = 208 octets
- Zone de système = 1088 octets
  - Zone de gestion de programme = 632 octets
  - Zone de gestion de variable = 456 octets

Etant donné que la carte RAM elle-même est équipée d'une zone de gestion de programme indépendante et d'une zone de gestion de variable, on peut passer à l'exécution d'un programme différent en changeant simplement de carte RAM. Si la carte RAM insérée dans la fente 1 est utilisée pour des données et que cette carte est remplacée par plusieurs autres, il conviendra d'apporter une attention particulière aux combinaisons de zones de programme et de zones de donnée. (Voir page 19)

★ **Topographie mémoire**



Les commandes SYSTEM sont très pratiques pour déterminer le nombre d'octets utilisés par chaque zone de la carte RAM. (Voir page 220)

4. Types de cartes RAM et nombre d'octets

Type	Capacité de stockage	Octets
RC-2	2K octets	2048 octets
RC-4	4K octets	4096 octets

* Une carte RAM RC-4 (4 kilo-octets) vous sera fournie à l'achat de votre FX-750P.

Le FX-750P est équipé de deux fentes pour les cartes RAM, vous pouvez donc utiliser soit une RC-2 (2 kilo-octets), soit une RC-4 (4 kilo-octets), soit les deux combinées.

Nombre d'octets selon la combinaison des cartes RAM

- Lorsque vous utilisez deux cartes RAM avec le FX-750P, la manière de les combiner est facultative; mais une fois que des données et un programme ont été introduits selon une certaine combinaison, c'est celle-ci que vous devrez toujours utiliser ainsi que les deux mêmes fentes. Si vous utilisez une autre carte RAM en combinaison, ou seulement une carte RAM, ou si vous introduisez une carte dans la mauvaise fente, une erreur apparaîtra à l'affichage (voir page 21).

Combinaison	Fente 0	Fente 1	Capacité (octets)	Zone de l'utilisateur (octets)	Zone de système ordinaire
1	RC-2	–	2048	752	Zone de gestion de programme 632 octets
2	RC-4	–	4096	2800	
3	RC-2	RC-2	4096	2800	Zone de variable fixe 208 octets
4	RC-2	RC-4	6144	4848	
5	RC-4	RC-2	6144	4848	Zone de gestion de variable 456 octets
6	RC-4	RC-4	8192	6896	Total 1296 octets

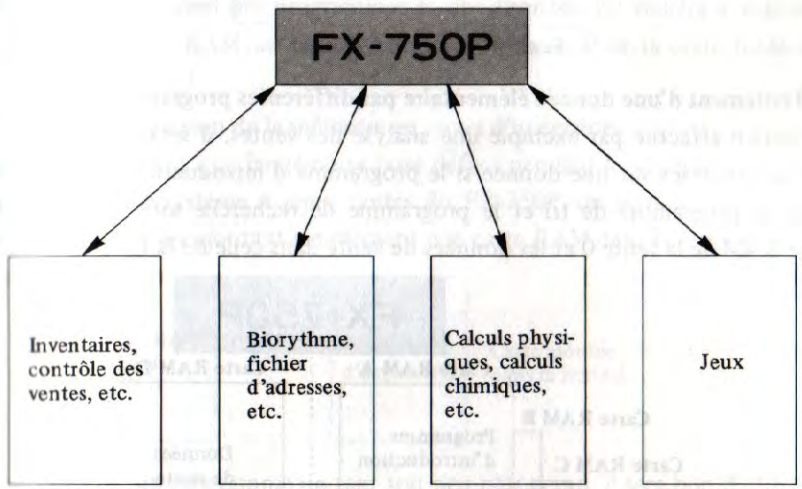
5. Méthode d'utilisation selon les combinaisons de cartes RAM

L'utilisation exclusive de chaque carte RAM pour stocker un programme particulier ou un groupe de données permet d'accéder au programme ou aux données désirés aussi facilement que si l'on recherchait une page dans la table des matières d'un livre.

Nous allons maintenant expliquer les différentes combinaisons possibles pour utiliser les cartes RAM.

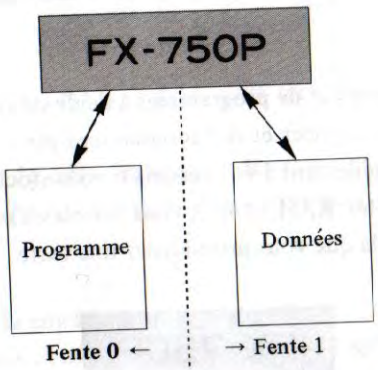
- **Classification de données et de programmes à l'aide de cartes RAM**

Lorsque vous créez des données et des programmes pour le FX-750P, l'ordinateur pourra répondre rapidement à vos besoins si vous stockez ces données et ces programmes sur des cartes RAM et que vous les classifiez par cartes avec des indices; il suffira pour cela que vous introduisiez une carte.



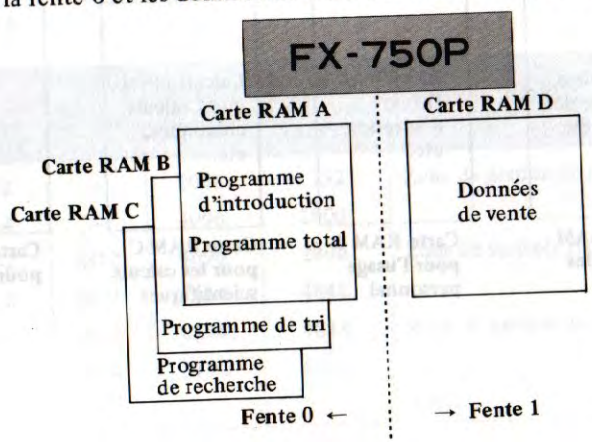
■ **Utilisation de deux cartes RAM en combinaison**

Puisque le FX-750P est équipé de deux fentes d'insertion de cartes, on peut naturellement l'utiliser comme mémoire amplifiée lorsque deux cartes sont insérées. L'ordinateur offrira une large possibilité d'utilisation générale, puisque les cartes pourront aussi être utilisées séparément pour des programmes ou des données.



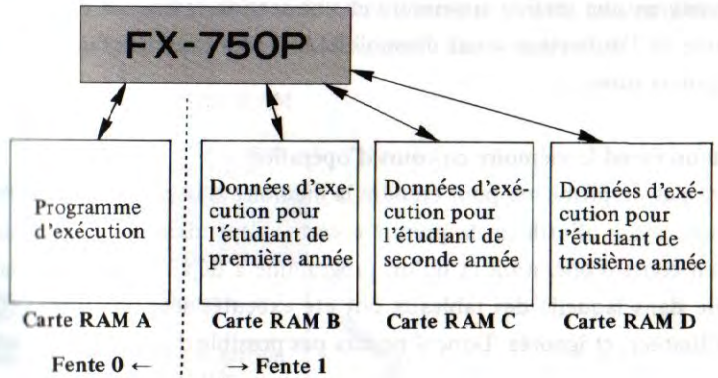
1. Traitement d'une donnée élémentaire par différentes programmes

Lorsqu'on effectue par exemple une analyse des ventes, il sera possible de faire diverses analyses sur une donnée si le programme d'introduction, le programme total, le programme de tri et le programme de recherche sont stockés dans la carte RAM de la fente 0 et les données de vente dans celle de la fente 1.



2. Traitement de données multiples avec un programme unique

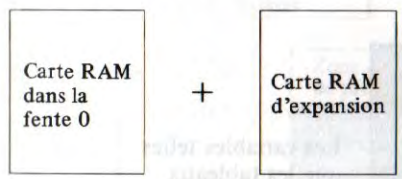
Pour les données multiples, les cartes RAM permettent un traitement beaucoup plus rapide que les bandes cassettes.



Lorsque l'on sépare ainsi des programmes et des données, on veillera à toujours bien insérer la carte RAM de programme dans la fente 0 et la carte RAM de données dans la fente 1.

3. Possibilité d'expansion de la mémoire en cours d'opération

Il arrive fréquemment que la mémoire fasse défaut pendant la création d'un programme. Grâce au système à deux fentes du FX-750P, on peut sans difficulté étendre la mémoire en ajoutant simplement une carte RAM dans la fente 1.



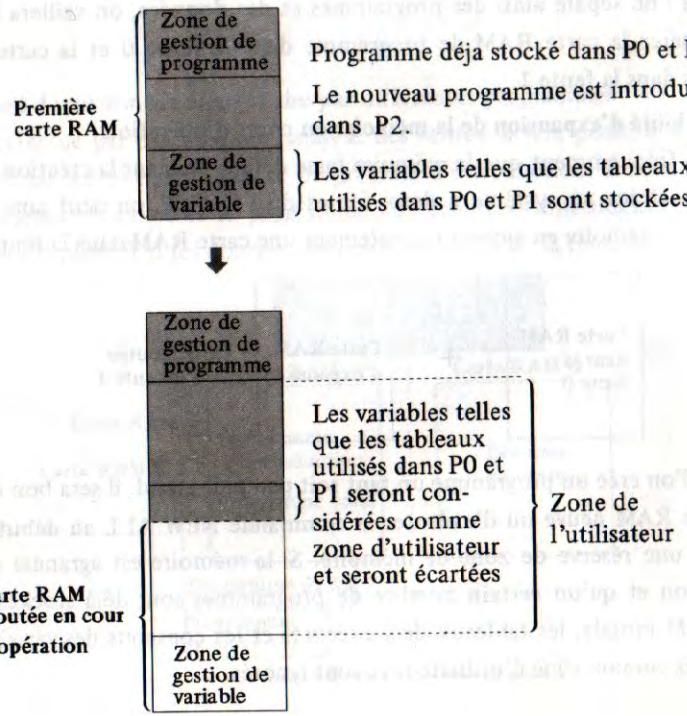
Lorsque l'on crée un programme un tant soit peu plus grand, il sera bon d'utiliser une carte RAM neuve ou d'exécuter la commande NEW ALL au début, afin de s'assurer une réserve de zone de mémoire. Si la mémoire est agrandie en cours d'opération et qu'un certain nombre de programmes sont déjà stockés dans la carte RAM initiale, les tableaux déjà exécutés et les contenus des variables sont considérés comme zône d'utilisateur et sont ignorés.

6. Zone de système et zone de l'utilisateur lorsqu'on utilise deux cartes RAM

Nous avons déja signalé qu'en utilisant deux cartes RAM la zone de système serait séparée en une section supérieure et une section inférieure et qu'une plus grande zone de l'utilisateur serait disponible. Il faudra toutefois faire bien attention aux points suivants.

1. Lorsqu'on étend la mémoire en cours d'opération

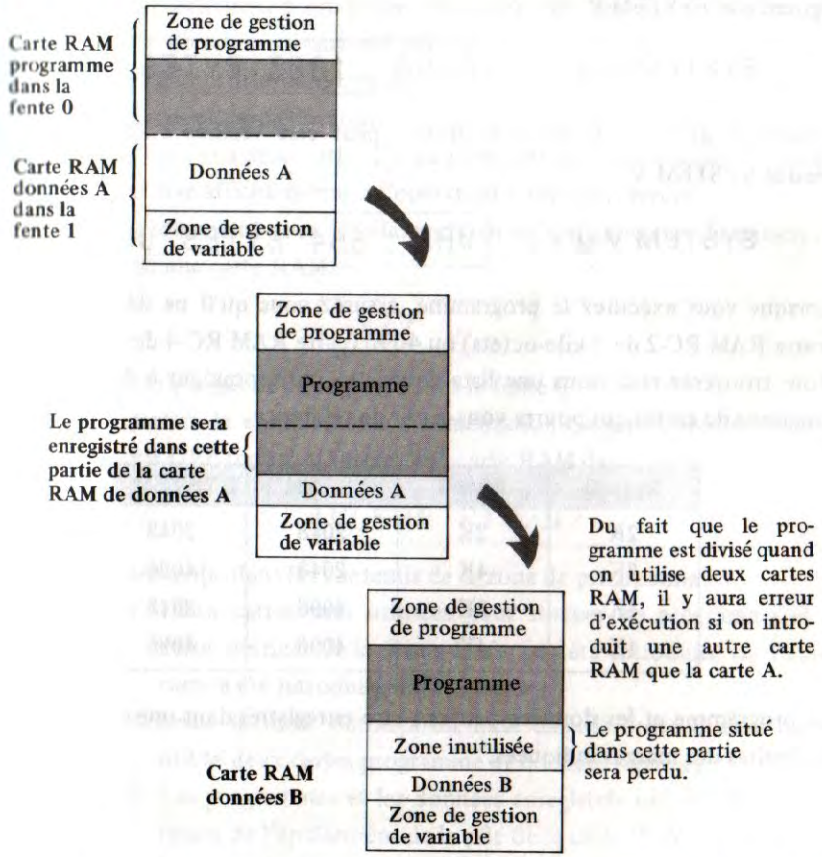
Il n'y aura pas de problème pour étendre la mémoire si la commande NEW ALL est exécutée avant d'utilider la première carte. Cependant, si la mémoire est agrandie en cours d'opération et qu'un programme a déjà été introduit, la zone de variable dans laquelle des tableaux ont été exécutés sera considérée comme zone d'utilisateur, et ignorée. Donc il ne sera pas possible d'appeler les données.



2. Lorsqu'on utilise séparément deux cartes RAM

Il n'y aura aucun problème lorsque deux cartes RAM seront utilisées comme un ensemble, la carte RAM de la fente 1 servant de mémoire étendue. Mais lorsqu'on les utilise séparément comme carte données et carte programme, la capacité de stockage de chaque carte RAM ne devra pas être dépassée.

Contenus initialisés de carte RAM



Il convient aussi de faire bien attention en disposant les zones; en effet, si des données étaient stockées dans une zone de programme, le traitement ne pourrait plus se faire correctement.

Lorsqu'on utilise ainsi deux cartes RAM séparément, les programmes et les données dépassant 2048 ou 4096 octets (caractères) ne doivent pas être stockés dans une carte RAM. C'est pourquoi il existe une commande de système (SYSTEM) pour vérifier si un programme et des données ne dépassent pas les capacités de la carte RAM.

La quantité de programme enregistré utilisé peut être vérifiée à l'aide de la commande SYSTEM P.

SYSTEM P↵

```
PROG  1003 BYTES USED
```

L'étendue de zone de variable utilisée peut être vérifiée à l'aide de la commande SYSTEM V.

SYSTEM V↵

```
VAR  664 BYTES USED
```

Lorsque vous exécutez le programme, assurez vous qu'il ne dépasse pas 2048 (carte RAM RC-2 de 2 kilo-octets) ou 4096 (carte RAM RC-4 de 4 kilo-octets). Vous trouverez ci-dessous une liste de valeurs correspondant à différentes combinaisons de cartes qui pourra vous servir de référence.

Fente 0	Fente 1	SYSTEM P	SYSTEM V
2K	2K	2048	2048
2K	4K	2048	4096
4K	2K	4096	2048
4K	4K	4096	4096

Le programme et les données peuvent être enregistrés dans une carte RAM dans les limites des marges indiquées.

7. Messages d'erreur consécutifs à l'insertion ou à l'enlèvement d'une carte RAM

Le FX-750P offre la possibilité d'effectuer des contrôles automatiques qui permettent de vérifier si une carte RAM a été correctement positionnée (par rapport aux zones de programme et de variable) après qu'on ait inséré une carte RAM et mis le commutateur d'alimentation électrique sur marche (ON).
Si la carte est correctement positionnée, "READY P0" apparaîtra à l'affichage. Par contre, si le message suivant est affiché, remettez correctement la carte RAM. Il faut aussi faire attention lorsqu'on répond au message d'erreur Y/N? car tous les programmes et données seront effacés si l'on appuie sur "Y".
D'autre part si l'on a mis deux cartes qui peuvent être utilisées indépendamment, READY P0 peut être affiché même si l'opération n'est pas correcte.

* Vérifier rigoureusement tous les éléments en ce qui concerne l'insertion ou l'enlèvement d'une carte RAM.

```
???
```

Cause: il n'y a pas de carte RAM dans la fente 0.
Remède: mettre le commutateur d'alimentation électrique de l'ordinateur sur OFF (arrêt) et insérer une carte RAM dans la fente 0.

```
CLEAR P:Y/N?
```

Cause: anomalie dans les contenus de la zone de programme
Exemples:
1. Deux cartes sont utilisées pour stocker un programme et la carte destinée à la fente 1 n'a pas été introduite ou l'autre carte a été introduite dans la fente 1.
2. Seule la carte données est mise dans la fente 0 alors qu'on utilise deux cartes programme et données séparées.
3. Les programmes et les données enregistrés ont été détruits en raison de l'épuisement de la pile de la carte RAM.

Remède:
1. Remettre les deux cartes dans la bonne fente.
2. Appuyer sur la touche [Y] (dans ce cas tous les programmes sont effacés et "READY P0" est affiché).
3. Remplacer la pile de la carte RAM.

```
CLEAR D:Y/N?
```

Cause: anomalie dans les contenus de la zone de variable.

Exemples:
1. Deux cartes sont utilisées pour stocker des données, et la carte destinée à la fente 1 n'a pas été introduite, ou l'autre carte est mise dans la fente 0 par erreur.
   dans la fente 0.
2. Alors qu'on utilise des cartes programme et données séparées, la carte programme est mise dans la fente 0 mais la carte données n'est pas mise dans la fente 1.
3. Le format est détruit en raison de l'épuisement de la pile de la carte RAM.

Remède:
1. Remettre les deux cartes RAM dans les fentes qui leur correspondent.
2. Appuyer sur la touche [Y] (dans ce cas tous les contenus de variable sont effacés et "READY P0" est affiché).
3. Remplacer la pile de la carte RAM.

```
NEWALL Y/N?
```

Cause: anomalie dans les contenus de la zone de programme et de la zone de données.

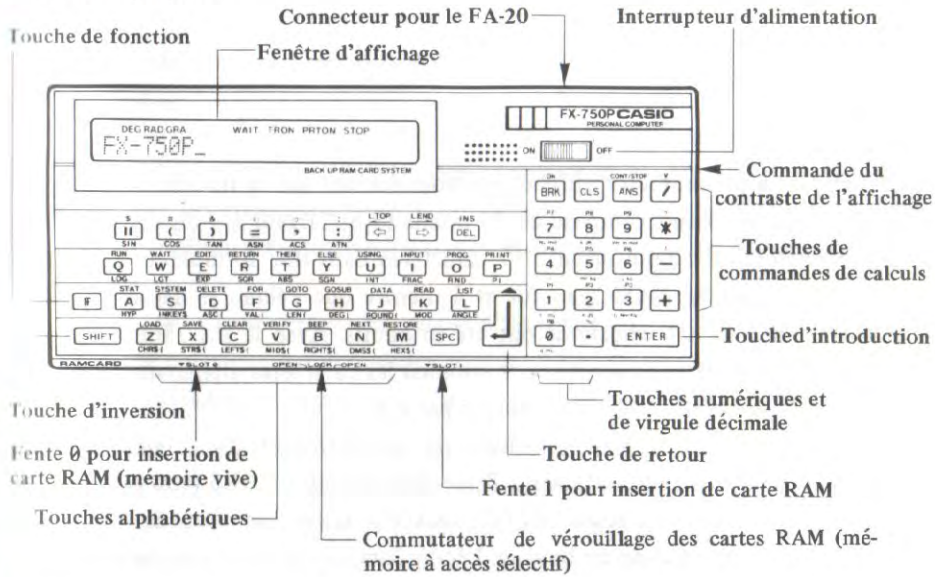
Exemple: Les cartes programme et données ont été inversées.

Remède:
1. Mettre la carte programme dans la fente 0 et la carte données dans la fente 1.
2. Appuyer sur la touche [Y] (dans ce cas tous les programmes et contenus de variable seront effacés).

*** Les cartes RAM conçues pour le FX-750P sont exclusivement réservées à cet ordinateur, et ne peuvent être utilisées sur un autre.**

1-3 Utilisation de votre FX-750P

Sur le clavier de votre FX-750P, les touches numériques et les touches de commandes de calculs sont disposées à droite de la touche de retour [↵], tandis que les touches alphabétiques (disposition classique des claviers de machine à écrire) et les touches de symboles se trouvent à gauche de cette touche. Pour comprendre bien clairement les différentes fonctions de la FX-750P, nous vous recommandons de les mettre toutes en pratique au fur et à mesure qu'elles vous seront expliquées.

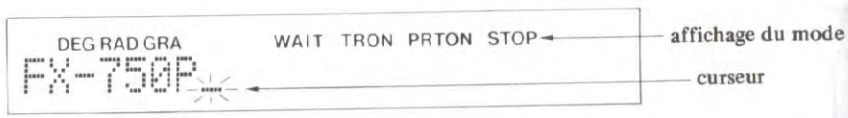


ON OFF **Interrupteur d'alimentation:**
Pousser cet interrupteur vers la gauche pour mettre en marche. Le symbole READY P0 sera alors affiché pour indiquer que votre FX-750P est prêt à recevoir l'introduction des données. Cet interrupteur sert aussi à rétablir l'alimentation lorsque le FX-750P s'est mis en "coupure automatique d'alimentation".

*** Fonction de coupure automatique de l'alimentation**
Il s'agit d'une fonction d'économie d'électricité; elle coupe automatiquement l'alimentation électrique au cas où l'on aurait oublié d'arrêter l'appareil 6 minutes après l'opération de la dernière touche. Pour remettre en marche, appuyer simplement sur la touche [ON BRK] et remettre l'interrupteur sur ON (Marche). Même

si l'alimentation est interrompue, le contenu de la mémoire et le programme ne s'effacent pas; toutes les spécifications d'angle et de mode ("WAIT", "TRON", etc.) seront par contre annulées. Cette fonction ne peut pas être effectuée lorsqu'un programme est en cours d'exécution.

■ L'affichage



24 caractères peuvent être affichés horizontalement sur la fenêtre d'affichage, chaque caractère étant composé de 35 points (5 en largeur x 7 en hauteur). On peut toutefois introduire jusqu'à 79 caractères dans une formule de calcul ou dans un programme BASIC, au fur et à mesure que l'écran se déroule vers la gauche si plus de 24 caractères ont été intoduits. Avec le FX-750P, on peut utiliser un total de 128 caractères, y compris les caractères alphanumériques et les symboles spéciaux. (Zéro sera indiqué par Ø.)

* Les caractères graphiques ne peuvent pas être introduits directement par le clavier, ils doivent être affichés à l'aide de la fonction CHR$ (voir page 145).

Les affichages d'unités d'angle "DEG", "RAD", "GRA" ou de mode "TRON", "STOP", etc. apparaissent en haut de l'écran lorsque ces modes particuliers ont été spécifiés.

- **DEG** (Mode degré): Ce mode est désigné par [F] [ANGLE] [0] [↵] . C'est une unité d'angle mesurant l'angle droit en 90°. Elle fonctionne dès que l'interrupteur est sur ON. (Voir page 47)
- **RAD** (Mode radian): Ce mode est désigné par [F] [ANGLE] [1] [↵] . C'est une unité d'angle mesurant l'angle droit en $\frac{\pi}{2}$ radians.
- **GRA** (Mode grade): Ce mode est désigné par [F] [ANGLE] [2] [↵] . C'est une unité d'angle mesurant l'angle droit en 100 grades.
- **TRON** (Mode analyse): Ce mode indique le déroulement du programme d'exécution et est désigné par [T] [R] [O] [N] [↵]. L'introduction de [T] [R] [O] [F] [F] [↵] annule le mode d'analyse. (Voir page 251)

- **PRTON** (Mode impression): C'est le mode d'impression, désigné par [SHIFT] [PRINT] [O] [N] [↵] . L'introduction de [SHIFT] [PRINT] [O] [F] [F] [↵] annule le mode d'impression. (Voir page 243)
- **STOP**: Apparaît quand l'execution d'un programme est interrompue. (Voir page 249)
- **WAIT**: Apparaît lorsque l'exécution d'un programme est provisoirement interrompue, pour afficher le contenu d'une instruction PRINT (Impression). (Voir page 252)

• Curseur

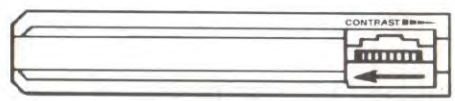
Quand l'appareil est sur ON et que des caractères ou une donnée numérique ont été introduits, le symbole "READY PØ" disparaît et le symbole " _ " clignote sur la droite du caractère affiché. C'est ce symbole qu'on appelle "curseur." Il indique la position du caractère qui suit et se déplace vers la droite, caractère par caractère.

On l'utilise aussi pour corriger les erreurs d'introduction.

Bien qu'on ne puisse pas introduire plus de 79 caractères par ligne, le curseur clignote en affichant le symbole " – " jusqu'au 71ème caractère, il affiche ensuite le symbole " ▒ " pour vous avertir que vous approchez de la fin de la ligne.

Réglage du contraste

Une commande de l'intensité lumineuse est à votre disposition sur le côté droit de votre FX-750P. Si vous souhaitez un affichage plus lumineux, tournez dans le sens de la flèche.



* Si l'intensité lumineuse de l'affichage reste faible, même à la position maximale de réglage, remplacez les piles dès que possible car il est probable qu'elles soient épuisées. (Voir page 2)

■ Fonction des touches

Votre FX-750P vous permet d'accomplir des fonctions multiples à l'aide d'une seule touche, simplement en changeant de mode. Appuyez simplement sur une touche si vous désirez uniquement le mode direct, appuyez simultanément sur une touche et sur la touche [SHIFT] pour opérer en mode de décalage et appuyez simultanément sur une touche et sur la touche [F] pour obtenir le mode Fonction.

* Ici, [SHIFT][ ] exige que l'on appuie sur ces 2 touches en même temps.

Les fonctions de la touche [Q] par exemple, sont les suivantes dans chacun des modes:

Fonctions des touches

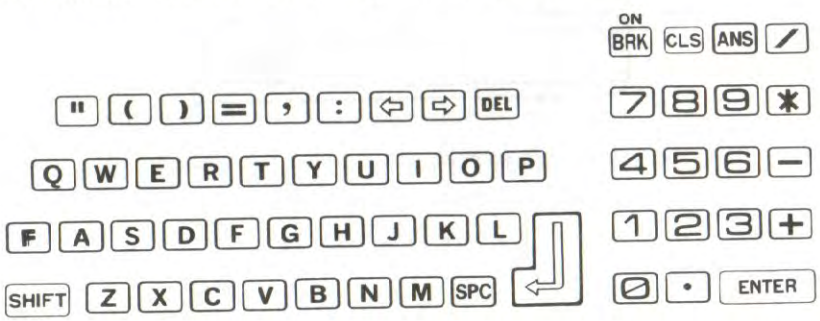
RUN
[Q]
LOG

[Q] ..... Q ..... Mode direct
[SHIFT][Q] ..... RUN ..... Mode décalage
[F][Q] ..... LOG ..... Mode fonction

Cela veut dire qu'avec une seule touche, vous pouvez introduire Q en mode direct, RUN (exécution de programme) en mode décalage et LOG (logarithme naturel) en mode de fonction. Vous trouverez ci-dessous la liste des touches principales et de leurs différents modes d'utilisation.

1. Mode direct

Dans ce mode, le caractère ou le symbole indiqué par la touche sera introduit, ou bien la fonction sere accomplie, dès qu'on aura appuyé sur la touche.



[0]~[9], [.] Touches numériques et de virgule décimale

- Appuyez sur ces touches lorsque vous désirez introduire un chiffre ou une virgule décimale.

[+] [−] [*] [/] Touches de commande de calcul

- Utilisez ces touches pour les calculs arithmétiques. Les touches [*] et [/] tiennent ici lieu de × et ÷, respectivement.

[(] [)] Touches de parenthèses

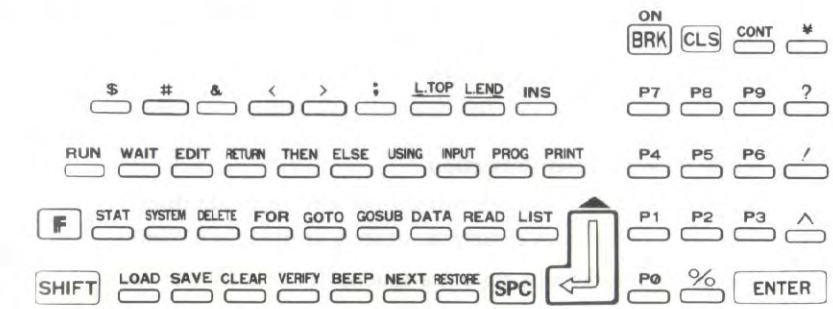
- Appuyez sur ces touches pour introduire une mise entre parenthèses.

[=] Touche égale

- Cette touche sert à indiquer des conditions égales d'affectation, d'instruction ou de jugement, dans une instruction IF (instruction conditionnelle).

2. Mode de décalage ([SHIFT][ ])

Une commande en BASIC (GOTO, PRINT, BEEP, etc.) ou le symbole indiqué au dessus d'une touche seront affichés, si l'on appuie simultanément sur la touche en question et sur la touche [SHIFT] .

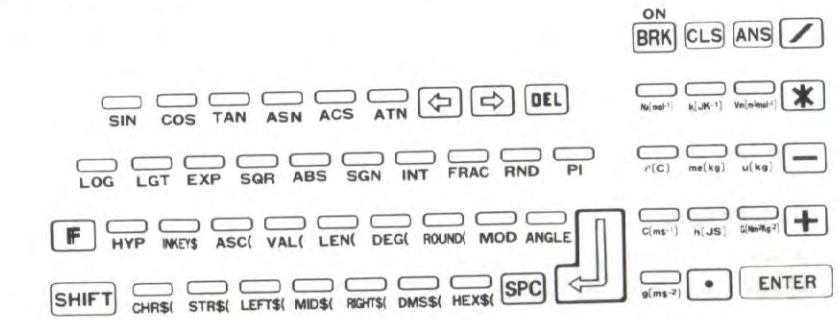


[SHIFT][P0] ~ [SHIFT][P9] **Touches de zone de programme**

- Utilisez ces touches pour désigner une zone de programme. Si un programme a déja été écrit dans cette zone, il sere exécuté.

3. Mode de fonction (F)

La fonction numérique (SIN, PI, etc.), la fonction de caractère (ASC, CHR$, etc.) ou les constantes scientifiques indiquées par une touche seront affichées, si on appuie simultanément sur la touche en question et sur la touche F . On trouvera au chapitre 2 des explications plus détaillées de ces opérations.



F 0 ~ F 9 **Constantes scientifiques**

- Utilisez ces touches pour faire apparaître des constantes scientifiques telles que la constante de Plank ou la constante d'Avogadro.

4. Touches d'édition et touches spéciales

INS
DEL **Touche de suppression/insertion**

- Supprime le caractère juste au dessus du curseur en mode direct, et déplace d'un espace vers la gauche les caractères situés à droite afin d'éliminer l'espace.
- Si l'on appuie simultanément sur cette touche et sur la touche SHIFT , tous les caractères situés au dessus et à droite du curseur seront déplacés vers la droite, pour faire un espace.

HOME
Touche gauche de curseur/Touche de retour à la ligne

- Chaque fois qu'on appuie sur cette touche, le curseur se déplace d'un caractère vers la gauche. Si cette touche est maintenue enfoncée, le curseur se déplacera de façon continue jusqu'à l'extrémité gauche de la ligne.
- Si l'on appuie simultanément sur cette touche et sur la touche SHIFT , le curseur reviendra au début de la ligne.

END
Touche de curseur à droite/Touche fin de ligne

- Chaque fois qu'on appuie sur cette touche, le curseur se déplace d'un caractère vers la droite. Si cette touche est maintenue enfoncée, le curseur se déplacera de façon continue jusqu'à l'extrémité droite de la ligne.
- Si l'on appuie simultanément sur cette touche et sur la touche SHIFT , le curseur se déplace jusqu'à la fin de la ligne.

CLS **Touche d'effacement de l'écran**

- Efface l'affichage et envoie le curseur à l'extrême gauche de l'écran.

CONT/STOP
ANS **Touche de réponse**

- Utilisez cette touche pour afficher le résultat d'un calcul manuel exécuté précédemment, ou le résultat des valeurs après l'exécution d'une instruction PRINT ou LPRINT, ou encore la valeur de la constante scientifique demandée.
- Cette touche interrompt temporairement l'exécution du programme. Pour reprendre l'opération, appuyer sur cette touche en même temps que sur la touche SHIFT .

Touche de retour

- Lorsque vous êtes en train d'écrire un programme, appuyez sur cette touche à la fin de chaque ligne. Cette touche sert également à exécuter une commande à l'intérieur d'un programme.
- Si l'on appuie sur cette touche en mème, temps que sur la touche [SHIFT] du mode EDIT, on peut faire apparaît la ligne précédente.

[ENTER] **Touche d'introduction**

- Même fonction que (=) et donne le résultat d'un calcul manuel. Cette touche a également la même fonction que [↵] , pendant l'exécution d'un programme, lorsqu'on attend l'introduction des données.

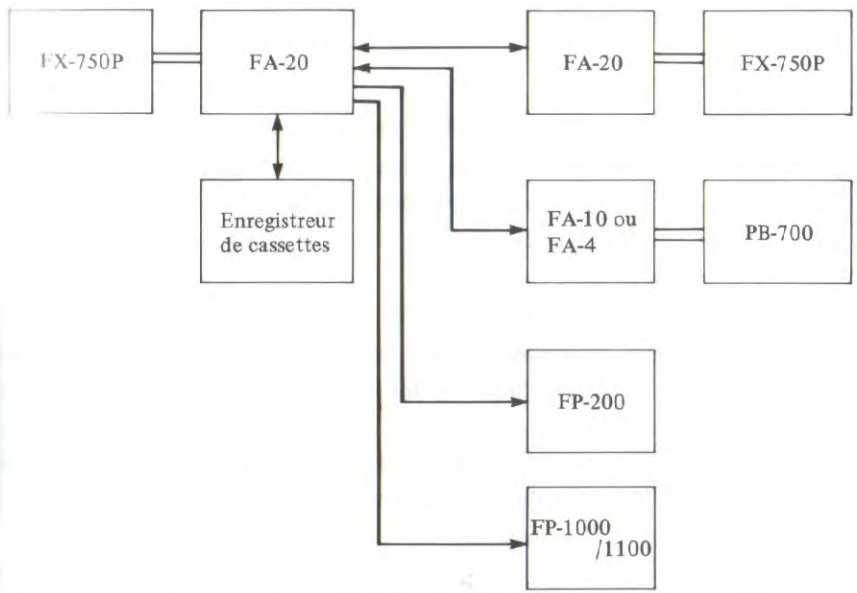
ON [BRK] **Touche de pause**

- Utilisez cette touche lorsque vous désirez interrompre l'exécution d'un programme.
- A utiliser également pour rétablir le courant quand la fonction coupure automatique d'alimentation a été activée.

1-4 Option

Le FX-750P peut parfaitement être utilisé seul, mais une imprimante de caractères avec une liaison cassette < FA-20 > est disponible en option (ne relier l'ordinateur avec aucun autre appareil que le FA-20).

La section imprimante du FA-20 peut imprimer les listes de programme, les résultats de calcul, etc.
La section liaison cassette du FA-20 est prévue pour être raccordée à un enregistreur de cassettes, de façon à ce que le programme et les données préparés par le FX-750P puissent être mémorisés sur une bande, et enregistrés à partir d'une bande. Elle permet également une intercommunication entre le FX-750P raccordé à un autre FA-20 et le PB-700 raccordé au FA-10 ou FA-4, ainsi que le transfert au FP-1000/1100 et au FP-200.



Pour le raccordement entre le FX-750P, le FA-20 et l'enregistreur de cassettes, ainsi que pour les points qui demandent une attention particulière, se reporter au manuel d'instructions fourni avec le FA-20. On ne trouvera décrit ci-après que le fonctionnement de base.

■ La section imprimante

En cours de programmation, il arrive fréquemment que des modifications du contenu du programme ou un dépistage d'erreurs deviennent nécessaires. Dans ce cas, la possibilité de sortir la liste de programme sur imprimante s'avère très pratique.
Le stockage de l'impression est aussi très commode. Pour imprimer un programme ou des données diverses, utiliser la commande LLIST.

Commande d'impression	Contenu
LLIST	Imprime le programme de la zone de programme actuellement spécifiée.
LLIST ALL	Imprime tous les programmes de toutes les zones de programme (P0 à P9).
LLIST V	Imprime les noms de variable enregistrée et les noms de variable de tableau déclarée.

■ Section liaison cassette

L'une des caractéristiques de la carte RAM, c'est sa disponibilité immédiate.
L'opération peut démarrer dès insertion de la carte dans l'unité centrale. La carte RAM est disponible sous diverses formes, selon l'usage auquel elle est destinée. Cependant, il est recommandé d'enregistrer des programmes et des données sur une cassette dans les cas suivants.

- Lorsqu'on change la pile d'une carte RAM.
- Lorsqu'on charge des programmes et des données dans une autre carte RAM.
- Lorsqu'on charge des programmes et des données pour constituer une carte de réserve.

Voici des instructions pour la mémorisation de programme et de données sur une cassette, et inversement, pour l'introduction dans le FX-750P de programme ou de données enregistrés sur une cassette.

- Stockage **de programme (SAVE)**

Lorsque l'on mémorise ou que l'on charge un programme, il est nécessaire que le câble CMT (câble de raccordement à l'enregistreur de cassettes) soit correctement connecté au FA-20 et à l'enregistreur de cassettes (pour la méthode de connexion, consulter le manuel d'instructions du FA-20).
Pour le stockage de programme, il y a trois commandes: "SAVE", "SAVE ALL" et "SAVE, A". "SAVE" sert à stocker le programme dans la zone de programme désignée. "SAVE ALL" stocke tous les programmes dans toutes les zones de programme de P0 à P9.
"SAVE, A" permet d'enregistrer sur une cassette le programme de la zone de programme désignée avec le format de code ASCII. Les enregistrements en code ASCII permettent l'introduction avec le FP-1000/1100 ou le FP-200.

Exemples:

```
SAVE↵
SAVE "CASIO"↵
SAVE ALL↵
SAVE ALL "AA"↵
SAVE ,A↵
SAVE "CASIO" ,A↵
```

Lorsque l'on stocke des programmes, les caractères entre guillemets constituent le nom de fichier. Le nom de fichier est facultatif, mais nous recommandons son emploi, car il permettra de spécifier un programme particulier quand, par la suite, on aura besoin de le charger. Un nom de fichier utilise jusquià huit caractères ou symboles.

- Chargement **de programme (LOAD)**

Pour le chargement de programme, il existe quatre commandes: "LOAD", "LOAD ALL", "LOAD,A" et "LOAD,M". Lorsqu'on emploie une commande LOAD, elle doit correspondre au format de la commande SAVE qui a été utilisé lors du stockage du programme. Dans le cas de la commande "LOAD,M", le

programme mémorisé avec le format de code ASCII peut être fusionné avec le programme qui est dans l'unité centrale. Lorsqu'on utilise la commande "LOAD" ou LOAD,A", le programme précédent est effacé. Les commandes "LOAD ,M" et "LOAD "Nom de fichier", M" permette de charger un programme supplémentaire. Si le même numéro de ligne existe déja, c'est celui qui aura été chargé en dernier qui aura la priorité et sera retenu.

Exemples:

LOAD↵
LOAD"CASIO"↵
LOAD ALL↵
LOAD ALL"AA"↵

LOAD,A↵
LOAD"CASIO",A↵
LOAD,M↵
LOAD"CASIO",M↵

- **Relation entre SAVE et LOAD**

Stockage	LOAD	LOAD "Nom de fichier"	LOAD ALL	LOAD ALL "Nom de fichier"	Affichage pendant chargement
SAVE	○	×	×	×	PFB
SAVE "Nom de fichier"	○	○	×	×	Nom de fichier PFB
SAVE ALL	×	×	○	×	AFB
SAVE ALL "Nom de fichier"	×	×	○	○	Nom de fichier AFB
SAVE ,A	×	×	×	×	PFA
SAVE "Nom de fichier" ,A	×	×	×	×	Nom de fichier PFA

Stockage	LOAD ,A	LOAD "Nom de fichier" ,A	LOAD ,M	LOAD "Nom de fichier" ,M	Affichage pendant chargement
SAVE	×	×	×	×	PFB
SAVE "Nom de fichier"	×	×	×	×	Nom de fichier PFB
SAVE ALL	×	×	×	×	AFB
SAVE ALL "Nom de fichier"	×	×	×	×	Nom de fichier AFB
SAVE ,A	○	×	○	×	PFA
SAVE, "Nom de fichier" ,A	○	○	○	○	Nom de fichier PFA

○ – Peut être chargé. × – ne peut pas être chargé.

- **Stockage et chargement de données (PUT, GET)**

On ne peut pas stocker ou charger des contenus de variables en se servant d'une commande SAVE ou LOAD. Pour stocker des contenus de variable, employer la commande PUT. Pour charger des contenus de variable, employer la commande GET.

Exemple de stockage de données:

PUT "CASIO" (nom de fichier) A,B,C (nom de variable)

Exemple de chargement de données

GET "CASIO" (nom de fichier) A,B,C (nom de variable)

Le nom de fichier peut être omis, mais nous vous recommandons de l'utiliser, car il vous permettra de distinguer facilement les programmes lors du chargement.

Attention

Certaines caractéristiques de l'enregistreur de cassettes ou la qualité de la bande cassette peuvent parfois empêcher une mémorisation précise des programmes et des données. Il sera donc bon de prendre l'habitude de vérifier la précision de la mémorisation après chaque opération de stockage, en chargeant sur l'unité principale le programme ou les données enregistrés sur la cassette.

Il peut y avoir de nombreuses causes à la mauvaise qualité du stockage. Voici les principales:

- Mauvaises connections, câbles, etc.
- Mauvaises opérations de stockage ou de chargement.
- Mauvais réglage du volume de l'enregistreur de cassettes au moment du chargement.
- Problèmes dûs à la bande cassette elle-même: tache ou autre endommagement de la bande.
- La tête de l'enregistreur n'est pas propre.

CHAPITRE 2

FONCTIONS DE CALCUL

Ce chapitre a pour but de vous faire connaitre les opérations de base, depuis les calculs manuels les plus simples jusqu'à l'utilisation des nombreuses fonctions numériques, fonctions de caractères, fonctions statistiques et constantes scientifiques que le FX-750P met à votre disposition.

Les fonctions numériques et de caractères peuvent être facilement exécutées par l'intermédiaire de fonctions à une seule touche.

Dans ce chapitre, nous utiliserons la méthode suivante pour désigner les opérations de touches.

- Les caractères alphabétiques, les chiffres et les commandes ne seront pas encadrés.
  Exemple:
  [C][O][S][(][P][I][/][3][)][ENTER]
  ➡ COS(PI/3)[ENTER]
- Bien que les commandes à touche unique et les fonctions à touche unique ainsi que les touches [SHIFT] et [F] psent aussi être utilisées avec les touches alphabétiques, à partir de maintenant nous les indiquerons exclusivement avec les touches alphabétiques.
  Exemple:
  [SHIFT] RUN ➡ RUN ⬅ [R][U][N]
  [F] LOG ➡ LOG ⬅ [L][O][G]

2-1 Avant d'effectuer les calculs

■ Ordre de priorité des calculs

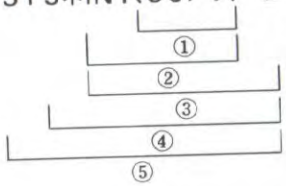
Tous les calculs obéissent à un ordre de priorité, mais cet ordre est inscrit dans la mémoire du FX-750P; les calculs seront donc automatiquement effectués dans l'ordre de priorité et l'affichage du résultat désiré sera obtenu en appuyant simplement sur les touches correspondant à l'expression numérique.

Ordre de priorité

1. Eléments entre parenthèses
2. Fonctions (sin, cos, tan, etc.)
3. Puissances (^)
4. Sélection des signes (affiche uniquement "–")
5. Multiplication et division (∗ , /)
6. MOD (calcul restant): laisse de côté les fractions des valeurs numériques.
7. Addition et soustraction (+ et –)
8. Opérateurs de relation (>, <, >=, < > etc.)

* Si l'ordre de priorité est identique, les calculs commenceront à partir de la gauche.

Example: 3+5∗INT(86/7)^2=723



■ Utilisation de l'opérateur de relations

Les opérateurs de relation servent d'expressions conditionnelles dans les instructions IF (voir page 235).

=	Ce qui est à gauche est égal à ce qui est à droite
<>,><	Ce qui est à gauche n'est pas égal à ce qui est à droite
<	Ce qui est à gauche est plus petit que ce qui est à droite
>	Ce qui est à gauche est plus grand que ce qui est à droite
=>,>=	Ce qui est à gauche est supérieur ou égal à ce qui est à droite
=<,<=	Ce qui est à gauche est inférieur ou égal à ce qui est à droite

Exemples:

```
IF A>30 THEN 500
```

Le programme passe alors à la ligne 500 si A est supérieur à 30.

```
IF B$="6"THEN BEEP 0
```

Un "bip" de tonalité basse retentira si B$ est égal à 6.

```
IF CHR$(65)>CHR$(N)THEN W=3
```

Si CHR$(65), c'est à dire A est plus grand que CHR$(N), on attribue la valeur de 3 à W.

■ Méthodes de calcul

Compositions

	$\frac{x+y}{2}$	→ (x + y) / 2
	x^3	→ x ^ 3
	$x^2+2xy+y^2$	→ x ^ 2 + 2 ∗ x ∗ y + y ^ 2
	$(-y)^2$	→ (–y) ^ 2
	$(x^y)^2$	→ x ^ y ^ 2
	x^{y^2}	→ x ^ (y ^ 2)
Reste de	$\frac{x}{y}$	→ x MOD y

Exemples:

0.5 ^ 0	→ 1 (0.5⁰ = 1)
–0.5 ^ 0	→ –1 (–0.5⁰ = –1)
(–0.5) ^ 0	→ 1
0.5 ^ 0.5	→ 0.7071067812
(–0.5) ^ 0.5	→ MA ERROR
123 MOD 5	→ 3
10 MOD –6	→ 4
–10 MOD–6	→ –4

■ Opérateurs de chaîne de caractères < + >

Les chaînes de caractères peuvent être reliées par le signe +. La valeur numérique encadrée par des guillemets (") sera considérée comme une chaîne de caractères. Les opérateurs autres que + ne peuvent pas servir d'opérateurs de chaîne.

Exemple:

"AB"+"123" [ENTER] ⟶ AB123
A$="FX-" [↵]
B$="750P" [↵]
C$= A$+B$ [↵]
C$ [ENTER] ⟶ FX-750P

■ Affichages d'erreurs

1. Une tentative de division par zéro entraîne une erreur MA (MA ERROR).
2. Un dépassement de la capacité de calcul ($\pm 1 \times 10^{-99} \sim \pm 9{,}999999999 \times 10^{+99}$) entraîne une erreur de dépassement (OV ERROR).
3. Capacité de calcul des puissances

($x>0$, $y>0$)	$0 \wedge 0$	→ MA ERROR
	$(\pm x) \wedge 0$	→ 1
	$0 \wedge y$	→ 0
	$0 \wedge (-y)$	→ MA ERROR
	$(-x) \wedge (\pm y)$	→ Seulement quand y est un nombre entier. Sinon cela entraîne une erreur MA (MA ERROR).

La précision sera de ±1 au 7ème chiffre quand $1 - 10^{-6} < x < 1 + 10^{-6}$ en $x \wedge y$.

* Voir page 324 pour les affichages d'erreurs.

■ Nombre de chiffres d'entrée/sortie et de chiffres de calcul

Jusqu'à 79 caractères peuvent être introduits par ligne dans une formule de calcul tandis que des calculs internes sont effectués avec une mantisse à 12 chiffres plus un exposant à 2 chiffres.
Le résultat du calcul sera exposé avec une mantisse à 10 chiffres (le 11ème chiffre étant arrondi au nombre entier le plus proche) plus un exposant à 2 chiffres.
Cet affichage passera automatiquement aux affichages exponentiels, s'il s'agit de 10^{10} ou plus, ou de moins de 10^{-3}.
Exemples:

1.2345678912 [ENTER]	1.234567891
12345678912 [✱] 100 [ENTER]	1.234567891E12
1 [✱] 0.0001 [ENTER]	1E-04

■ Niveaux d'opération

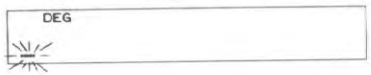
Valeurs numériques . . . . . 8 niveaux
Opérateurs de relations . . . . . 20 niveaux

* Les triples parenthèses seront égales à un opérateur d'un niveau.

(((123+456*((789+123*(456+_____
1 niveau 1 niveau 1 niveau

2-2 Calculs Manuels et corrections d'entrées

Avant de commencer un calcul manuel, appuyez sur la touche [CLS] pour effacer l'affichage.



Le curseur clignote à l'extrémité du côté gauche. Cela signifie que l'ordinateur attend l'introduction de données. Le mode DEG est réglé pour les calculs réclamant une spécification d'unité d'angle (fonctions trigonométriques etc.). Mais cet affichage de DEG n'interviendra pas sur les autres calculs.
Commensons par un simple calcul manuel.

Exemples:

$8+4.2\div5-2.5\times2=3.84$

Appuyez sur les touches correspondant à l'expression numérique. Utilisez la touche [*] pour la multiplication, la touche [/] pour la division et la touche [ENTER] pour "=".

Opération:

8 [+] 4.2 [/] 5 [−] 2.5 [*] 2

[ENTER]

```
8+4.2/5-2.5*2_
 3.84
```

On peut aussi poursuivre les calculs après avoir accompli une opération manuelle.

Opération:

123 [*] 456 [ENTER]

(Continuer à appuyer) [+] 789

[ENTER]

```
56088
56088+789_
56877
```

Si au cours d'un calcul vous souhaitez utiliser le résultat d'un calcul précédent, appuyez simplement sur la touche [ANS] .

Opération:

1.2 [+] 3.4 [ENTER]

11.5 [/] [ANS]

[ENTER]

```
 4.6
11.5/ 4.6_
 2.5
```


■ Modification des entrées (correction, suppression, insertion)

Utilisez les touches [←] , [→] , [INS DEL] pour modifier le contenu des entrées. Tant que le calcul manuel n'est pas terminé, des modifications partielles sont possibles; toutefois, on ne pourra plus effectuer de modification une fois que la touche [ENTER] aura été actionnée alors il vaut mieux effectuer une opération d'entrée à partir du début. Il est donc souhaitable que vous preniez l'habitude de revérifier votre entrée avant d'appuyer sur la touche [ENTER] .
Pour modifier un programme qui a déja été introduit, corrigez la ligne éditée et appuyez sur la touche [↵] qui convient après avoir actionné le mode EDIT et ([SHIFT] [EDIT]).

1) Correction

A l'aide des touches [←] ou [→] déplacez le curseur jusqu'à l'endroit à corriger et introduisez le caractère ou le symbole qui convient.

Exemple:

Correction de SIN 40 en SIN 30

```
SIN40_
```

Opération:

[←] [←] (Déplacez le curseur jusqu'au chiffre 4.)

Appuyez sur la touche [3] .

```
SIN40
SIN30_
```

2) Suppression

Amenez le curseur jusqu'au caractère ou symbole à supprimer; appuyez ensuite sur la touche [DEL] , le caractère sera éliminé et tous ceux qui se trouvent à la droite du curseur seront décalés d'un espacement afin de rétablir la mise en page.

Exemple:

Correction de LOKCATE en LOCATE.

```
LOKCATE_
```

Opération:

[←] [←] [←] [←] [←] (amenez le curseur au K)

Appuyez sur la touche [DEL] .

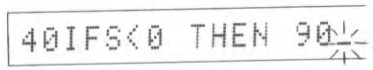
```
LOKCATE
LOCATE_
```

3) Insertion

Amenez le curseur à droite de l'endroit où l'insertion doit être faite et appuyez sur [SHIFT] [INS]. Un espace apparaît pour que le nouveau caractère ou symbole puisse être inséré.

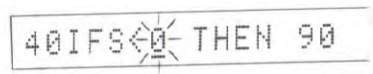
Exemple:

Correction de IF S < 0 en IF S <= 0

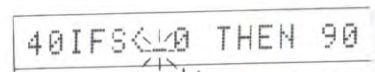


Opération:

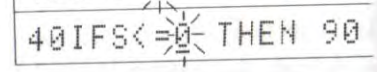
[←][←][←][←][←][←][←][←][←]
(amener le curseur au 0.)



[SHIFT] [INS]



Appuyer sur la touche [=].



■ Calculs avec utilisation de la fonction de mémoire

Les fonctions de mémoire permettent d'effectuer différents calculs mathématiques avec une grande commodité.

Exemple:

50000×0.75 =
80000×0.75 =

Ces calculs seront facilités si l'on met en mémoire la valeur dont l'introduction doit être répétée et qu'on l'appelle lorsque besoin est. Dans cet exemple, A = 0,75.

Dans ce cas, A est une variable. Une valeur numérique ou un caractère est assigné à une variable. Il convient de noter ici que "A = 0,75" est une instruction d'affectation et que "=" est une instruction signifiant que 0,75 est assigné à A. Le signe "=", tel qu'il est utilisé ici, ne doit donc pas être confondu avec le signe égal utilisé dans une formule de calcul.

Les opérations de touche sont les suivantes.

Opération:

A[=]0.75[↵]

Vous remarquerez bien que l'introduction n'est pas effectuée avec la touche [ENTER], mais avec la touche [↵], car ceci n'est pas une formule de calcul.

Vérifions maintenant si l'affectation correcte a été réalisée.

Opération:

A [ENTER]	0.75

Ensuite résolvons les formules de calcul données plus haut.

Opération:

50000[✱]A [ENTER]	37500
80000[✱]A [ENTER]	60000

* On trouvera une explication détaillée des variables au chapitre 3.

Il est important de comprendre bien clairement l'utilisation des touches [↵] et [ENTER], car une utilisation erronée de ces touches entraînerait une erreur de syntaxe (SN ERROR).

[ENTER]

Cette touche sert à afficher les résultats d'un calcul manuel ainsi qu'à confimer le contenu d'une variable.

[↵]

Cette touche sert à assigner des valeurs numériques et des expressions aux variables, à exécuter des commandes de programme en BASIC et à spécifier des positions comme par exemple le mode d'impression.

2-3 Fonctions numériques

Les fonctions numériques indiquées au tableau ci-dessous sont incorporées au FX-750P. Bien qu'il soit possible d'effectuer des entrées à l'aide des caractères alphabétiques, on peut aussi facilement introduire des équations complexes en utilisant la fonction á touche unique avec la touche [F], et des calculs techniques difficiles peuvent être effectués rapidement.

Fonction	Expression mathématique	Format	Signification
Fonctions trigonométriques	sin	SIN (expression numérique) * A partir d'ici X.	$-5400° < X < 5400°$ (30π rad, 6000 gra)
	cos	COS (X)	$-5400° < X < 5400°$ (30π rad, 6000 gra)
	tan	TAN (X)	$-5400° < X < 5400°$ (30π rad, 6000 gra) (à l'exclusion des cas où $\|X\| = (2n-1) \times 90°$) * n est un nombre entier.
Fonctions trigonométriques inverses	$\sin^{-1}$	ASN (X)	$\|X\| \leqq 1$, $-90° \leqq$ ASN (X) $\leqq 90°$
	$\cos^{-1}$	ACS (X)	$\|X\| \leqq 1$, $0° \leqq$ ACS (X) $\leqq 180°$
	$\tan^{-1}$	ATN (X)	$\|X\| < 10^{100}$, $-90° \leqq$ ATN (X) $\leqq 90°$
Fonctions hyperboliques	sin h	HYPSIN (X)	$\|X\| \leqq 230$
	cos h	HYPCOS (X)	$\|X\| \leqq 230$
	tan h	HYPTAN (X)	$\|X\| \leqq 10^{100}$, $-1 \leqq$ HYP TAN(X) $\leqq 1$
Fonctions hyperboliques inverses	$\sin h^{-1}$	HYPASN (X)	$\|X\| < 5 \times 10^{99}$
	$\cos h^{-1}$	HYPACS (X)	$1 \leqq X < 5 \times 10^{99}$
	$\tan h^{-1}$	HYPATN (X)	$\|X\| < 1$
Fonction exponentielle	e^x	EXP (X)	$-227 \leqq X \leqq 230$
Logarithme naturel	$\log_e x$	LOG (X)	X>0 la précision sera de ±1 au 8ème chiffre quand $1 - 10^{-6} < X < 1 + 10^{-6}$
Logarithme ordinaire	$\log_{10} x$	LGT (X)	X > 0 la précision sera de ±1 au 8ème chiffre quand $1 - 10^{-6} < X < 1 + 10^{-6}$
Racine carrée	$\sqrt{\ }$	SQR (X)	$X \geqq 0$
Valeur absolue	$\|x\|$	ABS (X)	Donne la valeur absolue de X.
Signe		SGN (X)	Donne { -1 quand X < 0, 0 quand X = 0, 1 quand X > 0. }
Nombre entier		INT (X)	Fonction de gauss: donne le plus grand nombre entier qui ne dépasse pas la valeur de X.

Fonction	Expression mathématique	Format	Signification
Fraction		FRAC (X)	Donne la fraction de X.
Arrondi		ROUND(x,y[,f],{+ / −})	Arrondit la valeur de x à la position 10^y.
		x,y: X f: 0 ou 1	(f est omis: } spécifie la $f = 0$: } position de 10^y $f = 1$: Spécifie la $y$ème position d'un nombre significatif
			Signe omis: arrondit au plus près +: arrondit au nombre supérieur −: écarte
Degrés/minutes /secondes	Nombres sexagésimaux → décimaux	DEG (d[,m[,s]]) d, m, s: X	Donne des valeurs converties en décimales
Constante circulaire	π	PI	Donne une valeur approximative de π à 3,1415926536
Nombre aléatoire		RND	0 < Nombre aléatoire < 1 Un nombre aléatoire à 10 chiffres est constitué.

(X) → Expression numérique

- Les parenthèses ne peuvent pas être omises pour ROUND et DEG, elles peuvent par contre être omises dans les autres cas où des valeurs numériques ou des variables sont utilisées comme (expression numérique).

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

Lorsque vous utilisez les fonctions trigonométriques ou trigonométriques inverses, spécifiez toujours les unités d'angle. (Aucune spécification ne sera nécessaire s'il n'y a pas à changer les unités d'angle.) Pour spécifier les degrés (DEG) utilisez ANGLE 0 ↵ , pour les radians (RAD) ANGLE 1 ↵ et pour les grades (GRA) ANGLE 2 ↵. Ces modes seront affichés en haut de l'écran.

- La relation entre DEG, RAD et GRA est la suivante:
  $90° = \frac{\pi}{2}$ (rad) = 100 grades.

- DEG

Exemple: sin45°

Opération: [F] [ANGLE] 0 ↵
[F] [SIN] 45 [ENTER]

DEG
0.7071067812

Fonctions numériques

Etant donné que l'utilisation des touches alphabétiques [S][I][N] pour SIN aura le même résultat que l'introduction de [F] SIN pour la commande de fonction à touche unique, nous omettrons dorénavant [F] et n'indiquerons que les caractères alphabétiques.

Exemple: sin 15.34166667

Opération: SIN15.34166667 [ENTER]

DEG
0.2645744261

Exemple: sin 15°20'30"

Opération: SIN DEG[(]15[)]20[)]30[)][ENTER]

DEG
0.2645744261

- **RAD**

Exemple: $\cos(\frac{\pi}{3}\text{rad})$

Opération: ANGLE 1 [↵]
COS[(]PI[/]3[)][ENTER]

RAD
0.5

- **GRA**

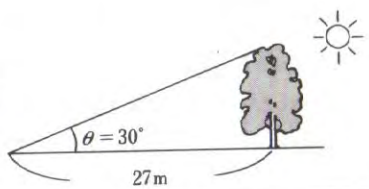
Exemple: $\tan(-35\text{gra})$

Opération: ANGLE 2 [↵]
TAN[−]35 [ENTER]

GRA
-0.6128007881

Faisons une pause et amusons nous à mesurer la hauteur d'un arbre. La hauteur des arbres, telle qu'elle est indiquée dans le schéma, peut être obtenue facilement si nous connaissons la longueur de leur ombre et l'angle θ.

Etant donné que $\tan\theta = \frac{\text{hauteur}}{\text{base}}$, la hauteur sera égale à base × tan θ. Nous spécifions alors DEG comme unité d'angle et nous introduisons base = 27 m et $\theta = 30°$.



Opération:

[F] ANGLE [0][↵]
27[*]TAN30 [ENTER]

15.58845727

Autrement dit notre arbre a une hauteur d'environ 15,6 m.

- **Fonctions hyperboliques (sinh, cosh, tanh) et fonctions hyperboliques inverses ($\sinh^{-1}$, $\cosh^{-1}$, $\tanh^{-1}$)**

Exemple: sinh 3.6

Opération: HYP SIN3.6 [ENTER]

18.28545536

Exemple: cosh 1.5 − sinh 1.5

Opération: HYP COS1.5[−]HYP SIN 1.5 [ENTER]

0.2231301602

Exemple: Valeur de x quand $\tanh 4x = 0{,}88$

$$x = \frac{\tanh^{-1}0.88}{4}$$

Opération: HYP ATN0.88[/]4 [ENTER]

0.3439419141

- **Fonctions logarithmiques ($\log_{10}$, $\log_e$) et fonctions exponentielles (e^x, x^y)**

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

Opération: LGT1.23 [ENTER]

0.08990511144

Exemple: $\log_e 90$

Opération: LOG90 [ENTER]

4.49980967

Exemple: $\log_{10} 456 \div \log_e 456$

Opération: LGT456[/]LOG456 [ENTER]

0.4342944819

Exemple: $e^{4.5}$ (pour obtenir l'anti-logarithme de log naturel 4,5)

Opération: EXP4.5 [ENTER]

90.0171313

Exemple: $10^{1.23}$ (pour obtenir l'anti-logarithme de log ordinaire 1,23)

Opération: 10 [SHIFT] △ 1.23 [ENTER]

16.98243652

Exemple: $\log_{10}\sin 40^\circ + \log_{10}\cos 35^\circ$
En outre quel est l'anti-logarithme du résultat?
(calcul logarithmique de sin 40° x cos 35°)

Opération:

ANGLE0↵	
LGT SIN40 + LGT COS35 ENTER	-0.2785679838
10 SHIFT ^ ANS ENTER	0.5265407845

■ Autres fonctions

Racine carrée < **SQR** >

Exemple: $\sqrt{2} + \sqrt{5}$

Opération: SQR2 + SQR5 ENTER → 3.65028154

Signes < **SGN** >

Exemple:

"1" est donné dans le cas d'un nombre positif, "−1" dans le cas d'un nombre négatif et "0" dans le cas de 0.

Opération:

SGN6 ENTER	1
SGN0 ENTER	0
SGN(−)8 ENTER	-1

Valeur absolue < **ABS** >

Exemple:

Valeur absolue de 1 − 78,9 ÷ 5,61

Opération:

ABS (1 − 78.9 / 5.61) ENTER → 13.06417112

* Pour la fonction ABS, prendre la valeur absolue de x, car si une valeur négative était attribuée à x, en SQRx ou LGTx, cela provoquerait une erreur MA (MA ERROR).

Exemple:

Trouvez la racine carrée de la valeur absolue de −26.

Opération:

SQR (ABS (−) 26) ENTER → 5.099019514

Nombre entier < **INT** >

Exemple:

Trouvez la partie entière de $\frac{7800}{96}$

Opération:

INT (7800 / 96) ENTER → 81

Fractions < **FRAC** >

Exemple:

Trouvez la partie fractionnaire de $\frac{7800}{96}$

Opération:

FRAC (7800 / 96) ENTER → 0.25

Arrondi < **ROUND** >

Exemple:

Arrondir 16666 à la position 10^2 (= 100)

Opération:

ROUND (1 6 6 6 6 , 2 , (−)) ENTER → 16000

Exemple:

Trouvez le quotient de 147 ÷ 13 arrondi aux 3 chiffres significatifs.

Opération:

ROUND (147 / 13, 4, 1, +) ENTER → 11.3

Angle < **DEG** >

(Sexagésimales → décimales)

Exemple:

Convertir 15°20'30" en décimales

Opération:

DEG (15 , 20 , 30) ENTER

```
15.34166667
```

Constante circulaire (π) < **PI** >

Exemple:

Trouver la surface d'un cercle dont le rayon est de 14 m

Opération:

PI ✱ 14 SHIFT △ 2 ENTER

```
615.7521601
```

Nombres aléatoires < **RND** >

Exemple:

Créer un nombre aléatoire

Opération:

RND ENTER

```
0.3413855482
```


2-4 Fonctions de caractères

Le FX-750P est équipé des fonctions de caractères indiquées sur le tableau ci-dessous.

Num	Format	Fonction
ASC	ASC(X$)	Donne le code ASCII du premier caractère de l'expression de caractères.
CHR$	CHR$(I)	Donne le caractère correspondant au code ASCII par l'intermédiaire d'une expression numérique. $0 \leq I < 256$
DMS$	DMS$(I)	Convertit la décimale en sexagésimale. $\|I\| < 10^{100}$
HEX$	HEX$(I)	Donne la chaîne de caractères de l'hexadécimale obtenue par conversion d'une décimale. $-32769 < I < 65536$
INKEY$	INKEY$	Permet l'entrée d'un caractère à partir du clavier.
LEN	LEN(X$)	Donne le nombre des caractères contenus dans une expression de caractères. $0 \leq X\$ \leq 79$
VAL	VAL(X$)	Convertit une expression de caractères en valeur numérique.
LEFT$	LEFT$(X$, I)	Recherche le nombre de caractères spécifiés par une expression numérique à partir de la gauche d'une expression de caractères. $0 \leq I < 256$
MID$	MID$(X$, I, J)	Recherche le nombre de caractères spécifiés par l'expression numérique No. 2 à partir de la position spécifiée par l'expression numérique No. 1.
RIGHT$	RIGHT$(X$, I)	Recherche le nombre de caractères spécifié par une expression numérique à partir de la droite d'une expression de caractères. $0 \leq I < 256$
STR$	STR$ (I)	Convertit une expression numérique en chaîne de caractéres.

X$ → Expression de caractères
I → Expression numérique ou expression numérique 1.
J → Expression numérique 2.

< **CHR$** >

Exemple:

Afficher des caractères graphiques

Opération:

F CHR$(223) ENTER

```
▫
```

Opération:

F CHR$(235) ENTER

```
♣
```

* Il n'est pas nécessaire d'appuyer sur la touche (pour les fonctions telles que CHR$ qui réclament des parenthèses, car elles possèdent déja la parenthèse

gauche "(". Par contre, n'oubliez pas d'appuyer sur la touche) pour fermer les parenthèses.

> Etant donné que, pour la fonction de caractère, on peut indifféremment utiliser la fonction à touche unique ou les touches alphabétiques, à partir de maintenant nous n'indiquerons plus que les caractères alphabétiques.

< DMS$ >

Exemple:

Convertir 15,34166667 en nombre sexagésimal

Opération:

DMS$ (15.34166667) ENTER

```
15°20'30
```

< HEX$ >

Exemple:

Convertir 250 en une chaîne de caractères exprimée en hexadécimales.

Opération:

HEX$ (250) ENTER

```
00FA
```

Pour l'utilisation des autres fonctions de caractères, se référer au Chapitre 5: DETAIL DES COMMANDES ET FONCTIONS.

■ Constante hexadécimale < &H >

Conversion en décimales d'un nombre hexadécimal comprenant 1 à 4 chiffres.

Exemple:

Convertir 00FA obtenu par **< HEX$ >** en décimales.

Opération:

SHIFT & H00FA ENTER

```
250
```

Exemple:

Affichage du caractère M.

Opération:

CHR$ (SHIFT & H4D) ENTER

```
M
```


2-5 Calculs statistiques

Il va sans dire que le calcul statistique est indispensable, tant dans le travail de bureau que le travail technique, pour analyser et obtenir de nouvelles données à partir des données déja recueillies. Grâce aux fonctions énumérées dans le tableau ci-dessous, le FX-750P peut effectuer facilement des calculs statistiques ardus et donner sans délai des coefficients corrélatifs et des valeurs estimées.

- Etant donné qu'il n'y a pas de fonctions à touche unique, appuyez sur les touches correspondant au format.
- Sur introduction de STAT LIST ↵ les statistiques de base dont le nom est suivi du signe ★ sont affichées en ordre séquentiel avec leur valeur.
  Appuyez sur ↵ ou ENTER si vous désirez interrompre l'affichage, appuyez à nouveau sur la même touche si vous voulez rétablir l'affichage. Pour les sorties imprimées, introduisez STAT LLIST ↵.

Format		Fonction
CNT★	n	Nombre de données statistiques traitées
SUMY★	Σy	Total des données y
SUMX★	Σx	Total des données x
SUMXY★	Σxy	Somme des produits des données x par les données y
SUMX2★	Σx^2	Somme des carrés des données x
SUMY2★	Σy^2	Somme des carrés des données y
SDX	$x\sigma_{n-1}$	Valeur de l'écart type de l'échantillon de la donnée x $\sqrt{\frac{n\Sigma x^2-(\Sigma x)^2}{n(n-1)}}$
SDY	$y\sigma_{n-1}$	Valeur de l'écart type de l'échantillon de la donnée y $\sqrt{\frac{n\Sigma y^2-(\Sigma y)^2}{n(n-1)}}$
SDXN	$x\sigma_n$	Valeur de l'écart type démographique de la donnée x $\sqrt{\frac{n\Sigma x^2-(\Sigma x)^2}{n^2}}$
SDYN	$y\sigma_n$	Valeur de l'écart type démographique de la donnée y $\sqrt{\frac{n\Sigma y^2-(\Sigma y)^2}{n^2}}$
LRA	A	Limite de la constante de régression linéaire $\frac{\Sigma y-\mathrm{LRB}\cdot\Sigma x}{n}$
LRB	B	Coefficient de régression linéaire $\frac{n\cdot\Sigma xy-\Sigma x\cdot\Sigma y}{n\cdot\Sigma x^2-(\Sigma x)^2}$
COR	r	Coefficient de corrélation $\frac{n\Sigma xy-\Sigma x\cdot\Sigma y}{\sqrt{\{n\Sigma x^2-(\Sigma x)^2\}\{n\Sigma y^2-(\Sigma y)^2\}}}$
EOX (X)	$\hat{x}$	Valeur estimée (Valeur de x estimée à partir de la valeur de y) $\mathrm{EOX}(y_n)=\frac{y_n-\mathrm{LRA}}{\mathrm{LRB}}$
EOY (X)	$\hat{y}$	Valeur estimée (Valeur de y estimée à partir de la valeur de x) $\mathrm{EOY}(x_n)=\mathrm{LRA}+x_n\cdot\mathrm{LRB}$

(X) → Expression numérique

La mémoire conserve les données qu'elle contient tant que STAT CLEAR ↵ n'a pas été programmé. Il faut donc toujours appuyer sur ces touches pour supprimer les données antérieures, avant d'introduire de nouvelles données.

■ Introduction de données statistiques

- Données statistiques à 1 variable
  - Donnée individuelle
    Données STAT ↵
  - Données multiples de valeur identique
    Données STAT SHIFT ; Fréquence ↵
- Données statistiques à 2 variables
  - Données individuelles
    Données x STAT , Données y ↵
  - Données multiples de valeur identique
    Données x STAT , Données y SHIFT ; Fréquence ↵

Les données sont introduites à l'aide de la touche ↵, mais le résultat de calculs statistiques tels que la déviation type sont obtenus à l'aide de la touche ENTER. Veillez à bien utiliser la bonne touche, sinon il en résulte une erreur SN (SN ERROR). Les valeurs initiales peuvent être établies pour les statistiques de base au moyen de l'instruction d'affectation.

Exemple: SUMX=123↵

D'après l'état des expéditions indiqué sur le tableau, trouver les déviations type et déterminer les écarts d'expéditions pour les produits x et y

produit \ Date	4	5	6	7	8
x	2	2	5	8	8
y	1	5	5	5	9

Opération:

STAT CLEAR↵
STAT2,1↵ STAT2,5↵ STAT5,5↵
STAT8,5↵ STAT8,9↵
STAT LIST↵ (Les statistiques de base seront automatiquement affichées.)

(Nombre de données)	CNT 5
(Total des données y)	SUMY 25
(Total des données x)	SUMX 25
(Total des données x et y)	SUMXY 149
(Total du carré des données x)	SUMX2 161
(Total du carré des données y)	SUMY2 157
Ultérieurement	READY P0
(Moyenne des données x) SUMX/CNT ENTER	5
(Moyenne des données y) SUMY/CNT ENTER	5
(Déviation type de x) SDXN ENTER	2.683281573
(Déviation type de y) SDYN ENTER	2.529822128

Bien que les valeurs totales et moyennes des produits x et y soient identiques selon les résultats des calculs, les variations d'expédition augmenteront pour le produit x avec l'augmentation de sa déviation type.

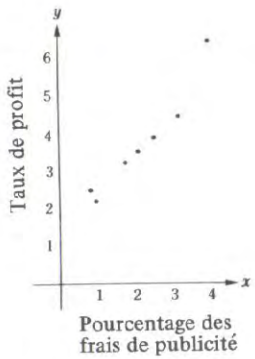
Effectuer ensuite un calcul de régression sur le couple de données et obtenir le coefficient de corrélation et les valeurs estimées.

Exemple:

Le tableau ci-dessous indique le pourcentage de frais publicitaires (frais publicitaires/frais d'opération x 100) et le pourcentage des profits d'opération (profits d'opération/ventes x 100) de 7 succursales d'une même chaîne de super-marchés pour l'année fiscale écoulée. Voyons maintenant si ces frais publicitaires ont été rentables.

	1	2	3	4	5	6	7
Pourcentage des frais publicitaires	0.8	2.1	2.5	1.8	3.1	4.0	1.0
Pourcentage des profits d'opération	2.5	3.4	3.7	3.2	4.3	6.3	2.3

A partir de ce tableau, préparons un shéma de points.



Le shéma de points montre que les profits augmentent proportionnellement aux frais de publicité. La ligne qui joint les points tracés sur le schéma est appelée courbe de regression; il s'agit dans ce cas d'une régression linéaire, puisque cette ligne est presque droite. Dans les courbes de régression, nous avons aussi les courbes logarithmiques, les courbes exponentielles et les courbes des puissances; toutes ces courbes peuvent être sélectionnées en fonction des relations qu'on désire établir entre les donnée x et y.

On sait aussi que le coefficient de corrélation (r) est compris dans la gamme $-1 \leqq r \leqq 1$ et que la corrélation sera positive quand $0 < r \leqq 1$ et négative quand $-1 \leqq r < 0$. Il n'y aura pas de corrélation quand $r = 0$.

Sachant cela, introduisons une donnée pour les 7 magasins et obtenons la donnée statistique qui concerne chacun d'entre eux.

Opération:

STAT CLEAR ↵

STAT0.8 ▷ 2.5 ↵ STAT2.1 ▷ 3.4 ↵ STAT2.5 ▷ 3.7 ↵

STAT1.8 ▷ 3.2 ↵ STAT3.1 ▷ 4.3 ↵ STAT4.0 ▷ 6.3 ↵

STAT1.0 ▷ 2.3 ↵

(Terme constant de la régression linéaire: A)	LRA ENTER	1.174221646
(Coefficient de la régression linéaire: B)	LRB ENTER	1.142512973
(Coefficient de corrélation: r)	COR ENTER	0.9628252383

D'après la valeur de r, nous savons maintenant que la corrélation entre x et y est positive. Essayons maintenant de déterminer quel devrait être le pourcentage de frais publicitaires pour un taux de profit de fonctionnement de 5,7%; essayons également d'estimer le taux de profit d'opération correspondant à un pourcentage de frais publicitaires de 4,5%.

Opération:	EOX5.7 ENTER	3.961248986
	EOY4.5 ENTER	6.315530022

La réponse obtenue nous dit que le pourcentage des frais publicitaires pour un taux de profit de fonctionnement de 5,7% sera de 3,96%, et que le taux de profit de fonctionnement correspondant à des frais publicitaires de 4,5% sera approximativement de 6,32%.

▪ Calculs de régression logarithmique, régression exponentielle et régression de puissances

A l'aide des données indiquées dans le tableau, effectuer les divers calculs de régression. Mettre l'année fiscale en x et les réalisations en y.

Anné fiscale	Réalisations
54	5.810
55	5.637
56	6.736
57	7.938
58	8.169

• Calcul de régression logarithmique

La formule de régression est la suivante: $y = A + B \cdot \ln x$. Introduire le log (ln) de x pour la donnée x et introduire la donnée y, comme dans la régression linéaire. Pour Σx on obtiendra le calcul de $\Sigma \ln x$, pour Σx^2: $\Sigma(\ln x^2)$ et pour Σxy: $\Sigma \ln xy$.

Opération:

STAT CLEAR ↵

STAT LOG54 ▷ 5810 ↵ STAT LOG55 ▷ 5637 ↵

STAT LOG56 ▷ 6736 ↵ STAT LOG57 ▷ 7938 ↵

STAT LOG58 ▷ 8169 ↵

(Terme constant de régression A)	LRA ENTER	-151086.8602
(Coefficient de régression B)	LRB ENTER	39240.6409
(Coefficient de corrélation r)	COR ENTER	0.9461867989
(Coefficient de décision r^2)	SHIFT △ 2 ENTER	0.8952694584

- **Calcul de régression exponentielle**

La formule de régression est la suivante: $y = A \cdot e^{B \cdot x}$ ($\ln y = \ln A + B \cdot x$). Introduire le log (ln) de y pour la donnée y et introduire la donnée x, comme dans la régression linéaire. Pour le terme de décision A on obtiendra le calcul de lnA, pour le total SUMY: $\Sigma \ln y$ et pour le total des carrés SUMY 2: $\Sigma x \cdot \ln y$.

Opérations:

STAT CLEAR↵
STAT 54▷LOG5810↵ STAT 55▷LOG5637↵
STAT 56▷LOG6736↵ STAT 57▷LOG7938↵
STAT 58▷LOG8169↵

(Terme constant de régression A)	EXP LRA ENTER	21.93154256
(Coefficient de régression B)	LRB ENTER	0.102384121
(Coefficient de corrélation r)	COR ENTER	0.9442661562

- **Calcul de régression des puissances**

La formule de régression est la suivante: $y = A \cdot x^B$ ($\ln y = \ln A + B \cdot \ln x$).
Introduire les logarithmes à la fois pour les donnés de x et de y. Pour le terme constant A on sera le calcul de lnA, pour Σx: $\Sigma \ln x$, pour Σx^2: $\Sigma (\ln x)^2$, pour Σy: $\Sigma \ln y$, pour Σy^2: $\Sigma (\ln y)^2$ et pour Σxy: $\Sigma (\ln x \cdot \ln y)$.

Opération:

STAT CLEAR↵
STAT LOG54▷LOG5810↵ STAT LOG55▷LOG5637↵
STAT LOG56▷LOG6736↵ STAT LOG57▷LOG7938↵
STAT LOG58▷LOG8169↵

(Terme constant de régression A)	EXP LRA ENTER	6.651154824E-07
(Coefficient de régression B)	LRB ENTER	5.725355325
(Coefficient de corrélation r)	COR ENTER	0.9433168782

2-6 Constantes scientifiques

Le FX-750P est équipé de 10 constantes scientifiques qu'on peut rappeler à l'aide des touches numériques (0 ~ 9) en mode de fonction.

Opération	Nom	Symbole	Valeur numérique	Unité
F 0	Vitesse d'accélération d'un corps en chute libre	g	9.80665	ms^{-2}
F 1	Vitesse de la lumière (dans l'espace)	c	299792458	ms^{-1}
F 2	Constante de Plank	h	6.626176×10^{-34}	Js
F 3	Constante de la gravitation	G	6.672×10^{-11}	Nm^2kg^{-2}
F 4	Charge élémentaire	e	$1.6021892 \times 10^{-19}$	C
F 5	Masse de l'électron	me	9.109534×10^{-31}	kg
F 6	Masse atomique	u	$1.6605655 \times 10^{-27}$	kg
F 7	Constante d'Avogadro	N_A	6.022045×10^{23}	mol^{-1}
F 8	Constante de Boltzmann	k	1.380662×10^{-23}	JK^{-1}
F 9	Volume de 1 kg-mole sous STP.	V_m	0.02241383	m^3mol^{-1}

Exemple:

Supposer qu'une source de lumière ultra violette ayant une fréquence de vibration de $\nu = 1{,}16 \times 10^{15}$ (H_2) irradie des photons de fréquence fixe et calculer l'énergie de ces photons.
Si nous désignons l'énergie des photons par E et la constante de Plank par h, nous avons: $E = 1{,}16 \times 10^{15} \times h$ (appuyer sur F 2) d'après la formule de l'énergie des photons $E = h\nu$.

Opération:

1.16E15 ✱ F 2 ENTER → 7.68636416E-19

Exemple:

Trouver la valeur moyenne de l'énergie du mouvement translationnel que possède une molécule d'un gaz idéal à 0°. Si nous désignons la valeur moyenne de l'énergie cinétique par $<K>$, la température absolue par T (0° = 273) et la constante de Boltzmann par k, nous avons $<K> = 3 \div 2 \times 273 \times k$ (introduire [F][8] pour obtenir ce programme), d'après l'équation $<K> = 3/2\ kT$, qui donne la valeur moyenne de l'énergie cinétique du mouvement thermique.

Opération:

3 / 2 * 273 * [F][8] ENTER

```
5.65381089E-21
```

Comment acquérir des connaissances scientifiques grâce au FX-750P

• Vitesse de la lumière

On die que la planète Jupiter, qui est la plus grosse planète du système solaire, est éloignée de 778.300.000 km du soleil. Etablissons combien de temps il faut à la lumière solaire pour atteindre Jupiter. Si nous désignons le temps par t, la distance par S et la vitesse de la lumière par c, nous aurons $t = \frac{S}{c}$, conformément à la formule de la vitesse de la lumière: $c = \frac{S}{t}$. Si nous substitutions maintenant les valeurs numériques aux variables, nous aurons:

$$t = \frac{778300000000\,[m]}{\text{Vitesse de la lumière}\ ([F][1])} = \frac{7.783 \times 10^{11}\,[m]}{[F][1]}$$

Opération:

7.783E11 / [F][1] ENTER

```
2596.129353
```

Le temps recherché est donc approximativement de 2.596 secondes, soit 43 minutes et 16 secondes.

• Vitesse et cycle de révolution d'un satellite

Calculons la vitesse et le cycle de révolution d'un satellite qui se déplace sur une orbite circulaire autour de la terre à une distance de 5.500 km.

Estimons le rayon de la terre à 6.400 km et sa masse à 6×10^{24} kg.

Désignons par v la vitesse du satellite, par T son cycle de révolution, par M la masse de la terre ($= 6 \times 10^{24}$), par m la masse du satellite, par r son rayon de révolution ($= (6400 + 5500) \times 10^3 = 1.19 \times 10^7$), par G la constante gravitationnelle (appuyer sur [F][3] pour l'obtenir).

Etant donné que le satellite développe un mouvement circulaire uniforme, nous pouvons établir une équation de l'équilibre des forces où gravité universelle = force centrifuge. Autrement dit, nous pouvons formuler cette équation de la manière suivante: $G\frac{mM}{r^2} = \frac{mv^2}{r}$.

De cette équation, nous déduisons: $v = \sqrt{\frac{GM}{r}}$ $\quad T = \frac{2\pi r}{v}$

Pour obtenir la vitesse v, il suffit maintenant de remplacer les variables par leurs valeurs numériques et de calculer

$$v = \sqrt{\frac{G \cdot 6 \times 10^{24}}{1.19 \times 10^7}}$$

Opération:

SQR ([F][3] * 6E24 / 1.19E7) ENTER

```
5800.028977
```

Ainsi que l'indique l'affichage, nous obtenons comme résultat une vitesse d'environ 5800 m/sec.

Calculons ensuite T.

$$T = \frac{2\pi \cdot 1.19 \times 10^7}{v}$$

Si on continue l'opération des touches à partir du calcul de la vitesse, le résultat sera obtenu à l'aide de la touche [ANS].

Opération:

2 [*] PI [*] 1.19E7 [/] [ANS] [ENTER]

```
12891.29855
```

Le résultat obtenu donne un cycle de révolution d'environ 12.891 secondes, soit:

Opération:

DMS$ [(] [ANS] [/] 3600 [)] [ENTER]

```
3°34'51.3
```

approximativement 3 heures, 34 minutes et 51 secondes.

CHAPITRE 3

REFERENCE DE "BASIQUE"

Le chapitre précédent vous a permis de vous familiariser avec les fonctions de calcul, si bien que vous savez maintenant que le FX-750P est aussi une calculatrice qui peut effectuer des calculs scientifiques.

Au chapitre trois, vous allez vous familiariser avec les programmations qui vous permettront d'utiliser pleinement tout l'éventail des nombreuses fonctions du FX-750P.

3-1 A propos des programmes

Si nous pensons aux dernières réalisations scientifiques, il est probable que ce soit la navette spaciale qui nous vienne en premier à l'esprit. Si on la compare aux énormes fusées qui ont été lancées dans le passé, on peut dire que la navette spaciale, par sa légèreté et sa compacité est une invention digne de notre époque de préservation des ressources.

Comme nous le savons tous, la navette spaciale est contrôlée par des ordinateurs. Mais certains de ces contrôles méritent d'être explicités:

- La navette est lancée en position verticale et ses deux propulseurs (fusées chargées du lancement initial) quittent l'orbite à une altitude de 44 km.
- Une fois détachés, les propulseurs tombent en mer dans une zone programmée, où ils pourront être récupérés pour réutilisation.
- Le moteur principal brûle pendant huit minutes et le réservoir extérieur se détache à une altitude de 109 km. Ceci se passe juste avant que la navette n'entre en orbite.
- Dès qu'elle entre en orbite, la navette spatiale tourne autour de la terre à une vitesse de 7.743 m/sec.
- Lorsqu'elle quitte son orbite, son taux de décélération est de 91 m/sec.

Il existe une quantité d'autres contrôles complexes commandés par ordinateur. Ces ordinateurs sont eux-mêmes soumis à des programmes de précision, dont on peut donc dire qu'ils sont extrêmement précis et qu'ils constituent des instructions de travail détaillées destinées à permettre à l'ordinateur d'exécuter correctement sa tâche. Une fois que l'ordinateur a mémorisé le programme, il fonctionne en toute fiabilité conformément au programme en question. Si nous voulons confier un peu de notre travail quotidien à un ordinateur, comment allons nous procéder pour lui donner les instructions convenables?

3-2 L'organigramme

Pour donner à l'ordinateur des instructions correctes, nous devons d'abord analyser ce que nous désirons qu'il fasse.

Nous pouvons analyser mentalement un travail simple, mais si nous entreprenons d'analyser mentalement un travail plus compliqué, cela sera très difficile voire impossible; il se peut même qu'une étape importante nous échappe.

C'est pourquoi nous mettons en général les points importants par écrit, ou utilisons des schémas pour clarifier notre analyse. Ainsi il nous est possible d'avoir une vue d'ensemble, avec une base systématique. Nous pouvons aussi rechercher les erreurs et les omissions commises en cours d'analyse.

▪ Préparation de l'organigramme

Il existe de nombreuses méthodes pour expliquer facilement et efficacement les procédés de travail. L'une d'entre elles est la méthode de l'organigramme qui permet de classer les différents procédés de travail en blocs simples, chaque bloc indiquant la signification du procédé qui lui correspond. Pour faire apparaître les relations procédurales et le déroulement du travail en cours, il suffit de rattacher ces blocs par des lignes.

Prenons maintenant un exemple simple pour expliquer comment préparer l'organigramme.

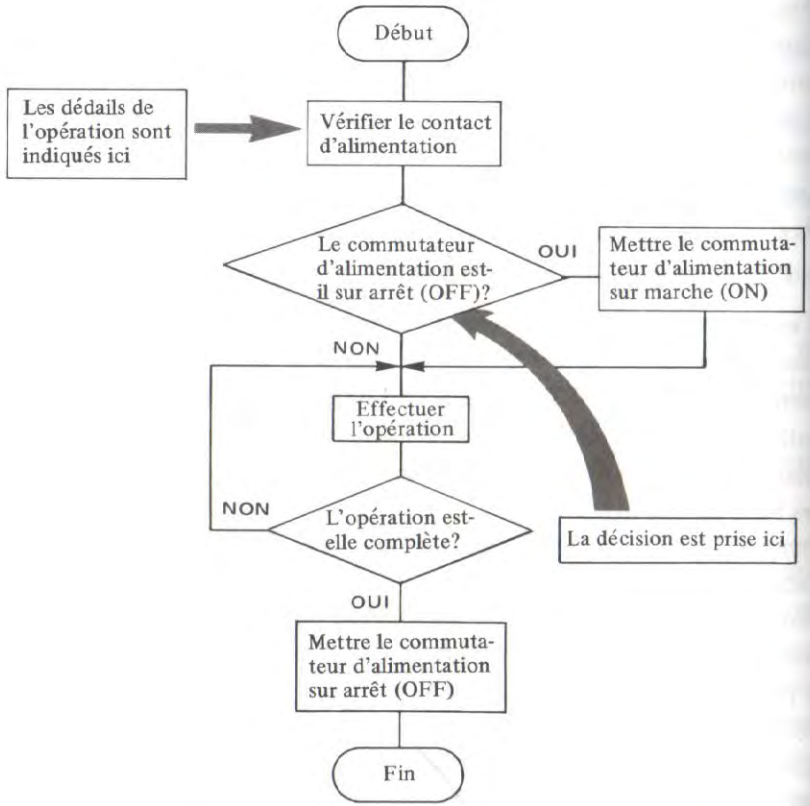
Pour utiliser le FX-750P, vous devez toujours mettre le commutateur d'alimentation sur marche. Examinons maintenant cette opération de mise sous tension et la préparation d'un organigramme.

Pour utiliser l'ordinateur, le commutateur doit être mis sur ON (marche). Lorsque nous avons terminé, il doit être mis sur OFF (arrêt). Si le commutateur est déjà sur ON, ne rien changer. Le commutateur ne sera jamais mis sur OFF pendant le cours d'une opération.

De ce qui vient d'être dit, nous déduisons la marche à suivre suivante.

1. Vérifier le commutateur d'alimentation électrique.
2. Le mettre sur marche (ON) s'il est sur arrêt (OFF). S'il est déja sur marche, passer à l'étape suivante.
3. Effectuer l'opération.
4. Si l'opération est incomplète, revenir à la phase 3. Si elle est complète, passer à la phase suivante.
5. Mettre le commutateur d'alimentation sur arrêt (OFF).

Etablissons maintenant un organigramme basé sur les phases 1 à 5.



Des opérations complexes peuvent ainsi être comprises sans difficulté, grâce à l'utilisation des organigrammes. La préparation des programmes utilisés pour les ordinateurs est basée sur le même type de procedé que celui qu'on suit dans la préparation des organigrammes.

Vous trouverez ci-dessous quelques uns des symboles fréquemment utilisés dans la préparation des organigrammes (voir page 330).

Limite 

Ce symbole de limites est utilisé au début et à la fin d'un organigramme.

Entrée manuelle 

Indique une introduction par opération manuelle

Traitement 

Indique l'exécution d'une opération dans le bloc.

Affichage 

Ce symbole est utilisé pour afficher les résultats d'un calcul, etc.

Connexion

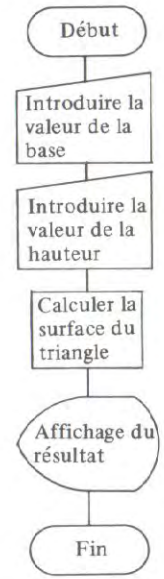
C'est une flèche qui relie deux symboles. Le travail se poursuit selon la direction indiquée par la flèche.

Etant donné que les organigrammes sont, en principle, écrits de haut en bas ou de gauche à droite, si les flèches sont omises, le travail se déroulera soit vers le bas soit vers la droite.

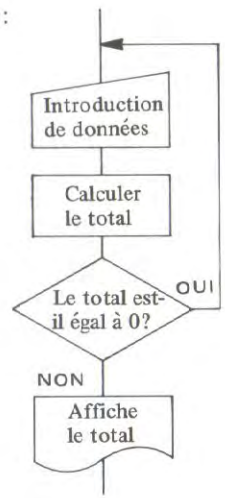
Décision 

Au fur et à mesure que l'opération se déroule, des décisions deviennent nécessaires. Les décisions, les interrogations ou les comparaisons sont formulées en relation avec les éléments des blocs. Il y a en général deux sorties (branches) ou plus.

Exemple:



Exemple:

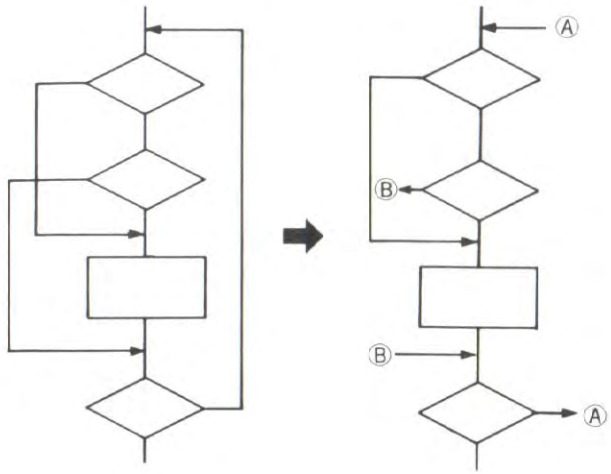


Document

Ce symbole est utilisé pour envoyer les résultats d'un calcul et différents documents à l'impression.

Connecteur ○

Ceci indique une sortie vers un endroit extérieur à l'organigramme ou une entrée dans l'organigramme.



Symbole de définition

Indique une opération définie ailleurs, comme par exemple un sous-programme.

Exemple



3-3 La programmation

Jusqu'ici, toutes les explications données concernaient l'analyse du travail et la préparation de l'organigramme. Nous allons maintenant passer à la préparation réelle d'un programme et voir comment faire fonctionner l'ordinateur.

Il existe de nombreux types de langages d'ordinateurs. Les ordinateurs de grande dimension utilisent en général comme langage le COBOL ou le FORTRAN, tandis que l'usage du BASIC est très largement répandue pour les ordinateurs personnels. Pour le FX-750P, c'est le BASIC qui est utilisé. Ce langage a été développé à l'intention des débutants, dans une optique de simplicité de programmation et les débutants qui travaillent avec un ordinateur pour la première fois n'auront donc aucune difficulté à l'apprendre.

▪ Préparation d'un programme échantillon

Créons un programme à partir d'un exemple simple comme: "Obtenir la racine cubique x de la valeur numérique y"

1. Tâche analytique

1) Introduire un certain nombre y à partir du clavier

2) L'équation qui donnera la racine cubique x sera la suivante:

$$x = e^{\frac{\log_e y}{3}}$$

$$y = x^3$$

Si nous prenons le log de chacun des côtés, nous avons:

$$\log_e y = 3 \log_e x$$

Donc:

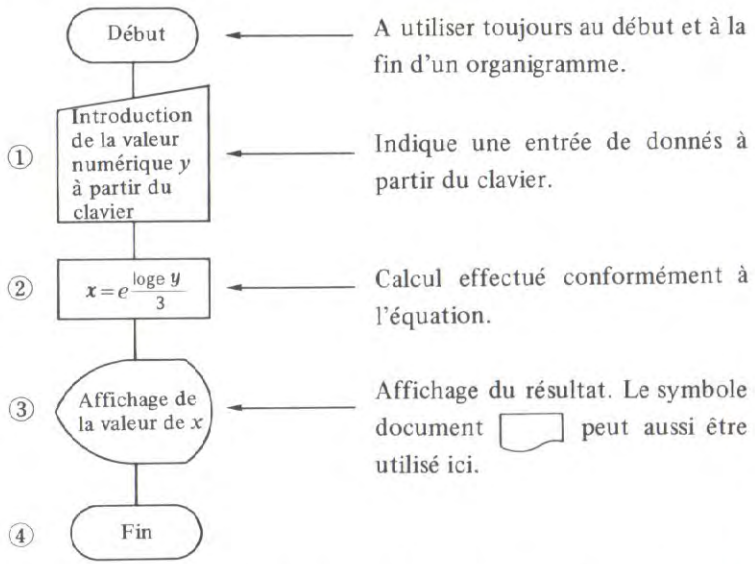
$$\log_e x = \frac{\log_e y}{3}$$

$$x = e^{\frac{\log_e y}{3}}$$

3) Le résultat x du calcul de la racine cubique est affiché.

2. Conception de l'organigramme

Si nous traduisons les résultats de l'analyse ci-dessus sous forme d'organigramme, nous obtiendrons ceci:



A utiliser toujours au début et à la fin d'un organigramme.

Indique une entrée de donnés à partir du clavier.

Calcul effectué conformément à l'équation.

Affichage du résultat. Le symbole document ⌐¬ peut aussi être utilisé ici.

Etant donné qu'il y a quatre traitements, ①, ②, ③ et ④, qui feront l'objet de la programmation, créons un programme pour ces traitements.

3. Programmation

Ecrivons un programme correspondant aux phases 1 à 4 de l'organigramme.

```
① 10 INPUT Y
② 20 X=EXP(LOGY/3)
③ 30 PRINT X
④ 40 END
```

← Introduction d'une valeur numérique à partir du clavier.

← Calculs effectués à partir de l'équation.

← Affichage des résultats calculés.

← Fin de l'exécution.

La programmation permet ainsi de venir à bout du calcul difficile d'une racine cubique en 4 lignes seulement.

Nous allons maintenant développer un peu l'explication de ce programme.

Les numéros 10, 20, 30, et 40 en tête de chaque ligne indiquent la séquence du traitement par l'ordinateur. Normalement, la séquence part du plus petit numéro, comme dans l'exemple ci-dessus, mais il n'est pas forcément nécessaire de commencer par la ligne 10. La séquence peut commencer par n'importe quel chiffre, petit ou grand; il est par contre important que le programme soit écrit en respectant l'ordre de l'organigramme.

Le mot INPUT, à la ligne 10, est une des commandes de BASIC; Il signifie que l'ordinateur est prêt à recevoir une entrée. Dans ce programme, il indique que nous pouvons introduire la valeur y dont la racine cubique doit être extraite.

Le mot PRINT à la ligne 30 est aussi une commande en BASIC. Il commande l'affichage; dans ce programme, il commande l'affichage des résultats du calcul.

Dans le cas de la ligne 20, la méthode formelle d'écriture est la suivante:

```
20 LET X=EXP(LOG Y/3)
```

LET est aussi une commande en BASIC; on l'appelle une instruction d'affectation, car sa fonction est d'affecter le contenu du côté droit à la variable du côté gauche. Etant donné que cette commande n'est pas impérative, elle est en général omise comme dans l'exemple précédent.

Cette instruction est destinée à mettre à exécution un calcul de fonction et EXP à donner la fonction exponentielle e^x de x. LOG donne le logarithme naturel log e^x de x. La barre oblique (/) a ici la signification de ÷. En langage BASIC, des caractères comme X et Y sont appelés des variables et des valeurs numériques comme 3 sont appelées des constantes.

Se reporter au chapitre 5, DETAIL DES COMMANDES ET FONCTIONS, pour la signification et la gamme d'entrée des fonctions.

Non seulement les programmes créés à partir du langage BASIC utilisent des commandes faciles à comprendre, mais encore ils appliquent des équations qui nous servent dans nos activités de tous les jours.

3-4 Les variables

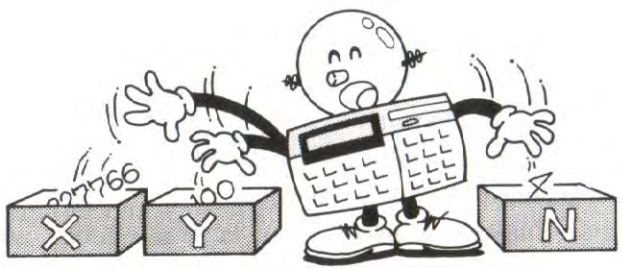
Vous savez maintenant comment créer un programme basé sur un organigramme; nous pouvons donc passer à l'explication des fonctions des caractères X et Y (variables) utilisés dans le programme.
Des équations générales comme

$$x = e^{\frac{\log y}{n}}$$

signifient que les valeurs de droite et celles de gauche sont égales; mais, en langage BASIC, la signification est différente: cela veut dire assigner le contenu du côté droit au côté gauche. L'expression peut être $x = e^{\frac{\log y}{n}}$ mais elle ne peut pas être exprimée par

$$e^{\frac{\log y}{n}} = x$$

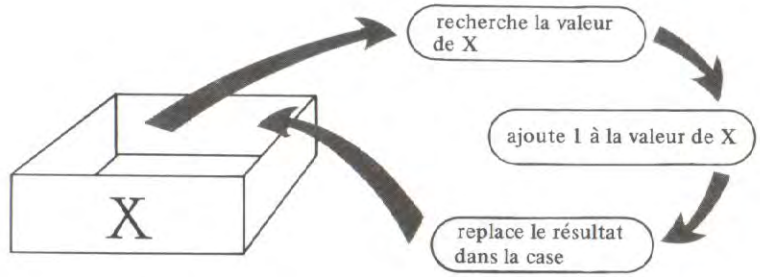
et naturellement des équations telles que 1 = x ne peuvent pas être utilisées. La raison en est que, lorsque des variables telles que X et Y sont utilisées dans des ordinateurs, elles indiquent la place de stockage des valeurs numériques ou des caractères. Afin de mieux comprendre, imaginons simplement que l'ordinateur est équipé de cases marquées X et Y, dans lesquelles sont mises les valeurs numériques.



Il en résulte que, lorsqu'on calcule $x = e^{\frac{\log y}{n}}$, les valeurs numériques qui étaient dans les cases N et Y en sont extraites afin d'être calculées. Le résultat est mis dans la case X. Si nous utilisons cette méthode, l'opération peut en fait être formulée ainsi:

X = X + 1

Bien que ceci ne puisse pas être établi comme une relation d'identité, étant donné que le contenu du côté droit doit être affecté au côté gauche, l'ordinateur accomplira ce qui suit.
1) Addition de 1 à la valeur numérique prise dans la case X.
2) Le résultat est replacé dans la case X.
La méthode qui consiste par exemple à ajouter 1 à la valeur de X est largement utilisée dans les ordinateurs.
La composition de X = X + 1



• Utilisation des variables de tableau numérique

Une variable de tableau est une variable dont la fonction a été étendue. Cette variable de tableau est hautement efficace quand de nombreuses variables sont utilisées dans le même but.
Si nous voulons par exemple traiter des informations pour une période de 1 mois, un maximum de 31 variables sera requis. La préparation du programme devient compliquée si nous essayons d'utiliser des variables comme
A1, A2, A3, . . . , A0, B1, B2, . . . , B0, . . . , C9, D1
aussi nous utilisons simplement une variable de tableau, c'est à dire une variable à laquelle un nombre a été attaché.

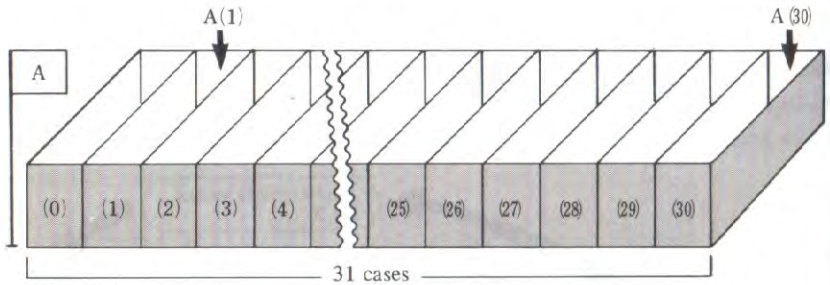
Exemple:

```
A(30)
```

Cette variable peut être utilisée à l'aide de la commande DIM.

```
10 DIM A(30)
```

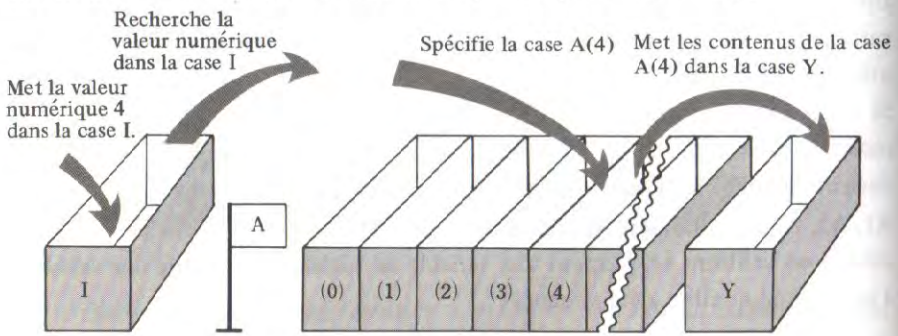
Dans cet exemple, nous allons créer une variable de tableau appelée A, pourvue de 31 cases allant de 0 à 30. Admettons simplement qu'il y a 31 cases appelées A et qu'elles sont différenciées par les numéros (paramètres) allant de 0 à 30 qui leur ont été attribués.



Pour mettre les valeurs numériques dans ces cases ou les en retirer, il suffit de changer les paramètres comme A(0), A(30), etc. La création de programme est simplifiée par la possibilité d'utiliser des variables entre parenthèses à la place des numéros.

Exemple:

Rechercher dans la case du même numéro que lorsque I = 4, Y = A(I).



Vous comprenez sans doute maintenant que l'utilisation des variables de tableau permet de traiter facilement de grandes quantités de données.

Vous trouverez dans l'annexe applications du chapitre 4 des explications plus

approfondies sur les variables, accompagnées d'exemples concrets de programmes.

▪ Différents types de variables

Les variables peuvent être classées en variables numériques qui contiennent des valeurs numériques et en variables à caractères qui contiennent des chaînes de caractères. Jusqu'ici nous n'avons parlé que des variables numériques dans lesquelles des valeurs numériques sont stockées pour les calculs. Nous arrivons maintenant aux variables à caractères, constituées d'une lettre de l'alphabet à laquelle un signe S a été attaché, comme A$ ~ Z$.

Les différents types de variables dont le FX-750P permet l'utilisation sont indiqués ci-dessous.

Variables numériques

- Variables fixes: A, B, etc. (jusqu'à 12 chiffres)
- Variables marquées: A1, X9, etc. (jusqu'à 12 chiffres)
- Variables de tableau:
  - Tableau numérique en demie précision
    A! (10), etc. (jusqu'à 5 chiffres)
  - Tableau numérique en simple précision
    A (10), etc. (jusqu'à 12 chiffres)

Variables à caractères

- Variables fixes: A$, B$, etc. (jusqu'à 7 caractères)
  Exemple: A$ = "FX-750P"
- Variables marquées: A1$, X9$, etc. (jusqu'à 16 caractères)
  Exemple: B1$ = "DEVIATION"
- Variables de tableau: Un nombre de caractères allant de 0 à 79 peut être spécifié.

Variables fixes (A à Z ou A$ à Z$)

Il y a 26 variables numériques fixes allant de A à Z et 26 variables à caractères fixes allant de A$ à Z$. Pour la variable fixe, on doit utiliser exclusivement, soit la case valeur numérique, soit la case chaîne de caractères. Si nous essayons d'utiliser une variable numérique fixe et une variable à caractères fixes ayant toutes deux le même nom (par exemple A et A$), cela provoquera une erreur UV ERROR.

Variables marquées

Les noms de variables à deux caractères peuvent être composés à l'aide des valeurs numériques et des majuscules de l'alphabet. Si toutefois on utilise des variables ayant 3 caractères ou plus (sans compter le $), cela provoquera une erreur SN ERROR en cours d'exécution du programme.

En ce qui concerne les variables marquées, les règles suivantes doivent être respectées.

1. Le premier caractère doit être une lettre majuscule de l'alphabet.
2. On ne peut pas utiliser les mots réservés (IF, ON, TO, PI, etc. voir page 329).
3. Une variable numérique marquée (AB, A1, etc.) peut stocker 12 chiffres pour la mantisse et 2 chiffres pour les exposants.
4. Un maximum de 16 caractères peut être stocké dans la variable à caractères marquée.
5. On peut utiliser un total de 40 variables marquées, y compris les variables de tableau. Toutefois si l'on dépasse cette limite de 40 variables, il en résultera une erreur VA ERROR et le programme sera interrompu. Les variables en surnombre devront alors être annulées à l'aide de la touche CLEAR (voir page 225) ou ERASE (voir page 229).

* Le nom de la variable qui est enregistré peut être confirmé en appuyant sur LIST V ⏎ .

Demie-précision et simple décision

En général des calculs internes sont faits avec une mantisse à 12 chiffres. Lorsque ! est joint à une variable de tableau on obtient une demie-précision (jusqu'à 5 chiffres) qui permet de stocker d'avantages de données dans la mémoire, dans le cas où une précision de calcul à 5 chiffres est suffisante pour le stockage des données. La spécification d'une précision simple requier 8 octets par variable numérique alors que la définition d'une demie-précision ne demande que 4 octets. En d'autres termes, la spécification d'une demi-précision permet d'utiliser efficacement les mémoires limitées.

3-5 Introduction et exécution d'un programme

Les explications données jusqu'ici concernaient les aspects théoriques du programme. Etait-ce difficile? Peut être été un peu pour certains d'entre vous mais, comme on dit, c'est en pratiquant qu'on se perfectionne et vous serez surpris de constater à quel point tout cela est facile, dès que vous vous mettrez à utiliser vraiment l'ordinateur.

Exécutons maintenant sur le FX-750P le programme que nous avions conçu pour extraire une racine cubique. Il nous faudra tout d'abord faire enregistrer le programme par l'ordinateur. En ce qui concerne l'enregistrement des programmes, diverses méthodes ont été développées récemment, comme par exemple l'introduction vocale (reconnaissance de la tonalité) et l'introduction de caractères écrits (reconnaissance des structures), utilisées toutes deux dans le domaine des ordinateurs de grande dimension. Toutefois les ordinateurs personnels utilisent la méthode orthodoxe d'introduction à partir du clavier.

▪ Avant d'introduire le programme

Introduisons le programme de calcul de la racine cubique à partir du clavier. Auparavant, les opérations suivantes sont nécessaires pour préparer l'introduction d'un programme.

Opération:

NEW	NEW_
⏎	READY P0

NEW est une commande destinée à annuler le programme stocké dans la zône de programme à utiliser. Cette opération est celle qu'on exécute en premier, car il est nécessaire que la zône dans laquelle le nouveau programme doit être introduit soit vidée. On doit prendre garde à ne pas confondre NEW et NEW ALL, car NEW ALL annule les programmes et les contenus des variables dans toutes les zônes de programme.

READY P0 à l'affichage indique la zône de programme P0 à utiliser.
Le FX-750P dispose de 10 zônes de programmes numérotées de P0 à P9. Des programmes indépendants peuvent être introduits dans chaque zône de programme. C'est une fonction de division d'une très grande commodité qui permet une utilisation libre de chaque zône dans les limites de la capacité de la mémoire. Il faut cependant que les variables utilisées soient communes à toutes les zônes de programme.

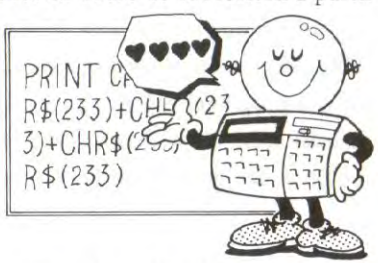
■ Introduction du programme

Opération:

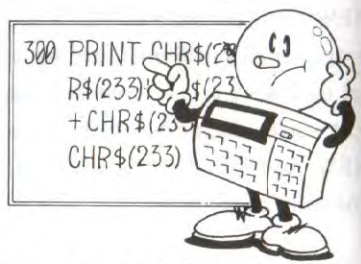
10 [SHIFT] [INPUT] Y	10INPUT Y_
↵	10 INPUT Y

Etant donné que la commande à touche unique par l'intermédiaire de la touche [SHIFT] (par exemple [SHIFT] [INPUT]) a le même résultat que l'introduction de [I][N][P][U][T] à l'aide des touches alphabétiques, dorénavant nous omettrons [SHIFT] et effectuerons les indications seulement avec les lettres alphabétiques.

La touche ↵ sert à stocker le programme dans l'ordinateur au moment choisi. Cette utilisation a une signification en rapport avec le numéro "10" introduit auparavant dans la commande; ce n'est donc pas la même fonction que celle de la touche ↵ utilisée pendant l'exécution de la commande NEW. Le numéro dont nous venons de parler est appelé numéro de ligne, et les programmes sont stockés en ordre de succession à partir des plus petits numéros de lignes.



S'il n'y a pas de numéro de ligne, l'opération sera exécutée immédiatement.



S'il y a un numéro de ligne, elle sera stockée en tant que programme.

Introduisons les lignes qui restent de la même manière.

Opération:

20 X [=] EXP [(] LOG Y [)] 3 [)] ↵	20 X=EXP(LOGY/3)
30 PRINT X ↵	30 PRINT X
40 END ↵	40 END

■ Vérification des programmes d'introduction

Pour visionner les programmes d'introduction, programmer LIST ↵ .
LIST est la commande d' "affichage du programme".
Les programmes peuvent être vérifiés, puisqu'ils seront affichés dans l'ordre de séquence depuis la ligne 10 jusqu'à la ligne 40.
Pour garder l'affichage, appuyer sur la touche ↵ ou sur la touche [ENTER] .
L'affichage restera stationnaire jusqu'à ce qu'on appuie à nouveau sur les touches ↵ ou [ENTER] .

■ Correction du programme

Si, pendant que vous êtes en train de revoir le programme à l'aide de la commande LIST, vous vous apercevez qu'une erreur d'introduction a été commise, mettez en mode EDIT et effectuez la correction. EDIT est une commande destinée à corriger un programme; en appuyant sur EDIT ↵ , vous obtiendrez l'affichage du programme à partir de la première ligne. Dans notre cas, c'est la ligne 10 qui sera affichée.

Opération:

EDIT ↵	10 INPUT Y_

Vous remarquerez que, contrairement au cas de l'affichage par l'intermédiaire de la commande LIST, le curseur clignote au bout de la ligne. La méthode à suivre pour corriger en mode EDIT est la même que celle qui a été expliquée au chapitre 2 pour la fonction d'opération. Déplacez simplement le curseur jusqu'à l'endroit à corriger et corrigez. Annulez (utiliser [DEL]) ou insérez (utiliser [INS]).

Appuyez ensuite sur la touche ↵ ; la ligne corrigée sera écrite et la ligne suivante sera affichée. Si on appuie sur SHIFT ↵ après la correction, la ligne corrigée sera écrite et la ligne précédente affichée. Appuyez sur la touche ↵ pour les lignes qui n'ont pas d'erreur. Appuyez sur BRK pour annuler le mode EDIT.

■ Exécution de programme

Maintenant que nous avons finalement introduit un programme sans erreurs, passons à la vraie utilisation de l'ordinateur et exécutons ce programme. Nous devons pour cela accomplir les opérations suivantes.

Opération:

RUN ↵

```
?
```

RUN est une commande d' "exécution de programme". Cette commande passée, un point d'interrogation (?) apparaît sur l'écran. Cet affichage est une demande d'introduction, étant donné que la commande INPUT (introduction) de la ligne 10 a été exécutée. Introduisons par exemple 64.

Opération:

64 ENTER
(On peut aussi utiliser ↵ . De même par la suite)

```
WAIT
4
```

La racine cubique de 64 est calculée à la ligne 20, et le résultat est affiché à l'aide de la commande PRINT en ligne 30. L'apparition du mot WAIT (attendre) en haut de l'affichage signifie que la commande PRINT a été exécutée et que l'ordinateur est en attente d'une entrée à partir de la touche ↵ ou de la touche ENTER .

Opération:

ENTER

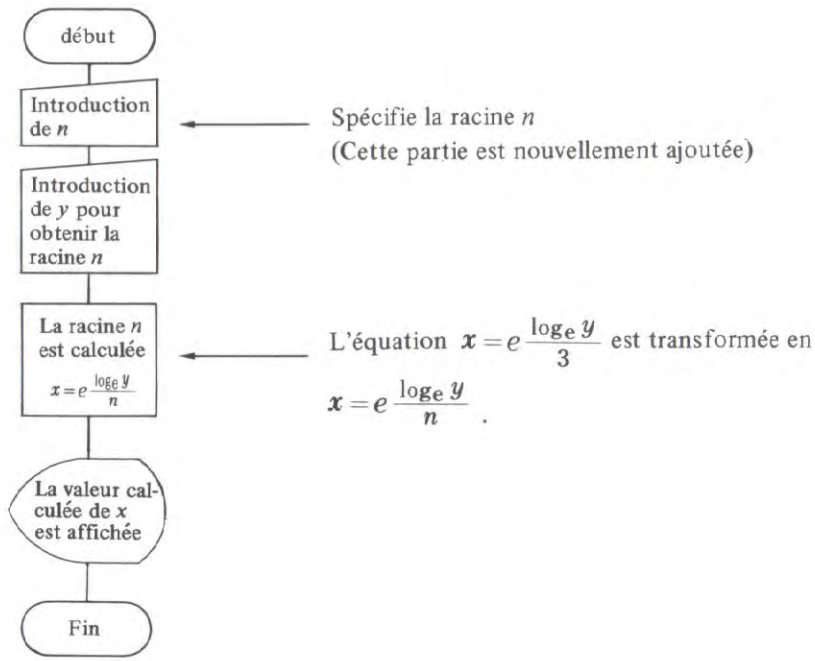
```
READY P0
```

Cette commande est destinée à ramener le programme exécuté à son état originel. Nous suggérons que vous exécutiez quelques programmes supplémentaires en prenant d'autres valeurs.

3-6 Modification de programme

Dans certains cas, il sera nécessaire de modifier le programme, afin qu'il corresponde aux exigences du travail en cours. Puisque le programme actuellement stocké est l'extraction de la racine cubique, modifions le pour obtenir la racine n. Pour ce faire, il suffit de remplacer la constante 3, valable pour la racine cubique, par une variable N et de calculer ensuite la racine n.

Organigramme



Le programme correspondant à cet organigramme sera le suivant.

```
5  INPUT"N=";N
10 INPUT"Y=";Y
20 X=EXP(LOGY/N)
30 PRINT X
40 END
```

C'est ce qu'on appelle une chaîne de caractères; si elle est entre guillemets, elle sera traitée comme un caractère, même s'il s'agit de valeurs numériques.
Lorsqu'on utilise une chaîne de caractères, il faut mettre un point virgule avant la variable.

Revoyons maintenant la partie modifiée.
D'abord, la ligne 5 a été ajoutée. Cela a été fait dans le but d'introduire N pour la racine *n*. Si 5 a été choisi comme numéro de ligne, c'est parcequ'il était nécessaire d'exécuter cette ligne avant la ligne 10 et qu'il fallait donc un numéro plus petit que 10. N'importe quel chiffre compris entre 1 et 9 aurait pu être utilisé.

```
5 INPUT"N=";N
```

"N =" ; a été introduit à la suite de la commande INPUT. C'est une chaîne de caractères destinée à montrer la question posée par l'ordinateur, et N = ? sera affiché pour indiquer que l'ordinateur est en mode d'attente pour des instructions concernant la racine *n*. Si nous mettons une virgule à la place du point virgule, le point d'interrogation (?) ne sera pas affiché.
La partie "Y =" a également été ajoutée à la ligne 10.
Etant donné qu'à la ligne 20, /3 a été changé en / N, le calcul est passé de la racine cubique à la racine *n*.

■ Méthode de modification de programme

1) Ajouter la partie d'introduction de N pour la racine *n*

Opération:

5 INPUT ["]N[=]["][SHIFT][:] N[↵]

```
5 INPUT "N=";N
```

2) Rappeler la ligne 10 à l'aide du mode EDIT

Opération:

EDIT10[↵]

```
10 INPUT Y_
```

3) Amener le curseur à la position Y. Introduire de nouveau "Y =" ; Y.

Opération:

[←]["]Y[=]["][SHIFT][:] Y[↵]

```
20 X=EXP(LOGY/3)_
```

4) Lorsque la ligne 20 apparaîtra à l'affichage, amener le curseur au 3 et introduire N.

Opération:

[←][←]N[↵]

```
30 PRINT X_
```

5) La ligne 30 est affichée mais ne demande aucune modification.
Appuyer sur la touche [BRK] et annuler le mode EDIT.

Opération:

[BRK]

```
READY P0
```

Une fois les modifications effectuées, affichez de nouveau le programme, à l'aide de la commande LIST et vérifiez que les modification ont été correctement exécutées.

■ Exécution du programme modifié

L'exécution du programme modifié se fait par l'intermédiaire de la commande RUN.

Exemple:
Trouver la 4ème racine de 100
Le commentaire N = ?, qui veut dire N de la racine *n*, apparaîtra à l'affichage.

Opération:

RUN[↵]

```
N=?
```

Introduire le chiffre 4, puisque c'est la 4ème racine que nous voulons obtenir.

Opération:

4[ENTER]

```
Y=?
```

Maintenant, introduire 100.

Opération:

100[ENTER]

```
3.16227766
```
WAIT

La 4ème racine de 100 est calculée et affichée. WAIT apparaîtra également en haut de l'affichage.

* Le programme d'obtention de la nème racine vous aidera à vous familiariser avec la programmation. Le calcul manuel ayant recours à " ^ " donne le même résultat.

Exemple: 100^(1/4) [ENTER]

Après avoir examiné les résultats du calcul, appuyez sur la touche [ENTER] ou [↵] et l'exécution du programme sera terminée.

3-7 Précautions à respecter pour les programmes pratiques du chapitre 4

Dans l'annexe APPLICATIONS du chapitre 4, des exemples pratiques sont donnés et le langage BASIC est expliqué à partir d'une utilisation réelle. Comme il est difficile d'introduire de longs programmes sans faire d'erreurs, une liste des opérations d'introduction et des précautions à respecter, et des affichages d'erreurs et de leur correction a également été prévue au chapitre 4 pour vous aider effectuer vos introductions sans problème.

■ Opérations d'introduction et précautions à respecter

1. Spécification de la zône de programme
   Etant donné que le FX-750P est équipé de fonctions de division de programme, commencez par spécifier dans laquelle des 10 zones (P0 ~ P9) l'introduction doit être faite.

   Exemple:
   Spécifier P3 comme zone de programme.

   Opération:

   [SHIFT] [PROG] 3 [↵]

   ```
   READY P3
   ```

2. Videz la zone spécifiée du programme et des variables qu'elle peut contenir.

   Opération:

   NEW[↵] (efface le programme)
   CLEAR[↵] (efface toutes les variables)

3. Pour l'introduction, utilisez la commande à touche unique et assurez vous bien de ne pas oublier la position.
4. Appuyez toujours sur la touche [↵] et stocker après l'introduction de chaque ligne.
5. Si vous remarquez une erreur d'introduction au cours de l'introduction d'un programme, conformez-vous aux indications suivantes.

- Avant d'appuyer sur la touche ↵ .... Servez-vous des touches ⇦ et ⇨ pour amener le curseur à l'endroit qui doit être corrigé; corrigez l'introduction.
- Après avoir appuyé sur la touche ↵ .... Rappelez la ligne à corriger en utilisant EDIT, No. de la ligne, ↵ et déplacer le curseur à l'aide des touches ⇦⇨ pour l'amener jusqu'à l'endroit à corriger. Corrigez l'introduction et appuyez sur la touche ↵ .
- Vous avez oublié d'introduire une ligne .... L'introduire à un endroit qui convient pendant l'opération d'introduction.
- Pour supprimer une ligne entière .... Introduire DELETE No. de la ligne ↵ ou No. de la ligne ↵ .

6. Contrôle de l'impression

```
200 PRINT USING"###
    #.###";R;" OHM"
210 PRINT "X=";
220 PRINT USING"###
    #,###";X;" OHM"
230 END
```

L'espacement est automatiquement inséré (il n'est pas nécessaire d'appuyer sur la touche SPC .)

Appuyer sur la touche SPC pour obtenir un espacement entre guillemets (").

Ne pas confondre le point virgule (;) avec les doubles points (:).

Faire attention aux points (.) et aux virgules (,). Utiliser la touche décimale pour les points.

7. Après avoir effectué l'introduction, contrôlez la liste de programme à l'aide de LIST ↵ .
   Appuyez sur ↵ ou ENTER pour arrêter l'affichage et appuyez à nouveau sur les mêmes touches pour remettre l'affichage. Si, en cours de contrôle, vous remarquez une erreur d'introduction, appuyez sur la touche BRK , rappelez la ligne à corriger à l'aide de EDIT No. de ligne ↵ et, après avoir corrigé l'introduction, appuyez sur la touche ↵ .

■ Affichages d'erreurs et mesures correctives.

Il peut arriver qu'une erreur apparaisse à l'affichage et que l'exécution du programme soit arrêtée, ou bien même si le programme est exécuté, que le résultat obtenu ne soit pas exact. Cela est dû à des erreurs ou défauts insérés dans le programme. La correction de ces erreurs est appelée mise au point.

```
SN ERROR P0-20
```

No. de la ligne dans laquelle l'erreur a été commise

Zône de programme dans laquelle l'erreur a été commise

Type d'erreur

Les affichages d'erreur en cours d'exécution de programme indiqueront le type d'erreur, la zône de programme et le numéro de la ligne.
SN ERROR indique une erreur de syntaxe qui est affichée à la suite d'une erreur de composition dans le programme. 21 erreurs sont répertoriées, y compris OV ERROR et BS ERROR; reportez vous à la liste des messages d'erreur page 324, voyez de quel type d'erreur il s'agit et déterminez la cause.
Introduisez EDIT 20 ↵ pour afficher la ligne à corriger et, après correction, appuyez sur la touche ↵ . En appuyant sur la touche BRK , vous lancerez le mode EDIT et READY P*n* apparaîtra à l'affichage pour permettre la continuation de l'exécution du programme.
Les erreurs qui vont à l'encontre de la grammaire BASIC seront affichées, mais il faut tout de même être très attentif lorsqu'on effectue une introduction, car ERROR n'apparaîtra pas à l'affichage si une expression ou une équation fausses ont été introduites, mais les résultats obtenus seront faux.

CHAPITRE 4

APPLICATIONS

Grâce aux explications et aux notions de langage et de programmation en BASIC qui vous ont été données dans les chapitres précédents, nous pensons que vous avez acquis une bonne connaissance générale des ordinateurs.
Vous trouverez dans ce chapitre des programmes pratiques permettant d'utiliser pleinement les possibilités du FX-750P, ainsi que des commentaires se rapportant aux commandes en BASIC.

4-1 Conversion des devises étrangères

Lorsque nous voyageons à l'étranger, que ce soit pour le tourisme ou pour les affaires, il nous faut souvent convertir des devises, et c'est une tâche ennuyeuse.
Ce programme affiche immédiatement le montant d'une devise spécifiée en son équivalent dans toutes les autres devises, sur simple introduction des taux de change de ces devises. Il peut également afficher le montant de chaque devise étrangère convertie en une devise spécifiée.
Nous allons maintenant convertir entre elles des sommes exprimées en dollar U.S., yen japonais, marks allemands, francs français et livres anglaises.

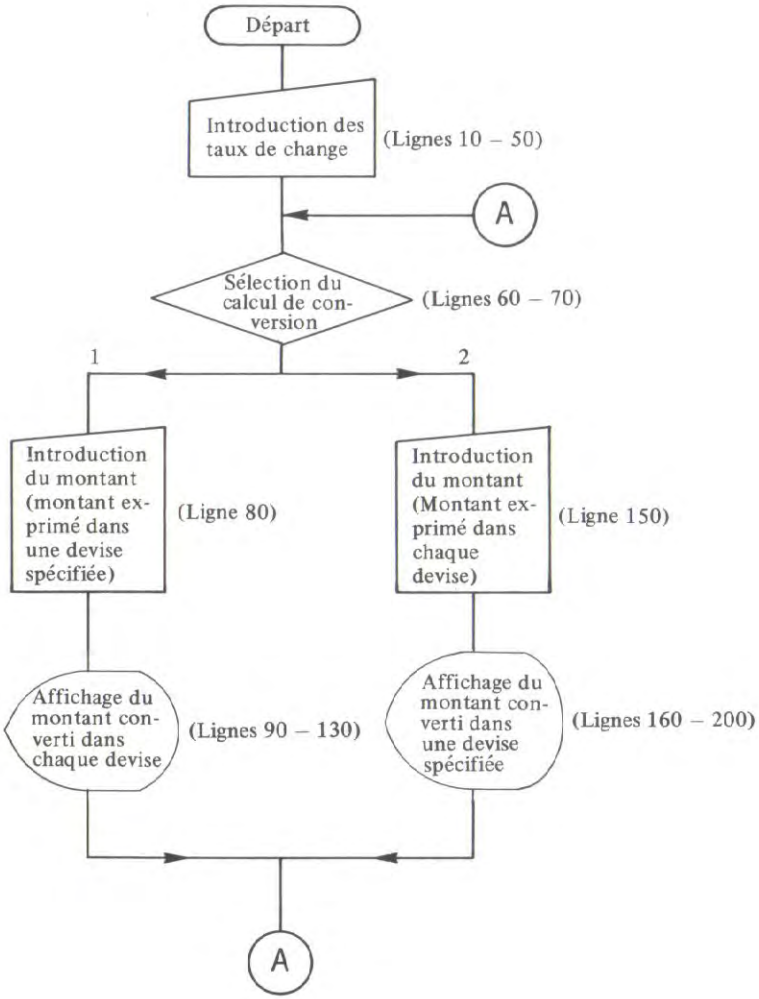
Commandes

PRINT、INPUT、GOTO、IF ~ THEN ~ ELSE

Liste de programme

```
10 INPUT "DOLLAR="
   ;A
20 INPUT "YEN=";B
30 INPUT "MARK=";C
40 INPUT "FRANC=";
   D
50 INPUT "POUND=";
   E
60 INPUT "1 OR 2";
   X
70 IF X=2 THEN 150
80 INPUT "AMOUNT="
   ;Y
90 PRINT Y*A;"DOLL
   AR"
100 PRINT Y*B;"YEN"
110 PRINT Y*C;"MARK
   "
120 PRINT Y*D;"FRAN
   C"
130 PRINT Y*E;"POUN
   D"
140 GOTO 60
150 INPUT "AMOUNT="
   ;Y
160 PRINT Y;"DOLLAR
   ";Y/A
170 PRINT Y;"YEN";Y
   /B
180 PRINT Y;"MARK";
   Y/C
190 PRINT Y;"FRANC"
   ;Y/D
200 PRINT Y;"POUND"
   ;Y/E
210 GOTO 60
```


Organigramme



Introduisez les taux de change d'une devise spécifiée et convertissez en devises de votre pays ou d'un autre pays. Nous utiliserons 1 dollar U.S. comme montant de référence pour nos calculs; ceux-ci n'auront d'ailleurs qu'une valeur approximative, en raison de la fluctuation quotidienne des taux de change. Pour obtenir des conversions précises, il faut donner la valeur 1 à la devise de base et introduire les taux de change du jour.

Introduction des données

Devise de l'unité	Taux de change
Dollar U.S.	$ 1
Yen japonais	¥ 225
Mark allemand	DM2.62
Franc français	FF7.94
Livre anglaise	ST. £.0.66

Opération:

RUN ↵

```
DOLLAR=?
```

Mettre dollar U.S. pour 1, puisque nos conversions seront faites à partir d'un dollar.

1 ↵

```
YEN=?
```

Introduire le taux de change du dollar en yen.

225 ↵

```
MARK=?
```

Introduire le taux de change du dollar en marks.

2.62 ↵

```
FRANC=?
```

Introduire de la même manière les taux de change du franc et de la livre.

0.66 ↵

```
1 OR 2?
```


Toutes les données nécessaires sont maintenant introduites. Supposons que nous avons acheté un article d'une valeur de U.S.$3.675. Choisissez maintenant 1, si vous désirez que le montant de votre achat soit converti dans chacune des autres devises.

1 ↵

```
AMOUNT=?
```

Introduisez le montant en dollars.

3675 ↵

```
3675DOLLAR
```

↵

```
826875YEN
```

Sa conversion en yens japonais apparaitra à l'affichage.

↵

```
9628.5MARK
```

Les montants en francs et en livres seront affichés de la même manière.

↵

```
1 OR 2?
```

Choisissez 2 si vous désirez maintenant connaître la conversion en dollars d'une somme exprimée en chacune des devises.

2 ↵

```
AMOUNT=?
```

La conversion en dollars d'une somme de 80.000 de chacune des devises sera affichée.

80000 ↵

```
80000DOLLAR 80000
```

↵

```
80000YEN 355.5555556
```

↵

```
80000MARK 30534.35115
```

⋮

↵

```
1 OR 2?
```

Etant donné que ce programme n'a pas de fin, appuyez sur BRK pour l'arrêter.

■ Affichons des caractères (commande PRINT)

```
90 PRINT Y*A;"DOLLAR"
```

Commande PRINT Contenus de l'affichage

La commande PRINT à la ligne 90 permet l'affichage du résultat de A multiplié par Y et de DOLLAR.
La commande PRINT peut être utilisée en commande unique, mais normalement on l'utilise avec des expressions numériques et des chaînes de caractères.

Vous trouverez ci-dessous des exemples de diverses méthodes d'utilisation.

Effacement de l'affichage seulement à l'aide d'une instruction unique PRINT.

```
10 PRINT
RUN↵
```

```
READY P0
```

Affichage d'une valeur numérique si elle est précédée d'une instruction PRINT.

```
10 PRINT 18
RUN↵
```

```
 18
```

Affichage d'une chaîne de caractères si elle est mise entre guillements " ".

```
10 PRINT "CASIO"
RUN↵
```

```
CASIO
```

Affichage du résultat de calcul d'une éventuelle expression numérique.

```
10 PRINT 18*21
RUN↵
```

```
 378
```


Les chaînes de caractère seront affichées en utilisant le symbole + comme lien. Servez vous de la touche [SPC] pour insérer un espace.

```
10 PRINT "CASIO "+
   "FX-750P"
RUN↵
```

```
CASIO FX-750P
```

Affichage d'une valeur assignée dans le cas où il existe une variable.

```
10 A=3.19
20 PRINT A
RUN↵
```

```
 3.19
```

Affichage de messages entre guillements " " dans le cas où il existe une variable de caractère.

```
10 A1$="COMPUTER"
20 PRINT A1$
RUN↵
```

```
COMPUTER
```

Lors de l'affichage d'une valeur numérique, elle apparaîtra précédée d'un espace si elle est positive et précédée du signe – si elle est négative.
Si l'instruction PRINT est exécutée, la donnée sera affichée et l'exécution du programme s'arrêtera tandis que WAIT apparaîtra à l'affichage. Si un délai a été instauré à l'aide de la commande WAIT (voir page 252), l'exécution du programme est suspendue conformément à ce qui a été spécifié.
Si aucun délai n'a été instauré, l'exécution du programme est arrêtée, à moins qu'on appuie sur [ENTER] ou ↵.

```
DEG                WAIT
 3.19
```

En cas d'affichage d'une donnée multiple à l'aide d'une instruction unique PRINT, ponctuez cette donnée avec des virgules (,) ou des points-virgules (;).
Si vous employez la virgule, les données seront affichées une par une.

```
10 FOR I=1 TO 3
20 PRINT I,I*5
30 NEXT I
```

RUN↵

1 5
↵
⋮
15
↵

Si vous employez le point-virgule, elles apparaîtront en ligne. Dans ce cas, le point-virgule n'obéira pas à la commande WAIT.

```
10 FOR I=1 TO 3
20 PRINT I;I*5
30 NEXT I
```

RUN↵

1 5
↵
2 10
↵
3 15

La commande PRINT sert à afficher le résultat des calculs et les caractères. (Voir page 241)

■ Introduisons une donnée (commande INPUT)

```
10 INPUT "DOLLAR=";A
```

commande INPUT — Message — Variable

Cette ligne de programme affiche le message DOLLAR = ?, dans l'attente d'une introduction de donnée à partir du clavier par l'intermédiaire de l'instruction INPUT à la ligne 10. La donnée introduite est assignée à la variable A.

La commande INPUT ne peut être utilisée qu'en combinaison avec une variable comme 10 INPUT A. Cependant, étant donné que seul un point d'interrogation sera affiché, une instruction de message est nécessaire pour faciliter la clarification de la donnée à introduire.

```
10 INPUT "AMOUNT="
   ;A
```

L'instruction du message AMOUNT réclame l'introduction du montant.

RUN↵

AMOUNT=?

Dans un tel cas le message et la variable doivent être séparés par un point-virgule ou une virgule.

Ponctuer avec un ; pour afficher un ? après un message. Ponctuer avec une , si vous n'avez pas besoin d'un ?. (Dans le cas présent un message est nécessaire. S'il n'y avait pas de message, cela provoquerait une erreur SN ERROR).

La composition de la donnée et le type de la variable doivent être identiques. C'est à dire que des caractères ne peuvent pas être stockés dans une variable qui contient des valeurs numériques. Si des données de types différents sont introduites, une erreur TM ERROR sera affichée. Appuyez alors sur les touches ↵ ou ENTER pour effectuer une nouvelle entrée. (Remarque: des introductions seront toujours possibles, même en cas d'erreur TM ERROR.)

ABCDE↵

↵

1200↵

TM ERROR
AMOUNT=?
READY P0

Etant donné que des équations de calcul peuvent être utilisées comme introductions de données, les données introduites doivent d'abord être calculées.

Les équations de calcul peuvent être introduites telles qu'elles sont.

```
• 5 B=20
10 INPUT "DATA=";A
20 PRINT A
```

RUN↵

10+25+37*3↵

DATA=?
146

On peut également introduire des fonctions et des variables. Les variables toutefois doivent être préenregistrées.

Exemple: $10+\sqrt{81}+\sin 30^\circ+B=10+9+0.5+20$

```
RUN↵
10+SQR81+SIN30+B↵
```

```
DATA=?
 39.5
```

On peut introduire des données dans des variables multiples par l'intermédiaire de l'instruction INPUT. Dans ce cas les variables sont ponctuées par une virgule. On doit introduire les variables l'une après l'autre.

```
10 INPUT "AMOUNT="
   ;A,B
```

```
RUN↵
100↵
150↵
```

```
AMOUNT=?
?
READY P0
```

Si l'on appuie sur 100, 150 ↵ cela provoquera une erreur TM ERROR.

Quatre variables peuvent aussi être introduites les unes à la suite des autres. Un signe moins (−) est nécessaire pour l'introduction des nombres négatifs.

```
10 INPUT "AMOUNT="
   ;A,B,C,D
20 PRINT "DATA=";A
   ;B;C;D
```

```
RUN↵
4↵
−8↵
 5↵
 7↵
```

```
AMOUNT=?
?
?
?
DATA= 4 -8  5  7
```


Des affichages souples de caractères peuvent être obtenus grâce à la possibilité d'utiliser des variables à caractères comme messages par l'intermédiaire de la commande INPUT.

```
10 A$="(APPLE)"
20 INPUT "PRICE"+A
   $;X
30 A$="(GRAPE)"
40 INPUT "PRICE"+A
   $;Y
50 A$="(PEACH)"
60 INPUT "PRICE"+A
   $;Z
```

Remplacez le nom des produits par le message (PRICE) correspondant.

```
RUN↵
120↵
140↵
250↵
```

```
PRICE(APPLE)?
PRICE(GRAPE)?
PRICE(PEACH)?
READY P0
```

Le nom de l'article peut être changé et affiché en suivant les indications données ci-après.

> La donnée entrée à partir du clavier est assignée à la variable par la commande INPUT. (Voir page 236)

■ Pour modifier le déroulement du programme (Commande GOTO)

```
140 GOTO 60
```

Commande GOTO — Spécifie le numéro de la ligne à laquelle saute le programme.

Quand le programme est exécuté jusqu'à la ligne 140, la commande GOTO le force à sauter à la ligne 60.

L'exécution du programme saute à la ligne 10.

```
10 PRINT "CASIO,";
20 GOTO 10
```

RUN↵

```
CASIO,CASIO,CASIO,CASIO,
```

CASIO apparaîtra sans fin à l'affichage. Pour arrêter l'exécution, appuyez sur [BRK] .

Affiche 1,5 fois le numéro introduit.

```
10 INPUT "AMOUNT="
   ;X
20 Y=X*1.5
30 PRINT Y
40 GOTO 10
```

RUN↵

```
AMOUNT=?
```

100↵

```
 150
```

De cette manière, la commande GOTO peut accomplir le même processus n'importe quel nombre de fois.

Si un numéro de ligne qui n'existe pas dans le programme est spécifié, cela provoquera une erreur UL ERROR.

```
20 GOTO 200
30 GOTO 10
```

RUN↵

```
UL ERROR P0-20
```

La commande GOTO fera augmenter la numérotation et deviendra une source de confusion si la structure du programme lui-même n'a pas été soigneusement examinée lors de sa conception. Ultérieurement, la structure du programme deviendra difficile à comprendre, même pour l'auteur du programme en question.

La commande GOTO sert à faire sauter l'exécution d'un programme à un numéro de ligne spécifié. (Voir page 234)

■ Expression conditionnelle (IF ~ THEN)

```
70 IF X=2 THEN 150
```

Condition — Numéro de ligne auquel sauter (ou numéro de commande à exécuter) lorsque la condition est réalisée.

L'exécution du programme saute à la ligne 150 si 2 est assigné à la variable X de la ligne 70. La ligne suivante sera exécutée si X n'est pas égal à 2. L'instruction IF permet un contrôle conditionnel du déroulement du programme.

Utilisation de l'instruction IF

1) IF (condition) THEN (commande)

```
10 A=20:B=100
20 IF A=20 THEN B=
   B/2
30 PRINT B
```

Si le contenu de A est 20, le contenu de B sera la moitié.

RUN↵

```
 50
```

2) IF (condition) THEN (commande 1) ELSE (commande 2)

```
10 INPUT A
20 IF A<100 THEN P
   RINT "ABC" ELSE
    PRINT "XYZ"
```

Affiche ABC (exécute la commande 1) si A est inférieur à 100 et affiche XYZ (commande 2) si A est supérieur ou égal à 100.

RUN↵

```
?
```

120↵

```
XYZ
```

On peut aussi utiliser un numéro de ligne à la place d'une commande.

```
10 INPUT A
20 IF A<100 THEN 3
   0 ELSE 50
30 PRINT "ABC"
40 END
50 PRINT "XYZ"
60 END
```

Saute à la ligne 30 si A est inférieur à 100 et saute à la ligne 50 si A est supérieur ou égal à 100.

```
RUN↵
?
90↵
ABC
```

La commande de la ligne 50 sera exécutée si une valeur égale ou supérieure à 100 est introduite.

```
140↵
XYZ
```

On peut aussi utiliser ces deux formules en combinaison.

```
10 INPUT A
20 IF A<100 THEN B
   =A*2: GOTO 30 E
   LSE B=A/2: GOTO
    50
30 PRINT B
40 END
50 PRINT B
60 END
```

Si A est plus petit que 100 à la ligne 20, l'exécution sautera à la ligne 30 avec B égal à deux fois la valeur de A.

```
RUN↵
?
68↵
 136
```


Si A est égal ou supérieur à 100, la commande sautera à la ligne 50 avec B égal à la moitié de la valeur de A.

```
RUN↵
?
134↵
67
```

Il existe 9 opérateurs relationnels pour les expressions conditionnelles:

=, <>, ><, >, <, >=, =>, <=, =<

```
10 A=50
20 IF A<>20 THEN P
   RINT "END"
30 END
```

END est affiché si A n'est pas égal à 20.

Conformez vous aux indications suivantes pour programmer les conditions AND (et) et OR (ou).

```
10 A=10:B=20
20 IF A=10 THEN IF
    B=20 THEN PRIN
   T "HELLO!"
```

"HELLO!" sera affiché quand A = 10 et B = 20.

```
10 INPUT A
20 IF A=1 THEN 40
   ELSE IF A=7 THE
   N 40
30 GOTO 10
40 PRINT "GOOD"
```

"GOOD" sera affiché si la valeur d'introduction est 1 ou 7.

IF ~ THEN peuvent être utilisés ensemble de cette manière.

Les commandes IF ~ THEN ~ ELSE prennent des décisions en fonction des contenus d'exécution basés sur les conditions spécifiées. (Voir page 235)

4-2 Calculs de circuits électriques (Calcul de l'impédance)

Lorsqu'on conçoit des circuits électriques, il est difficile de calculer leur impédance. Le programme que nous allons voir maintenant facilite cette tâche, et se déroule de la manière suivante.

1. Il affiche l'impédance et la déclinaison si on introduit la valeur de la résistance et la réactance.
2. Il affiche la valeur de la résistance et la réactance si on introduit l'impédance et la déclinaison.

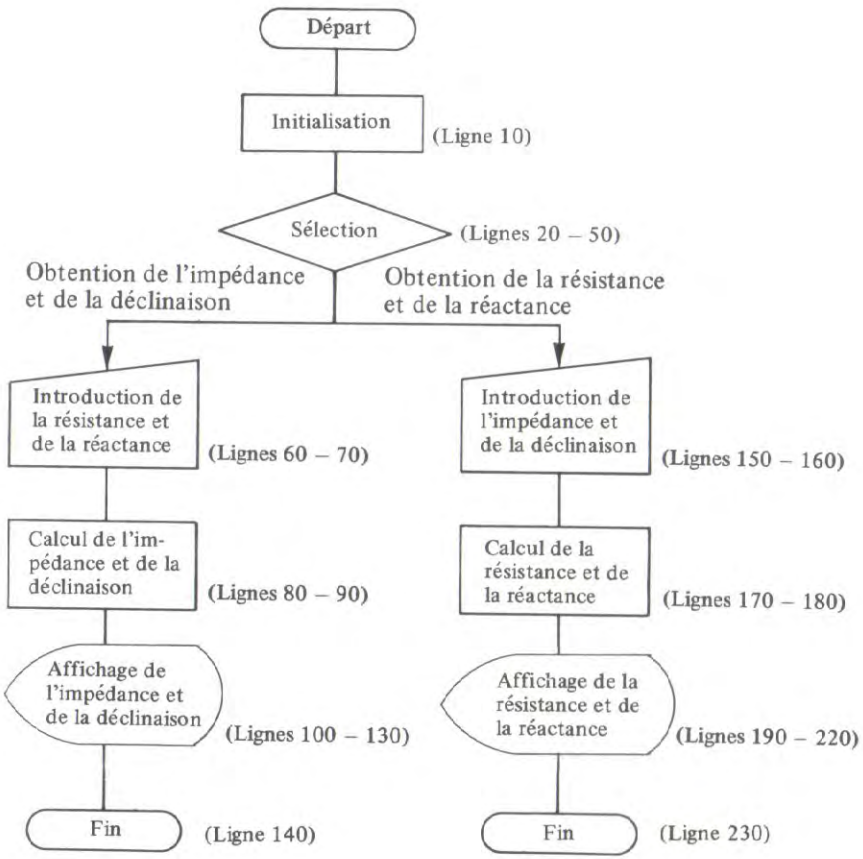
Commandes et fonctions à utiliser

SQR、SIN、ATN、USING、ANGLE、LET

Liste de programme

```
 10 ANGLE 0
 20 INPUT "(1)..Z,S
    (2)..R,X";N
 30 IF N<1 THEN BEE
    P : GOTO 20
 40 IF N>2 THEN BEE
    P : GOTO 20
 50 IF N=2 THEN 150
 60 INPUT "R=";R
 70 INPUT "X=";X
 80 LET Z=SQR(R^2+X
    ^2)
 90 LET S=ATN(X/R)
100 PRINT "Z=";
110 PRINT USING"###
    #.###";Z;" OHM"
120 PRINT "S=";
130 PRINT USING"###
    #.###";S
140 END
150 INPUT "Z=";Z
160 INPUT "S=";S
170 LET R=Z*COSS
180 LET X=Z*SINS
190 PRINT "R=";
200 PRINT USING"###
    #.###";R;" OHM"
210 PRINT "X=";
220 PRINT USING"###
    #.###";X;" OHM"
230 END
```


Organigramme



L'impédance et la déclinaison peuvent être obtenues en introduisant la résistance et la réactance ou, inversement, la résistance et la réactance peuvent être obtenues en introduisant l'impédance et la déclinaison.

Donnée 1	
Résistance	5 Ω
Réactance	15 Ω

Donnée 2	
Impédance	16 Ω
Déclinaison	72°

Opération:

RUN ↵

```
(1)..Z,S (2)..R,X?
```

Obtention de l'impédance et de la déclinaison en premier.

1 ↵

```
R=?
```

La résistance est de 5 Ω.

5 ↵

```
X=?
```

L'impédance sera affichée si on introduit la réactance de 15 Ω.

15 ↵

```
Z=  15.811 OHM
```

Obtention de la déclinaison.

↵

```
S=  71.565
```

↵

```
READY P0
```

RUN ↵

```
(1)..Z,S (2)..R,X?
```

Obtention de la résistance et de la réactance.

2 ↵

```
Z=?
```


L'impédance est de 16 Ω.

16 ↵

```
S=?
```

Affichage de la résistance après introduction d'une déclinaison de 72°.

72 ↵

```
R=   4.944 OHM
```

Affichage de la réactance si on appuie ensuite et à nouveau sur ↵.

↵

```
X=  15.217 OHM
```

■ La fonction SQR

```
80 LET Z=SQR(R^2+X^2)
```

Fonction SQR Argument

La racine carrée de la somme de R^2 et X^2 est obtenue et assignée à la variable Z à la ligne 80. La fonction SQR est équivalente au signe racine carrée ($\sqrt{\ }$) utilisé en mathématiques.

Voici un exemple d'utilisation de la fonction SQR.
Trouver la racine carrée de 25.

```
10 A=25
20 B=SQRA
30 PRINT A;B
```

RUN ↵

```
 25  5
```

En ce qui concerne les fonctions numériques telles que la fonction SQR, si l'argument est une expression numérique, il doit être mis entre parenthèses.

```
10 A=SQR200+25
20 B=SQR(200+25)
30 X=200
40 C=SQRX+25
50 D=SQR(X+25)
60 PRINT "A=";A
70 PRINT "B=";B
80 PRINT "C=";C
90 PRINT "D=";D
```

Aux lignes (lignes 10 et 40) où les expressions numériques ne sont pas entre parenthèses, les calculs de racine carrée ne s'appliqueront qu'à la première valeur numérique ou variable. Pour les expressions numériques mises entre parenthèses, les racines carrées seront extraites après calcul de ces expressions.

```
RUN↵

↵

↵

↵
```

```
A= 39.14213562
B= 15
C= 39.14213562
D= 15
```

Il est impossible d'extraire la racine carrée d'une nombre négatif. Autrement dit, l'expression numérique doit être supérieure ou égale à 0. Si l'on essaye d'obtenir la racine carrée d'un nombre négatif, une erreur MA ERROR apparaîtra à l'affichage.

```
10 A=25
20 B=-25
30 PRINT SQRA
40 PRINT SQRB
```

```
RUN↵

↵
```

```
 5
MA ERROR P0-40
```

La fonction SQR extrait la racine carrée des valeurs numériques (voir page 268).

■ Les fonctions SIN, COS, TAN

```
170 LET R=Z*COSS
                └── Fonction COS
180 LET X=Z*SINS
                └── Fonction SIN
```

La résistance et la réactance sont obtenues respectivement aux lignes 170 et 180 en effectuant les calculs de COS et SIN des valeurs numériques stockées dans la variable S, et en les multipliant par Z. On peut donc utiliser dans un programme des fonctions trigonométriques telles que SIN et COS. Les parenthèses ne sont pas requises pour les SIN, COS et TAN des valeurs numériques si celles-ci sont stockées dans des variables, elles le sont par contre pour l'obtention des SIN, COS et TAN d'expressions numériques.

```
10 X=30:Y=15
20 A=SINX+Y
30 B=SIN(X+Y)
40 PRINT "SIN X+Y=
   ";A
50 PRINT "SIN(X+Y)
   =";B
```

Calculé comme $\mathrm{SIN}\,30^\circ + 15 = 0.5 + 15 = 15.5$

```
RUN↵
```

```
SIN X+Y= 15.5
```

Calculé comme $\mathrm{SIN}\,(30+15)^\circ = \mathrm{SIN}\,45^\circ = 1/\sqrt{2} = 0.7071067812$

```
↵
```

```
SIN(X+Y)= 0.7071067812
```

Les trois types suivants d'unités d'angle peuvent être utilisés dans les fonctions trigonométriques.

1. DEGRE (DEG)
2. RADIAN (RAD)
3. GRADE (GRA)

Ces unités sont spécifiées par la commande ANGLE.

ANGLE 0 . . . . . DEG
ANGLE 1 . . . . . RAD
ANGLE 2 . . . . . GRA

Les valeurs numériques qui peuvent être données doivent être incluses dans les limites suivantes:

DEG . . . . . $-5400° <$ Expression numérique $< 5400°$
RAD . . . . . $-30\pi <$ Expression numérique $< 30\pi$
GRA . . . . . $-6000 <$ Expression numérique < 6000

Si l'on donne des valeurs numériques qui ne sont pas comprises dans ces limites, une erreur BS ERROR apparaîtra à l'affichage.

```
10 ANGLE 0:REM DEG
   REE
20 A=SIN(10000)
30 PRINT A
```

Ce cas excède nettement les limites indiquées de $-5400° <$ Expression numérique $< 5400°$.

RUN↵

```
BS ERROR P0-20
```

La fonction TAN est aussi disponible pour obtenir la tangente x d'une valeur numérique donnée. Si l'angle donné est un multiple impair de $90°$ ($\frac{\pi}{2}$ RAD, 100 GRA), le résultat du calcul sera infini lorsqu'on utilise cette fonction. MA ERROR apparaîtra à l'affichage.

SIN obtient le sinus x d'une valeur numérique donnée.
COS obtient le consinus x d'une valeur numérique donnée.
TAN obtient le tangente x d'une valeur numérique donnée. (Voir page 265)

■ Les fonctions ASN, ACS, ATN

```
90 LET S=ATN(X/R)
         Argument  Fonction ATN
```

La déclinaison est obtenue à la ligne 90 par le calcul de l'arc tangente ($\mathrm{Tan}^{-1}x$) de la valeur numérique obtenue en divisant la réactance (X) par la résistance (R). ATN est une fonction qui sert à obtenir les arcs tangents. On dit que c'est une fonction trigonométrique inverse, car elle calcule des angels à partir de valeurs numériques données, contrairement aux fonctions trigonométriques dans lesquelles des valeurs sont calculées à partir d'angles donnés.

$y = \sin x \longleftrightarrow x = \sin^{-1} y$ (Arc sinus – ASN)
$y = \cos x \longleftrightarrow x = \cos^{-1} y$ (Arc cosinus – ACS)
$y = \tan x \longleftrightarrow x = \tan^{-1} y$ (Arc tangente – ATN)

Dans le cas de ASN et ACS, l'argument est limité à |argument| $\leqq 1$.
Les résultats peuvent être obtenues dans les limites suivantes.

$-90°$ ($-\frac{\pi}{2}$ RAD、 -100 GRA) $\leqq$ ASN $\leqq 90°$ ($\frac{\pi}{2}$ RAD、 100 GRA)
$0°$ (0 RAD、 0 GRA) $\leqq$ ACS $\leqq 180°$ (π RAD、 200 GRA)
$-90°$ ($-\frac{\pi}{2}$ RAD、 -100 GRA) $\leqq$ ATN $\leqq 90°$ ($\frac{\pi}{2}$ RAD、 100 GRA)

ASN sert à obtenir l'arc sinus ($\mathrm{Sin}^{-1}x$) d'une valeur numérique donnée, ACS sert à obtenir l'arc cosinus ($\mathrm{Cos}^{-1}x$) et ATN l'arc tangent ($\mathrm{Tan}^{-1}x$). (Voir page 266).

■ La commande ANGLE

```
10 ANGLE 0
```
Commande ANGLE — Spécifie degré (DEG)

Les trois unités d'angle suivantes peuvent être spécifiées avec la commande ANGLE, lorsqu'on effectue des calculs trigonométriques et des calculs trigonométriques inversés.

ANGLE 0 . . . . . DEGRE (DEG)
ANGLE 1 . . . . . RADIAN (RAD)
ANGLE 2 . . . . . GRADE (GRA)

```
10 REM DEGREE
20 ANGLE 0
30 A=SIN30
40 REM RADIAN
50 ANGLE 1
60 B=SIN30
70 REM GRADIENT
80 ANGLE 2
90 C=SIN30
100 PRINT "SIN30(DE
   G)=";A
110 PRINT "SIN30(RA
   D)=";B
120 PRINT "SIN30(GR
   A)=";C
```

Les résultats seront différents selon l'unité d'angle spécifiée, même pour le même SIN 30 .

```
RUN↵
SIN30(DEG)= 0.5
↵
SIN30(RAD)=-0.9880316241
↵
SIN30(GRA)= 0.4539904997
```

La fonction de la commande ANGLE est de spécifier les unités d'angle (DEG, RAD, GRA). (Voir page 222)

■ Réarrangement de format d'affichage à l'aide de la fonction USING

```
110 PRINT USING"####.###";Z;" OHM"
130 PRINT USING"####.###";S
```
Commande USING — Format d'affichage

USING "####.###" est écrit après la commande PRINT aux lignes 110 et 130. La fonction USING sert ici à réarranger le format pour faciliter la lecture, dans le cas où la valeur à afficher, par exemple Z ou S, comporte jusqu'à 10 chiffres après la virgule décimale. Cette fonction permet d'afficher le nombre désiré de chiffres. Vous trouverez ci-après une comparaison entre l'utilisation et la non-utilisation de la fonction USING.

1) **Lorsque USING n'est pas utilisé.**

```
10 A=SQR15
20 B=SQR30
30 PRINT A;
40 PRINT B
```

```
RUN↵
3.872983346 5.477225575
```

2) **Lorsque USING est utilisé.**

```
10 A=SQR15
20 B=SQR30
30 PRINT USING"##.
   ##";A;
40 PRINT USING"##.
   ##";B
```

```
RUN↵
3.87 5.48
```

USING arrondit donc la valeur affichée au nombre de chiffres spécifié.
Cette fonction peut aussi servir à rendre l'affichage plus facile à lire, comme indiqué ci-dessous.

```
10 A=SQR15
20 B=SQR30
30 PRINT USING"##.
   ##";"A=";A;
40 PRINT USING"##.
   ##";",B=";B
```
→ Le message, A =, est ajouté.

RUN↵

```
A= 3.87,B= 5.48
```

Nous avons toujours les symboles # et . après USING.

```
####.##
```

On appelle ce type de chaîne de caractères une chaîne de caractères de format; on s'en sert pour afficher des valeurs numériques. En plus de ces symboles, le symbole & est aussi utilisé pour l'affichage de caractères.

```
10 A=SQR30
20 PRINT USING"###          ——→ Affiche les valeurs numériques
   .###";A
30 B$="ABCDEFG"
40 PRINT USING"&&&          ——→ Affiche les caractères
   &&";B$
```

L'affichage sera constitué de 3 nombres entiers et de l'emplacement de 3 nombres décimaux.

RUN↵

```
 5.477
```

Affiche des caractères en nombre égal au nombre de symboles &.

↵

```
ABCDE
```

Dans le cas d'un affichage de valeurs multiples avec le même format, la spécification de USING n'est pas obligatoire pour chaque valeur.

```
10 A=SQR15
20 B=SQR40
30 C=SQR90
40 PRINT USING"##.
   ###";A;B;C
```

Les valeurs de A, B et C sont affichées selon une chaîne de caractères de format unique.

RUN↵

```
 3.873 6.325 9.487
```


Pour l'utilisation de la fonction USING, prendre bonne note des points suivants.

1. Si des caractères autres que #, . et & sont utilisés dans la chaîne de caractères de format, cela provoquera une erreur SN ERROR.
2. # et . pour les valeurs numériques et & pour les caractères ne peuvent pas être utilisés dans la même chaîne de caractères de format. S'ils l'étaient, cela provoquerait une erreur SN ERROR.
3. Chaîne de caractères de format pour les valeurs numériques.
   - # . . . . . Spécifie les chiffres de la valeur numérique
   - . . . . . Spécifie la virgule décimale
   - ^ . . . . . Spécifie les exposants

- Un maximum de 13 # peut être spécifié avant la virgule décimale et de 9 # après la virgule décimale 13 # peuvent être spécifiés en tout (virgule décimale comprise), lorsque aussi bien la partie avant la virgule que la partie après la virgule décimale sont désignées.

```
##############.#########
```

Maximum de 13 chiffres avant la virgule décimale — Virgule décimale — Maximum de 9 chiffres après la virgule décimale

Maximum de 13 chiffres lorsqu'on spécifie les deux

- Le signe moins compte comme un chiffre de #.
- ^ est écrit à la fin de la chaîne de caractères de format.
- Au cas où les chiffres situés après la virgule décimale dépasseraient la longueur spécifiée par la chaîne de caractères, le chiffre suivant le chiffre spécifié serait arrondi au nombre entier le plus proche.

```
10 A=PI
20 PRINT A
30 PRINT USING"##.
   ###";A
```

RUN↵

```
 3.141592654
```

↵

```
 3.142
```

- Quand les chiffres situés avant la virgule décimale dépassent la longueur spécifiée par la chaîne de caractères, un signe % est mist au début et la sortie par le format n'est pas possible.

```
10 A=12345.6789
20 PRINT A
30 PRINT USING"###
   ##.##";A
40 PRINT USING"##.
   ####";A
```

```
RUN↵
 12345.6789
↵
 12345.68
↵
% 12345.6789
```

4. Chaînes de caractères de format pour caractères.
   Spécification de position de caractères &.

- On peut écrire autant de symboles & que l'on veut.
- Si le nombre de symboles & est inférieur à la sortie de la chaîne de caractères, les caractères sont affichés à partir du début et en fonction du nombre de symboles &.

```
10 A1$="ABCDEFGHI"
20 PRINT USING"&&&
   &&&";A1$
```

Bien que 9 caractères soient assignés à A1$, étant donné qu'il n'y a que 6 symboles & dans le format, seuls les 6 premiers caractères seront affichés à partir du début.

```
RUN↵
ABCDEF
```

- Si le nombre de symboles & est supérieur à la sortie de la chaîne de caractères, les symboles en surnombre sortent sous forme d'espacements.
- Les chaînes de caractères seront affichées avec justification à gauche.

5. La spécification USING n'est effective que dans une instruction PRINT ou LPRINT.
6. La fonction USING est aussi utilisée pour changer le format.

7. On annule une spécification USING par USING " " ; .

USING est utilisé avec la commande PRINT (ou la commande LPRINT) pour afficher (imprimer) des valeurs numériques et des chaînes de caractères selon des formats spécifiés. (Voir page 263)

▪ La commande LET

```
80 LET Z=SQR(R^2+X^2)
90 LET S=ATN(X/R)
   └─ Commande LET
```

Les programmes utilisent souvent des instructions d'affectation telles que celles de la ligne 80, dans ce cas, la commande LET peut être omise.

```
10 LET A=10
20 LET B=25
30 PRINT A;B
```
Ⓐ (A) et (B) ont la même signification.

```
10 A=10
20 B=25
30 PRINT A;B
```
Ⓑ

Remarque:

Une instruction d'affectation doit être une variable à gauche du signe = et une constante, des variables ou une expression à droite. Une erreur SN ERROR apparaîtra s'il y à une constante ou une expression à gauche.
Une erreur TM ERROR apparaîtra si la correspondance entre les côtés gauche et droit est incorrecte, c'est à dire si des caractères sont assignés à des variables numériques.

La commande LET est placée au début d'une instruction d'affectation pour indiquer qu'une affectation va être effectuée. (Voir page 237)

4-3 Analyse comparative

Lorsqu'on se livre à une étude de marché ou à une analyse des ventes (par domaines et par secteurs), on rassemble des données en plusieurs groupes. La probabilité de trouver deux données identiques étant très réduite, comment allons nous pouvoir établir une tendance globale à partir de ces différents éléments?

Ce programme est prévu pour réaliser des essais significatifs à partir de données introduites sur ces différents groupes d'informations.

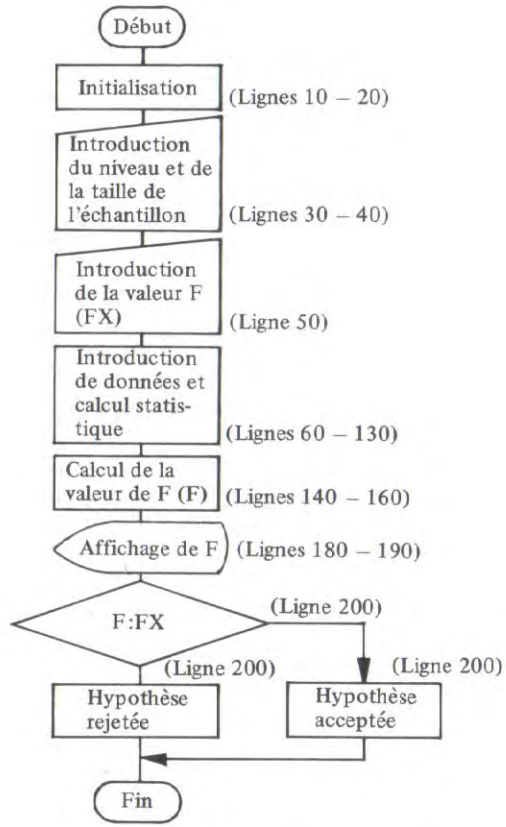
Commandes et fonctions

STAT CLEAR、STAT、SUMX(Y)、SUMX2(Y2)、SDX(Y)、SDXN(YN)

Liste de programme

```
10 CLEAR
20 STAT CLEAR
30 INPUT "LEVEL=";
   A
40 INPUT "SAMPLE S
   IZE=";N
50 INPUT "F=";FX
60 FOR I=1 TO A:B=
   0
70 FOR J=1 TO N
80 INPUT X
90 STAT X
100 B=B+X
110 NEXT J
120 C=C+B^2
130 NEXT I
140 D=SUMX^2/(A*N)
150 E=SUMX2-C/N
160 F=(C/N-D)*(A*N-
    A)/(A-1)/E
180 REM PRINT OUT
190 PRINT "FO=";F
200 IF F>FX THEN PR
    INT "REJECT!" E
    LSE PRINT "ACCE
    PT!"
```


Organigramme



Introduire les données relatives à chaque groupe et rechercher tout rapprochement significatif entre ces groupes.

Introduction des données

Nombre d'échantillons \ Nombre de groupes (niveau)	1	2	3	4
1	4	6	5	6
2	5	3	4	5

$F0.05(\frac{3}{4}) = 6.59$

Opération:

RUN↵

```
LEVEL=?
```

Le nombre de niveaux est nécessaire. Puisque dans notre cas il y a quatre niveaux, introduire 4.

4↵

```
SAMPLE SIZE=?
```

Le nombre d'échantillons est nécessaire. Puisqu'il y en a 2, introduire 2.

2↵

```
F=?
```

Pour la valeur de F0.05 (3/4) introduire 6,59 à partir du diagramme de répartition de F.

6.59↵

```
?
```

Introduire les données de chaque niveau en ordre séquentiel.

4↵

```
?
```

5↵

```
?
```

6↵

```
?
```

⋮

5↵

```
FO= 0.3333333
```

↵

```
ACCEPT!
```

La valeur de F sera calculée et affichée après introduction de la dernière donnée. Etant donné que la valeur de FO est inférieure à celle de F0.05 (3/4), il s'avère que la donnée appartient à la même source (population).

■ Préparation aux calculs statistiques (Commande STAT CLEAR)

```
20 STAT CLEAR
   └── Commande STAT CLEAR
```

A la ligne 20, les préparations sont effectuées pour l'accomplissement des calculs statistiques. La commande CLEAR efface tous les contenus des variables de la ligne 10, mais la commande STAT CLEAR n'efface que la mémoire qui concerne les calculs statistiques, c'est à dire qu'elle efface les statistiques de base (CNT, SUMX, SUMY, SUMXY, SUMX2, SUMY2).

> La commande STAT CLEAR a pour fonction l'initialisation en vue du calcul des statistiques de base (CNT, SUMX, SUMY, SUMXY, SUMX2, SUMY2). (Voir page 249)

■ La commande STAT

```
90 STAT X
   │    └── Expression numérique
Commande STAT
```

Des données sont introduites à la ligne 90. Il n'est pas nécessaire de préparer des programmes compliqués pour obtenir les déviations types, les valeurs estimées et les coefficients de corrélation, puique le FX-750P incorpore des fonctions statistiques. Bien que cette fois ci, une seule donnée variable, X, ait été introduite, on peut aussi introduire des données deux par deux.

```
10 STAT CLEAR
20 FOR I=1 TO 10
30 STAT I
40 NEXT I
50 PRINT "XBAR=";S
   UMX/CNT
```

→ La commande incorporée STAT facilite les additions et les calculs de moyennes en ce qui concerne les nombres de 1 à 10.

RUN↵

```
XBAR= 5.5
```

```
10 STAT CLEAR
20 FOR I=1 TO 5
30 READ X,Y
40 STAT X,Y
50 NEXT I
60 PRINT "XBAR=";S
   UMX/CNT
70 PRINT "YBAR=";S
   UMY/CNT
80 DATA 2,4,3,6,5,
   11,9,20,13,27
```

Elle permet aussi une lecture aisée en 5 ensembles de données X, Y et le calcul de la moyenne des valeurs correspondantes de X et Y.

RUN↵

↵

```
XBAR= 6.4
YBAR= 13.6
```

* Les valeurs moyennes peuvent être obtenues en divisant la somme des données par le nombre de données (SUMX/CNT, SUMY/CNT).

La commande STAT a pour fonction d'introduire les calculs statistiques. (Voir page 248)

■ Les fonctions SUMX et SUMX2

```
140 P=SUMX^2/(A*N)
```
SUMX de la statistique de base

```
150 E=SUMX2-C/N
```
SUMX2 de la statistique de base

Les statistiques de base SUMX et SUMX2 sont utilisées aux lignes 140 et 150 pour calculer la valeur de F.
SUMX obtient la somme des données à variable unique qui sont introduites à l'aide de la commande STAT, et SUMX2 obtient la somme des carrés de ces données.

$$SUMX = \sum_{i=1}^{n} x_i$$

$$SUMX2 = \sum_{i=1}^{n} x_i^2$$

(1)

```
10 CLEAR
20 FOR I=1 TO 10
30 A=A+I:B=B+I^2
40 NEXT I
50 PRINT A;B
```

(2)

```
10 STAT CLEAR
20 FOR I=1 TO 10
30 STAT I
40 NEXT I
50 PRINT SUMX;SUMX
   2
```

Les programmes (1) et (2) obtiennent la somme et la somme des carrés pour les nombres 1 à 10. Le programme (2) utilise les fonctions statistiques.
On se sert de SUMY et SUMY2 pour les données introduites par paires.

(1)

```
10 CLEAR
20 FOR I=1 TO 5
30 READ X,Y
40 A=A+X:B=B+X^2
50 C=C+Y:D=D+Y^2
60 NEXT I
70 PRINT A;B
80 PRINT C;D
90 END
100 DATA 3,5,4,9,2,
    1,6,4,6,9
```

(2)

```
10 STAT CLEAR
20 FOR I=1 TO 5
30 READ X,Y
40 STAT X,Y
50 NEXT I
60 PRINT SUMX;SUMX
   2
70 PRINT SUMY;SUMY
   2
80 END
100 DATA 3,5,4,9,2,
    1,6,4,6,9
```

Dans les programmes (1) et (2) les données X et Y sont lues, et la somme et la somme des carrés sont calculées. Le programme (2) qui utilise les fonctions SUMX et SUMY est nettement plus facile à préparer.

SUMX et SUMX2 ont pour fonction d'obtenir respectivement la somme et la somme des carrés des données à variable unique introduites à l'aide de la commande STAT. (Voir page 275)

* On peut utiliser SUMY et SUMY2 pour les données groupées par paires.

■ Les fonctions SDX(Y) et SDXN(YN)

En plus des sommes et des valeurs moyennes, on utilise souvent les déviations types (déviations types de l'échantillon et deviations types de la population) dans les données statistiques.

```
10 STAT CLEAR
20 READ X
30 IF X=-99 THEN 6
   0                    ——→ Lecture des données.
40 STAT X
50 GOTO 20
60 PRINT "SUM=";SU      ——→ Affichage de la somme.
   MX
70 PRINT "MEAN=";S      ——→ Affichage de la valeur moyenne.
   UMX/CNT
80 PRINT "SD=";SDX      ——→ Affichage de la déviation type de l'échantillon.
90 END
100 DATA 63,87,71,6
    5,63
110 DATA 67,73,63,8
    5,78
120 DATA -99
```

RUN ↵ (somme)	SUM= 715
↵ (valeur moyenne)	MEAN= 71.5
↵ (déviation type de l'échantillon)	SD= 9.107384062

On peut ainsi effectuer sans difficulté les calculs compliqués de déviation type. De même quand des donnés ont été extraites comme échantillon d'un vaste groupe, on peut obtenir immédiatement la déviation type de ce groupe (déviation type de la population).


```
10 STAT CLEAR
20 READ X
30 IF X=-99 THEN 6
   0                    ——→ Lecture des données.
40 STAT X
50 GOTO 20
60 PRINT "SUM=";SU      ——→ Affichage de la somme.
   MX
70 PRINT "MEAN=";S      ——→ Affichage de la valeur moyenne.
   UMX/CNT
80 PRINT "SD=";SDX      ——→ Affichage de la déviation type de l'échantillon.
90 PRINT "SDN=";SD      ——→ Affichage de la déviation type de la population.
   XN
95 END
100 DATA 63,87,71,6
    5,63
110 DATA 67,73,63,8
    5,78
120 DATA -99
```

RUN ↵ (somme)	SUM= 715
↵ (valeur moyenne)	MEAN= 71.5
↵ (déviation type de l'échantillon)	SD= 9.107384062
↵ (déviation type de la population)	SDN= 8.640023148

On peut se servir de SDY et SDYN dans le cas de données groupées par paires.

```
10 STAT CLEAR
20 READ X,Y
30 IF X=-99 THEN 6
   0                    ——→ Lectures des données.
40 STAT X,Y
50 GOTO 20
60 PRINT " SUM(X)=
   ";SUMX
70 PRINT "MEAN(X)=
   ";SUMX/CNT
80 PRINT "  SD(X)=
   ";SDX                ——→ Affichage de la somme, de la valeur moyenne,
90 PRINT " SDN(X)=          de la déviation type de l'échantillon et de la
   ";SDXN                  déviation type de la population en ce qui con-
                           cerne les données X.
```

```
100 PRINT " SUM(Y)=
    ";SUMY
110 PRINT "MEAN(Y)=
    ";SUMY/CNT
120 PRINT "  SD(Y)"
    ;SDY
130 PRINT " SDN(Y)=
    ";SDYN
140 END
200 DATA 929,50,925
    ,59
210 DATA 854,51,818
    ,48
220 DATA 931,51,849
    ,52
230 DATA -99,-99
```

Affichage de la somme, de la valeur moyenne, de la déviation type de l'échantillon et de la déviation type de la population en ce qui concerne les données Y.

RUN ↵	(somme)	SUM(X)= 5306
↵	(valeur moyenne)	MEAN(X)= 884.3333333
↵	(déviation type de l'échantillon)	SD(X)= 49.79022663
↵	(déviation type de la population)	SDN(X)= 45.45205045
↵	(total)	SUM(Y)= 311
↵	(valeur moyenne)	MEAN(Y)= 51.83333333
↵	(déviation type de l'échantillon)	SD(Y)= 3.763863265
↵	(déviation type de la population)	SDN(Y)= 3.435921356

Les fonctions SDX et SDY servent à obtenir la déviation type de l'échantillon des données statistiques. SDXN et SDYN permettent le calcul de la déviation type de la population des données statistiques. (Voir page 277)

4-4 Analyse de régression (régression linéaire)

On fait en général un tracé des données recueillies sur du papier quadrillé pour se faire une idée générale des tendances.
Dans ce programme on suppose que les données recueillies suivent une certaine tendance et on obtient une ligne droite qui représente au plus près ces données.

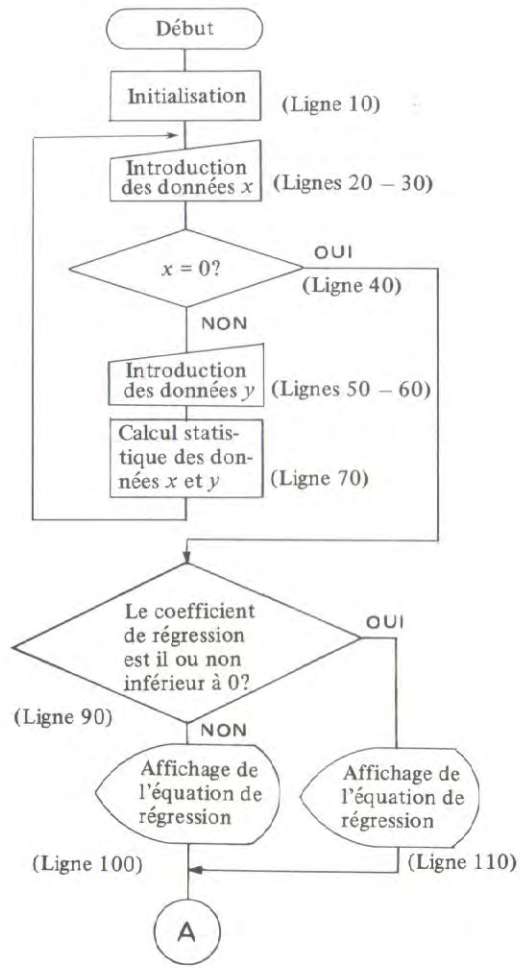
Fonctions

CNT、LRA、LRB、EOX、EOY、COR

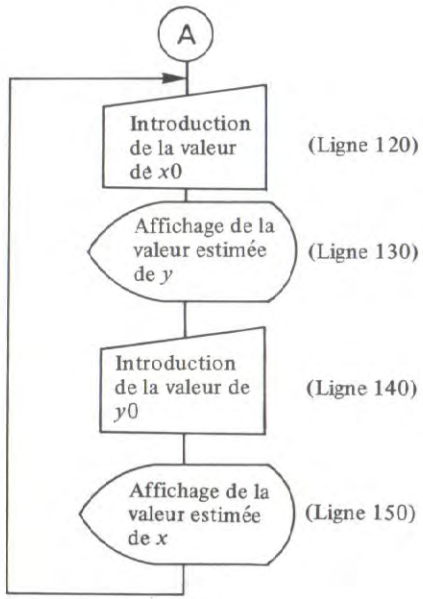
Liste de programme

```
10 STAT CLEAR
20 PRINT "DATA #";
   CNT+1;",X=";
30 INPUT X
40 IF X=0 THEN 90
50 PRINT "DATA #";
   CNT+1;",Y=";
60 INPUT Y
70 STAT X,Y
80 GOTO 20
90 IF LRB(0 THEN 1
   10
100 PRINT "Y=";LRA;
   "+";LRB;"*X": G
   OTO 120
110 PRINT "Y=";LRA;
   LRB;"*X"
120 INPUT "X=";X0
130 PRINT "Y=";EOY(
   X0)
140 INPUT "Y=";Y0
150 PRINT "X=";EOX(
   Y0)
160 GOTO 120
```

Organigramme



Organigramme



Introduire les données x et y par ensembles et obtenir la ligne de régression la plus conforme à la masse des données. Obtenir la valeur estimée de y pour des valeurs données de x et la valeur estimée de x pour des valeurs données de y.

Introduction de données

x	258.2	370.8	442.3	501.6
y	190.2	240.7	303.1	412.8

Opération:

RUN ↵

```
DATA # 1,X=?
```

Introduire les données dans l'ordre à partir de la gauche. Introduire en premier les données x.

258.2 ↵

```
DATA # 1,Y=?
```

Introduire les données y.

190.2 ↵

```
DATA # 2,X=?
```

Introduire les données x et les données y suivantes.

370.8 ↵

```
DATA # 2,Y=?
```

240.7 ↵

```
DATA # 3,X=?
```

⋮

⋮

Introduire les dernières données y.

412.8 ↵

```
DATA # 5,X=?
```

Introduire 0 et l'équation de régression linéaire sera affichée.

```
0 ↵        Y=-54.44575618+ 0.867558
                  ⋮ Déplacements de l'écran.
           44575618+ 0.8675586654*X
```

Obtenir ensuite la valeur estimée.

```
↵          X=?
```

Introduire 100 pour x et la valeur estimée de y sera affichée.

```
100 ↵      Y= 32.31011036
```

Ensuite, introduire 100 pour y afin d'obtenir la valeur estimée de x.

```
↵          Y=?
100 ↵      X= 178.0234148
↵          X=?
⋮
```

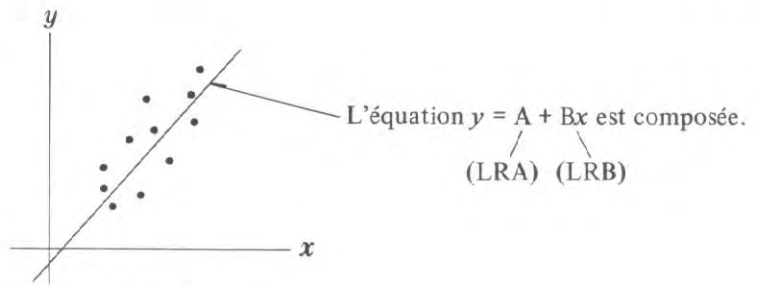
Etant donné que ce programme est sans fin, appuyer sur la touche [BRK] pour le terminer.

■ Les fonctions LRA et LRB

```
100 PRINT "Y=";LRA;"+";LRB;"*X":GOTO120
```

LRA : Obtention de la constante de régression linéaire.
LRB : Obtention du coefficient de régression linéaire.

Une équation de régression linéaire correspondant précisément aux données introduites sera affichée à ligne 100.



Nuage de points (exemple)

On obtient la constante A et le coefficient B dans l'équation de régression linéaire $y = A + Bx$ à l'aide des expressions suivantes.

(Terme constant de l'expression de régression) $A = \dfrac{\Sigma y - \mathrm{LRB} \cdot \Sigma x}{n}$

(Coefficient de l'expression de régression) $B = \dfrac{n\Sigma xy - \Sigma x \cdot \Sigma y}{n\Sigma x^2 - (\Sigma x)^2}$

> Les fonctions LRA et LRB servent à obtenir la ligne droite qui correspond le plus précisément aux données par paires introduites à l'aide de la commande STAT. (Voir page 278)

■ La fonction CNT

```
20 PRINT "DATA #";CNT+1;",X=";
```

CNT : Fonction statistique CNT

La ligne 20 montre l'ordre d'introduction habituel des données x. CNT (n), qui est une statistique de base, augmente le nombre des données statistiques d'une unité à chaque introduction de donnée.
Les avantages de cette fonction sont très appréciables lorsque l'on traite un grand nombre de données, car elle dispense de les compter à l'avance.

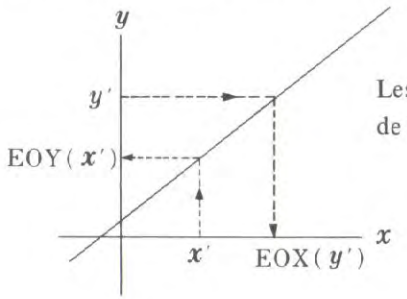
> La fonction CNT a pour fonction de stocker le nombre des données soumises à un traitement statistique. (Voir page 275)

■ Les fonctions EOX et EOY

```
130 PRINT "Y=";EOY(X0)
```
Obtention de la valeur de y pour le x de même coordonnées dans une équation linéaire.

```
150 PRINT "X=";EOX(Y0)
```
Obtention de la valeur de x pour le y de même coordonnées dans une équation linéaire.

La ligne 130 donne la valeur de y pour un x donné dans l'équation linéaire $y = A + Bx$. La ligne 150 donne la valeur de x pour un y donné.



$$y = A + Bx$$

Les équations suivantes donnent les valeurs de EOX et EOY:

$$EOX(yn) = \frac{yn - LRA}{LRB}$$

$$EOY(xn) = LRA + xn \cdot LRB$$

Les fonctions EOX et EOY permettent de calculer la valeur estimée de y pour un x donné et celle de x pour un y donné, à partir d'une ligne de régression obtenue par LRA et LRB. (Voir page 279, 280)

■ La fonction COR

Les coefficients (r) de corrélations deviennent souvent problématiques lorsqu'on cherche une ligne de régression; on peut les exprimer ainsi:

$$r = \frac{n \cdot \Sigma x \cdot y - \Sigma x \cdot \Sigma y}{\sqrt{\{n \cdot \Sigma x^2 - (\Sigma x)^2\}\{n \cdot \Sigma y^2 - (\Sigma y)^2\}}}$$

Cette programmation est extrêmement compliquée. Dans le FX-750P la fonction COR est incorporée, si bien qu'il permet un calcul immédiat du coefficient (r) de corrélation, ce qui évite la composition ingrate de formules de calcul compléxes.


```
10 STAT CLEAR
20 READ X,Y
30 IF X=0 THEN 60
40 STAT X,Y
50 GOTO 20
60 PRINT "COR=";CO
   R
70 END
80 DATA 10,21,15,2
   9,20,42
90 DATA 47,92,12,2
   2,0,0
```

Lecture des données et exécution des calculs statistiques.

Affichage immédiat du coefficient de corrélation.

RUN↵

```
COR= 0.9983837729
```

Le coefficient de corrélation correspondant à ces données sera donc de 0,998 environ.

La fonction COR permet de trouver le coefficient de corrélation entre deux données en paires introduites à l'aide de la commande STAT. (Voir page 279)

4-5 Gestion du budget familial

Si nous examinons nos achats hebdomadaires au super-marché d'un point de vue global, quel genre de marchandises achetons nous le plus souvent? Si nous avons dépensé inconsidérément cette semaine, il nous faudra être un peu plus prudent la semaine prochaine. Il suffit d'introduire le genre des articles et leur prix pour que ce programme calcule et affiche le montant total et le pourcentage de chaque article. Ce programme s'avère aussi pratique quand nous avons besoin d'additionner des données par groupes.

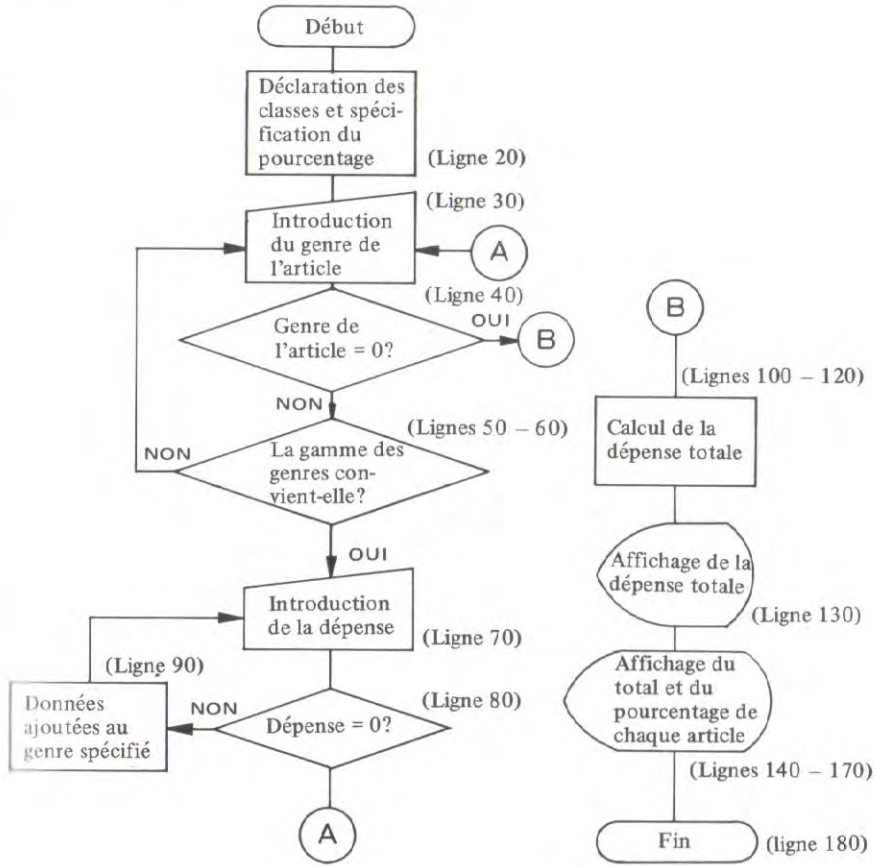
Commandes et fonctions

CLEAR、DIM、CHR$、ROUND、FOR ~ NEXT ~ STEP/NEXT

Liste de programme

```
 5 CLEAR
10 N=5
20 DIM A(N):P$=CHR
   $(37)
30 INPUT "ITEM(0,1
   ,2,3,4,5)=";X
40 IF X=0 THEN 100
50 IF X<0 THEN BEE
   P : GOTO 30
60 IF X>N THEN BEE
   P : GOTO 30
70 INPUT "EXPENSE?
   (END=0)",Y
80 IF Y=0 THEN 30
90 A(X)=A(X)+Y: GO
   TO 70
100 FOR I=1 TO N
110 A(0)=A(0)+A(I)
120 NEXT I
130 PRINT "TOTAL=";
    A(0)
140 FOR I=1 TO N
150 B=ROUND(A(I)/A(
    0)*100,-3)
160 PRINT "NO.";I;"
    =";A(I);"(";B;P
    $;")"
170 NEXT I
180 END
```


Organigramme



Classer les articles achetés par genres et obtenir le pourcentage en introduisant chaque montant. Cinq articles seulement sont ici pris en considération, mais il suffit de remplacer N = 5 par N = 10 à la ligne 10 pour que ce programme puisse servir pour 10 articles.

Introduction des données

Article	Epicerie (1)	Divers (2)	Papèterie (3)	Vêtements (4)	Autres (5)
Montant (Données)	$15	35	21	250	110
	5	18	16	44	85
			2	6	

Opération:

RUN↵

```
ITEM(0,1,2,3,4,5)=?
```

Classer les articles selon leur genre (genre 1 à genre 5).

1↵

```
EXPENSE?(END=0)
```

Le genre 1 concerne l'épicerie dont nous avons aujourd'hui effectué des achats pour $15,00 et $5,00.

15↵

```
EXPENSE?(END=0)
```

5↵

```
EXPENSE?(END=0)
```

Puisqu'il n'y a pas d'autre achat pour le genre 1, introduire 0.

0↵

```
ITEM(0,1,2,3,4,5)=?
```

Spécifier le genre 2.

2↵

Continuer d'introduire les données concernant chaque article de la même manière. Introduire 0 après avoir terminé l'introduction des dernières données.

0↵

```
TOTAL=607
NO.1= 20( 3.29%)
NO.2= 53( 8.73%)
NO.3= 39( 6.43%)
NO.4= 300( 49.42%)
NO.5= 195( 32.13%)
```

Le pourcentage et le montant total de chaque article seront affichés comme indiqué.

▪ Effacement de toutes les données (commande CLEAR)

```
5 CLEAR
```

Commande CLEAR

Tous les contenus stockés dans des variables sont effacés à la ligne 5. Etant donné que la commande CLEAR efface toutes les données stockées, ne l'utilisez pas lorsque vous exécutez un programme basé sur des données stockées.
Le programme suivant obtient le montant total des données introduites.

```
10 CLEAR
20 INPUT "DATA=";X
30 IF X=0 THEN 50
40 Y=Y+X: GOTO 20
50 PRINT "TOTAL=";
   Y
```

Exécuter d'abord ce programme.

Introduire 0 pour avoir le montant total.

RUN↵	DATA=?
10↵	DATA=?
20↵	DATA=?
0↵	TOTAL=30

Ensuite effacer la ligne 10.

```
20 INPUT "DATA=";X
30 IF X=0 THEN 50
40 Y=Y+X: GOTO 20
50 PRINT "TOTAL=";
   Y
```

On ne pourra pas obtenir les résultats corrects des calculs effectués si l'instruction CLEAR est annulée à la ligne 10. Ceci est dû à ce que la valeur numérique 30 a déja été assignée à Y au cours d'un calcul précédent.

RUN↵	DATA=?
10↵	DATA=?
20↵	DATA=?
0↵	TOTAL= 60

Si la commande CLEAR est exécutée pendant la boucle FOR ~ NEXT, une erreur FO ERROR apparaitra à l'affichage; en effet la pile de la boucle FOR (emplacement de retour) sera aussi effacée et la boucle FOR ~ NEXT ne pourra pas être maintenue.

```
10 FOR I=1 TO 10
20 PRINT I
30 CLEAR
40 NEXT I
```

RUN↵	1
↵	FO ERROR P0-40

La commande CLEAR efface toutes les valeurs stockées dans les variables. (Voir page 225)

■ Qu'est-ce qu'un tableau? (DIM)

```
20 DIM A(N):P$=CHR$(37)
```

Commande DIM — Indice — Variable de tableau

La commande DIM déclare les variables de tableau comme à la ligne 20; elle peut déclarer des variables multiples en une seule fois.

Exemple: `DIM A(5),B(7),C(30)`

1) Indice

A la ligne 20, DIM déclare l'utilisation du tableau à une dimension appelé A. L'indice aura N comme valeur maximale (dans ce cas ce sera 5, puisque 5 a été assigné à N à la ligne 10 (voir page 136)). Cela veut dire que les 6 cases de A(0), A(1), A(2), A(3), A(4) et A(5) ont été préparées à la ligne 20, puisqu'en BASIC les indices partent de 0. On peut aussi utiliser des variables comme indices de variables de tableau en plus des valeurs numériques.

Cependant, si la variables comprend une ou des fractions, cette ou ces fractions seront automatiquement ignorées.

Exemple:

Si les données stockées dans l'indice 3 de la variable de tableau A sont affectées à X, cela peut être exprimé de la manière suivante:

`X=A(3)`, `I=3:X=A(I)` et `J=3.5:X=A(J)`

Ces trois types de programmation aboutissent à la même instruction de mémoire.

Si l'indice spécifié dépasse la gamme d'indices de la variable de tableau déclarée par la commande DIM, cela provoquera une erreur UV ERROR.

```
10 DIM A(5)
20 A(0)=0
30 A(1)=10
40 A(2)=20
```

→ La gamme des indices de la commande DIM est la suivante: 0 ≦ indice ≦ 5.

```
50 A(3)=30
60 A(4)=40
70 A(5)=50
80 A(6)=60
```

La gamme spécifiée des indices est dépassée à la ligne 10, puisque 6 a été spécifié.

```
RUN
UV ERROR P0-80
```

En outre, la valeur maximale de l'indice qu'on peut déclarer à l'aide de la commande DIM est de 255. Les indices qui dépassent cette valeur ($0 \leq$ indice < 256) ne peuvent pas être utilisés.

2) Dimension

On peut spécifier jusqu'à deux indices pour une variable de tableau en ponctuant avec une virgule (,). C'est ce qu'on appelle un tableau à deux dimensions.

Exemple:

tableau à une dimension

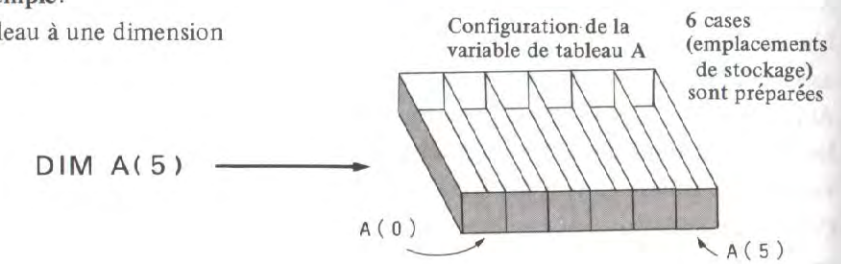
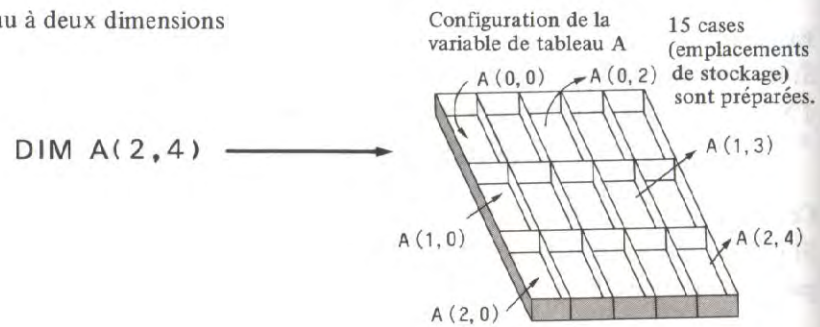


tableau à deux dimensions



3) Noms des variables de tableau

Pour les noms des variables, on peut utiliser toutes les lettres majuscules de l'alphabet, de A jusqu'à Z. A la suite de ces lettres, on peut ajouter les symboles $ et ! lorsqu'on spécifie des tableaux de caractères ou des tableaux de demie-précision.

Tableau en simple précision

A(i) ~ Z!(i) Précision: mantisse de 12 chiffres et exposant de 2 chiffres
A(i, j) ~ Z!(i, j) Mémoire: 8 octets pour chaque

Tableau en demie précision

A!(i) ~ Z!(i, j) Précision: Mantisse de 5 chiffres et exposant de 2 chiffres
A!(i, j) ~ Z!(i, j) Les caractères qui dépassent la mantisse de 6 chiffres seront ignorés lors de l'exécution des instructions d'affectation.

Mémoire: 4 octets pour chaque. On peut économiser la capacité de stockage si 5 chiffres ou moins sont suffisants.

Tableau de caractères

A$(i) ~ Z$(i) Capacité: Peut stocker jusqu'à 16 caractères pour chaque.
A$(i, j) ~ Z$(i, j) Le nombre de caractères qui peut être spécifié en vue de stockage doit être compris dans les limites suivantes: $0 \leq$ nombre de caractères < 80

Mémoire: (Nombre de caractères stockés + 1) octets

Exemple:

Lorsqu'on désire stocker un maximum de 20 caractères:

```
DIM A$(5)*20
```

On ne peut pas utiliser ensemble A(i) et A(i, j), A!(i) et A!(i, j), A$(i) et A$(i, j). De même lorsque A(i) a été déclaré, on ne peut plus déclarer A(j). Sinon, dans tous ces cas, une erreur DD ERROR (déclaration en double) apparaitra à l'affichage.

```
10 DIM A(10)
20 A(0)=12:A(1)=35
   :A(2)=27
30 DIM A(3,4)
40 A(0,1)=75:A(1,1
   )=19:A(2,1)=34
```

→ Le tableau A(i) a été déclaré.

→ A(i,j) du même nom de tableau a été déclaré.

```
RUN↵
DD ERROR P0-30
```

```
10 DIM A(10)
20 FOR I=0 TO 10
30 A(I)=I*I
40 NEXT I
50 DIM A(20)
60 FOR I=0 TO 20
70 A(I)=I*I
80 NEXT I
```

→ Le tableau A(i) a été déclaré.

→ Le tableau A(i) a été déclaré à nouveau car on a besoin d'un tableau légèrement plus grand.

```
RUN↵
DD ERROR P0-50
```

Le même tableau ne peut pas être défini à nouveau. S'il doit l'être, servez vous d'abord de la commande CLEAR ou ERASE. Faîtes attention lorsque vous utilisez ces commandes, en effet la commande CLEAR effacera toutes les valeurs stockées dans les variables et la commande ERASE annulera toutes les données stockées dans les variables spécifiées.

CLEAR – efface toutes les données stockées.

ERASE A – annule toutes les données stockées dans la variable de tableau A. (Voir page 229)

```
10 DIM A(3),B(5)
20 FOR I=0 TO 2
30 A(I)=I+100:B(I)
   =I+200
40 NEXT I
50 FOR I=0 TO 2
60 PRINT A(I);B(I)
70 NEXT I
80 ERASE A
90 DIM A(5)
100 FOR I=1 TO 2
110 A(I)=I+300
```

→ Déclare les tableaux A et B.

→ Efface le tableau A car un tableau A plus grand est nécessaire.

→ Déclare à nouveau le tableau A.


```
120 NEXT I
130 FOR I=0 TO 2
140 PRINT A(I);B(I)
150 NEXT I
```

Les valeurs suivantes sont stockées dans les tableaux A et B.

```
RUN↵
 A   B
100 200
101 201
102 202
```

En raison de l'annulation opérée par la commande ERASE, le tableau A (0) ne comporte aucune donnée.

```
↵
0 200
```

Les valeurs numériques des tableaux A(1) et A(2) sont affichées.

```
↵
301 201
↵
302 202
```

La commande DIM a pour fonction de déclarer la dimension et les indices maximum d'une variable de tableau. Cette déclaration est toujours demandée avant d'utiliser la variable de tableau. (Voir page 227)

▪ La fonction CHR$

```
20 DIM A(N):P$=CHR$(37)
```

Fonction CHR$ — Code de caractères

A la ligne 20, le 37ème caractère du code de caractères est assigné à P$ pour afficher le symbole %.

```
10 PRINT CHR$(37)
20 PRINT CHR$(39)
30 PRINT CHR$(64)
```

→ %

→ Guillemet simple

→ Symbole de prix de l'unité

```
40 PRINT CHR$(91)  ——→ Parenthèse gauche
50 PRINT CHR$(93)  ——→ Parenthèse droite
60 PRINT CHR$(95)  ——→ Curseur (soulignement)
70 PRINT CHR$(135) ——→ Curseur (carré)
```

```
RUN↵
↵
↵
↵
↵
↵
↵
```

Des caractères qui ne figurent pas au clavier peuvent ainsi être affichés grâce à la fonction CHR$, pourvu qu'ils soient définis au tableau des codes de caractères. (Voir page 328)

Avec la fonction CHR$, on peut utiliser des valeurs numériques comprises dans les limites suivantes: $0 \leqq$ valeur numérique < 256.

La fonction CHR$ opère la conversion entre une valeur numérique donnée et le caractère qui lui correspond. (Voir page 253)

*On trouvera la liste de ces valeurs numériques et des caractères correspondants dans le code à usage international ASCII (Code Américain Standard d'Echange d'Informations).

■ La fonction ROUND

```
150 B=ROUND(A(1)/A(0)*100,-3)
```

Fonction ROUND — Nombre à arrondir — Position de chiffre

A la ligne 150, le total de chaque classe est divisé par le total général et puis multiplié par 100. Le pourcentage de chaque article par rapport au total est ensuite affiché.

Une valeur numérique est arrondie à la troisième décimale.

Comme il existe un certain nombre d'autres manières d'utiliser la commande ROUND, nous allons en expliquer quelques unes à l'aide de programmes simples.

```
10 A=1234567
20 B=4.56789
30 C=ROUND(A,2)
40 D=ROUND(B,-2)
50 PRINT C
60 PRINT D
```

Les valeurs numériques 1234567 et 4,56789 sont respectivement affectées à A et B. La valeur numérique de A est arrondie à la position 10^2 et le résultat est affecté à C. La valeur numérique de B est arrondie à la position 10^{-2} (position 1/100) et le résultat est affecté à D.

L'affichage donnera donc des valeurs numériques arrondies respectivement à la position des centaines et des centièmes.

```
RUN↵
↵
```

```
1235000
4.6
```

Lorsque ROUND (x,y) est spécifié de cette façon, les résultats du calcul de la valeur numérique ou de l'expression numérique donnée seront arrondis à la position 10^y.

```
10 A=123456789
20 B=ROUND(A,3,0)
30 C=ROUND(A,3,1)
40 PRINT B
50 PRINT C
```

La valeur numérique 123456789 est affectée à A. la valeur numérique de A sera arrondie à la position 10^3 et le résultat affecté à B. Le troisième chiffre significatif de la valeur numérique de A sera aussi arrondi et le résultat affecté à C. B est affiché avec sa valeur numérique arrondie à la position 10^3 et C est affiché avec le chiffre significatif arrondi à 2 chiffres.

```
RUN↵                    123460000
↵                       120000000
```

Lorsqu'on utilise ROUND (x,y,0 ou 1) de cette manière, x est arrondi à la position 10^y de x dans le cas de 0 et le yème chiffre significatif à partir de la gauche sera affiché pour x dans le cas de 1.

(1)

```
10 A=123456789
20 B=ROUND(A,3,0,+
   )
30 C=ROUND(A,3,0,-
   )
40 D=ROUND(A,3,0)
50 PRINT A
60 PRINT B
70 PRINT C
80 PRINT D
```

(2)

```
10 A=123456789
20 B=ROUND(A,3,1,+
   )
30 C=ROUND(A,3,1,-
   )
40 D=ROUND(A,3,1)
50 PRINT A
60 PRINT B
70 PRINT C
80 PRINT D
```

Arrondi au chiffre spécifié si le signe est +.

Ecarté au chiffre spécifié si le signe est −.

(1)

La valeur numérique 123456789 est affectée à A.

Arrondir la valeur numérique de A à la position 10^3 et affecter le résultat à B.

Ecarter aussi la valeur numérique de A à la position 10^3 et affecter le résultat à C.

Arrondir ensuite la valeur numérique de A à la position 10^3 et affecter le résultat à D.

```
RUN↵                    123456789
↵                       123460000
↵                       123450000
↵                       123460000
```


(2)

Le traitement, ici, sera effectué au $y$ème chiffre significatif, tandisqu'il le sera à la position 10^y dans le cas de (1).

```
RUN↵                    123456789
↵                       130000000
↵                       120000000
↵                       120000000
```

Remarque:

lorsque l'on spécifie des chiffres destinés à être arrondis, leur valeur numérique doit être conforme à: lvaleur de spécification du chiffrel < 100.

BS ERROR sera affiché si l'on spécifie une valeur supérieure à 100.

La fonction ROUND d'arrondit une valeur numérique à un chiffre spécifié.
(Voir page 272)

- **Répétition d'une opération (commande FOR ~ NEXT)**

```
100 FOR I=1 TO N
110 A(0)=A(0)+A(I)
120 NEXT I
```

L'exécution du traitement A(0) = A(0) + A(I) est répétée (N fois) jusqu'à ce que la valeur de I dépasse N aux lignes 100 ~ 120.

De la même manière, l'exécution des instructions des lignes 150 et 160 se répète aux lignes 140 à 170 jusqu'à ce que la valeur de I passe de 1 à N (N fois).

```
140 FOR I=1 TO N
150 B=ROUND(A(I)/A(0)*100,-3)
160 PRINT "NO";I;"=";A(I);"(";B;P$;")"
170 NEXT I
```

On a souvent recours à cette méthode de répétition d'une exécution un nombre déterminé de fois lors de la création d'un programme.

La commande FOR ~ NEXT a généralement le format suivant.

FOR variable de contrôle = valeur initiale TO valeur finale STEP incrément
NEXT variable de contrôle

La valeur de la variable de contrôle est d'abord établie à sa valeur initiale et l'incrément spécifié à l'aide de STEP est ensuite ajouté à la variable de contrôle.
La commande située entre FOR et NEXT sera alors exécutée si la valeur de la variable de contrôle est inférieure à la valeur finale. (c'est ce qu'on appelle une boucle.)
L'exécution sautera à l'instruction suivante de NEXT dans le cas où la valeur de la variable de contrôle dépasse la valeur finale. (Cela veut dire que la boucle est achevée.)
Si STEP est omis l'incrément sera automatiquement établi à 1.

```
10 FOR I=1 TO 10 S
   TEP 2
20 PRINT "*";
30 NEXT I
```

Si l'incrément est établi à 2, la valeur de I sera modifiée par étapes de 2, comme par exemple 1,3,5,7,9 et la boucle sera achevée à 11, puisque ce nombre dépasse la valeur finale de 10. Cinq symboles ∗ seront alors affichés.

RUN↵

```
*****
```

```
10 FOR I=1 TO 10
20 PRINT "*";
30 NEXT I
```

Si l'incrément est omis, la valeur de I avance par étapes simples, comme par exemple 1,2,3,4,5 ... et la boucle sera achevée à 11 puique ce nombre est plus grand que la valeur finale de 10. 10 symboles ∗ seront affichés dans ce cas.

RUN↵

```
**********
```


La valeur initiale, la valeur finale et l'incrément peuvent être des nombres négatifs ou des fractions, mais on doit faire attention à la taille.
C'est à dire que la boucle est exécutée une seule fois quand l'incrément est positif, même si la valeur finale est inférieure à la valeur initiale ou négative, même si la valeur finale est supérieure à la valeur initiale.

```
10 FOR I=1 TO 5
20 FOR J=1 TO 3
30 PRINT I*J;
40 NEXT J
50 PRINT ""
60 NEXT I
```

On utilise souvent les commandes FOR ~ TO ~ STEP ~ NEXT en emboîtement.
Ce qu'on appelle emboîtement, c'est la spécification de boucles multiples FOR ~ NEXT.
Dans ce cas les opérations se dérouleront comme suit:
Lorsque I a pour valeur 1, J = 1,2,3 I∗J = 1,2,3
Lorsque I a pour valeur 2, J = 1,2,3 I∗J = 2,4,6
Lorsque I a pour valeur 3, J = 1,2,3 I∗J = 3,6,9
Lorsque I a pour valeur 4, J = 1,2,3 I∗J = 4,8,12
Lorsque I a pour valeur 5, J = 1,2,3 I∗J = 5,10,15

```
10 FOR I=1 TO 5
20 FOR J=1 TO 3
30 PRINT "I=";I;",
   J=";J
40 NEXT I
50 NEXT J
```

On doit faire attention lorsqu'on positionne les commandes FOR et NEXT. Si une position incorrecte est spécifiée, FO ERROR sera affiché.

RUN↵

```
I= 1, J= 1
      ⋮
I= 5, J= 1
FO ERROR P0-50
```

L'emboîtement de boucles FOR ~ NEXT est possible jusqu'à 6 niveau. Si un nombre de niveaux d'emboîtement supérieur à 6 est spécifié, NO ERROR sera affiché.

La commande FOR ~ TO ~ STEP ~ NEXT répète l'exécution du programme un nombre spécifié de fois entre les commandes FOR et NEXT. (Voir page 230)

4-6 Emploi du temps hebdomadaire

Il n'est pas rare d'entendre quelqu'un raconter qu'il a brusquement été réveillé au beau milieu de sa sieste par un client qui lui téléphonait parcequ'il avait oublié l'heure de son rendez-vous. Il y a de bonnes chances pour que vous ayez, vous aussi, fait cette expérience au moins une fois ou deux.
Ce programme est destiné à aider ceux qui cherchent une méthode rapide leur permettant d'établir un emploi du temps, même pour la semaine en cours.

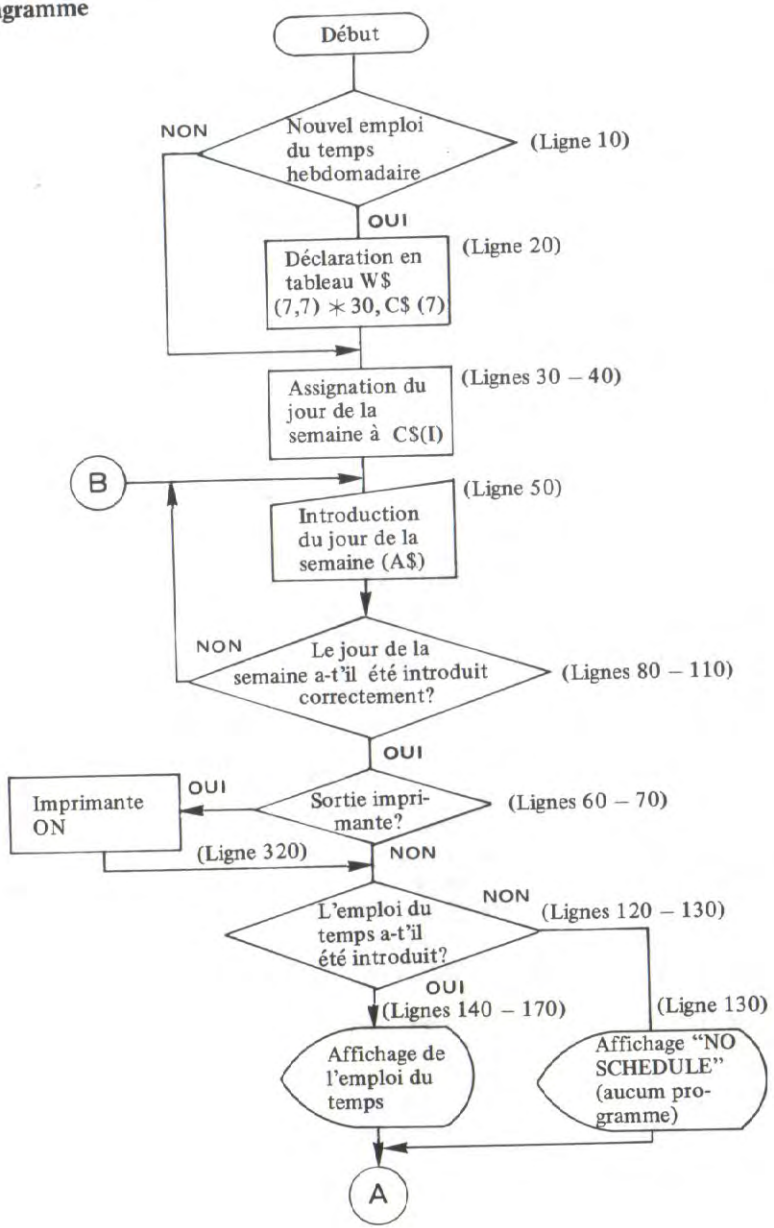
Commandes et fonctions

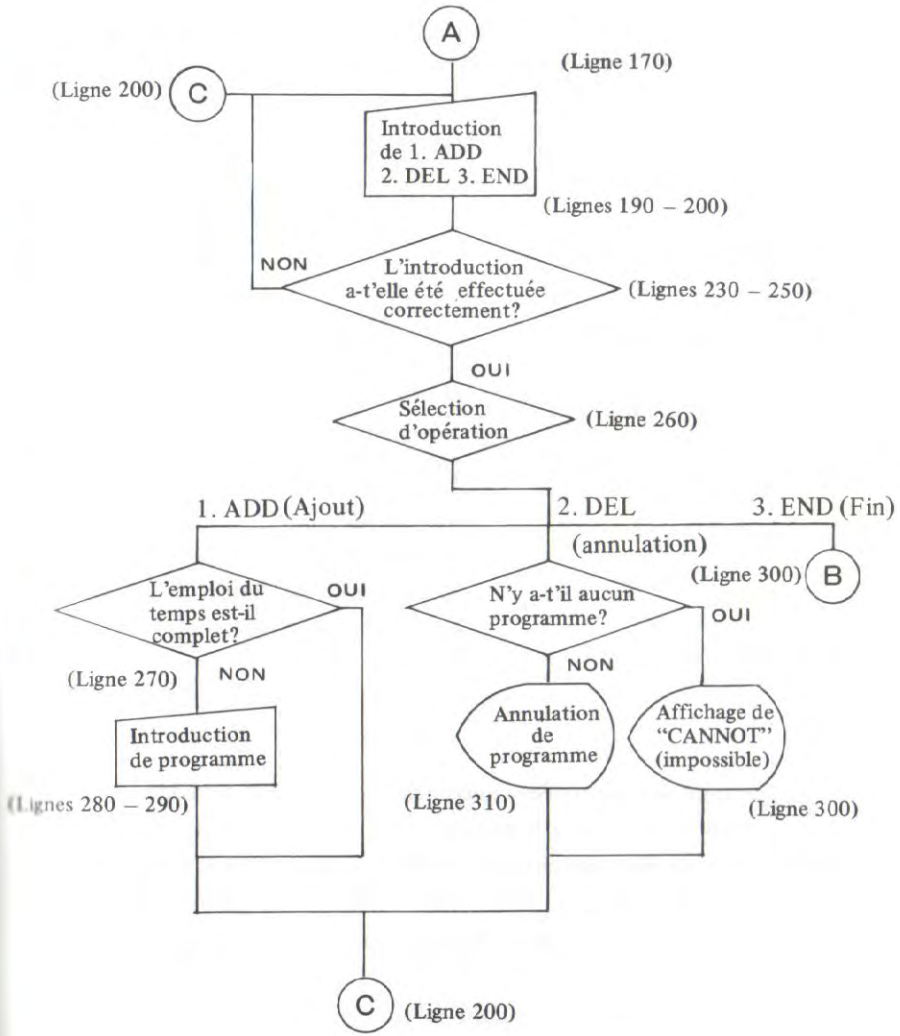
READ、DATA、RESTORE、VAL、STR$、INKEY$、ASC

Liste de programme

```
 10 INPUT "FIRST (Y
    /N)";X$:IF X$="
    N" THEN 30
 20 CLEAR :DIM W$(7
    ,7)*30,C$(7)
 30 FOR I=1 TO 7:RE
    AD C$(I):NEXT I
 40 DATA SUN,MON,TU
    E,WED,THU,FRI,S
    AT
 50 INPUT "DAY=";A$
 60 INPUT "PRINTER
    (Y/N)";N$
 70 IF N$="Y" THEN
    GOSUB 320
 80 X=0:FOR I=1 TO
    7
 90 IF A$=C$(I) THE
    N X=I
100 NEXT I
110 IF X=0 THEN BEE
    P : GOTO 50
120 MA=VAL(W$(X,0))
130 IF MA=0 THEN PR
    INT "NO SCHEDUL
    E": GOTO 180
140 FOR I=1 TO MA
150 PRINT W$(X,I);
160 A$=INKEY$:IF A$
    ="" THEN 160
170 PRINT :NEXT I
180 PRINT OFF
190 PRINT "1..ADD 2
    ..DEL 3..END";
200 A$=INKEY$:IF A$
    ="" THEN 200
210 Y=ASC(A$)
220 Y=Y-48
230 IF Y<0 THEN 200
240 IF Y>3 THEN 200
250 PRINT
260 ON Y GOTO 270,3
    00,50
270 IF MA=7 THEN BE
    EP :PRINT "NO S
    PACE!": GOTO 19
    0
280 MA=MA+1:INPUT W
    $(X,MA)
290 W$(X,0)=STR$(MA
    ): GOTO 190
300 IF MA=0 THEN BE
    EP :PRINT "CANN
    OT!": GOTO 190
310 MA=MA-1:W$(X,0)
    =STR$(MA): GOTO
     190
320 PRINT ON :PRINT
    A$
330 RETURN
```

Diagramme


Début
Nouvel emploi du temps hebdomadaire
NON
OUI
(Ligne 10)
Déclaration en tableau W$ (7,7) * 30, C$ (7)
(Ligne 20)
Assignation du jour de la semaine à C$(I)
(Lignes 30 – 40)
B
Introduction du jour de la semaine (A$)
(Ligne 50)
Le jour de la semaine a-t'il été introduit correctement?
NON
(Lignes 80 – 110)
OUI
Sortie imprimante?
OUI
Imprimante ON
(Lignes 60 – 70)
(Ligne 320)
NON
L'emploi du temps a-t'il été introduit?
NON
(Lignes 120 – 130)
OUI
(Lignes 140 – 170)
(Ligne 130)
Affichage de l'emploi du temps
Affichage "NO SCHEDULE" (aucum programme)
A


A
(Ligne 170)
(Ligne 200)
C
Introduction de 1. ADD 2. DEL 3. END
(Lignes 190 – 200)
NON
L'introduction a-t'elle été effectuée correctement?
(Lignes 230 – 250)
OUI
Sélection d'opération
(Ligne 260)
1. ADD (Ajout)
2. DEL (annulation)
3. END (Fin)
(Ligne 300)
B
L'emploi du temps est-il complet?
OUI
(Ligne 270)
NON
Introduction de programme
(Lignes 280 – 290)
N'y a-t'il aucun programme?
OUI
NON
Annulation de programme
(Ligne 310)
Affichage de "CANNOT" (impossible)
(Ligne 300)
C
(Ligne 200)

Préparez un emploi du temps hebdomadaire en introduisant des données quotidiennes. Avant d'imprimer les données, s'assurer que l'imprimante (FA-20) est correctement connectée.

- Utiliser 3 caractères alphabétiques pour introduire les jours de la semaine.

SUN (dim), MON (lun), TUE (mar), WED (mer), THU (jeu), FRI (ven), SAT (sam)

- Vous pouvez utiliser jusqu'à 30 caractères par programme et introduire 7 programmes par jour.

Introduction des données

Monday (Lundi)	10 h	Meeting with Mr. Jones (r.d.v. avec Mr. Jones)
	14 h	Conference (Conférence)
	16 h	Go to Company A (Aller à la société A)
Tuesday (Mardi)	13 h	Conference (Conférence)
	15 h	Meeting with Mr. Smith (r.d.v. avec Mr. Smith)

Opération

RUN↵ | FIRST (Y/N)?

L'affichage demande si les données concernent une nouvelle semaine. Appuyer sur la touche [Y] pour introduire la première donnée de la semaine. Appuyer sur la touche [N] pour vérifier ou changer (ADD (ajouter), DELE (annuler)) un horaire précédemment introduit. Faîtes bien attention lorsque vous utilisez la touche [Y] : Toutes les données introduites auparavant seront effacées lorsque vous appuyez sur cette touche.

Etant donné que nous voulons introduire ici un emploi du temps qui concerne une nouvelle semaine, appuyer sur les touches suivantes.

Y↵ | DAY?

Introduire le jour de la semaine.

MON↵ | PRINTER (Y/N)?

Cet affichage demande si l'emploi du temps introduit doit être imprimé. Vu que rien n'a encore été introduit, appuyer sur la touche [N] .

N↵ | NO SCHEDULE

L'affichage indique qu'il n'y a aucun programme prévu pour MON (lundi).

↵ | 1..ADD 2..DEL 3..END

Choisir l'une des opérations suivantes et appuyer sur la touche correspondante.

[1] ADD····· Introduction de données
[2] DEL····· Annulation de données
[3] END····· Termine l'opération

Il y a dans nôtre cas une donnée à introduire, choisir donc 1.

1 | ?

Intoduire le programme pour lundi en ordre séquentiel, en n'utilisant pas plus de 30 caractères par élément.

10H[SPC]JONES↵	1..ADD 2..DEL 3..END
1	?
14H [SPC]CONFERENCE↵	1..ADD 2..DEL 3..END
1	?
16H [SPC]A[SPC]CO.↵	1..ADD 2..DEL 3..END

Lorsque le programme pour MON (lundi) a été introduit, appuyer sur 3.

3 | DAY?

Introduire ensuite le programme pour TUE (mardi).

TUE ↵	PRINTER (Y/N)?
N ↵	NO SCHEDULE
↵	1..ADD 2..DEL 3..END
1	?
13H SPC CONFERENCE ↵	1..ADD 2..DEL 3..END
1	?
15H SPC SMITH ↵	1..ADD 2..DEL 3..END

Interrompre un instant l'introduction de données et imprimer les données qui ont été introduites jusqu'ici. Ces données seront affichés et imprimées à chaque fois qu'on appuiera sur la touche ↵ .

3	DAY?
MON ↵	PRINTER (Y/N)?

C'est normalement sur la touche [Y] qu'on appuie ici, mais appuyez sur la touche [N] si vous ne vous servez pas de l'imprimante. L'affichage et les opérations de touches sont les mêmes.

Y ↵	10H JONES
↵	14H CONFERENCE
↵	16H A CO.

L'impression se fera comme indiqué ci-dessous.

```
MON
10H JONES
14H CONFERENCE
16H A CO.
```


(Opération suivante)

↵	1..ADD 2..DEL 3..END
3	DAY?

Les jours de la semaine et les données sont introduits de cette manière. Toutefois, comme ce programme est sans fin, appuyer sur BRK pour le terminer.

BRK	READY P0

Ensuite, nous allons appeler les données stockées et modifier l'emploi du temps.

RUN ↵	FIRST (Y/N)?
N ↵	DAY?

Modifier l'emploi du temps de TUE (mardi).

TUE ↵	PRINTER (Y/N)?

Cette fois ci nous n'imprimerons pas.

N ↵	13H CONFERENCE
↵	15H SMITH
↵	1..ADD 2..DEL 3..END

Pour modifier, attendre que tout l'emploi du temps de TUE (mardi) soit affiché. Changer l'heure du rendez-vous avec Mr. Smith et le mettre à quatre heures. Pour réaliser ceci, commencer par annuler la donnée en cours et puis introduire la nouvelle donnée.

2	1..ADD 2..DEL 3..END

L'affichage est le même qu'auparavant, sauf que 15H SMITH a été effacé. Les données introduites seront effacées une par une à partir de la fin, chaque fois qu'on appuiera sur la touche [2]. Il n'est pas possible de spécifier une donnée particulière pour effacement.
Pour introduire la nouvelle donnée, effectuer l'opération suivante:

```
1
16H  [SPC]SMITH[↵]
3
```

```
?
1..ADD 2..DEL 3..END
DAY?
```

Appuyez sur [BRK] lorsque vous souhaitez terminer le programme.

■ Introduction de données dans un programme (Commande READ)

```
30 FOR I=1 TO 7:READ C$(I):NEXT I
```

READ : Introduit les données spécifiées par l'instruction de données dans la variable de tableau C$(I) (Commande READ)

```
40 DATA SUN,MON,TUE,WED,THU,FRI,SAT
```

DATA : Instruction de données
SUN,MON,TUE,WED,THU,FRI,SAT : Données

SUN, MON ... SAT sont stockés dans la variable de tableau C$(I) à la ligne 30. Pour exécuter la commande READ, il faudra toujours une instruction DATA comme indiqué à la ligne 40. Une des données de l'instruction de données sera mémorisée à chaque fois que la commande READ sera exécutée.

```
10 FOR I=1 TO 4
20 READ A$
30 PRINT A$
40 NEXT I
50 DATA MOUNT,SKY,
   RIVER,SEA
```

→ La commande READ sera exécutée quatre fois par l'instruction FOR ~ NEXT.

```
RUN[↵]
[↵]
[↵]
[↵]
```

```
MOUNT
SKY
RIVER
SEA
```


Bien que la variable A$ reste toujours la même, les données continueront de changer. (les données de l'instruction DATA à la ligne 50 seront introduites dans l'ordre séquentiel.)
La commande READ peut aussi introduire des données dans des variables multiples en une seule fois.

```
10 READ A$,B$,C$
20 PRINT A$
30 PRINT B$
40 PRINT C$
50 DATA SUNDAY,MON
   DAY,TUESDAY
```

10 READ A$,B$,C$ → Introduction de donnée dans les variables de caractères A$, B$ et C$.
20 PRINT A$ / 30 PRINT B$ / 40 PRINT C$ → Affichage des données introduites.

```
RUN[↵]
[↵]
[↵]
```

```
SUNDAY
MONDAY
TUESDAY
```

Si le nombre des données de l'instruction DATA dépasse celui des variables introduites à l'aide de la commande READ, les données en surnombre seront ignorées.

```
10 READ A$,B$
20 PRINT A$
30 PRINT B$
40 DATA RED,BLUE,G
   REEN,YELLOW
```

RED (rouge) et BLUE (Bleu) sont introduits pour A$ et B$ dans ce programme. Les données en reste GREEN (vert) et YELLOW (jaune) ne sont pas affichées.

```
RUN[↵]
[↵]
[↵]
```

```
RED
BLUE
READY P0
```

Cependant, si une autre commande READ est exécutée, les données restantes seront alors introduites.

Pour afficher GREEN et YELLOW aussi, modifier le programme de la façon suivante.

```
10 READ A$,B$
20 PRINT A$
30 PRINT B$
40 READ C$,D$
50 PRINT C$
60 PRINT D$
70 DATA RED,BLUE,G
   REEN,YELLOW
```

Une nouvelle commande READ est ajoutée.

```
RUN↵
↵
↵
↵
```

```
RED
BLUE
GREEN
YELLOW
```

Une erreur DA ERROR se produira si le nombre des données spécifié par l'instruction de données est inférieur à celui des variables introduites à l'aide de la commande READ.

```
10 READ A$
20 READ B$
30 READ C$
40 READ D$
50 DATA BLACK,WHIT
   E,GRAY
```

La commande READ est exécutée quatre fois.

Il n'y a que trois données disponibles.

Une erreur DA ERROR sera affichée si l'on essaye d'introduire la quatrième donnée pendant l'exécution de ce programme.

```
RUN↵
```

```
DA ERROR P0-40
```

Si vous souhaitez, à une ligne particulière, lire une des instructions DATA, utilisez la commande RESTORE.


```
10 READ A,B,C
20 PRINT A;B;C
25 REM
30 RESTORE 300
40 READ A,B,C
50 PRINT A;B;C
55 REM
60 RESTORE 200
70 READ A,B,C
80 PRINT A;B;C
90 END
100 DATA 1,2,3
200 DATA 10,20,30
300 DATA 11,22,33
```

Les données sont lues à partir du début de l'instruction DATA à l'aide d'une commande READ ordinaire.

Introduction de données de l'instruction DATA à la ligne 300.

Introduction de données de l'instruction DATA à la ligne 200.

```
RUN↵
↵
↵
```

```
1 2 3
11 22 33
10 20 30
```

La commande READ a pour fonction d'introduire dans des variables les données spécifiées par l'instruction DATA. (Voir page 245)

■ La fonction VAL

```
120 MA=VAL(W$(X,0))
```

Founction VAL

Contient des chaînes de caractères à convertir en valeurs numériques

L'expression ci-dessus à la ligne 120 signifie que les caractères stockés dans le tableau de caractères W$ seront convertis en une valeur numérique et seront affectés à une variable numérique enregistrée MA.

La fonction VAL convertit une chaîne de caractères en une valeur numérique de cette façon.

```
10 A$="12345"
20 X=VAL(A$)
30 PRINT X;X*2
```

Une chaîne de caractères numériques a été affectée à A$.

La valeur numérique de A$ est affectée à X.

On peut donc se servir de X pour les calculs.

```
RUN↵
```

```
12345 24690
```

Tous les caractères autres que les caractères numériques seront traités comme 0 par la fonction VAL. Cependant, les notations hexadécimales (&HOO) sont des exceptions (exemple: VAL ("&HF") ENTER 15). Si un caractère numérique se trouve au début d'une chaîne mixte de caractères numériques et non numériques, ces caractères seront convertis en valeurs numériques. Si c'est un caractère non numérique qui se trouve au début, les caractères seront traités comme 0.

```
10 A$="ABCDEFG"
20 X=VAL(A$)
30 PRINT X
```

→ Chaîne de caractères alphabétiques.

Une chaîne de caractères alphabétiques est convertie à la valeur numérique 0.

```
RUN↵
```

```
0
```

```
10 A$="246XYZ"
20 B$="XYZ789"
30 C$="45LMN69"
40 X=VAL(A$)
50 Y=VAL(B$)
60 Z=VAL(C$)
70 PRINT "X=";X
80 PRINT "Y=";Y
90 PRINT "Z=";Z
```

→ Les trois premiers caractères affectés à A$ sont des caractères numériques.

→ Les trois derniers caractères affectés à B$ sont des caractères numériques.

→ Des caractères numériques se trouvent au début et à la fin de C$.

Dans les chaînes de caractères qui commencent par un caractère numérique, les premiers caractères numériques sont convertis en valeurs numériques.

```
RUN↵
↵
↵
```

```
X= 246
Y= 0
Z= 45
```


Cette fonction est très pratique.

```
10 INPUT X$
20 Y=VAL(X$)
30 IF Y<1 THEN 10
40 IF Y>10 THEN 10
50 PRINT Y
```

Le programme ci-dessus est un programme qui ne peut accepter des données que si elles sont comprises dans les limites 1 ≦ donnée ≦ 10.

Autrement dit, si une donnée non numérique est introduite dans X$, Y sera converti en 0 par la fonction VAL. De même, puisque les données numériques introduites seront converties en valeurs numériques par la fonction VAL, Y peut être jugé conformément à cette valeur quand des caractères numériques sont introduits.

> La fonction VAL convertit une chaîne de caractères donnée en valeur numérique. (Voir page 254)

- **La fonction STR$**

```
290 W$(X,0)=STR$(MA):GOTO160
```

Fonction STR$ — Convertit les valeurs numériques stockées dans MA en caractères.

Les valeurs numériques stockées dans la variable MA sont converties en chaînes de caractères et stockées dans le tableau de caractères W$ à la ligne 290. La fonction STR$ convertit ainsi les valeurs numériques en chaînes de caractères, tandis que la fonction VAL convertit les chaînes de caractères en valeurs numériques.

Un espacement sera placé au début d'une chaîne de caractères obtenue par conversion d'une valeur numérique à l'aide de la fonction STR$, si la valeur numérique est soit un nombre positif soit 0. Un symbole (−) sera placé au début s'il s'agit d'un nombre négatif.

Il conviendra aussi de faire bien attention au genre de variables de caractères à stocker dans la chaîne de caractères lorsqu'on utilise des fractions décimales comprenant un grand nombre de chiffres.

7 caractères maximum peuvent être utilisés pour A$ ~ Z$. (Dans ce cas le symbole occupera l'espace d'un caractère.)

16 caractères maximum peuvent être utilisés pour A1$ ~ Z9$. (Dans ce cas le symbole occupera l'espace d'un caractère.)

```
10 A=37.169
20 B=-34.885
30 X$=STR$(A)
40 Y$=STR$(B)
50 PRINT "DATA=";X
   $;"(X$)"
60 PRINT "DATA=";Y
   $;"(Y$)"
```

Un espacement sera placé au début si le nombre est positif. Un symbole (−) sera placé au début si le nombre est négatif.

RUN↵

```
DATA= 37.169(X$)
DATA=-34.885(Y$)
```

```
10 A=548.3271
20 X$=STR$(A)
30 PRINT "DATA=";X
   $;"(X$)"
```

La valeur de X$ est théoriquement "548, 3271", mais un maximum de sept caractère peut être stocké dans X$ lorsqu'on exécute la fonction STR$.

Une erreur ST ERROR se produit donc à la ligne 20.

RUN↵

```
ST ERROR P0-20
```

En passant de X$ à X1$ on peut stocker jusqu'à 16 caractères dans ce programme.

```
10 A=548.3271
20 X1$=STR$(A)
30 PRINT "DATA=";X
   1$;"(X1$)"
```

RUN↵

```
DATA= 548.3271(X1$)
```

```
10 A=-3.1415926535
   *10^-99
20 X1$=STR$(A)
30 PRINT "DATA=";X
   1$;"(X1$)"
```


Voici un exemple d'utilisation de tous les 16 caractères de X1$.

RUN↵

```
DATA=-3.141592654E-99(X1
```

Défilement de l'écran

```
TA=-3.141592654E-99(X1$)
```

La fonction STR$ convertit des valeurs numériques données en chaînes de caractères. (Voir page 255)

- **La fonction INKEY$**

```
200 A$=INKEY$:IF A$=""THEN 200
```

Fonction INKEY$

Un caractère est affecté à partir du clavier

Un caractère est affecté dans A$ à partir du clavier et si le résultat est nul (" "), la commande retourne à la ligne 200. C'est à dire qu'il n'y aura pas d'exécution tant qu'une touche du clavier n'aura pas été actionnée.

On utilise cette méthode pour les jeux, quand on manipule les touches ou quand on ne veut pas appuyer sur les touches ↵ et ENTER après une introduction par le clavier.

La différence entre INKEY$ et INPUT est indiquée ci-dessous.

Utilisation	INKEY$	INPUT
Affichage en course d'exécution de commande	Aucun affichage	? (Message de demande d'introduction. On peut ne pas afficher "?".)
Introduction de données	Touche actionnée lorsqu'on exécute une commande (Pas d'introduction sans appuyer sur cette touche)	Introduction de données jusqu'à ce que la touche ↵ soit actionnée.
Genres des données	1 caractère en tant que caractère	Nombre de caractères et de chiffres compris dans les limites d'une variable comme un caractère ou une valeur numérique.
Exécution de la commande suivante	Exécution immédiate (↵ n'est pas nécessaire)	En suspens jusqu'à ce que soit ↵ actionnée.

```
10 PRINT "CONTINUE
   (Y/N)?";
20 A$=INKEY$:IF A$
   ="" THEN 20 ELS
   E BEEP 0
30 IF A$="Y" THEN
   60
40 IF A$="N" THEN
   80
50 GOTO 20
60 PRINT :PRINT "C
   ASIO"
70 GOTO 10
80 PRINT :PRINT "E
   ND"
```

RUN ↵

```
CONTINUE(Y/N)?
```

L'opération suivante sera accomplie lorsqu'on appuiera sur les touches [Y] ou [N] .

Y

↵

```
CASIO
CONTINUE(Y/N)?
```

CASIO sera affiché si l'on appuie à nouveau sur la touche [Y], CONTINUE (Y/N) ? sera répété si l'on appuie sur ↵ . Pour terminer le programme appuyer sur la touche [N] .

N

↵

```
END
READY P0
```

```
10 PRINT "PRESS AN
   Y KEY..";
20 R=INT(RND*10)+1
30 A$=INKEY$:IF A$
   ="" THEN 20
40 PRINT
50 PRINT "NUMBER="
   ;R
60 GOTO 10
```


Lorsque ce programme est exécuté, PRESS ANY KEY ... (appuyer sur n'importe quelle touche) apparaît à l'affichage et un nombre aléatoire sera affiché si on appuie sur une touche. Ce genre de programme peut aussi être utilisé pour les jeux.

> INKEY$ a pour fonction de donner une affectation à un caractère conformément au code ASCII quand on appuie sur une touche. Nul (" ") est affecté si l'on n'appuie sur aucune touche. (Voir page 261)

▪ La fonction ASC

```
210 Y=ASC(A$)
      └─ Fonction ASC
    └─ Le code ASCII correspondant stocké dans A$
       est affecté à Y.
```

Le code ASCII d'un caractère stocké dans la variable de caractrès A$ sera ainsi affecté à la variable Y. Si une chaîne de caractères est stockée dans A$, seul le premier caractère du code ASCII sera affecté. 0 sera affecté si rien n'est stocké dans A$.

```
10 FOR I=1 TO 6
20 READ A$
30 S=ASC(A$)
40 PRINT A$;"=";S
50 NEXT I
60 DATA A,B,C,0,1,
   2
```

Lit les caractères A, B, C, 0, 1 et 2 et affiche les codes ASCII correspondants.

RUN ↵

↵

↵

↵

```
A= 65
B= 66
C= 67
0= 48
```

↵

```
1= 49
```

↵

```
2= 50
```

```
10 A$="ABCDEFG"
20 B$=""
30 X=ASC(A$)
40 Y=ASC(B$)
50 PRINT "A$=";X
60 PRINT "B$=";Y
```

RUN↵

```
A$= 65
```

↵

```
B$= 0
```

Le code ASCII du premier caractère (A) de la chaîne de caractères stockée dans A$ est affiché. 0 sera affecté à B$ puisque rien n'y est stocké.
La fonction ASC convertit un caractère affecté en la valeur numérique qui lui correspond (code ASCII). (La fonction CHR$ est une fonction correspondante. Voir page 253).

> La fonction ASC convertit le premier caractère d'une chaîne de caractères donnée en la valeur numérique qui lui correspond. (Voir page 253)
> * Consulter le tableau des codes de caractères (voir page 328).

4-7 Contrôle $\overline{X}$-R

Nul n'ignore que les voitures que nous conduisons tous les jours exigent les plus hautes qualités de précision. Le contrôle de la qualité joue un rôle extrêmement important aussi bien pour les produits industriels que dans d'autres domaines.
La méthode de contrôle $\overline{X}$-R est l'une des méthodes qui permettent de maintenir une haute qualité de produits. C'est elle que nous utiliserons dans ce programme.
Dans un diagramme de contrôle $\overline{X}$-R, le contrôle $\overline{X}$ est une méthode qui permet d'obtenir la gamme moyenne de tolérance à partir de la moyenne de plusieurs mesures de valeurs (montant de la production par périodes spécifiées de temps) prises au hasard comme échantillons de chaque groupe.
Le contrôle R permet d'obtenir la moyenne de la diffusion de données tolérée à partir du coefficient R*i* (R*i* = maximum − minimum) de chaque groupe.

Commande

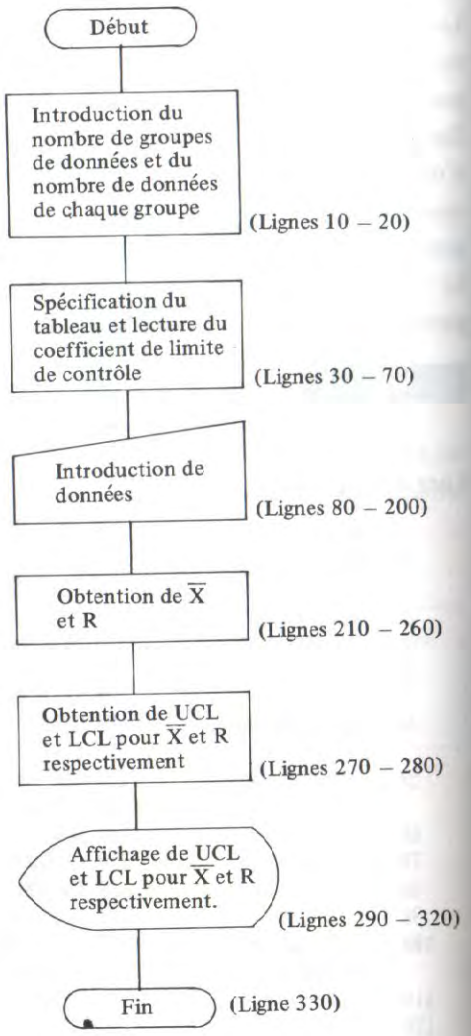
CLS

Liste de programme

```
 5 CLEAR
10 INPUT "NO. OF G
   ROUP=";N
20 INPUT "NO. OF D
   ATA/GROUP=";M
30 DIM A(N),R(N),Q
   (10,3)
40 PRINT "READING
   DATA...";
50 FOR I=1 TO 10:F
   OR J=1 TO 3
60 READ Q(I,J)
70 NEXT J:NEXT I
80 CLS
90 FOR I=1 TO N
100 STAT CLEAR :MA=
   0:MI=10000
110 FOR J=1 TO M
120 PRINT "GRP:";I;
   ",DATA";J;"=";
130 INPUT X
140 STAT X
150 IF X>MA THEN MA
   =X
160 IF X<MI THEN MI
   =X
170 NEXT J
180 A(I)=SUMX/CNT
190 R(I)=MA-MI
200 NEXT I
210 STAT CLEAR
220 FOR I=1 TO N
230 STAT A(I),R(I)
240 NEXT I
250 XB=SUMX/CNT
260 RB=SUMY/CNT
270 U1=Q(N,3)*RB:L1
   =Q(N,2)*RB
280 U2=XB+Q(N,1)*RB
   :L2=XB-Q(N,1)*R
   B
290 BEEP :PRINT "XB
   AR=";ROUND(XB,-
   3)
300 PRINT "LCL=";RO
   UND(L2,-3);",UC
   L=";ROUND(U2,-3
   )
310 BEEP :PRINT "RB
   AR=";ROUND(RB,-
   3)
320 PRINT "LCL=";RO
   UND(L1,-3);",UC
   L=";ROUND(U1,-3
   )
330 END
340 DATA0,0,0
350 DATA1.88,0,3.26
   8
360 DATA1.023,0,2.5
   74
370 DATA0.729,0,2.2
   88
380 DATA0.577,0,2.1
   14
390 DATA0.483,0,2.0
```

```
   04
400 DATA0.419,0.076
    ,1.924
410 DATA0.373,0.136
    ,1.864
420 DATA0.337,0.184
    ,1.816
430 DATA0.308,0.223
    ,1.777
```

Organigramme



Introduire les données échantillon et obtenir les limites de contrôle.

Données introduites

Groupe / Echantillon	1	2	3
1	10,3	10,2	10,2
2	10	10,1	10
3	10,3	10,2	10,3
4	10,4	10,3	10,3

Opération

RUN↵

```
NO. OF GROUP=?
```

Introduire le nombre de groupes.

3↵

```
NO. OF DATA/GROUP=?
```

Introduire le nombre de données.

4↵

```
READING DATA...
```

Les données ont été enregistrées.

```
GRP: 1,DATA 1=?
```

Introduire les données du groupe 1.

10.3↵

```
GRP: 1,DATA 2=?
```

10↵

```
GRP: 1,DATA 3=?
```

10.3↵

```
GRP: 1,DATA 4=?
```

10.4↵

```
GRP: 2,DATA 1=?
```

⋮ ⋮

Affichage de la valeur de X lorsque la quatrième donnée du groupe 3 est introduite.

```
10.3↵                    XBAR= 10.22
```

Affichage des lignes de LCL et UCL (Contrôle $\overline{X}$: limite supérieure de contrôle et limite inférieure de contrôle)

```
↵                    LCL= 9.91, UCL= 10.52
```

Affichage de la valeur moyenne de R (Moyenne de Rs (maximum – minimum))

```
↵                    RBAR= 0.3
```

Affichage des lignes LCL et UCL (Contrôle R: limite supérieure de contrôle et limite inférieure de contrôle)

```
↵                    LCL=0, UCL= 0.77
```

La qualité de ce groupe peut être considérée comme excellente puisque les données introduites ne sortent pas des limites 9,91 ≦ données ≦ 10,52.

- **La commande CLS**

```
80 CLS
   └── Commande CLS
```

La commande CLS sert à la ligne 80 pour effacer le contenu de l'affichage. La commande CLS peut donc être utilisée ainsi pour vider l'affichage.

```
10 FOR I=0 TO 22
20 LOCATE I:PRINT
   "#";
30 FOR J=1 TO 5:NE
   XT J
40 CLS
50 NEXT I
```

Dans ce programme, le symbole # se déplace de la gauche vers la droite.
(Chaque fois que # se déplace d'un caractère vers la droite il y a effacement à l'affichage. Ce processus se répète 23 fois)

La commande CLS a pour fonction d'effacer le contenu de l'affichage.
(Voir page 226)

4-8 Analyse ABC

Il circule, dans le secteur de la vente au détail, une variété considérable de produits. Dresser un inventaire des ventes par types et saisir les canaux par lesquels circulent tous ces produits devient une tâche effrayante.
Grâce à ce programme il sera possible de classer des ventes par ordre décroissant et aussi de leur affecter des positions A, B et C en introduisant simplement le nom d'un produit et le montant de ses ventes.
Dans ce programme le classement est établi comme suit:

Catégorie A . . . . . Produits qui représentent 80% du total et figurent parmi les 10 premiers du classement.
Catégorie B . . . . . Produits qui représentent 15% du total, mis à part les produits de la catégorie A
Catégorie C . . . . . Tous les autres produits

Donc, si l'on se base sur des ventilations de probabilités données approximativement par une courbe normale, les données seront classées dans l'ordre suivant, lorsqu' on les utilise pour l'analyse scienfifique.
Comprises dans les limites de ± 1σ (Catégorie A)
Comprises dans les limites de ± 2σ (Catégorie B)
Comprises dans les limites de ± 3σ (Catégorie C)

Commandes et fonctions

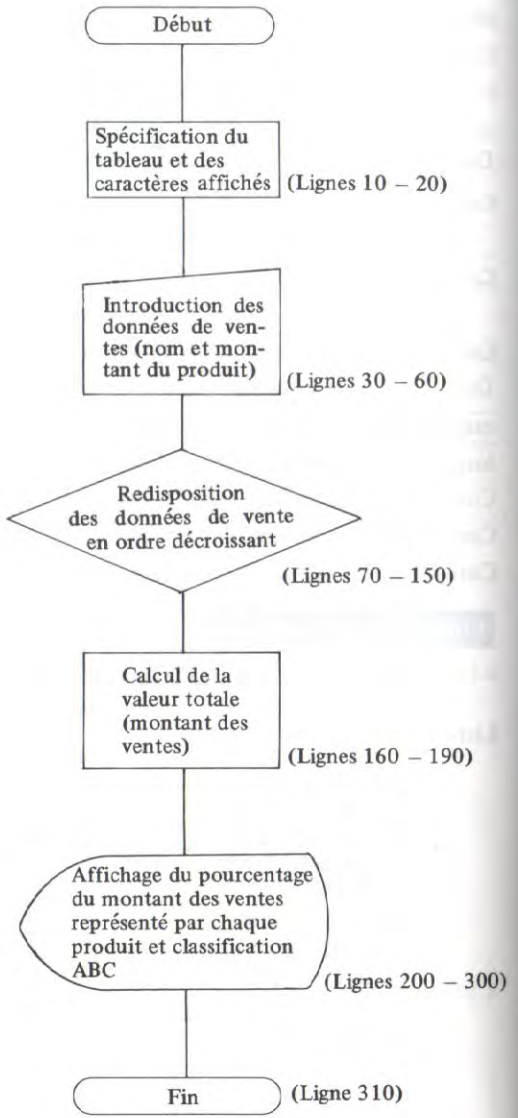
REM、LEFT$、MID$、RIGHT$

Liste de programme

```
10 INPUT "NO. OF I
   TEM=";N
20 DIM S(N),B$(N):
   P$=CHR$(37):C$=
   "ABC"
30 FOR I=1 TO N
40 PRINT "NO.";I;:
   INPUT " NAME=";
   B$(I)
50 PRINT "NO.";I;:
   INPUT " SALES="
   ;S(I)
60 NEXT I
70 REM SORT DATA
80 F=0
90 FOR I=1 TO N-1
100 IF S(I)>=S(I+1)
    THEN 140
110 X=S(I):S(I)=S(I
    +1):S(I+1)=X
120 X$=B$(I):B$(I)=
    B$(I+1):B$(I+1)
    =X$
130 F=1
140 NEXT I
150 IF F=1 THEN 70
160 REM CALC & DISP
170 G=0
```

```
180 FOR I=1 TO N:G=
    G+S(I):NEXT I
190 T=0
200 PRINT "NAME SAL
    ES TOTAL (";P$;
    ")"
210 FOR I=1 TO N
220 T=T+S(I)
230 PRINT LEFT$(B$(
    I),5);
240 PRINT S(I);T;
250 R=T/G*100
260 PRINT ROUND(R,-
    2);
270 IF R<80 THEN IF
    ·I<=10 THEN PRI
    NT LEFT$(C$,1):
     GOTO 300
280 IF R<95 THEN PR
    INT MID$(C$,2,1
    ): GOTO 300
290 PRINT RIGHT$(C$
    ,1)
300 NEXT I
310 END
```

Organigramme



Effectuer l'analyse ABC en introduisant le nom du produit (ou son numéro) et le montant global des ventes.

Numéro du produit	Montant des ventes
001	5703
002	4328
003	3984
004	8905
005	2305
006	2996
007	4848
008	2636
009	2095
010	5510

(Ne pas utiliser une représentation de position à trois chiffres.)

Opération

RUN↵

```
NO. OF ITEM=?
```

Introduire le nombre total d'articles.

10↵

```
NO. 1 NAME=?
```

Introduire le numéro du produit (ou son nom).

001↵

```
NO. 1 SALES=?
```

Introduire le montant des ventes pour 001.

5703↵

```
NO. 2 NAME?
```

⋮

Effectuer toutes les introductions de la même manière jusqu'au produit No. 10.

Une fois toutes les données introduites, le FX-750P mettra environ 20 secondes pour les classer et afficher les résultats.

```
NAME SALES TOTAL (%)
004 8905 8905 20.6A
001 5703 14608 33.7A
010 5510 20118 46.5A
        ⋮
005 2305 41215 95.2C
009 2095 43310 100C
```

Les numéros de produits seront classés par ordre séquentiel à partir des montants de vente les plus élevés. Ils seront affichés en même temps que les cumuls, les pourcentages et les classifications par catégories.

Ordre	No de produit	Montant des ventes	Cumul	Pourcentage cumul/total	Catégorie
1	004	8905	8905	20.6	A
2	001	5703	14608	33.7	A
3	010	5510	20118	46.5	A
4	007	4848	24966	57.6	A
5	002	4328	29294	67.6	A
6	003	3984	33278	76.8	A
7	006	2996	36274	83.8	B
8	008	2636	38910	89.8	B
9	005	2305	41215	95.2	C
10	009	2095	43310	100	C

▪ Pour une compréhension facile du programme (Commande REM)

```
 70 REM SORT DATA
160 REM CALC & DISP
```

└ Commande REM

Aux lignes 70 et 160, les commandes n'affectent pas directement l'exécution du programme lui-même. Ces commandes sont simplement utilisées pour rendre le programme plus clair et plus intelligible. En effet, plus un programme devient long, plus il est difficile à comprendre pour d'autres personnes. C'est pourquoi on introduit des commentaires à l'intérieur du programme à l'aide de la commande REM; on peut ainsi comprendre le programme plus vite et plus facilement.

```
10 REM AVERAGE PRO
   GRAM
20 FOR I=1 TO 5
30 READ X:REM READ
   DATA
40 S=S+X:REM CALCU
   LATE TOTAL
50 NEXT I
60 REM CALCULATE A
   VERAGE
70 AV=S/5
80 REM PRINT OUT
90 PRINT "AVE=";AV
100 END
110 REM ORIGINAL DA
   TA
120 DATA 54,48,68,4
   2,59
```

Le programme deviendra très facile à comprendre si des commentaires de ce genre y sont insérés.

La commande REM a pour fonction d'insérer des commentaires dans le programme. (Voir page 247)

■ La fonction LEFT$

```
270 IF R<80 THEN IF I<=10
    THEN PRINT LEFT$(C$,1):GOTO 300
```

Fonction LEFT$: recherche le premier caractère à partir de la gauche de la chaîne de caractères stockée dans C$.

Comme indiqué à la ligne 270, la fonction LEFT$ peut rechercher un nombre spécifié de caractères à partir du côté gauche de la chaîne de caractères stockée dans une variable à caractères.

```
10 A$=LEFT$("ABCDE
   FG",4)
20 PRINT A$
```

Recherche quatre caractères à partir de la gauche de la chaîne de caractères "ABCDEFG".

RUN↵

```
ABCD
```

```
10 A$="SUMMER"
20 B$=LEFT$(A$,4)
30 PRINT B$
```

Recherche quatre caractères à partir du côté gauche de la chaîne de caractères et les introduit dans A$.

RUN↵

```
SUMM
```

```
10 A$=LEFT$("ABCD"
   ,8)
20 PRINT A$
```

Si la chaîne de caractères à rechercher est plus longue que la chaîne de caractères originelle, c'est cette dernière qui sera affectée à A$.

RUN↵

```
ABCD
```

```
10 A$=LEFT$("",5)
20 PRINT A$
```

S'il n'y a pas de chaîne de caractères, c'est nul ("") qui sera affecté à A$.

RUN↵ (Nul est affiché)

La fonction LEFT$ recherche les chaînes de caractères qui ont la longueur spécifiée, à partir du côté gauche d'une chaîne de caractères donnée. (Voir page 258)

■ La fonction MID$

```
280 IF R<95 THEN PRINT MID$(C$,2,1)
```

Fonction MID$: recherche 1 caractère vers la droite à partir du deuxième caractère de la chaîne de caractères stockée dans C$.

Comme indiqué à la ligne 280, la fonction MID$ peut rechercher le nombre spécifié de caractères à partir d'une position définie de la chaîne de caractères stockée dans une variable à caractères.

```
10 A$=MID$("ABCDEF
   G",4,2)
20 PRINT A$
```

Recherche deux caractères à partir du quatrième caractère (D) depuis la gauche de la chaîne de caractères "ABCDEF".

RUN↵

```
DE
```

```
10 A$=MID$("ABCDEF
   G",3,7)
20 PRINT A$
```

Si le nombre des caractères situés à droite de la position spécifiée est inférieur à celui des caractères à rechercher, tous les caractères seront retenus à pertir de cette position.

RUN↵

```
CDEFG
```

```
10 A$=MID$("ABCDEF
   G",10,3)
20 PRINT A$
```

Si la position spécifiée dépasse la longueur de la chaîne de caractères donnée, nul ("") sera affecté à A$.

RUN↵ (Nul est affiché)

```
10 A$=MID$("ABCDEF
   G",3)
20 PRINT A$
```

Si la précision de la longueur de la chaîne de caractères à rechercher a été omise, tous les caractères qui se trouvent après la position spécifiée seront retenus.

RUN↵

```
CDEFG
```

La fonction MIDS recherche le nombre spécifié de caractères vers la droite de la position spécifiée dans une chaîne de caractères donnée. (Voir page 260)

■ La fonction RIGHT$

```
290 PRINT RIGHT$(C$,1):GOTO 300
```

Fonction RIGHT$: recherche 1 caractère à partir du côté droit de la chaîne de caractères stockée dans C$.

Comme indiqué à la ligne 290, la fonction RIGHTS peut rechercher le nombre spécifié de caractères à partir du côté droit d'une chaîne de caractères stockée dans une variable de caractères.

```
10 A$=RIGHT$("ABCD
   EFG",4)
20 PRINT A$
```

→ Recherche quatre caractères à partir de la droite de la chaîne de caractères "ABCDEF".

```
RUN↵
```

```
DEFG
```

```
10 A$=RIGHT$("ABCD
   EFG",10)
20 PRINT A$
```

→ Si la longueur de la chaîne de caractères à rechercher est supérieure à celle de la chaîne de caractères originelle, c'est cette dernière qui sera affectée à A$.

```
RUN↵
```

```
ABCDEFG
```

```
10 A$=RIGHT$("",5)
20 PRINT A$
```

→ S'il n'y a pas de chaîne de caractères, c'est nul ("") qui sera affecté à A$.

La fonction RIGHT$ recherche une chaîne de caractères de la longueur spécifiée à partir du côté droit d'une chaîne de caractères donnée. (Voir page 259)

4-9 Calculs chimiques (calcul de l'atténuation de la radiation)

Tous les pays sont récemment devenus extrèmement sensibles à tout ce qui touche à la radioactivité. Au lieu de se contenter de craindre les radiations, ne vaudrait-il pas mieux apprendre à connaître leur vraie nature?

Ce programme sélectionne le type de la radiation, effectue les calculs que nous allons indiquer et affiche les résultats.

1. Combien d'heures s'est-il déja écoulé?
2. Quelle sera la quantité de radiations émises dans un certain laps de temps à partir de maintenant?
3. Quelle était l'intensité initiale de la radiation?

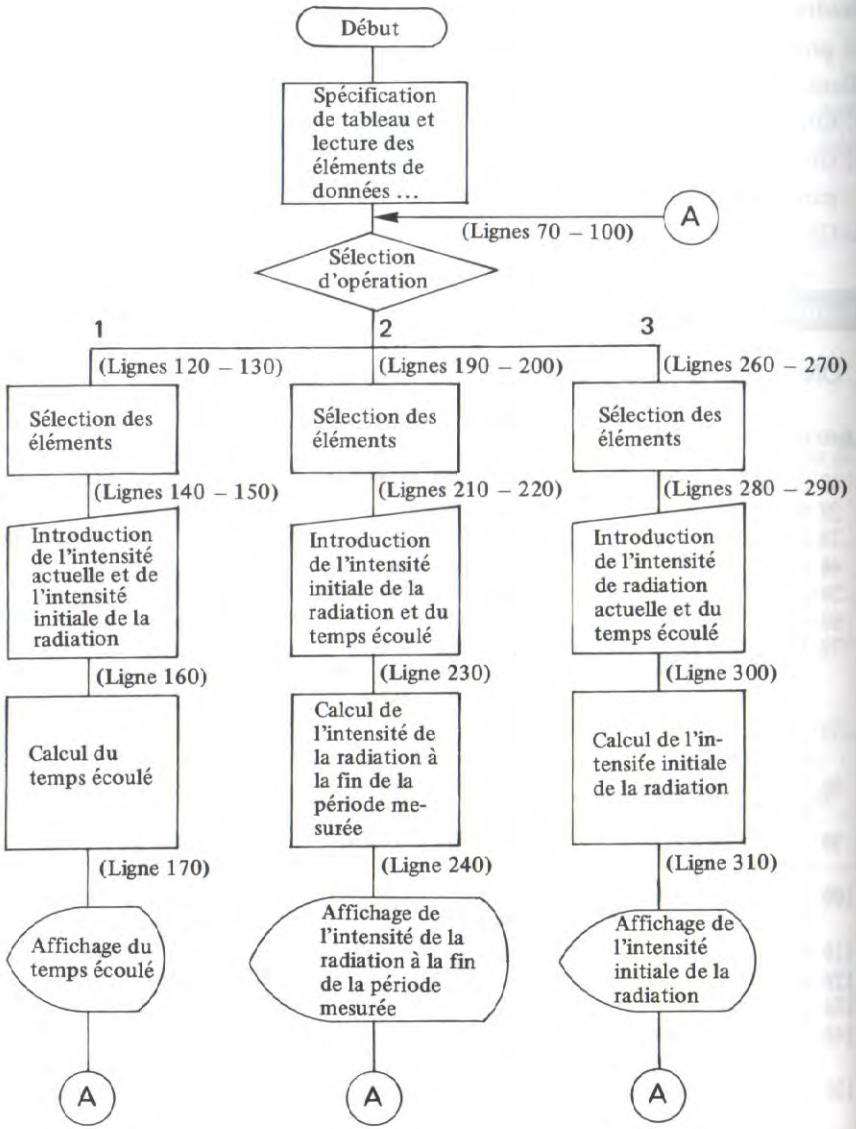
Commandes et fonctions

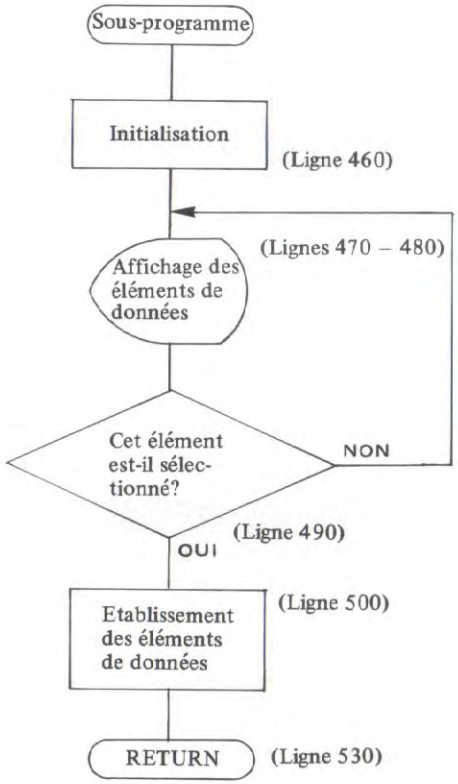
ON ~ GOTO、LOG、GOSUB / RETURN

Liste de programme

```
10 CLEAR
20 N=10
30 DIM M$(N),T(N)
40 FOR I=1 TO N
50 READ M$(I),T(I)
60 NEXT I
70 INPUT "1=TIME,2
   =AFTER,3=BEFORE
   ",X
75 IF X>3 THEN BEE
   P : GOTO 60
80 IF X<1 THEN BEE
   P : GOTO 70
90 IF X>3 THEN BEE
   P : GOTO 70
100 ON X GOTO 120,1
   90,260
110 END
120 REM TIME
130 GOSUB 440
140 INPUT "INITIAL=
   ";B
150 INPUT "PRESENT=
   ";C
160 T=Y*LOG(C/B)/LO
   G(1/2)
170 PRINT "TIME=";T
180 GOTO 70
190 REM AFTER
200 GOSUB 440
210 INPUT "INITIAL=
   ";B
220 INPUT "TIME=";T
230 C=B*(1/2)^(T/Y)
240 PRINT "AFTER=";
   C
250 GOTO 70
260 REM BEFORE
270 GOSUB 440
280 INPUT "PRESENT=
   ";C
290 INPUT "TIME=";T
300 B=C/(1/2)^(T/Y)
310 PRINT "BEFORE="
   ;B
320 GOTO 70
330 REM DATA FOR  I
   SOTOPE
340 DATA H3,1.07415
   12E5
350 DATA C14,5.0194
   8E7
360 DATA CO60,46103
   .88
370 DATA CO57,6480
380 DATA I125,1440
390 DATA I131,193.2
400 DATA CS137,2628
   00
410 DATA CR51,667.2
420 DATA F18,1.8283
430 DATA P32,342.72
440 REM SELECT DATA
450 PRINT "SELECT I
   SOTOPE!"
460 Y=0
470 FOR I=1 TO N
480 PRINT M$(I);
490 Z$=INKEY$:IF Z$
   ="" THEN 490
500 IF Z$=CHR$(13)
   THEN PRINT :Y=T
   (I): GOTO 530
510 PRINT :NEXT I
520 GOTO 460
530 RETURN
```

Organigramme





Sélectionner l'élément et l'opération. Introduire 2 des 3 facteurs concernant la valeur initiale, la valeur actuelle et le temps écoulé afin d'obtenir la valeur du facteur restant.

Introduction des données

Données 1

Nom de l'élément	CR51
Intensité initiale de la radiation	100 Curies
Intensité actuelle de la radiation	50 Curies

Données 2

Nom de l'élément	CR51
Intensité initiale de la radiation	100 Curies
Temps écoulé	deux semaines

Donnée 3

Nom de l'élément	CR51
Intensité actuelle de la radiation	70,5 Curies
Temps écoulé.	une semaine

La radioactivité et sa période de demi-valeur. (Unité: Heure)

* Les noms des éléments seront affichés dans l'ordre suivant.

H3 (Hydrogène lourd) 1,0741512E5
C14 (Carbone 14) 5,01948E7
CO60 (Cobalt 60) 64103,88
CO57 (Cobalt 57) 6480
I125 (Iode 125) 1440
I131 (Iode 131) 193,2
CS137 (Cesium 137) 262800
CR51 (Krypton 51) 667,2
F18 (Fer 18) 1,8283
P32 (Phosphore 32) 342,72

Opération:

RUN↵

```
1=TIME, 2=AFTER, 3=BEFORE
```

En spécifiant 1, on obtient l'estimation du temps écoulé à partir de l'intensité actuelle de la radiation.
En spécifiant 2, on obtient l'estimation de l'intensité de la radiation à la fin de la période mesurée.
En spécifiant 3, on obtient l'estimation de la l'intensité initiale de la radiation à partir de l'intensité actuelle et du temps écoulé.
Sélectionnons 1.

1↵

```
SELECT ISOTOPE!
```

Appuyer sur la touche ENTER jusqu'à ce que le nom d'élément CR51 apparaisse à l'affichage.

ENTER

```
H3
```

⋮

ENTER

```
CR51
```

Appuyer alors sur la touche ↵ et effectuer l'opération suivante.

↵

```
INITIAL=?
```

Introduire 100 Curies comme intensité initiale de la radiation du Krypton 51.

100↵

```
PRESENT=?
```

L'intensité actuelle de la radiation est de 50 Curies.

50↵

```
TIME= 667.2
```

On estime que la radiation prend environ 27,8 jours (667,2 heures ÷ 24) pour descendre de 100 Curies à 50 Curies.

↵

```
1=TIME, 2=AFTER, 3=BEFORE
```

Obtenir ensuite la mesure de la radiation après écoulement d'un laps de temps spécifié à partir de l'intensité actuelle de la radiation.

```
2↵                SELECT ISOTOPE!
ENTER             H3
ENTER             C14
                    :
```

Sélectionner le krypton 51.

```
ENTER             CR51
```

Passer à l'opération suivante.

```
↵                 INITIAL=?
```

La mesure initiale de la radiation est de 100 Curies.

```
100↵              TIME=?
```

Fixons dans notre cas le temps écoulé à deux semaines (24 heures x 14 jours).

```
24*14↵            AFTER= 70.53459231
```

L'affichage montre que l'intensité estimée de la radiation est d'environ 70,5 Curies au bout de deux semaines.

Pour obtenir la mesure initiale de la radiation à partir de l'intensité de la radiation actuelle et du temps écoulé, appuyer sur les touches de la manière suivante.

```
↵                 1=TIME,2=AFTER,3=BEFORE
3↵                SELECT ISOTOPE!
ENTER             H3
  :                 :
```



```
ENTER             CR51
↵                 PRESENT=?
```

Introduire l'intensité actuelle de la radiation.

```
70.5↵             TIME=?
```

Mettre le temps écoulé à une semaine (24 heures X 7 jours).

```
24*7↵             BEFORE= 83.94368628
```

L'affichage indique que l'intensité initiale de la radiation est estimée à environ 83,9 Curies.

Appuyer sur la touche BRK pour terminer le programme.

■ La commande GOTO de spécification de destination

```
100 ON X GOTO 120,190,260
    └──┘      └──────────┘
Commande ON ~ GOTO   Destination
```

La commande à ligne 100 indique le saut inconditionnel à la ligne 120 si X a pour valeur 1, à la ligne 190 s'il a pour valeur 2 et à la ligne 260 s'il a pour valeur 3.

La valeur de X est introduite à la ligne 70.

```
70 INPUT "1=TIME,2=AFTER,3=BEFORE";X
```

Par exemple, 2 sera stocké dans X si nous choisissons la deuxième opération (quelle est l'intensité de la radiation après un certain laps de temps?). L'exécution du programme est amené à la ligne (ligne 190) spécifiée par la seconde destination à la ligne 100. La commande ON ~ GOTO peut ainsi spécifier le numéro de la ligne à exécuter par la valeur affectée à la variable.

```
10 INPUT X
20 ON X GOTO 100,200,300 ──→ Saute à la ligne 100 si X a pour valeur 1, à la ligne 200 s'il a pour valeur 2 et à la ligne 300 s'il a pour valeur 3.
30 PRINT "RE-ENTER
   "
```

```
 40 GOTO 10
100 PRINT "BOOK"
110 GOTO 10
200 PRINT "WATCH"
210 GOTO 10
300 PRINT "RADIO"
310 GOTO 10
```

Si la valeur de X est supérieure au numéro de la destination, ou si c'est une fraction, la commande ON ~ GOTO provoquera l'exécution de la ligne suivante.

> La commande ON ~ GOTO a pour fonction la spécification de la destination par le contenu de la variable placée après ON. (Voir page 240)

■ Exécution simultanée d'opérations similaires (GOSUB/RETURN)

```
130 GOSUB 440
 :
200 GOSUB 440
 :
270 GOSUB 440
 :   Commande GOSUB   Noméro de ligne
 :
440 REM SELECT DATA   <- Exécution d'un sous-programme à l'aide de GOSUB
450 PRINT"SELECT ISOTOPE!"
 :
530 RETURN
    Commande RETURN
```

L'instruction GOSUB 440 est introduite à chacune des lignes 130, 200 et 270. Il y a en fait trois cas où des isotopes radioactifs sont sélectionnés dans ce programme. Cette opération est destinée à choisir 1 élément parmi un grand nombre d'autres éléments. Si nous essayons de créer un programme séparé pour chaque cas, il en résultera donc un programme extrêmement long.

Il serait donc bien commode de pouvoir combiner les programmes, et aussi de pouvoir les stocker afin d'être en mesure de les appeler quand c'est nécessaire. Les programmes des lignes 440 – 530 sont des sous-programmes créés dans ce but.


```
440 REM SELECT DATA
450 PRINT "SELECT I
    SOTOPE!"
460 Y=0
470 FOR I=1 TO N
480 PRINT M$(I);
490 Z$=INKEY$:IF Z$
    ="" THEN 490
500 IF Z$=CHR$(13)
    THEN PRINT :Y=T
    (I): GOTO 530
510 PRINT :NEXT I
520 GOTO 460
530 RETURN
```

} Sous-programmes

Les programmes créés dans ce but se terminent toujours par une commande RETURN.

Autrement dit, les exécutions qui vont de la ligne spécifiée à l'aide de GOSUB jusqu'à la commande RETURN sont considérées comme un programme séparé appelé sous-programme.

```
 60 R1=0.2:R2=0.6:R
    3=A
 70 GOSUB 210
 80 D=D+P
 90 R1=0.4:R2=0.9:R
    3=B
100 GOSUB 210
110 D=D+P
120 R1=0.4:R2=0.8:R
    3=C
130 GOSUB 210
140 D=D+P
150 E=E+D
```

Sous-programme (210~RETURN)

La commande RETURN renvoie l'exécution à l'instruction ou à la ligne qui suit celle à laquelle la commande GOSUB a été exécutée.

Dans l'exemple ci-dessus, l'exécution saute à la ligne suivante, puisqu'il n'y a pas d'instruction correspondant à GOSUB 210.

```
10 A=7
20 GOSUB 100
30 PRINT "C=";C
```

```
 40 END
100 C=A*2
110 B=A/2
120 PRINT "B=";B
130 RETURN
```

Etant donné qu'il y a une commande RETURN, les valeurs de B et C seront toutes deux affichées ici.

RUN↵

```
B= 3.5
C= 14
```

↵

```
10 A=100
20 B=250
30 RETURN
```

S'il y a seulement une commande RETURN et pas de commande GOSUB, il se produira une erreur GS ERROR.

RUN↵

```
GS ERROR P0-30
```

La commande GOSUB peut spécifier non seulement des numéros de lignes mais encore des zones de programme.
Supposons par exemple que le programme suivant se trouve dans la zone de programme 0.

```
10 A=10
20 GOSUB PROG 1 ——→ Saute à la zone de programme 1 (sous-programme) et revient après exécution du programme.
30 PRINT B
40 END
```

Supposons aussi que le programme suivant est dans la zone de programme 1.

```
10 B=A*2/5
20 RETURN
```

La valeur de B sera calculée et affichée si le programme est exécuté à partir de la zone de programme 0.

RUN↵

```
4
```

La commande GOSUB permet aussi d'appeler un sous-programme parmi les différents autres sous-programmes.


```
10 A=15
20 GOSUB 100 ——→ Appelle le sous-programme débutant à la ligne 100.
30 PRINT "C=";C
40 END ——→ Les programmes situés aprè la ligne 100 seront exécutés ici s'il n'y a pas de commande END et il se produira une erreur GS ERROR.
100 REM SUBROUTINE-
    1
110 B=A*A*A
120 GOSUB 1000 ——→ Appelle le sous-programme débutant à la ligne 1000.
130 RETURN
1000 REM SOBROUTINE-
     2
1010 C=SQR(B)
1020 RETURN
```

$\sqrt{15^3}$ sera calculé et affiché ici.

RUN↵

```
C= 58.09475019
```

Présenté sous une forme facile à comprendre, le programme sera comme indiqué ci-dessous.

```
10 A=15
20 GOSUB 100
30 PRINT "C=";C
40 END
```

```
100 REM SUBROUTINE-
    1
110 B=A*A*A
120 GOSUB 1000
130 RETURN
```

```
1000 REM SOBROUTINE-
     2
1010 C=SQR(B)
1020 RETURN
```

Les sous-programmes appelés à l'aide de la commande GOSUB sont liés les uns aux autres de cetté manière. Le FX-750P permet de spécifier jusqu'à 12 niveaux de sous-programmes. S'il y a plus de 12 niveaux cela provoquera une erreur NO ERROR.

```
10 GOSUB 20 —— 1
15 END
20 GOSUB 30 —— 2
25 RETURN
30 GOSUB 40 —— 3
35 RETURN
40 GOSUB 50 —— 4
45 RETURN
50 GOSUB 60 —— 5
55 RETURN
60 GOSUB 70 —— 6
65 RETURN
```

```
 70 GOSUB 80 ——7
 75 RETURN
 80 GOSUB 90 ——8
 85 RETURN
 90 GOSUB 100——9
 95 RETURN
100 GOSUB 110——10
105 RETURN
110 GOSUB 120——11
115 RETURN
120 GOSUB 130——12
125 RETURN
130 GOSUB 140——13 ┐→ A partir de là on ne peut plus utiliser GOSUB.
135 RETURN                │
140 GOSUB 150——14 ┘
145 RETURN
```

La commande GOSUB/RETURN sert à exécuter le programme à l'instruction ou à la ligne qui suit la ligne qui contient la commande GOSUB, après avoir appelé et exécuté un sous-programme. (Voir page 233)

▪ La commande END

```
110 END
    └─ commande END
```

Si vous souhaitez terminer un programme, introduire END à l'endroit désiré. Introduire aussi END à la fin d'un programme similaire à celui de la ligne 110. (L'absence de END n'affectera pas ici l'exécution, mais il vaut mieux prendre l'habitude d'introduire END.)

La commande END arrête l'exécution d'un programme de la même manière que la commande STOP. Cependant, lorsqu'on utilise la commande STOP, on peut reprendre l'exécution du programme en utilisant la commande CONT, tandis que cette opération n'est pas possible si l'on se sert de la commande END. On trouvera ci-dessous un exemple d'utilisation de la commande END.


```
10 PRINT "ABC";
20 GOTO 50
30 PRINT "GHI"
40 END
50 PRINT "DEF";
60 GOTO 30
```

RUN↵	ABCDEFGHI
↵	READY P0

Si ce programme est exécuté, 40 END étant effacé, DEFGHI apparaîtra continuellement à l'affichage à chaque fois qu'on appuiera sur la commande ↵. Appuyer sur [BRK] pour terminer ce programme.

RUN↵	ABCDEFGHI
↵	DEFGHI

La commande END a pour fonction de terminer l'exécution d'un programme. (Voir page 229)

▪ La fonction LOG

```
160 T=Y*LOG(C/B)/LOG(1/2)
          └─ Fonction LOG ─┘
```

A la ligne 160, le temps écoulé est obtenu à partir des intensités initiale et actuelle de la radiation d'un corps radioactif. On se sert pour cela de la fonction LOG. La fonction LOG permet le calcul des logarithmes naturel; en mathématiques, on l'exprime sous la forme $\log_e x$.

```
10 FOR I=1 TO 10
20 X=LOG(I
30 PRINT "LOG(";I;
   ")=";X
40 NEXT I
```

```
RUN↵                LOG( 1)= 0
↵                   LOG( 2)= 0.6931471806
⋮                       ⋮
↵                   LOG( 10)= 2.302585093
```

Les valeurs à calculer à l'aide de la fonction LOG doivent être conformes à: 0 < expression numérique. Dans le cas suivant, une erreur MA ERROR apparaîtra à l'affichage.

```
10 A=-5
20 X=LOGA
30 PRINT X
```

```
RUN↵                MA ERROR P0-20
```

De même que la fonction LOG calcule le logarithme naturel $\log_e x$, la fonction LGT calcul le logarithme ordinaire $\log_{10} x$.

```
10 FOR I=1 TO 10
20 X=LGTI
30 PRINT "LGT(";I;
   ")=";X
40 NEXT I
```

```
RUN↵                LGT( 1)= 0
↵                   LGT( 2)= 0.3010299957
⋮                       ⋮
↵                   LGT( 10)= 1
```

La fonction LOG obtient le logarithme naturel des valeurs numériques données. (Voir page 269)

4-10 Jeu: devinez les chiffres

Voici un jeu vous amusera peut-être lorsque vous faites une pause au cours de votre travail ou de vos études. Il est relativement simple il s'agit de deviner le chiffre entre 1 et 20 que le FX-750P a choisi au hasard.
L'ordinateur choisit un chiffre grâce à la fonction de nombre aléatoire de ce programme. Une fois que vous aurez maîtrisé ce jeu, vous devriez être capable de trouver le chiffre en moins de quatre réponses.

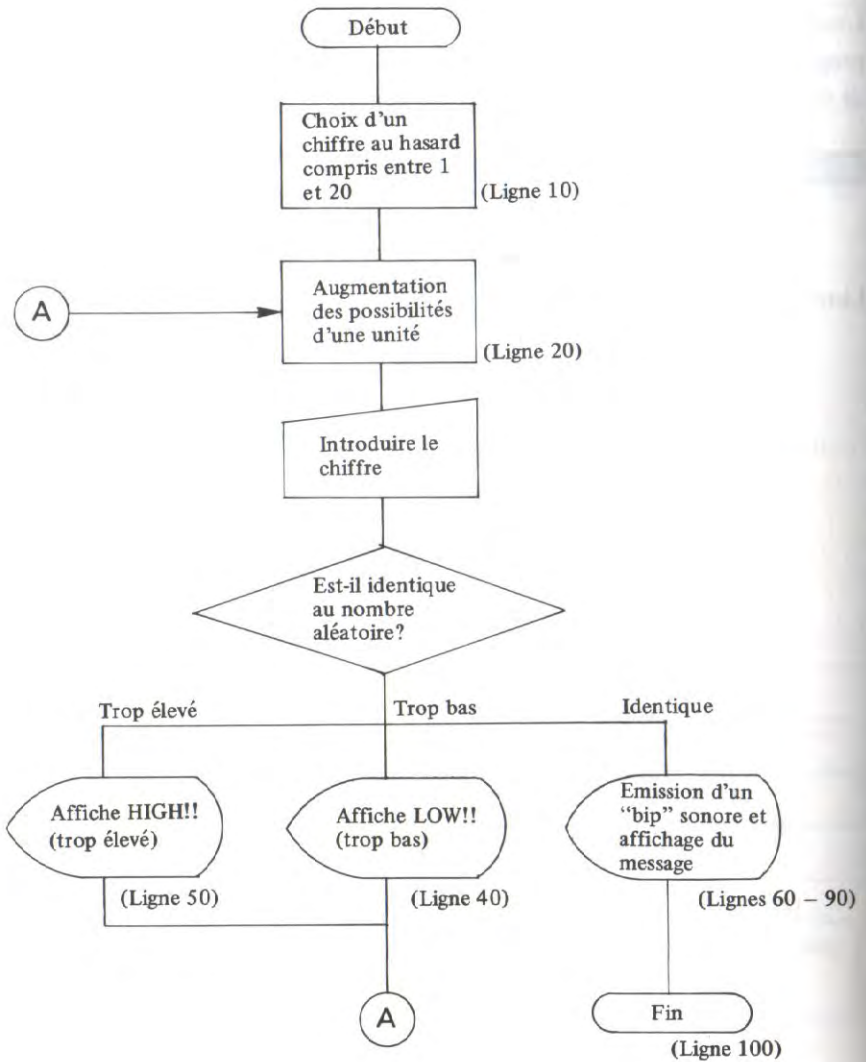
Commandes et fonctions

RND、INT、BEEP、WAIT

Liste de programme

```
10 A=INT(RND*20+1)
   :C=0
20 C=C+1
30 INPUT "NUMBER!.
   .";B
40 IF B<A THEN PRI
   NT "LOW!!: ";:
   GOTO 20
50 IF B>A THEN PRI
   NT "HIGH!!: ";:
    GOTO 20
60 BEEP
65 WAIT 30
70 PRINT "CONGRATU
   LATION!!"
80 PRINT " YOU GUE
   SSED IT!!"
90 PRINT "YOU GOT
   IT IN";C;"TRIES
   "
100 END
```

Organigramme



Deviner et introduire le chiffre compris entre 1 et 20 choisi par le FX-750P.

Opération:

RUN ↵

```
NUMBER!..?
```

Proposer le chiffre 10 par exemple.

10 ↵

```
HIGH!!: NUMBER!..?
```

Puisque 10 est trop élevé, essayons 3.

3 ↵

```
LOW!!: NUMBER!..?
```

3 étant trop bas, la réponse juste doit se situer entre 3 et 10; essayons 5.

5 ↵

↵

```
CONGRATULATION!!
YOU GUESSED IT!!
```

(vous avez deviné)

Vous avez réussi en 3 essais.

```
YOU GOT IT IN 3TRIES
```

■ La fonction RND

```
10 A=INT(RND*20+1):C=0
      └─ Fonction RND
```

La fonction RND peut servir à différents usages sans qu'on ait besoin de la modifier; on l'utilise entre autres pour générer des nombres aléatoires compris dans des limites spécifiées (0 à 100, 100 à 200, etc.) en créant une expression numérique comme indiqué à la ligne 10. On obtient un nombre aléatoire plus grand que 1 et plus petit que 21 en générant des nombres aléatoires à partir de 0 jusqu'à moins de 20 et en ajoutant 1.
Nous utiliserons ici un programme simple pour montrer comment un nombre aléatoire est généré.

```
10 A=RND ──────→ Génére un nombre aléatoire et le stocke dans A.
20 PRINT A ────→ Affiche le nombre aléatoire stocké dans A.
30 GOTO 10
```

```
RUN↵
```

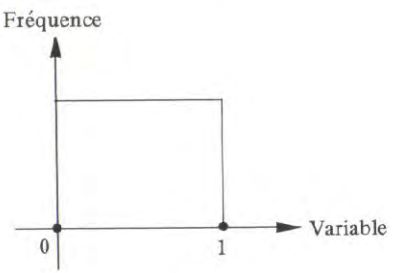
```
0.9828790089
```

```
↵
```

```
0.4737026837
```

⋮

On appelle nombres aléatoires uniformes les nombres aléatoires affichés successivement comme indiqué ci-dessus. Leur utilisation est très répandue pour les jeux et les simulations. (Les nombres aléatoires uniformes sont des nombres aléatoires générés de façon uniforme.)



Afin d'établir des références, un contrôle du EX-750P a été effectué avec les résultats suivants.

```
10 STAT CLEAR
20 FOR I=1 TO 1000
30 STAT RND
40 NEXT I
50 PRINT "MEAN=";S
   UMX/CNT
60 PRINT "SD=";SDX
```

```
RUN↵
```

```
MEAN= 0.490399328
```

La moyenne de 1000 nombres aléatoires

```
↵
```

```
SD= 0.2837680059
```

Ecart type

Utiliser l'expression suivante pour créer des nombres aléatoires compris dans les limites de 15 ≦ valeur numérique ≦ 50.

```
10 R=INT(RND*35+1)+15
20 PRINT R
```

La fonction RND génère des nombres aléatoires comprenant jusqu'à 10 chiffres, plus grands que 0 mais plus petits que 1 ($0 < X < 1$). (Voir page 274)

■ La fonction INT

```
10 A=INT(RND*20+1):C=0
     └─ Fonction INT
        └─ La partie nombre entier doit être extraite
```

A la ligne 10, un nombre fractionnaire compris entre 1 et 20 est généré et converti en nombre entier.
En plus des expressions numériques, dans les parenthèses on peut aussi utiliser des valeurs numériques et des variables.

Nous allons maintenant voir comment opère la fonction INT.

```
10 A=RND*10
20 PRINT A;",";INT
   (A)
30 GOTO 10
```

La fonction RND génère un nombre aléatoire plus petit que 1 et plus grand que 0 et le résultat est multiplié par 10.
Affichage du nombre aléatoire multiplié par 10 et de la valeur numérique après exécution de la fonction INT.

RUN↵

```
 5.737355616,  5
 1.375856229,  1
```

↵

Vous remarquerez que la partie entière de la valeur numérique a été extraite et affichée à droite.
Quand la valeur numérique est un nombre négatif, la fonction INT donne un entier inférieur d'une unité à l'entier de la valeur numérique.
Modifier le programme précédent de façon à ce qu'un nombre négatif apparaisse à la ligne 10 et exécuter le programme.

```
10 A=-RND*10
20 PRINT A;",";INT
   (A)
30 GOTO 10
```

RUN↵

```
-4.314976866,-5
-9.389450762,-10
```

↵

L'affichage montre que le nombre entier à droite est inférieur d'une unité à la partie nombre entier de la valeur numérique.

> La fonction INT donne le plus grand nombre entier qui ne dépasse pas la valeur numérique donnée. (Voir page 270)

■ La commande BEEP

```
60 BEEP
   └─Commande BEEP
```

A la ligne 60, un avertisseur sonore se déclenche quand le nombre correct est introduit. On peut produire des effets sonores intéressants avec la commande BEEP, car la tonalité change selon qu'on choisisse BEEP (ou BEEP 0) ou BEEP 1.
On obtiendra un effet sonore intéressant en modifiant, par exemple, la ligne 60 comme indiqué ci-dessous.

```
60 BEEP 0:BEEP 1:B
   EEP 0:BEEP 1
```

> La commande BEEP peut engendrer des tonalités hautes ou basses. BEEP (ou BEEP 0) produit une tonalité basse et BEEP 1 produit une tonalité haute. (Voir page 223)

■ Contrôle du temps d'affichage (commande WAIT).

```
65 WAIT 30
   │    └─ Indication du temps d'affichage (n)
   └─ Commande WAIT
```

Le temps d'affichage des données est établi à l'aide de la commande WAIT à la ligne 65. La commande WAIT peut établir de cette manière un temps d'affichage optionnel pour les données à sortir par l'intermédiaire de la commande PRINT.
La durée établie à l'aide de la commande WAIT sera à peu près égale à 0,05 secondes × chiffre indiquant le temps (*n*).

```
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
```

Dans le programme ci-dessus, on doit appuyer sur les touches ↵ ou ENTER à chaque fois qu'une valeur numérique est affichée.

```
 5 WAIT 10
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
```

Quand on utilise la commande WAIT, des valeurs numériques sont automatiquement affichées en ordre séquentiel. Pour changer le temps établi par la commande WAIT, il suffit d'exécuter une autre commande WAIT.

```
 5 WAIT 10 ——————→ Etablit le temps d'affichage.
10 FOR I=1 TO 10
20 PRINT I
30 NEXT I
40 WAIT 50 ——————→ Modifie le temps d'affichage.
50 FOR I=1 TO 10
60 PRINT I
70 NEXT I
```

La commande WAIT peut utiliser tout chiffre conforme à: $0 \leqq n \leqq 999$. Si un chiffre supérieur à 999 est établi, le programme se déroulera comme s'il n'y avait pas eu de commande WAIT.

Si l'on appuyait sur les touches ↵ ou ENTER pendant un affichage de données, de nouvelles données seraient affichées, même si le temps établi par la commande WAIT ne s'était pas encore complètement écoulé. D'autre part, le temps sera maintenu, même si on fait sauter l'exécution à une zone de programme différente à l'aide des commandes GOTO PROG ou GOSUB PROG si la commande WAIT est utilisée dans la zone de programme P0.

La commande WAIT établit le temps d'affichage.(Voir page 252)

CHAPITRE 5

DETAIL DES COMMANDES ET FONCTIONS

Symboles

On trouvera ci-dessous la liste des indications utilisées pour décrire la grammaire.

- Les mots en caractères gras . . . doivent toujours être écrits sous cette forme.
- $\begin{Bmatrix} \times\times\times\times \\ \bigcirc\bigcirc\bigcirc\bigcirc \end{Bmatrix}$ . . . . . Pour choisir un des eléments pris entre les accolades $\{\ \}$ et l'écrire.
- $\begin{bmatrix} \times\times\times\times \\ \bigcirc\bigcirc\bigcirc\bigcirc \end{bmatrix}$ . . . . . Les éléments entre crochets $[\ ]$ peuvent être omis.
- $\bigcirc\bigcirc\bigcirc\bigcirc^*$ . . . . . Les éléments accompagnés d'un astérisque * en haut et à droite peuvent être écrits de façon répétitive.
- Expression numérique . . . . Désigne une valeur numérique, une expression de calcul ou une variable numérique comme 10, 2 + 3, A, S * Q, etc.
- Chaîne de caractères . . . . . Désigne une constante de caractères ou une variable de caractères comme "ABC", X$, N$ + M$, etc.
- Paramètre . . . . . Désigne un élément accompagnant une commande.
- Ⓟ . . . . . Peut être exécuté dans un programme.
- Ⓜ . . . . . Ne peut être exécuté que manuellement.
- Ⓐ . . . . . Peut être exécuté soit manuellement soit dans un programme.
- Ⓕ . . . . . Fonction qui peut être exécutée soit dans un programme soit manuellement.

Exemple:

DATA [data] [, [data]]*

Etant donné que toutes les données ressortent avec des crochets [ ], on pourra aussi se contenter d'écrire simplement "DATA". Etant donné que, [data] est accompagné de [ ]*, on pourra écrire cet élément de façon répétitive. On pourra donc écrire: "DATA data, data, ...". Si l'on oublie le premier [data], on peut aussi écrire "DATA, data, data, ...".

Exemple:

GOTO $\begin{Bmatrix} \textbf{numéro de ligne} \\ \textbf{PROG numéro de la zone de programme} \end{Bmatrix}$

Ceci exprime les deux formats suivants.

1) GOTO numéro de ligne

2) GOTO PROG numéro de la zône de programme

5-1 Commandes manuelles

CONT Ⓜ

Fonction:

Continue l'exécution d'un programme temporairement arrêté.

Explication:

Reprend l'exécution à partir du numéro de ligne qui suit la ligne à laquelle le programme a été arrêté à l'aide de la commande STOP ou de la touche STOP .

Exemple:

Insertion de la commande STOP à des endroits choisis pour simplifier le dépistage d'erreurs lorsque l'on crée un programme.

Supposons que nous créons un programme comme le suivant pour obtenir le volume d'un cylindre.

```
10 INPUT "R";R
20 INPUT "H";H
30 D=R^2*PI
40 STOP
50 A=D*H
60 PRINT "A=";A
70 END
```

Exécuter ce programme à l'aide de la commande RUN.

RUN ↵

```
R?
```

Commencer par introduire 10, par exemple, lorsque l'ordinateur demande le rayon du cylindre.

10 ↵

```
H?
```

L'ordinateur demande maintenant la hauteur H, introduisons par exemple 20.

20 ↵

```
STOP P0-40
```

La ligne 30 est exécutée et l'affichage montre que l'exécution a été arrêtée à la ligne 40 avec la commande STOP.
Vérifier si les lignes de programme antérieures à la ligne 40 ont été exécutées correctement.
Affichage de la variable D qui donne la surface circulaire d'un cylindre calculée à la ligne 20.

D [ENTER]

```
314.1592654
```

Vérifier l'exactitude de la réponse en comparant la valeur de la variable D à la surface d'un cercle dont le rayon est 10, surface qui a été vérifiée par un calcul manuel.
Redémarrer l'exécution du programme après la ligne 40 en appuyant sur CONT [↵] .
Le volume sera affiché quand les derniers résultats auront été obtenus et que le programme sera terminé.

CONT [↵]

```
A= 6283.185307
```

La commande STOP est insérée entre les sections de ce programme pour vérification et dépistage d'erreurs.
La commande CONT peut servir à redémarrer l'exécution d'un programme arrêté à l'aide de la touche [STOP] . Cependant, il se produira une erreur FC ERROR si la commande CONT est utilisée en cours de correction de programme ou pendant l'addition ou l'effacement d'une ligne à l'aide de la commande EDIT tandis que le programme est arrêté par la commande STOP ou la touche [STOP]. (Se reporter à la section qui concerne STOP).

DELETE

Ⓜ

{ [numéro de la ligne de départ / numéro de ligne] – [numéro de la ligne finale / numéro de ligne] }

Fonction:

Efface les lignes de programme.

Paramètre:

Numéro de ligne: $1 \leqq$ numéro de ligne < 10000

Explication:

Ne peut pas être utilisé pendant la spécification d'un mot de passe.

Exemple

DELETE 20 [↵] (efface seulement la ligne 20)
DELETE 50–100 [↵] (efface de la ligne 50 à la ligne 100)
DELETE 200– [↵] (efface toutes les lignes à partir de la ligne 200)
DELETE –80 [↵] (efface de la première ligne à la ligne 80)

EDIT [numéro de ligne] Ⓜ

Fonction:

Met la zone de programme désignée dans un mode qui perment la modification et la correction du programme et affiche celuici.

Paramètre:

Numéro de ligne: $1 \leqq$ numéro de ligne < 10000

Explication:

1) Ne peut pas être utilisé pendant la spécification d'un mot de passe.
2) Le mode EDIT sera annulé dans les cas suivants.
   1. Quand on appuie sur les touches [BRK] ou [CLS] .
   2. Si une erreur d'opération a été commise.
   3. Si le programme à afficher n'existe plus.
   4. Quand le courant est coupé (y compris alimentation automatique).

Exemples

EDIT ↵ (Affichage à partir de la première ligne.)
EDIT 50↵ (Affichage à partir de la ligne 50.)

LIST Ⓐ

({ [numéro de la ligne de départ] [–[numéro de la ligne finale]] / ALL / V })

LLIST

({ [numéro de la ligne de départ] [–[numéro de la ligne finale]] / ALL / V })

Fonction:

LIST: Affichage des contenus du programme situé dans la zône de programme désignée.

LLIST: Impression des contenus du programme situé dans la zone de programme désignée.

Paramètre:

Numéro de ligne: $1 \leqq$ numéro de ligne < 10000

ALL: Affichage (ou impression) de tous les contenus des programmes des zones de programme P0 à P9 en ordre séquentiel.

V: Affichage (ou impression) des noms des variables enregistrées et des noms des variables déclarées.

Explication:

1) Si l'on omet le paramétre, l'affichage (ou l'impression) du programme list commencera à la première ligne.
2) En exécutant la commande LIST, les contenus des programmes seront automatiquement affichés (ou imprimés) en ordre séquentiel. Appuyer sur ↵ lorsqu'on désire arrêter. Appuyer de nouveau sur ↵ lorsqu'on désire reprendre l'affichage à partir de la ligne suivante.
3) Appuyer sur la touche [BRK] pour arrêter les commandes LIST ou LLIST.

* Ne peut pas être utilisé au cours de la spécification d'un mot de passe.

Exemples

```
LIST          ↵ (affichage à partir de la première ligne)
LIST 20       ↵ (affichage de la ligne 20)
LIST 50—100   ↵ (affichage des lignes 50 à 100)
LIST 200—     ↵ (affichage à partir de la ligne 200)
LIST —80      ↵ (affichage de la première ligne à la ligne 80)
```

Ⓐ

LOAD [ALL] ["nom de fichier"]

LOAD ["nom de fichier"] [{ ,A / ,M }]

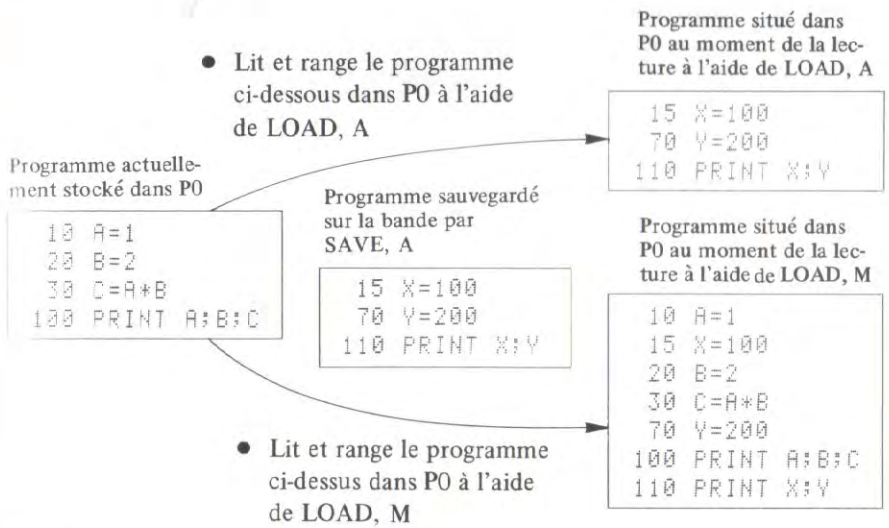
Fonction:

Chargement dans l'ordinateur d'un programme stocké sur une cassette.

Paramètre:

ALL:	Lecture des programmes de toutes les zones de programme de l'ordinateur.
Nom de fichier:	Chaîne de caractères (peut être omise) de 1 ≦ nom de fichier ≦ 8 caractères.
,A:	Charge un programme sauvegardé avec le format de code ASCII.
,M:	Fusionne avec le programme principal les programmes sauvegardés avec le format de code ASCII.

Dans le cas d'une opération LOAD ordinaire l'ancien programme sera perdu avec ,A tandis qu'il sera maintenu avec LOAD, M ou LOAD "nom de fichier", M, tant que les numéros de ligne seront différents.



Explication:

1) Si le nom de fichier est omis, lit le premier programme de même format qui apparaît après exécution de la commande SAVE.
2) Ne charge que les programmes ayant le même format que SAVE.
   Les commandes de même format correspondent ainsi: LOAD pour les programmes sauvegardés par SAVE et LOAD ALL pour les programmes sauvegardés par SAVE ALL. Les combinaisons de LOAD et SAVE sont indiquées dans le tableau suivant.

	LOAD	LOAD "nom de fichier"	LOAD ALL	LOAD ALL "nom de fichier"	LOAD ,A	LOAD "nom de fichier" ,A	LOAD ,M	LOAD "nom de fichier" ,M
SAVE	○	×	×	×	×	×	×	×
SAVE "nom de fichier"	○	○	×	×	×	×	×	×
SAVE ALL	×	×	○	×	×	×	×	×
SAVE ALL "nom de fichier"	×	×	○	○	×	×	×	×
SAVE ,A	×	×	×	×	○	×	○	×
SAVE "nom de fichier" ,A	×	×	×	×	○	○	○	○

○ – peut être chargé. × – ne peut pas être chargé.

3) Mot de passe

Si un programme muni d'un mot de passe est stocké sur une cassette, le mot de passe aussi est stocké. On trouvera ci-dessous des informations pour le chargement des programmes munis d'un mot de passe.

1. LOAD peut être exécuté quand un mot de passe n'est pas prévu pour le FX-750P. Dans ce cas, le mot de passe que l'on charge devient le mot de passe du FX-750P.
2. Quand le mot de passe prévu dans le FX-750P est identique au mot de passe du programme à stocker, le programme peut être stocké tel qu'il est. Dans ce cas, le mot de passe restera aussi.
3. Si le mot de passe du FX-750P et celui du programme à charger sont différents, une erreur **PR ERROR** est affichée dès que le chargement démarre et il ne peut pas continuer.

Exemple

```
LOAD "ABC" ↵
LOAD ALL ↵
```


NEW [ALL] Ⓐ

Fonction:

Efface le programme dans la zone de programme spécifiée en cours. Efface tous les programmes et variables.

Paramètre:

ALL: Efface tous les programmes et variables (y compris les variables statistiques) de toutes les zones de programme (P0 à P9).

Explication:

1) La commande NEW efface programme de la zone de programme désignée mais n'efface pas les variables. Elle ne peut pas être utilisée quand un mot de passe est assigné.
2) La commande NEW ALL accomplit ce qui suit.
   1. Met toutes les variables numériques à 0.
   2. Met toutes les variables de caractères à " " (chaîne vide).
   3. Libère les tableaux déclarés par la commande DIM.
   4. Efface toutes les variables enregistrées.
   5. Met ANGLE à 0 (DEG).
   6. Met la zone de programme à P0.

   On peut aussi utiliser NEW ALL lorsqu'on assigne un mot de passe.
   (Se reporter aux sections ANGLE, DIM et PASS)

Exemple

```
NEW ↵
NEW ALL ↵
```

PASS "Mot de passe" Ⓜ

Fonction:

Désigne ou annule un mot de passe.

Paramètre:

Mot de passe: chaîne de caractères de 1 ≦ mot de passe ≦ 8 caractères.

Explication:

1) Un mot de passe est établi pour toutes les zones de programme; il ne peut être libéré que lorsque le mot de passe affecté par la suite correspond à celui qui est déjà introduit. Si les deux mots de passe ne correspondent pas, il se produira une erreur PR ERROR (erreur de protection).
2) La création, la correction, l'annulation et l'effacement des programmes ne sont pas possibles lorsque le mot de passe a été établi avec cette commande (en cas d'utilisation des commandes DELETE, EDIT, LIST, LLIST, NEW ou SAVE, A, il se produira une erreur PR ERROR.)
3) Le mot de passe sera sauvegardé, même si l'alimentation électrique est coupée.
4) Si un programme muni d'un mot de passe est stocké à l'aide des commandes SAVE ou SAVE ALL, le mot de passe aussi sera stocké. Si un programme muni d'un mot de passe est introduit à partir d'une cassette à l'aide des commandes LOAD ou LOAD ALL, le mot de passe aussi sera introduit. Si un programme muni d'un mot de passe différent de celui de l'ordinateur est introduit, il se produira une erreur PR ERROR, (se reporter aux sections concernant SAVE, LOAD et NEW).

Remarque:

Si vous n'arrivez pas à vous souvenir d'un mot de passe affecté, exécutez la commande NEW ALL et videz tous les programmes et variables.

PROG Numéro de zone de programme Ⓐ

Fonction:

Spécifie une zone de programme entre P0 et P9.

Paramètre:

Numéro de zone de programme: 0 ≦ valeur numérique (expression numérique) < 10.

Explication:

Si un numéro spécifié de zone de programme (autre que 0 ≦ valeur numérique (expression numérique) < 10) n'existe pas, il se produira une erreur BS ERROR.

Exemple

PROG 2 ↵

(spécifie la zône de programme P2.)

```
READY P2
```

RUN [numéro de la ligne de départ] Ⓜ

Fonction:

Exécute un programme.

Paramètre:

Numéro de la ligne de départ de l'exécution: 1 ≦ numéro de ligne < 10000.

Explication:

1) Exécute à partir de la première ligne du programme au cas où le numéro de ligne est omis.
2) Si le numéro de ligne spécifié n'existe pas, exécute à partir du numéro de ligne supérieur le plus proche.
3) Ne peut pas être utilisé à l'intérieur d'un programme.

Exemple

RUN ↵
RUN 1000 ↵

SAVE [ALL] ["nom de dossier"] Ⓐ

SAVE ["nom de dossier"], A

Fonction:

Stocke un programme sur une cassette.

Paramètre:

ALL:	Stocke les programmes de toutes les zones de programme sur une cassette.
nom de fichier:	Chaîne de caractères de 1 ≦ nom de fichier ≦ 8 caractères, (peut être omis).
, A:	Stocke le programme qui se trouve dans la zone de programme désignée avec le format de code ASCII. Le stockage avec le format ASCII prendra plus de temps, mais les programmes doivent être stockés avec ce format pour exécuter la commande LOAD,M (voir LOAD). Ne peut pas être utilisé lorsqu'un mot de passe est affecté; en cas d'utilisation, il se produira une erreur PR ERROR.

(Se reporter à LOAD, VERIFY)

Exemple

SAVE "ABC" ↵
SAVE ALL ↵

STAT LIST Ⓐ

STAT LLIST

Fonction:

STAT LIST:	Affiche les statistiques de base.
STAT LLIST:	Imprime les statistiques de base.

Explication:

1) Les statistiques de base sont affichées dans l'ordre suivant.
CNT, SUMY, SUMX, SUMXY, SUMX2 et SUMY2 (voir page 275).
2) Si l'on désire arrêter momentanément l'affichage, appuyer sur ↵ ou ENTER . Appuyer à nouveau pour reprendre l'affichage.
3) En mode PRINT ON utiliser la commande STAT LLIST, puisque, dans ce mode, la commande STAT LIST ne peut pas être utilisée.

Exemple

STAT LIST ↵
STAT LLIST ↵

SYSTEM {P/V} Ⓜ

Fonction:

Affiche l'état d'une zone de programme, le nombre d'octets RAM restants et le nombre d'octets RAM utilisés.

Paramètre:

P: Affiche le nombre d'octets utilisés pour le programme.

V: Affiche le nombre d'octets utilisés pour les variables.

Explication:

SYSTEM ↵

```
P ♥123♥56789 1089BYTES
```

Etat de zone de programme — Nombre d'octets restants

Affiche un symbole représentant un coeur si un programme a déja été écrit dans la zone de programme. Ce symbole clignote dans la zone de programme désignée.

SYSTEM P ↵

```
PROG  1185 BYTES USED
```

Affiche le nombre d'octets utilisés pour le programme (y compris la zone de gestion de programme).

SYSTEM V ↵

```
VAR  1021 BYTES USED
```

Affiche le nombre d'octets utilisés pour les variables (y compris la zone de gestion de variables).

VERIFY ["nom de dossier"] Ⓜ

Fonction:

Vérifie l'état d'un programme ou des données stockées dans une cassette.

Paramètre:

Nom de dossier: Chaîne de caractères de 1 ≦ nom de dossier ≦ 8 caractères, (ceci peut être omis)

Explication:

Si le nom du fichier a été oublié, vérifie le nom du premier fichier qui apparaît après exécution de la commande. Si l'enregistrement n'a pas été correctement exécuté, il se produira une erreur RW ERROR.

Exemple

VERIFY ↵

VERIFY "PRG.5" ↵

5-2 Commandes programmées

ANGLE Expression numérique Ⓐ

Fonction:

Spécifie les unités d'angle.

Paramètre:

Expression numérique: $0 \leqq$ expression numérique < 3

Explication:

ANGLE 0 – DEGREE . . . . Expressions en degrés comme dans 90°, 360°, etc.

ANGLE 1 – RADIAN . . . . Expressions en radians comme dans $\pi/2$, 2π, etc.

ANGLE 2 – GRADE . . . . Expression des angles droits 90° par 100 comme dans 100, 400, etc.

Les limites sont les suivantes:

Degrés: |angle| < 5400°

Radians: |angle| < 30π radians

Grades: |angle| < 6000 grades

* 360° = 2π radians = 400 grades

A la mise sous tension, la commande ANGLE est sur la position ANGLE 0.

Exemple

```
10 INPUT R
20 FOR I=0 TO 2
30 ANGLE I
40 PRINT SINR
50 NEXT I
60 END
```


BEEP [{0 / 1}] Ⓐ

Fonction:

Déclenche un signal sonore.

Paramètre:

0: Tonalité basse

1: Tonalité haute

Explication:

Si le paramètre a été omis, fonctionnera comme si 0 avait été spécifié.

BEEP . . . . déclenche un signal sonore de tonalité basse.

BEEP 0 . . . . Idem.

BEEP 1 . . . . déclenche un signal sonore de tonalité haute.

Exemple

```
10 FOR I=1 TO 10
20 BEEP 0:BEEP 1
30 NEXT I
40 END
```

CHAIN ["nom de fichier"] Ⓐ

Fonction:

Charge un programme spécifié et exécute à partir de la première ligne.

Paramètre:

Nom de fichier: 1 ≦ nom de fichier ≦ 8 caractères.
(Peut être omis)

Explication:

1) Si le nom de fichier a été omis, charge et exécute le premier programme qui apparaît après exécution de la commande.
2) Le programme qui se trouve dans la zone de programme désignée pour le chargement en cours sera effacé.
3) La commande CHAIN ne permet pas l'introduction des programmes stockés à l'aide des commandes SAVE ALL ou SAVE,A.
4) Si le programme à charger comporte un mot de passe, celui-ci aussi sera introduit.
5) L'exécution de cette commande n'efface pas les contenus des variables.

Exemple

```
900 PRINT "PUSH 1-3
    ";
910 K$=INKEY$
920 IF K$="1" THEN
    CHAIN "PROG 1"
930 IF K$="2" THEN
    CHAIN "PROG 2"
940 IF K$="3" THEN
    CHAIN "PROG 3"
950 GOTO 910
```


CLEAR Ⓐ

Fonction:

Efface toutes les variables.

Explication:

Remet les contenus des variables fixes (A ~ Z, A$ ~ Z$) à 0 ou à " " (chaîne vide). Annule aussi l'enregistrement des variables enregistrées et libère les variables de rang définies. CLEAR ne peut pas être utilisée dans la boucle FOR ~ NEXT, car les piles d'emboîtements FOR ~ NEXT seront effacées et il ne sera pas possible de revenir à la commande FOR.

Exemple

```
10 CLEAR
20 DIM A(19)
30 FOR I=0 TO 19
40 INPUT A(I)
50 T=T+A(I)
60 NEXT I
70 FOR I=0 TO 19
80 PRINT A(I)/T*10
   0
90 NEXT I
100 END
```

* Introduire 20 données et obtenir le pourcentage de chacune.

CLS

Ⓐ

Fonction:

Efface tous les affichages et amène le curseur à sa position initiale (extrémité gauche).

Exemple:

```
10 X=10
20 CLS
30 LOCATE X:PRINT
   "*";
40 K$=INKEY$
50 IF K$="4" THEN
   X=X-1:BEEP 0
60 IF K$="6" THEN
   X=X+1:BEEP 1
70 GOTO 20
```

* Si on appuie sur la touche [4] , le "∗" se déplace vers la gauche ; si on appuie sur la touche [6] , il se déplace vers la droite.

Ⓐ

DIM **Nom de variable de tableau [!] (indice) [,nom de variable de tableau [!] (indice)]***

DIM **Nom de variable de tableau $ (indice) [∗ expression numérique] [, nom de variable de tableau $ (indice) [∗ expression numérique]]***

Fonction:

Déclare la définition d'une variable de tableau.

Paramètres:

Nom de variable de tableau:	Utilise n'importe quelle lettre de l'alphabet entre A et Z.
!:	Spécifie un tableau numérique en demie précision.
$:	Spécifie un tableau de caractères.
Indice:	Peut définir jusqu'à un tableau de 2 dimensions. $0 \leqq$ indice < 256
∗ Expression numérique:	Réservé aux chaînes de caractères; spécifie une chaîne de caractères de longueur définie. $0 \leqq$ expression numérique < 80.

Explication:

Tableaux numériques

1) Tableau numérique en simple précision
   Utilise 8 octets par variable et peut stocker une mantisse de 12 chiffres plus un exposant à deux chiffres.
2) Tableau numérique en demie précision
   Demande ! après un nom de variable de tableau. Une variable utilise 4 octets et peut stocker une mantisse de 5 chiffres plus un exposant de deux chiffres.

Tableaux de caractères

1) Tableau de caractères de longueur fixe
   Réclame $ après un nom de variable de tableau et "*expression numérique" est omis. Un tableau de 16 caractères peut être stocké dans une variable et chaque variable utilise 17 octets.
2) Tableau de caractères de longueur définie
   Spécifie la longueur maximale de la chaîne de caractères qui peut être stockée avec "*expression numérique" placé après l'indice; il faut un nombre de caractères + un octet par variable.

Remarque:

- Il se produira une erreur DD ERROR (définition en double) si on a un nombre de variables différent pour un même nom de variable de tableau. Dans ce cas, annuler la définition précédente en insérant une instruction ERASE ou CLEAR devant l'instruction DIM.
- S'il y a une différence dans la forme de spécification de tableau comme dans le cas de D!(5), DS(5) ou D(5), les variables seront classées séparément même si les noms de variables de tableau sont identiques.
- Déclarer toujours à l'aide de l'instruction DIM lorsqu'on utilise un tableau.
- Si le numéro des variables déclarées avec le même nom de variable d'ensemble est le même, on attribuera 0 aux variables numériques et rien aux variables de caractères.

```
10 CLEAR
20 DIM A(0)
30 A(0)=123
40 DIM A(0)
50 PRINT A(0)
```


END Ⓟ

Fonction:

Termine l'exécution d'un programme.

Explication:

Même s'il y a un programme après cette commande, il ne sera pas exécuté. Cette commande efface la boucle de commande GOSUB et les piles d'emboîtements FOR ~ NEXT.

On peut utiliser n'importe quel nombre de commandes END dans un programme.

ERASE Nom de variable [, nom de variable]* Ⓐ

Fonction:

Libère et efface les variables enregistrées définies et les variables de tableau.

Paramètre:

Nom de variable:	Nom de variable enregistrée ou nom de variable de tableau.

Explication:

Cette commande peut effacer les variables enregistrées ou les variables de tableau, confirmées par LIST V, avec un nom de variable pris comme unité. Si un nom de variable n'est pas enregistré, la commande sautera cette ligne et exécutera la ligne suivante.

Cette commande ne peut pas être utilisée dans une boucle FOR ~ NEXT, car la pile d'emboîtements FOR ~ NEXT sera effacée.

Exemple

```
10 ERASE D,E$
20 DIM D(10),E$(15
   )
```

FOR nom de la variable de contrôle = valeur initiale Ⓟ
TO valeur finale [**STEP** incrément]
NEXT nom de la variable de contrôle

Fonction:

Répète l'exécution des instructions entre FOR et NEXT tout en faisant passer les variables de contrôle de leur valeur initiale à leur valeur finale dans les incréments spécifiés.

Paramètres:

Nom de variable de contrôle: Nom de variable numérique. Ne peut pas utiliser de variables de tableau.

Valeur initiale: Expressions numériques ou variables numériques. Les nombres négatifs sont permis.

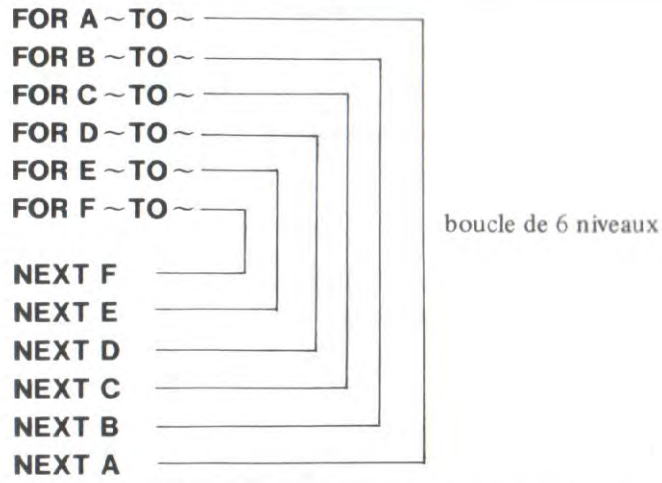
Valeur finale: Expressions numériques ou variables numériques. Les nombres négatifs sont permis.

Incrément: Expression numérique ou variable numérique. Les nombres négatifs sont permis. (Peut être omis.)

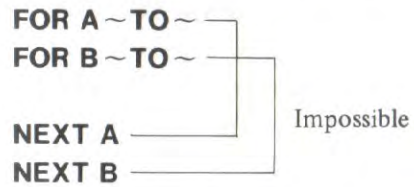
Explication:

1) 1 sera spécifié comme incrément si STEP a été omis.
2) Si la valeur initiale est plus grande que la valeur finale, les instructions qui se trouvent entre FOR et NEXT ne seront exécutées qu'une seule fois.
3) L'emboîtement de la boucle FOR ~ NEXT est possible jusqu'à 6 niveaux.

Structure d'emboîtement



4) L'emboîtement de la boucle FOR ~ NEXT doit être complet.



S'il y a croisement des boucles, comme dans le cas ci-dessus, il se produira une erreur FO ERROR.

5) Il peut y avoir de multiples commandes NEXT pour une seule commande FOR, mais il se produira une erreur FO ERROR si la commande NEXT est exécutée avant la commande FOR ou s'il n'y a pas de commande NEXT.
6) On peut sauter en dehors de la boucle FOR ~ NEXT, mais on ne peut pas sauter dans cette boucle à l'aide d'une instruction GOTO, etc. L'emboîtement continuera même après qu'on ait sauté hors de la boucle tant qu'il ne sera pas mis fin à l'instruction avec la commande NEXT.
7) Les commandes CLEAR et ERASE ne peuvent pas être exécutées dans la boucle FOR ~ NEXT.

(voir CLEAR, ERASE.)

GET ["nom de fichier"] variable [, variable]* Ⓐ

Fonction:

Introduit des données stockées sur une cassette dans des variables à l'aide de la commande PUT.

Paramètre:

Nom de fichier: Chaîne de caractères de 1 ≦ nom de fichier ≦ 8 caractères.
(Peut être omis)

Explication:

1) Introduit le premier fichier de données qui apparaît sur la cassette après exécution de la commande si le nom de fichier est omis.
2) Le nom de variable utilisé lorsqu'on stocke à l'aide de la commande PUT ne doit pas nécessairement correspondre au nom de variable qu'on introduit à l'aide de cette commande.
3) Si le nombre de données lues avec la commande GET est plus important que le données stockées, nombre de l'insuffisance de données stockées entrainera une erreur DA ERROR.

Exemple

GET A: Introduit les données dans la variable A.

GET "TEST" A,B,C: Introduit les données du fichier "TEST" dans les variables A,B et C.

```
P0
  10 A=1:B=2:C=3
  20 PUT "TEST"A,B,C
  30 END
P1
  10 GET "TEST"X,Y,Z
  20 PRINT X,Y,Z
  30 END
```

* Stocke sur une cassette les variables A, B et C avec P0. Les données sont ensuite introduites dans les variables X, Y et Z avec P1 et puis affichées.

GOSUB {Numéro de ligne / PROG numéro de zone de programme} Ⓟ

RETURN

Fonction:

GOSUB: Saute au sous-programme du numéro de ligne ou de la zone de programme spécifiés.

RETURN: Revient du sous-programme à l'instruction qui suit immédiatement GOSUB.

Paramètres:

Destination numéro de ligne: 1 ≦ numéro de ligne < 10000

Numéro de zone de programme: 0 ≦ numéro < 10

Explication:

1) On peut désigner un sous-programme à l'intérieur d'un autre sous-programme et effectuer jusqu'à 12 imbrications.
2) Il se produira une erreur UL ERROR si le numéro de ligne de destination n'existe pas ou s'il n'y a pas de programme dans la zone de programme spécifiée. Il doit toujours y avoir 1 programme dans cette zone.
3) Dans un sous-programme, il peut y avoir de multiples commandes RETURN, mais il doit toujours y en avoir au moins une.
4) Si la commande RETURN est exécutée avant la commande GOSUB, il se produira une erreur GS ERROR.

* Des expressions numériques peuvent être utilisées pour les destinations dans le numéro de ligne ou dans le numéro de zone de programme.

Exemple

```
GOSUB 1000
GOSUB N
GOSUB PROG 9
GOSUB PROG P
```

```
 10 PRINT "START"
 20 GOSUB 100
 30 PRINT "END"
 40 END
100 PRINT "LINE 100
    "
110 RETURN
```

GOTO {numéro de ligne / PROG numéro de zone de programme} Ⓟ

Fonction:

Saute inconditionnellement à la destination spécifiée.

Paramètres:

Numéro de ligne de destination: $1 \leqq$ numéro de ligne < 10000

Numéro de zone de programme: $0 \leqq$ numéro < 10

Explication:

Il se produira une erreur UL ERROR si le numéro de ligne de destination n'existe pas ou s'il n'y a pas de programme dans la zone de programme spécifiée.

* Des expressions numériques peuvent être utilisées pour le numéro de ligne de destination et le numéro de zone de programme.

```
GOTO 150
GOTO Z
GOTO S*10
GOTO PROG 5
GOTO PROG R
```

Exemple

```
 10 INPUT "PUSH 1-3
    ";A
 20 IF A<1 THEN 10
 30 IF A>3 THEN 10
 40 GOTO A*100
 50 END
100 PRINT "LINE 100
    ": GOTO 10
200 PRINT "LINE 200
    ": GOTO 10
300 PRINT "LINE 300
    ": GOTO 10
```


IF expression conditionnelle **THEN** {instruction [: instruction]* / destination de branchement} Ⓟ

[**ELSE** {instruction [: instruction]* / destination de branchement}]

★ Destination de branchement {numéro de ligne de destination / PROG numéro de zone de programme}

Fonction:

Exécute les instructions qui suivent THEN, si une expression conditionnelle est vérifiée exacte. Le programme saute aux instructions qui suivent THEN si ces instructions sont des destinations de bifurcation.

Exécute les instructions qui suivent ELSE, si une expression conditionnelle est vérifiée exacte. Le programme saute aux instructions qui suivent ELSE si ces instructions sont des destinations de bifurcation.

Paramètres:

Expression conditionnelle: (= expression de comparaison):
=, <>, ><, =<, <=, =>, >=

Instruction: On peut utiliser des instructions multiples.

Numéro de ligne: $1 \leqq$ numéro de ligne < 10000

Numéro de zone de programme: $0 \leqq$ numéro < 10

Explication:

1) Si ELSE est omis, la ligne suivante est exécutée lorsque l'expression conditionnelle est fausse.
2) S'il y a deux conditions ou plus, on peut utiliser une instruction IF après THEN ou ELSE.

Remarque:

Avant d'utiliser une variable enregistrée dans une instruction conditionnelle, il faut toujours définir cette variable à l'avance.

Exemple

```
10 INPUT X
20 IF X>10 THEN 10
30 PRINT X*X
```

* Ce programme calcule le carré d'un nombre plus petit que 10.

INPUT ["instruction de message" {;/,}] Ⓟ

nom de variable [, ["instruction de message"] {;/,} nom de variable]*

Fonction:

Introduit des données dans une variable à partir du clavier.

Paramétres:

Instruction de message: Une expression de caractères commençant par une constante de caractères.

Nom de variable: On peut utiliser toutes les variables.

Explication:

1) S'il y a un message, il sera affiché. S'il n'y a pas de message, seul un point d'interrogation "?" sera affiché.
2) Si un " ; " est placé après un message, un " ? " sera affiché, mais il ne le sera pas si un " , " est utilisé à la place.
3) Les touches [ENTER] et ↵ produiront le même effet si on les actionne pendant l'exécution d'une instruction INPUT.
4) Si une instruction INPUT comporte des données multiples, les touches [ENTER] ou ↵ les introduiront une par une.
5) Si l'on appuie sur [ENTER] ou ↵ sans introduire de données, une chaîne vide sera affectée à la variable (chaîne de caractères de longueur 0).
6) Si un caractère ou chaîne vide sont affectés à une variable numérique, une erreur TM ERROR apparaîtra à l'affichage.
7) On peut utiliser des expressions numériques (A ∗ 2, etc.) pour introduire des valeurs numériques.
8) On peut aussi utiliser des expressions de caractères commençant par une constante à caractères pour un message d'introduction.

   Exemple: INPUT "MOUNT"+A$;N
9) Un message d'introduction peut comporter jusqu'à 23 caractères, y compris le "?".
10) Les introductions seront incorrectes si le message d'instruction INPUT est affiché avec un décalage ou avec un 24ème caractère (y compris "?" seulement). Si cela se produit, comme pendant l'exécution d'une analyse de programme, appuyer sur [SHIFT] [CONT] (FC ERROR est affiché) et ramener le message à moins de 23 caractères.

Exemple

1)

```
10 INPUT A
20 B=2*A
30 PRINT B
```

2)

```
10 INPUT "H=";H,"N
   =";N
20 S=N*H/2
30 PRINT S
```

LET { variable numérique = expression numérique / variable à caractère = expression à caractères } Ⓐ

Fonction:

Affecte la valeur de l'expression de droite à la variable de gauche.

Explication:

1) La commande LET peut être omise. (On utilise normalement les instructions d'affectation avec omission de cette commande).
2) Des expressions numériques sont affectées aux variables numériques et des expressions à caractères (chaînes de caractères) aux variables à caractères. Il se produira une erreur TM ERROR si ceci n'est pas respecté.

Exemple

```
10 LET A=123
20 B=456
30 LET C$="CASIO"
40 D$="FX-750P"
50 PRINT A;B
60 PRINT C$;D$
```

LOCATE position de expression numérique Ⓟ

Fonction:

Spécifie la position du curseur.

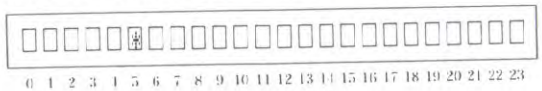
Paramètre:

Position: position horizontale sur l'écran d'affichage.
$0 \leqq$ expression numérique < 24

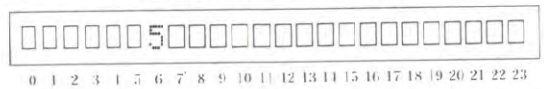
Explication:

1) Spécifie la position du curseur et affiche à partir de cette position s'il y a une instruction PRINT.
2) S'il y a une instruction INPUT après l'instruction LOCATE, un point d'interrogation "?" ou un message seront affichés à la position spécifiée.
3) Comme indiqué ci-dessous, les spécifications de position pourront aller de 0 à l'extrémité gauche à 23 à l'extrémité droite.

Dans le cas de **LOCATE 5 : PRINT"*"**



Dans le cas de **LOCATE 5 : PRINT 5**



Pour l'affichage d'une valeur numérique positive un espace est laissé ouvert au premier emplacement d'affichage pour remplacer le signe "+".

Exemple

```
10 X=11:CLS
20 LOCATE X:PRINT
   "*";
30 LOCATE X:PRINT
   " ";
40 K$=INKEY$
50 IF K$="4" THEN
   BEEP 1:X=X-1:IF
    X<0 THEN X=23
60 IF K$="6" THEN
   BEEP 0:X=X+1:IF
    X>23 THEN X=0
70 GOTO 20
```


* En appuyant sur la touche [4] on déplace le * vers la gauche, en appuyant sur la touche [6] on le déplace vers la droite.

ON expression numérique GOSUB destination [, destination]* Ⓟ

★ Destination { numéro de ligne de destination / PROG numéro de zone de programme }

Fonction:

La commande GOSUB provoque un saut à une destination spécifiée par la valeur de l'expression numérique.

Paramètre:

Numéro de ligne de destination: $1 \leqq$ numéro de ligne < 10000
Numéro de zône de programme: $0 \leqq$ numéro < 10

Explication:

1) Les expressions numériques et les destinations correspondent directement; c'est à dire que le saut se fera à la première destination si la valeur numérique est 1, à la deuxième destination, si elle est 2, et ainsi de suite.

```
ON A GOSUB 100,200,300
           Quand A = 1
           Quand A = 2
           Quand A = 3
```

2) On peut écrire n'importe quel chiffre de destination, pourvu qu'il ne dépasse pas les limites d'une ligne (79 caractères).
3) Il se produira une erreur UL ERROR si le numéro de ligne de destination n'existe pas ou s'il n'y a pas de programme dans la zone de programme spécifiée.
4) Si la valeur de l'expression numérique est inférieure à 1, ou s'il n'y a pas de destination, aucun saut ne sera effectué et l'instruction suivante sera exécutée (voir GOSUB).

Exemple

```
10 INPUT X
20 ON X GOSUB 200,
   300,400
30 GOTO 10
200 PRINT "SUB 200"
   :RETURN
300 PRINT "SUB 300"
   :RETURN
400 PRINT "SUB 400"
   :RETURN
```

ON expression numérique **GOTO** destination [, destination]* Ⓟ

★ Destination { numéro de ligne de destination / PROG numéro de zone de programme }

Fonction:

Effectue un saut inconditionnel à la destination spécifiée par la valeur d'une expression numérique.

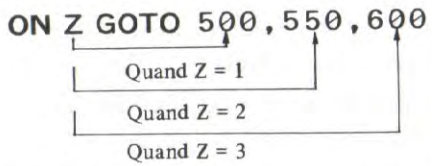
Paramètres:

Numéro de ligne de destination: $1 \leqq$ numéro de ligne < 10000

Numéro de zone de programme: $0 \leqq$ numéro < 10

Explication:

1) Les expressions numériques et les destinations correspondent directement; c'est à dire que le saut se fera à la première destination si la valeur numérique est 1, à la seconde destination si elle est 2, et ainsi de suite.



2) On peut écrire n'importe quel chiffre de destination, pourvu qu'il ne dépasse pas les limites d'une ligne (79 caractères).

3) Il se produira une erreur UL ERROR si le numéro de ligne de destination n'existe pas ou s'il n'y a pas de programme dans la zone de programme spécifiée.

4) Si la valeur de l'expression numérique est inférieure à 1 ou si le numéro de la destination de ligne n'est pas spécifié, la ligne de programme suivante est exécutée sans être branchée.

Exemple

```
10 INPUT A
20 ON A GOTO 100,1
   50,200
30 PRINT "OTHER"
40 GOTO 10
100 PRINT "A=1": GO
    TO 10
150 PRINT "A=2": GO
    TO 10
200 PRINT "A=3": GO
    TO 10
```

PRINT [résultats] [{ , / ; } résultats]* Ⓐ

LPRINT [résultats] [{ , / ; } résultats]*

Function:

PRINT:	Affiche les résultats
LPRINT:	Résultats vers l'imprimante

Paramètre:

Résultats:	fonctions de contrôle de sortie (TAB, USING), expression numérique et expression à caractères

Explication:

1) Un signe + ou un signe – est placé devant la valeur si le résultat est une expression numérique. Le signe +, cependant, apparaîtra sous la forme d'un espacement.

2) Si le résultat est une expression numérique dont la mantisse comprend plus de 10 chiffres, il sera arrondi au 11ème chiffre et affiché. Si c'est une expression numérique $| \geqq 10^{10}$ ou une expression numérique $| < 10^{-3}$, le signe pair (E) et deux chiffres exponentiels seront affichés.

3) Si le résultat est une expression à caractères, le nombre de caractères compris entre les guillemets ne pourra pas dépasser 79 y compris le numéro de ligne, les commandes et les espacements.
4) L'affichage continu de 2 expressions (ou chaînes de caractères) ou plus est possible si l'on ponctue avec une virgule (,) ou un point-virgule (;). Si c'est une virgule qu'on utilise, l'exécution du programme sera interrompue pendant le laps de temps spécifié à l'aide de la commande WAIT et le symbole WAIT s'allumera. Si l'on souhaite reprendre l'exécution du programme pendant la période d'attente, il suffit d'appuyer sur ↵ ou ENTER. Si c'est un point-virgule qu'on utilise, les résultats seront affichés de façon continue et le programme sera exécuté sans interruption (voir WAIT).
5) Quand tous les résultats sont omis (uniquement instruction PRINT), l'affichage est simplement effacé.
6) L'affichage ne s'arrêtera pas si l'instruction PRINT est exécutée pendant l'impression en mode impression (PRINT ON).
7) Lorsqu'il dépasse les limites de l'écran (25 caractères ou plus), l'affichage se déplace vers la gauche.
8) LPRINT envoie à la sortie imprimante les mêmes données que la commande PRINT mais n'entraîne aucune interruption du programme.

Remarque:

TAB ne peut être utilisé que dans le cas d'une commande LPRINT. Si on l'utilise dans une commande PRINT, cela provoquera une erreur SN ERROR.

Exemple

```
PRINT 5^3 ↵
PRINT B ↵
PRINT "X=";X ↵
LPRINT TAB(3);"ANSWER=";A ↵
```


PRINT ON
PRINT OFF

Ⓐ

Fonction:

PRINT ON:	Met en mode PRINT.
PRINT OFF:	Libère le mode PRINT.

Explication:

1) Impression des opérations manuelles.
   Les expressions numériques et les résultats des calculs manuels sont imprimés. En outre il est possible d'imprimer les resultats de commandes manuelles.
2) Impression par l'intermédiaire de la commande PRINT
   L'exécution de la commande PRINT en mode PRINT-ON permet l'impression des contenus de l'affichage.
   (On dispose d'un laps de temps plutôt court pour cet affichage)
3) Impression par l'intermédiaire de la commande INPUT
   Le message et le point d'interrogation sont imprimés pour faciliter l'introduction.
4) Impression en mode analyse
   Il est possible d'imprimer dans l'ordre analysé l'exécution d'une instruction; on dispose ainsi d'une méthode pratique de détection des erreurs de logique grâce à un enregistrement imprimé de l'exécution du programme.
5) Pendant l'exécution d'un programme, les messages d'erreur et la commande STOP sont imprimés.

Le symbole PRTON s'allumera lorsqu'on est en mode PRINT.
Si l'on met le mode PRINT sans qu'une imprimante soit raccordée, cela provoquera une erreur NR ERROR.

Remarque:

La commande STAT LIST ne peut pas être utilisée en mode PRINT ON. La remplacer par la commande STAT LLIST.

Exemple

```
10 PRINT ON              40 PRINT "SIN";I;"
20 ANGLE 0                  =";SINI
30 FOR I=0 TO 360        50 NEXT I
   STEP 15               60 PRINT OFF
```

PUT ["nom de fichier"] variable [, variable]* Ⓐ

Fonction:

Stocke des contenus de variable sur une cassette.

Paramètre:

Nom de fichier: chaîne de caractères de 1 ≦ nom de fichier ≦ 8 caractères.
(Peut être omis) (voir GET)

Exemple

PUT A ..... stocke le contenu de la variable A sur une cassette.

PUT "DATA" X , Y , Z ..... stocke le contenu des variables X, Y et Z sur une cassette sous le nom de fichier "DATA".

```
P0                       20 FOR I=0 TO 9
  10 FOR I=1 TO 10       30 GET A(I)
  20 PUT I               40 NEXT I
  30 NEXT I              50 FOR I=0 TO 9
  40 END                 60 PRINT A(I)
P1                       70 NEXT I
  10 CLEAR :DIM A(9)     80 END
```

* Le programme situé dans P0 stocke 10 données sur une cassette et le programme situé dans P1 introduit 10 données à partir de la cassette dans la variable de tableau A ( ) et les affiche. Dans une utilisation de ce genre, le nom de fichier est omis.

READ variable [, nom de variable]* Ⓟ

RESTORE [numéro de ligne]

DATA [donnée] [, [donnée]]*

Fonction:

READ: Introduit les contenus d'une instruction DATA dans la variable spécifiée.

RESTORE: Spécifie la position de l'instruction DATA introduite à l'aide de l'instruction READ.

DATA: Stocke des données.

Paramètres:

Nom de variable: On peut utiliser toutes les variables.

Numéro de ligne: Expression numérique. 1 ≦ numéro de ligne < 10000 (les fractions seront ignorées).

Données: Valeur numérique ou chaîne et constante à caractères.

Explication:

1) L'instruction READ sert à introduire les données d'une instruction DATA une par une dans une variable spécifique.
2) Les données d'une instruction DATA sont lues dans l'ordre croissant des numéros de ligne, à partir du début de l'instruction.
3) Si l'instruction DATA comprend un nombre de données inférieur à celui des variables introduites à l'aide de l'instruction READ, cele provoquera une erreur DA ERROR.
4) En cas d'omission d'un numéro de ligne de l'instruction RESTORE, les instructions READ qui viennent après seront exécutées à partir du début de l'instruction DATA qui apparaît la première dans le programme.

5) Si l'on spécifie un numéro de ligne avec l'instruction RESTORE, les instructions READ à exécuter après seront lues à partir de l'instruction DATA qui suit la ligne spécifiée.
6) Il se produira une erreur DA ERROR si le numéro de ligne spécifié avec l'instruction RESTORE n'existe pas ou s'il n'y a pas d'instruction DATA après le numéro de ligne spécifié.
7) En ponctuant avec une virgule, on peut écrire des données multiples dans une instruction DATA.
8) Si on exécute seulement une instruction DATA, rien ne se produit.
9) Si la donnée elle-même comprend des virgules ou un espacement au début, la mettre entre guillemets.
10) Si des données sont omises dans une instruction DATA, les chaînes de caractères seront considérées comme ayant une longueur 0 (chaîne vide).

    Exemple: DATA , , aura la même signification que DATA " " , " " , " ".
11) On peut écrire une instruction DATA à n'importe quel endroit du programme.

Remarque:

Si on spécifie un numéro de ligne par une instruction RESTORE, les données seront prises dans l'instruction DATA de la zone de programme précédente, même si l'exécution du programme a été déviée par une instruction GOTO ou GOSUB. Pour empêcher l'introduction de données à partir de la zone de programme précédente, insérer une spécification d'instruction RESTORE au début de la zone de programme de destination.

```
P0
  10 CLEAR
  20 RESTORE 100
  30 GOTO PROG 1
 100 DATA1,2,3
P1
  10 READ A,B,C
  20 PRINT A;B;C
  30 END
  40 DATA4,5,6
```

* Le résultat ici sera 1,2,3.

```
P0
  10 CLEAR
  20 RESTORE 100
  30 GOTO PROG 1
 100 DATA1,2,3
P1
   5 RESTORE
  10 READ A,B,C
  20 PRINT A;B;C
  30 END
  40 DATA4,5,6
```

* Le résultat ici sera 4,5,6.

REM [commentaire] Ⓟ

Fonction:

Insère des commentaires dans un programme.

Paramètre:

Commentaire: chaîne de caractères.

Explication:

Tous les caractères ou les symboles qu'on écrira après REM seront considérés comme des commentaires et ne seront donc pas exécutés. Les commandes ne seront pas exécutées, même si elles sont suivies d'une instruction multiple à la même ligne.

Exemple

```
10 REM TEST : A=50
                └→ ceci ne sera pas exécuté.

1000 REM *SUBROUTINE
       :
       :
       :
```

STAT valeur de la donnée *x* [; fréquence] Ⓐ

STAT valeur de la donnée *x*, valeur de la donnée *y* [; fréquence]

Fonction:

Introduit des données statistiques et la fréquence de ces données.

Paramètres:

Valeur de la donnée *x*:	Expression numérique
Valeur de la donnée *y*:	Expression numérique
Fréquence:	Expression numérique

Explication:

1) Introduire une donnée statistique à 1 variable avec l'instruction STAT *x* ; *n*. Si ; *n* est omis, *n* sera considéré comme égal à 1.
2) Introduire une donnée statistique à 2 variables avec l'instruction STAT *x*,*y* ; *n*. Si ; *n* est omis, *n* sera considéré être égal à 1.
3) Si on utilise une variable pour la valeur de la fréquence de chaque donnée, toujours définir les contenus de variable avant exécution de cette commande.

Exemple

```
10 STAT CLEAR
20 INPUT "DATA NO.
   ";N
30 FOR I=1 TO N
40 INPUT X,Y
50 STAT LGT(X),Y
60 NEXT I
70 STAT LIST
```

* Effectue un calcul de régression logarithmique utilisant une fonction statistique.

STAT CLEAR Ⓐ

Fonction:

Effectue l'initialisation des statistiques de base.

Explication:

1) Met à 0 CNT, SUMX, SUMY, SUMXY, SUMX2 et SUMY2.
2) Initialise les statistiques de bases et prépare au calcul suivant. Exécuter toujours cette commande lorsqu'on entreprend de nouveaux calculs statistiques (voir STAT).

STOP Ⓟ

Fonction:

Interrompt momentanément l'exécution d'un programme.

Explication:

1) La commande STOP affiche un message d'arrêt et allume le symbole STOP.
2) Après l'arrêt, l'exécution peut être reprise grâce à la commande CONT, pour effectuer les opérations suivantes:
   - Calculs manuels
     Exemple: A∗ 3 [ENTER]
   - Affectations manuelles de variable
     Exemple: X= 1 2 3 [↵]

- Contrôle manuel des contenus de variable
  Exemple: P [ENTER]
- Exécution des commandes suivantes.

ANGLE, BEEP, CLEAR, CLS, DIM, ERASE,
PRINT, LPRINT, PRINT ON, PRINT OFF,
STAT, STAT CLEAR, TRON, TROFF, WAIT

3) Si une erreur se produit à l'arrêt, la reprise de l'exécution est impossible avec CONT.
4) Si les opérations suivantes sont exécutées à l'arrêt, appuyer toujours sur la touche [BRK] avant de continuer.
   - Utilisation des commandes manuelles (telles que RUN, LOAD, LIST, etc., CONT non comprise).
   - Utilisation des commandes PUT et GET.
   - Exécution d'un programme dans les zones de programmes [SHIFT] [P0] à [SHIFT] [P9].

Exemple:

```
10 CLEAR :DIM D$(6
   ,2)
20 FOR I=0 TO 6
30 READ D$(I,1)
40 NEXT I
50 STOP
60 FOR I=1 TO 6
70 PRINT D$(I,1)::
   INPUT D$(I,2)
80 NEXT I
90 DATA SUNDAY,MON
   DAY,TUESDAY,WED
   NESDAY,THURSDAY
   ,FRIDAY,SATURDA
   Y
```

* Etant donné que l'exécution du programme s'arrêtera après l'introduction des données de l'instruction DATA, appuyer sur CONT ↵ pour faire repartir l'exécution.

TRON
TROFF

Ⓐ

Fonction:

TRON:	Spécifie le mode d'analyse.
TROFF:	Libère le mode d'analyse.

Explication:

Si on spécifie le mode d'analyse, le numéro de zone de programme (0 – 9) et le numéro de ligne en cours d'exécution seront successivement affichés comme indiqué ci-dessous.

```
[0:10] [0:20] [0:30]
```

Quand une ligne atteint l'extrémité de l'écran, l'affichage se déplace vers la gauche.

Exemple

```
TRON ↵
TROFF ↵
```

WAIT expression numérique Ⓟ

Fonction:

Spécifie un temps de pause pendant l'exécution d'une instruction PRINT.

Paramètre:

Expression numérique: $0 \leqq$ expression numérique < 999.
Si la valeur est supérieure à 999, le mode sera le même que sans spécification WAIT.

Explication:

1) Le symbole WAIT apparaitra pendant la pause, après l'instruction PRINT.
2) Pour reprendre l'exécution d'un programme, appuyer sur ↵ ou ENTER pendant la pause.
3) Si l'expression numérique est supérieure à 999 et dans le cas où il y a un point-virgule à la fin de l'instruction PRINT, l'exécution du programme n'obéira pas à l'instruction WAIT. S'il y a une virgule, l'exécution du programme sera arrêtée jusqu'à ce que la touche ↵ ou ENTER soit enfoncée.
4) Le temps fixé commencera avec l'exécution de la commande WAIT et se terminera avec la fin du programme. Il sera automatiquement libéré à la fin du programme.
5) Introduire WAIT 999 pour libérer le temps fixé pour un programme.
6) Le temps d'arrêt est d'environ 0,05 sec. × *n* (chiffre spécifié).

Exemple

```
10 WAIT 20
20 PRINT "1SECOND"
30 WAIT 400
40 PRINT "20SECOND
   "
50 WAIT 999
60 PRINT "STOP"
70 END
```


5-3 Fonctions de caractères

ASC ({"chaîne de caractères" / variable de caractères}) Ⓕ

Fonction:

Donne le code ASCII d'un caractère spécifié.

Explication:

Donne le code ASCII du premier caractère si le contenu d'une chaîne de caractères ou d'une variable de caractères comporte deux caractères ou plus (voir le tableau des codes de caractères, page 328).

Exemple

```
10 INPUT A$
20 X=ASC(A$)
30 PRINT X
```

CHR$ (Code ASCII / Expression numérique) Ⓕ

Fonction:

Donne le caractère d'un code ASCII spécifié.

Paramètre:

Code ASCII: Expression numérique comprise dans les limites de $0 \leqq$ expression numérique < 256.

Explication:

On peut aussi utiliser une variable comme paramètre. Si l'expression numérique comporte une partie fractionnaire, celle-ci sera ignorée. Si le caractère spécifié ne figure pas au tableau des codes de caractères, une chaîne vide sera affichée.

Exemple

```
10 PRINT CHR$(34);
   "GOOD";CHR$(34)
```

Affichage

```
"GOOD"
```

VAL ("chaîne de caractères" / variable de caractères) Ⓕ

Fonction:

Convertit une chaîne de caractères en valeur numérique.

Explication:

1) Si une chaîne de caractères commence par un caractère autre que le nom d'un nombre (y compris le symbole, la virgule décimale, la numération hexadécimale, le signe exponentiel E), "0" sera obtenu.
2) Si des caractères autres que des valeurs numérales sont utilisés à l'intérieur d'une chaîne de caractères, ces caractères et ceux qui viennent après seront ignorés.
3) S'il y a un espace au début d'une chaîne de caractères, il sera ignoré.
4) S'il y a trois numeraux ou plus à la suite du signe d'exposant "E", il se produira une erreur SN ERROR.

Exemple

```
VAL( "123") ENTER → 123
VAL ("&HFF") ENTER → 255
```


STR$ (expression numérique) Ⓕ

Fonction:

Convertit une valeur numérique en numéraux de chaîne de caractères.

Paramètre:

Expression numérique: Valeur numérique, expression de calcul, variable numérique, variable de tableau numérique.

Explication:

Un espace sera affecté à une valeur numérique positive et un signe "–" sera affecté à une valeur numérique négative.

Exemple

```
10 A=10/3
20 A1$=STR$(A)
30 PRINT A1$
```

DMS$ (expression numérique) Ⓕ

Fonction:

Convertit une chaîne de caractères décimale en son expression sexagésimale.

Paramètre:

Argument: Expression numérique comprise dans les limites de |argument| $< 10^{100}$.

Explication:

1) Les limites d'introduction sont: |argument| $< 10^{100}$, mais les limites de conversion en unités de degrés, minutes et secondes (heures, minutes, secondes) sont en fait: |expression numérique| < 1E6. Les valeurs numériques non comprises dans ces limites de conversion sont utilisées sous leur forme originelle comme chaînes de caractères.

Exemple

- Manuel

```
DMS$ ( 1.234 ) ENTER   1°14'2.4
```

- Programme

```
10 T1$=DMS$(1.234)
20 PRINT T1$
```


HEX$ (argument) Ⓕ

Fonction:

Convertit une décimale en une chaîne de caractères d'expression hexadécimale de quatre chiffres.

Paramètre:

Argument: Expression numérique comprise dans les limites de −32769 < argument < 65536.

Explication:

1) Exprime un nombre négatif comme complément de 2.
2) Les valeurs numériques égales et supérieures à 32768 sont converties par soustraction de 65535.

Exemple

```
10 A=127
20 PRINT HEX$(A)
```

LEFT$ ({expression de caractères / "chaîne de caractères" / variable de caractères} , nombre de caractères / expression numérique) Ⓕ

Fonction:

Recherche un nombre de caractères spécifié, à partir du côté gauche de la chaîne de caractères.

Paramètre:

Nombre de caractères: Expression numérique. On peut aussi utiliser des variables.
$0 \leqq$ nombre de caractères < 256.

Explication:

Si "0" est spécifié comme nombre de caractères à rechercher, une chaîne vide sera affichée. Si le nombre spécifié dépasse le nombre de caractères existants, la chaîne de caractères affectée sera elle-même recherchée.

Exemple

```
10 A$="ABCDEF"
20 PRINT LEFT$(A$,
   4)
```


RIGHT$ ({expression de caractères / "chaîne de caractères" / variable de caractères} , nombre de caractères / expression numérique) Ⓕ

Fonction:

Recherche un nombre de caractères spécifié, à partir du côté droit de la chaîne de caractères.

Paramètre:

Nombre de caractères: Expression numérique. On peut aussi utiliser des variables.
$0 \leqq$ nombre de caractères < 256.

Explication:

Si "0" est spécifié comme nombre de caractères à rechercher, une chaîne vide sera affichée. Si le nombre spécifié dépasse le nombre de caractères existants, la chaîne de caractères affectée sera elle-même recherchée.

Exemple

```
10 INPUT B$
20 PRINT RIGHT$(B$
   ,2)
30 GOTO 10
```

MID$ ({ expression de caractères / "chaîne de caractères" / variable de caractères } , position [, nombre de caractères]) Ⓕ

Fonction:

Recherche un nombre de caractères spécifié dans une chaîne de caractères, à partir d'une position spécifiée.

Paramètres:

Position: Expression numérique. Spécifie la position de caractère dans une chaîne de caractères en partant de la gauche. Peut aussi utiliser des variables.
$1 \leqq$ position < 256

Nombre de caractères: Expression numérique. Accepte aussi des variables.
$0 \leqq$ nombre de caractères < 256

Explication:

1) Si la spécification du nombre de caractères à rechercher a été omise, tous les caractères qui se trouvent après la position spécifiée seront recherchés.
2) Si le nombre de caractères spécifié est "0", une chaîne vide sera mise en place.
3) Si le nombre de caractères spécifié est plus grand que le nombre de caractères qui se trouvent après la position spécifiée, tous les caractères qui restent seront recherchés.
4) Si la position spécifiée dépasse la chaîne de caractères, une chaîne vide sera recherchée.
5) Si un paramètre comprend une fraction, celle-ci sera ignorée.

Exemple

```
10 A$="ABCDEF"
20 INPUT M,N
30 PRINT MID$(A$,M
   ,N)
```


LEN ({ expression de caractères / "chaîne de caractères" / variable de caractères }) Ⓕ

Fonction:

Donne la longueur (nombre de caractères) d'une chaîne de caractères.

Paramètre:

Expression de caractères: $0 \leqq$ numéro de caractères $\leqq 79$

Explication:

La longueur de l'expression de caractères etc. donnée doit être comprise dans les limites de 0 à 79 caractères.

Exemple

```
10 A$="BASIC"
20 PRINT LEN(A$)
```

INKEY$ Ⓕ

Fonction:

Introduit un caractère dans une variable de caractères à partir du clavier.

Explication:

1) On peut introduire des numéraux, des caractères alphabétiques ou des symboles sans qu'il soit nécessaire d'appuyer sur la touche ↵ .
2) Ni le "?" ni le curseur ne seront affichés et il n'y aura pas non plus d'état d'attente d'introduction.
   Si l'on appuie pas sur la touche, rien ne sera introduit (état nul) et la commande suivante sera exécutée. On utilise en général INKEY$ en combinaison avec l'instruction IF pour obtenir un état d'attente d'introduction.

Exemple

```
10 A$=INKEY$          30 BEEP
20 IF A$="" THEN 1    40 PRINT A$
   0
```

* Lorsqu'on appuie sur une touche, l'introduction sera confirmée par un "bip" sonore, et affichée.

5-4 Fonctions de Contrôle

TAB (expression numérique) Ⓐ

Fonction:

Imprime un nombre spécifié d'espaces.

Paramètre:

Expression numérique: 0 ≦ expression numérique ≦ 80 (fractions ignorées)

Explication:

1) Si le côté gauche de la position d'impression en cours est spécifié, TAB interviendra à la ligne suivante après un changement de ligne. Lorsqu'on effectue des sorties imprimées continues de caractères ponctués avec un point-virgule, TAB ne peut pas être utilisé après un changement de ligne.

* La fonction TAB ne peut servir que dans une instruction LPRINT. Si on l'utilise dans une instruction PRINT, il se produira une erreur SN ERROR.

Exemple

```
10 X$="BASIC"
20 LPRINT TAB(10);
   X$
```


USING "chaîne de caractères de format" Ⓐ

Fonction:

Spécifie les formats de sortie des instructions PRINT et LPRINT.

Paramètre:

Liste des chaînes de caractères de format

- # — Pour les valeurs numériques. Peut utiliser jusqu'à 13 chiffres avant la virgule décimale et jusqu'à 9 chiffres après la virgule décimale, mais le total ne doit pas dépasser 13 chiffres, virgule comprise.
- . — Position de la virgule décimale.
- ^ — Spécification d'exposant placée à la fin.
- & — Pour les caractères. Ne peut pas être employé en même temps que les caractères réservés aux valeurs numériques.

Explication:

1) Les valeurs numériques seront sorties avec justification à droite et les caractères avec justification à gauche.

2) Valeurs numériques

Si les fractions dépassent la longueur spécifiée, elles seront arrondies dans les limites de la valeur spécifiée.

Si le nombre spécifié de nombres entiers est dépassé (c'est à dire si une valeur numérique ne peut pas être exprimée dans les termes du format spécifié), un signe % sera ajouté à gauche et la sortie ne sera pas conforme au format.

Le signe moins compte comme un chiffre.

3) Caractères

Si une chaîne de caractères de sortie est plus longue que la longueur spécifiée, les caractères en surnombre seront éliminés.

Si la chaîne de caractères de sortie est inférieure au nombre de caractères spécifiés, des espaces seront insérés pour compenser la différence.

Exemple

```
10 A$="ABCDE"
20 B=12.35
30 PRINT USING"&&&
   ";A$;"=NAME"
40 PRINT USING"&&&
   &&&&";A$;"=NAME
   "
50 PRINT USING"###
   .#";B;"=DATA"
60 PRINT USING"###
   .###";B;"=DATA"
```

Opération

```
RUN ↵
ABC=NA

↵
ABCDE =NAME

↵
12.4=DATA

↵
12.350=DATA
```


5-5 Fonctions numériques

SIN argument Ⓕ

COS argument

TAN argument

Fonction:

SIN:	calcule le sinus de x, sin x.
COS:	calcule le cosinus de x, cos x.
TAN:	calcule la tangente de x, tan x.

Paramètre:

Argument: Expression numérique. Doit correspondre à l'unité d'angle choisie (voir ANGLE).

$-5400° <$ argument $< 5400°$ (degrés)

$-30\pi <$ argument $< 30\pi$ (radians)

$-6000 <$ argument < 6000 (grades)

Explication:

1) En plus d'un nombre réel on peut utiliser des variables ou des expressions numériques comme argument.

2) L'argument n'a pas besoin d'être mis entre parenthèses s'il est constitué d'un nombre réel ou d'une variable uniques. Si le résultat du calcul est l'argument, des parenthèses seront par contre nécessaires.

SIN X + Y . . . . . ajoute Y à la valeur de SIN X.

SIN (X + Y) . . . . . calcule le sinus de la somme de X et Y.

* En ce qui concerne TAN toutefois, les limites d'introduction excluent largument| = $(2n - 1) * 1$ angle droit.

(1 angle droit = 90° = $\pi/2$ rad = 100 grades)

Exemple

```
10 INPUT A
20 PRINT SINA
```

ASN argument Ⓕ

ACS argument

ATN argument

Fonction:

Ces fonctions trigonométriques inverses donnent l'angle à partir de la valeur d'une fonction trigonométrique.

ASN: calcule l'arcsinus, $\sin^{-1}x$.

ACS: calcule l'arccosinus, $\cos^{-1}x$.

ATN: calcule l'arctangente, $\tan^{-1}x$.

Paramètre:

Arguments: Expression numérique. ASN et ACS sont compris dans les limites de |argument| $\leqq 1$.
ATN est compris dans les limites de |argument| $< 10^{100}$.

Explication:

Les limites de sortie sont comme indiquées ci-dessous.

$-90° \leqq \text{ASNX} \leqq 90°$

$0° \leqq \text{ACSX} \leqq 180°$

$-90° \leqq \text{ATNX} \leqq 90°$

Exemple

```
10 INPUT X
20 PRINT ASNX
```


HYPSIN argument Ⓕ

HYPCOS argument

HYPTAN argument

Fonction:

Ces fonctions servent à obtenir les fonctions hyperboliques.

HYP SIN : $\sinh x = (e^x - e^{-x})/2$

HYP COS : $\cosh x = (e^x + e^{-x})/2$

HYP TAN : $\tanh x = (e^x - e^{-x})/(e^x + e^{-x})$

Paramètre:

Argument: Expression numérique.
HYP SIN et HYP COS doivent être compris dans les limites de |argument| $\leqq 230$. HYP TAN doit être compris dans les limites de |argument| $< 10^{100}$.

HYPASN argument Ⓕ

HYPACS argument

HYPATN argument

Fonction:

Ces fonctions servent à obtenir les fonctions hyperboliques inverses.

HYP ASN : $\sinh^{-1} x = \log(x + \sqrt{x^2+1})$

HYP ACS : $\cosh^{-1} x = \log(x + \sqrt{x^2-1})$

HYP ATN : $\tanh^{-1} x = \frac{1}{2}\log\frac{1+x}{1-x}$

Paramètre:

Argument: Expression numérique.
HYP ASN doit être compris dans les limites de |argument| $< 5 \times 10^{99}$. HYP ACS doit être compris dans les limites de $1 <$ argument $< 5 \times 10^{99}$. HYP ATN doit être compris dans les limites de |argument| < 1.

SQR argument Ⓕ

Fonction:

Donne la racine carrée d'un argument.

Paramètre:

Argument: Expression numérique. Argument $\geqq 0$

Exemple

```
10 A=256
20 B=SQRA
30 PRINT B
```


LGT argument Ⓕ

LOG argument

Fonction:

LGT: calcule le logarithme ordinaire de x, $\log_{10} x$.

LOG: calcule le logarithme naturel de x, $\log_e x$.

Paramètre:

Argument: Expression numérique. Argument > 0.

* La précision sera de ±1 au 8ème chiffre quand la valeur de x est $1 - 10^{-6} < x < 1 + 10^{-6}$.

Exemple

```
10 A=5
20 PRINT LGTA
```

EXP argument Ⓕ

Fonction:

Donne la valeur de la fonction exponentielle d'un argument (argument = x: e^x).

Paramètre:

Argument: Expression numérique.
$-10^{100} \leqq$ argument $\leqq 230.2585092$

Exemple

```
10 Y=EXP2.7
20 PRINT Y
```

ABS argument

Ⓕ

Fonction:

Donne la valeur absolue d'un argument.

Paramètre:

Argument: Expression numérique. |Argument| $< 10^{100}$.

Exemple

```
10 A=ABS-68
20 PRINT A
```

* Donne la valeur absolue de −68.

INT argument

Ⓕ

Fonction:

Donne le nombre entier le plus grand qui ne dépasse pas un argument (lorsque l'argument est positif, les fractions seront ignorées).

Paramètre:

Argument: Expression numérique. |Argument| $< 10^{100}$.

Exemple

- Calcul manuel

```
INT  4.4 [ENTER] →4
INT  −2.5[ENTER]→−3
```

- Calcul programmé

```
10 A=2468.135
20 PRINT INTA
```


FRAC argument

Ⓕ

Fonction:

Donne les fractions d'un argument. (Rejette les nombres entiers.)

Paramètre:

Argument: Expression numérique. |Argument| $< 10^{100}$.

Exemple

```
10 A=3.141592
20 B=FRACA
30 PRINT B
```

SGN argument

Ⓕ

Fonction:

Donne une valeur correspondant au signe de l'argument.

Paramètre:

Argument: Expression numérique.

Explication:

Si l'argument est plus grand que 0 (plus) on obtiendra 1.
Si l'argument est 0 on obtiendra 0.
Si l'argument est plus petit que 0 (moins) on obtiendra −1.

Exemple

```
10 INPUT X
20 Y=SGNX
30 PRINT Y
```

ROUND (expression numérique 1, expression numérique 2 [, {0/1}] [, {+/−}]) Ⓕ

Fonction:

Donne la valeur de l'expression numérique 1, qui, selon une méthode déterminée, est arrondie au chiffre désigné par l'expression numérique 2.

Paramètres:

Expression numérique 1: |expression numérique 1| $< 10^{100}$.
Expression numérique 2: |expression numérique 2| $< 10^{100}$.

Explication:

1) ROUND (x, y)
x est arrondi à la position 10^y.
2) ROUND (x,y, 0)
x est arrondi à la position 10^y.
ROUND (x,y,1)
x est arrondi au chiffre significatif y.
3) ROUND (x,y, {0/1}, signe)
Lorsque le signe est " + ", le chiffre sera arrondi au chiffre supérieur. (Voir alinéa 2 ci-dessus.)
Lorsque le signe est "–", la partie fractionnaire est supprimée. (Voir alinéa 2 ci-dessus.)

Remarque:

A l'exception de 0, toutes les valeurs d'un chiffre spécifié seront arrondies à l'unité supérieure dans une opération d'arrondissage à l'unité supérieure.

Exemple

Dans le cas suivant, c'est le premier chiffre après la virgule décimale qui a été spécifié.
12,30 → 13
12,03 → 12

Exemple

```
10 A=12345.67
20 X=ROUND(A,2)
30 Y=ROUND(A,2,1)
40 Z=ROUND(A,2,0,+
   )
50 PRINT X;Y;Z
```

DEG (degré expression numérique [, minute expression numérique [, seconde expression numérique]]) Ⓕ

Fonction:

Convertit les nombres sexagésimaux en nombres décimaux.

Paramètre:

Degré, minute, seconde: Expression numérique.
|Expression numérique| $< 10^{100}$.

Explication:

Dans le cas d'un angle négatif, un signe moins "–" sera ajouté à chaque degré, minute et seconde.

Exemple

- Calcul manuel.

DEG (12 , 34 , 56) ENTER

```
12.58222222
```

- Calcul programmé.

```
10 INPUT X,Y,Z
20 PRINT DEG(X,Y,Z
   )
30 END
```

PI Ⓕ

Fonction:

Donne la valeur de π (rapport de la circonférence d'un cercle à son rayon – 3,1415926536).

Remarque:

L'affichage deviendra 3,141592654.

Exemple

```
10 INPUT A
20 PRINT A*PI
30 GOTO 10
```

RND Ⓕ

Fonction:

Produit un nombre aléatoire ayant jusqu'à 10 chiffres dans lequel 0 < nombre aléatoire < 1.

Exemple

```
10 A=RND
20 PRINT A
```


5-6 Fonctions statistiques

CNT Ⓕ

Fonction:

Donne le nombre (n) de données traitées statistiquement.

SUMX Ⓕ

SUMY

Fonction:

Donne la somme totale des données.

SUMX: Σx

SUMY: Σy

SUMXY Ⓕ

Fonction:

Donne la somme des produits des données x et y.

SUMXY : $\sum xy$

SUMX 2 Ⓕ
SUMY 2

Fonction:

Donne la somme des carrés des données spécifiées.

SUMX2 : $\sum x^2$

SUMY2 : $\sum y^2$

SDX Ⓕ
SDY

Fonction:

Donne l'écart type de l'échantillon

SDX : $\sqrt{\frac{n\cdot\sum x^2-(\sum x)^2}{n(n-1)}}$ $(x\ \delta n-1)$

SDY : $\sqrt{\frac{n\cdot\sum y^2-(\sum y)^2}{n(n-1)}}$ $(y\ \delta n-1)$

(n est le nombre des données)

SDXN Ⓕ
SDYN

Fonction:

Donne l'écart type du parc.

SDXN : $\sqrt{\frac{n\cdot\sum x^2-(\sum x)^2}{n^2}}$ $(x\ \delta n)$

SDYN : $\sqrt{\frac{n\cdot\sum y^2-(\sum y)^2}{n^2}}$ $(y\ \delta n)$

(n est le nombre des données)

LRA Ⓕ

Fonction:

Donne le terme constant de régression linéaire.

$$\text{LRA} : \frac{\sum y - \text{LRB} \cdot \sum x}{n} \qquad (n \text{ est le nombre de données})$$

Exemple

```
10 Y=X*LRB+LRA
```

LRB Ⓕ

Fonction:

Donne le coefficient de régression linéaire.

$$\text{LRB} : \frac{n \cdot \sum xy - \sum x \cdot \sum y}{n\sum x^2 - (\sum x)^2} \qquad (n \text{ est le nombre de données})$$

COR Ⓕ

Fonction

Donne le coefficient de corrélation (r)

$$\text{COR} : \frac{n \cdot \sum xy - \sum x \cdot \sum y}{\sqrt{\{n \cdot \sum x^2 - (\sum x)^2\}\{n \cdot \sum y^2 - (\sum y)^2\}}}$$

(n est le nombre des données)

EOX (argument) Ⓕ

Fonction:

Donne une valeur estimée de x pour un y donné.

$$\text{EOX} : \frac{y - \text{LRA}}{\text{LRB}} \quad (\hat{x})$$

Paramètre

Argument: Expression numérique

EOY (argument) Ⓕ

Fonction

Donne une valeur estimée de y pour un x donné.

EOY : LRA + x · LRB ($\hat{y}$)

Paramètre

Argument: Expression numérique

5-7 Commandes diverses

■ Conversion hexadécimale

Convertit les nombres hexadécimaux de 1 à 4 chiffres qui suivent &H en nombres décimaux.

Exemple

- Calcul manuel

```
&H25A0 ENTER
```

```
 9632
```

- Calcul programmé

```
10 A=&H6F00
20 PRINT A
```

CHAPITRE 6

BIBLIOTHEQUE DE PROGRAMMES

6-1 Traitement d'exécution

(jusqu'a 10 sujets es 61 personnes)

Deux cartes RAM RC-4 sont nécessaires.

Ceci est un programme pour le classement des notes qui diminuera le travail des professeurs et leur fera gagner du temps. Ce programme peut être traité partout le monde sans difficulté et n'importe où.

Caractéristiques et détails de fonctionnement

1. Jusqu'à 61 étudiants peuvent être pris en considération et un maximum de 10 sujets peut être soumis à ce traitement. Ce programme calcule l'acquis (nombre total de points) et la valeur de l'écart pour chaque étudiant.
2. Un classement peut être réalisé par sujet particulier et pour l'ensemble des sujets.
3. Calcule la valeur de l'écart pour chaque sujet et pour l'ensemble des sujets.
4. Numéro de l'étudiant et noms des sujets sont d'abord enregistrés et le traitement se poursuit par la seule introduction des notes obtenues (on donne un numéro aux étudiants pour s'épargner la tâche difficile d'introduire leur nom).

Exemple:

3 sujets et 5 étudiants

Notes

Numéro de l'étudiant	Sujet		
	Anglais	Mathématiques	Sciences
1	75	58	80
2	63	70	78
3	81	72	61
4	48	51	80
5	53	63	40

Remarque 1.

Lorsque vous exécutez un programme pour la première fois, introduire CLEAR ↵ pour effacer toutes les données de la carte RAM.

Remarque 2.

La carte RAM de la fente 0 est la carte programme et la carte RAM de la fente 1 est la carte données. D'autres professeurs peuvent aussi utiliser la carte programme, il leur suffira de changer la carte données.

Tant pour le programme que pour les données, utiliser des cartes RAM RC-4 (4 kilo-octets).

Opération

RUN↵

```
*ACHIEVEMENT PROCESSING*
NUMBER OF SUBJECTS?
```

↵

Introduire le nombre de sujets

3↵

```
NUMBER OF STUDENTS?
```

Il y a cinq étudiants. (introduction possible d'un maximum de 61 étudiants)

5↵

```
SUBJECT INPUT
```

Destiné à l'introduction des noms de sujet.

↵

```
SUBJECT # 1=?
```

Introduire en ordre successif les caractères des mots anglais (ENGLISH), mathémartiques (MATH.) et sciences (SCIENCE) (on peut introduire des sujects comportant jusqu'à 20 caractères).

ENGLISH↵

```
SUBJECT # 2=?
```

MATH.↵

```
SUBJECT # 3=?
```

SCIENCE↵

```
1IN 2ED 3CAL 4SORT 5END
```

Le menu d'exécution sera affiché lorsque l'introduction des sujets sera terminée.

1 IN . . . . . A sélectionner pour l'introduction de notes.

2 ED . . . . . A sélectionner pour contrôle et correction des notes déjà introduites.

3 CAL . . . . . A sélectionner pour confirmation des notes, calculs de moyenne, etc.

4 SORT . . . . . A sélectionner pour classement par sujet ou pour l'ensemble des sujets.

5 END . . . . . A sélectionner pour terminer le programme ou pour changer la carte données.

```
1                ENGLISH ATTEND. 1=?
```

Ceci est une demande d'introduction des notes obtenues en anglais. Introduire 75, 63, 81, 48 et 53 dans l'ordre des numéros de pointage.

```
75↵              ENGLISH ATTEND. 2=?
 ⋮                      ⋮
53↵              MATH. ATTEND. 1=?
```

Ceci est une demande d'introduction des notes obtenues en mathématiques. Introduire 58, 70, 72, 51 et 63 dans l'ordre des numéros de pointage.

```
58↵              MATH. ATTEND. 2=?
 ⋮                      ⋮
63↵              SCIENCE ATTEND. 1=?
```

Introduire de même 80, 78, 61, 80 et 40 dans l'ordre des numéros de pointage pour les notes obtenues en sciences. Après introduction de la dernière note, le menu d'exécution sera à nouveau affiché.

```
80↵              SCIENCE ATTEND. 2=?
 ⋮                      ⋮
40↵              1IN 2ED 3CAL 4SORT 5END
```

Les calculs peuvent être exécutés immédiatement, nous allons cependant contrôler les notes introduites pour détecter une éventuelle erreur, toujours possible dans le cas ou l'on traite les notes d'un grand nombre d'étudiants.


```
2                1.LIST 2.EDIT 3.END
```

Sélectionner 1 pour vérifier les notes et 2 pour effectuer les corrections. Si, pendant la vérification des notes, vous découvrez une erreur d'introduction, noter le numéro du sujet et le numéro de pointage.

```
1                SUBJECT NUMBER 1 ENGLISH
↵                ATTEND. 1= 75
↵                ATTEND. 2= 63
 ⋮  Appuyer sur la touche ↵   ⋮  Les notes seront affichées successivement en anglais, mathématiques et sciences.
 ⋮  pour vérifier les données ⋮
 ⋮  dans l'ordre successif.   ⋮
↵                SUBJECT NUMBER 2 MATH.
 ⋮                      ⋮
↵                ATTEND. 5= 40
↵                1.LIST 2.EDIT 3.END
```

Une fois la vérification terminée, sélectionner 2 s'il y a des corrections à effectuer et 3 s'il n'y avait pas d'erreurs, pour revenir au menu d'exécution. Pour faciliter ces explications, nous changerons la note 51 du numéro de pointage 4 en mathématiques en 52.

```
2                SUBJECT NUMBER=?
2↵               ATTEND. NO.=?
4↵               MATH. ATTEND. 4=51
↵                MODIFY (Y/N)?
```

Changement en 52.

```
Y↵          MATH. ATTEND. 4=?
52↵         1.LIST 2.EDIT 3.END
3           1IN 2ED 3CAL 4SORT 5END
```

Sélectionner 3 et exécuter les calculs, une fois terminées les vérifications et corrections.

```
3           WAIT!!...
                Les calculs prendront environ 2 minutes
                pour 61 étudiants.
            ENGLISH
↵           AVERAGE 64.000
```

La moyenne de la classe en anglais est affichée.

```
↵           DEVIATION 14.035
```

L'affichage indique que l'écart type est de 14,035. Les notes et les valeurs d'écart seront ensuite affichées dans l'ordre des numéros de pointage à chaque fois qu'or appuiera sur la touche ↵ .

```
↵           ATTEND. 1= 75    57.84
                No. de pointage Note    Valeur de l'écart
↵           MATH.
                La note obtenue en mathématiques
                et la valeur de l'écart seront affichées.
↵           SCIENCE
                La note obtenue en sciences et la
                valeur de l'écart seront affichées.
```

Une fois terminé l'affichage de chaque sujet, la moyenne et les notes individuelles et la valeur de l'écart seront ensuite affichées pour l'ensemble des sujets.


```
↵           TOTAL
↵           AVERAGE 194.800
↵           DEVIATION 25.917
```

```
↵           ATTEND. 5=156    35.03
↵           1IN 2ED 3CAL 4SORT 5END
```

Nous allons maintenant classer les données.

```
4           SORT SUBJECT NO.=?
```

Si nous introduisons le numéro correspondant à un sujet à classer, les notes individuelles et les valeurs de l'écart seront affichées en ordre séquentiel à partir des valeurs les plus hautes. Mais nous allons effectuer un classement pour l'ensemble des sujets.

Introduire 11 pour classer tous les sujets (ceci parceque 11 est enregistré comme code de sortie de l'ensemble des sujets).

```
11↵         SORTING...
                L'ordinateur classe les données.
            TOTAL
↵           NO.  1   3   214    57.41
↵           NO.  2   1   213    57.02
            Classement  No. de pointage  Note  Valeur de l'écart
```

```
:
↵                NO. 5  5  156   35.03
↵                1IN 2ED 3CAL 4SORT 5END
```

Avec cette opération l'exécution est terminée.

```
5                1..CHANGE CARD 2..END
```

Appuyer sur la touche [2] pour arrêter l'exécution de ce programme. L'ordinateur sera en mode READY.
Si vous désirez effectuer une nouvelle exécution en introduisant une autre carte, sélectionnez 1.

```
1                POWER OFF & CHANGE CARD
```

Variables

Nom de variable	Contenu de variable
A!( )	Notes
B$( )	Nom de sujet
C!( )	Déviation type moyenne
I	Compteur
J	Compteur
U$	Attendre l'introduction du menu
W	Décision de l'introduction du menu
E	Correction des numéros de sujet
D	Correction des numéros d'étudiants
X	Nombre de sujets
Y	Nombre d'étudiants
K	Classement des numéros de sujet
H	Classement

Liste de programme

```
10 REM CHECK RAM C
   ARD
20 PRINT "*ACHIEVE
   MENT PROCESSING
   *"
30 IF Z$="DATA OK"
    THEN 130
40 INPUT "      O
   F SUBJECTS";X
50 IF X<11 THEN IF
    X>0 THEN 70
60 BEEP : GOTO 40
70 INPUT "NUMBER O
   F STUDENTS";Y
80 IF Y<62 THEN IF
    Y>0 THEN Y=Y-1
   : GOTO 100
90 BEEP : GOTO 70
100 DIM A!(12,60),B
   $(11)*10,C!(11,
   1)
110 Z$="DATA OK":PR
   INT "WAIT...";
120 PRINT
130 REM CHECK SUBJE
   CT
140 IF B$(11)="TOTA
   L" THEN 210
150 BEEP :PRINT "SU
   BJECT INPUT"
160 B$(11)="TOTAL"
170 FOR I=1 TO X
180 PRINT "SUBJECT
   #";I;"=";
190 INPUT B$(I)
200 NEXT I
210 REM SELECT JOB
220 PRINT "1IN 2ED
   3CAL 4SORT 5END
   ";
230 V$=INKEY$:IF V$
   ="" THEN 230
240 BEEP :W=VAL(V$)
   :IF W=0 THEN 23
   0 ELSE IF W>5 T
   HEN 230
250 PRINT
260 ON W GOTO 280,3
   50,610,1110,990
270 GOTO 220
280 REM INPUT DATA
290 FOR J=1 TO X
300 FOR I=0 TO Y
310 PRINT B$(J);" A
   TTEND.";I+1;"="
   ;
320 INPUT A!(J,I)
330 NEXT I:NEXT J
340 GOTO 220
350 REM EDIT DATA
360 PRINT "1.LIST 2
   .EDIT 3.END";
370 V$=INKEY$:IF V$
   ="" THEN 370
380 PRINT
390 BEEP :W=VAL(V$)
   :IF W=0 THEN 36
   0 ELSE IF W>3 T
   HEN 360
400 IF W=3 THEN 220
410 ON W GOSUB 430,
   500
420 GOTO 360
430 REM LIST DATA
440 FOR J=1 TO X
450 PRINT "SUBJECT
   NUMBER";J;" ";B
   $(J)
460 FOR I=0 TO Y
470 PRINT "ATTEND."
   ;I+1;"=";A!(J,I
   )
480 NEXT I:NEXT J
490 RETURN
500 REM EDIT DATA
510 INPUT "SUBJECT
   NUMBER=";E:IF E
   >X THEN BEEP :
   GOTO 510
520 IF E<1 THEN BEE
   P : GOTO 510
530 INPUT "ATTEND.
   NO.=";D:IF D>Y+
   1 THEN BEEP : G
   OTO 530
540 IF D<1 THEN BEE
   P : GOTO 530
550 PRINT B$(E);" A
   TTEND.";D;"=";A
   !(E,D-1)
560 INPUT "MODIFY (
   Y/N)";V$:IF V$=
   "Y" THEN 580
570 RETURN
580 PRINT B$(E);" A
   TTEND.";D;"=";
590 INPUT A!(E,D-1)
600 RETURN
610 REM CALC DATA
620 PRINT "WAIT!!..
   .";
630 FOR J=1 TO X
640 STAT CLEAR
650 FOR I=0 TO Y
660 STAT A!(J,I)
670 NEXT I
680 C!(J,0)=SUMX/CN
   T:C!(J,1)=SDX
690 NEXT J
700 STAT CLEAR
710 FOR I=0 TO Y
720 A!(11,I)=0:FOR
   J=1 TO X
730 A!(11,I)=A!(11,
   I)+A!(J,I)
740 NEXT J
```

```
750 STAT A!(11,I)
760 NEXT I
770 C!(11,0)=SUMX/C
    NT:C!(11,1)=SDX
780 PRINT
790 FOR J=1 TO X
800 BEEP :BEEP
810 PRINT B$(J)
820 PRINT "AVERAGE
    ";:PRINT USING"
    ###.###";C!(J,0
    )
830 PRINT "DEVIATIO
    N";:PRINT USING
    "###.###";C!(J,
    1)
840 FOR I=0 TO Y
850 PRINT "ATTEND."
    ;:PRINT USING"#
    #";I+1;
860 PRINT "=";:PRIN
    T USING"###";A!
    (J,I);"  ";
870 PRINT USING"###
    .##";50+10*(A!(
    J,I)-C!(J,0))/C
    !(J,1)
880 NEXT I:NEXT J
890 PRINT
900 PRINT B$(11)
910 PRINT "AVERAGE
    ";:PRINT USING"
    ###.###";C!(11,
    0)
920 PRINT "DEVIATIO
    N";:PRINT USING
    "###.###";C!(11
    ,1)
930 FOR I=0 TO Y
940 PRINT "ATTEND."
    ;:PRINT USING"#
    #";I+1;
950 PRINT "=";:PRIN
    T USING"###";A!
    (11,I);"  ";
960 PRINT USING"###
    .##";50+10*(A!(
    11,I)-C!(11,0))
    /C!(11,1)
970 NEXT I
980 PRINT : GOTO 22
    0
990 REM ANOTHER CAR
    D
1000 PRINT "1..CHANG
    E CARD 2..END";
1010 V$=INKEY$:IF V$
    ="" THEN 1010
1020 BEEP :W=VAL(V$)
    :IF W=0 THEN 10
    10 ELSE IF W>2
    THEN 1010
1030 ON W GOTO 1050,
    1100
1040 GOTO 1010
1050 PRINT
1060 BEEP :BEEP :BEE
    P
1070 PRINT "POWER OF
    F & CHANGE CARD
    ";
1080 V$=INKEY$:IF V$
    ="" THEN 1080
1090 GOTO 1080
1100 END
1110 REM SORT DATA
1120 INPUT "SORT SUB
    JECT NO.=";K
1130 IF K<=X THEN IF
     K>=1 THEN 1160
1140 IF K=11 THEN 11
    60
1150 BEEP : GOTO 112
    0
1160 PRINT "SORTING.
    ..";
1170 FOR I=0 TO Y:A!
    (12,I)=I+1:A!(0
    ,I)=A!(K,I):NEX
    T I
1180 FOR I=0 TO Y-1:
    FOR J=I+1 TO Y
1190 IF A!(0,J)<A!(0
    ,I) THEN 1220
1200 H=A!(0,I):A!(0,
    I)=A!(0,J):A!(0
    ,J)=H
1210 H=A!(12,I):A!(1
    2,I)=A!(12,J):A
    !(12,J)=H
1220 NEXT J:NEXT I
1230 BEEP :BEEP :PRI
    NT :PRINT B$(K)
1240 FOR I=0 TO Y
1250 PRINT "NO.";:PR
    INT USING"##";I
    +1;
1260 PRINT USING"###
    ";A!(12,I);"  "
    ;
1270 PRINT USING"###
    ";A!(0,I);"  ";
1280 PRINT USING"###
    .##";50+10*(A!(
    0,I)-C!(K,0))/C
    !(K,1)
1290 NEXT I
1300 GOTO 220
```


6-2 Programme de tabulation

Deux cartes RAM RC-4 sont nécessaires.

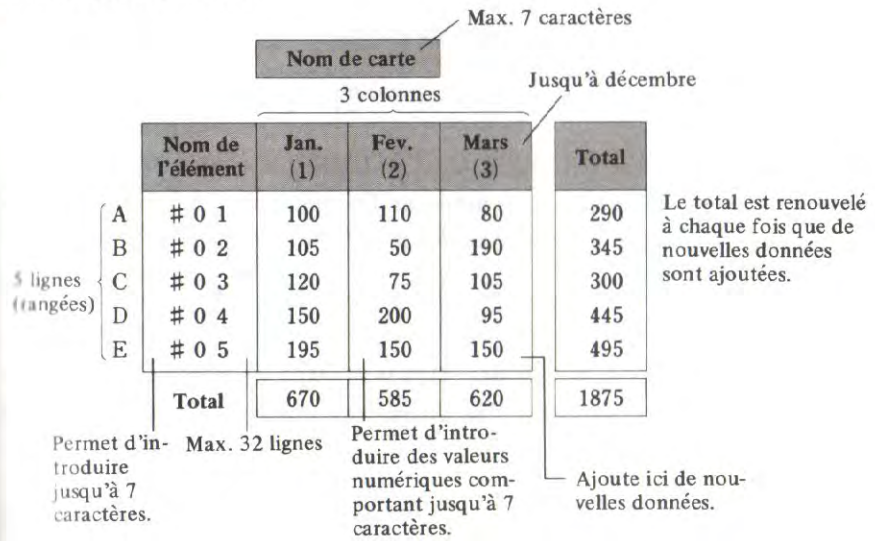
Les totaux horizontaux et verticaux constituent la base de tous les livres de compte. Ce programme, qui met à contribution les avantages de l'une des caractéristiques du FX-750P, utilise une carte de données et des cartes programmes. En changeant simplement les cartes données, on peut trouver le total de chacune. Les cartes données permettent aussi les additions mensuelles de données nouvelles. Ce programme peut donc être largement utilisé dans toutes sortes de domaines. Son format peut être librement choisi dans les limites suivantes: jusqu'à 32 lignes pour indiquer un nom de personne, un nom de produit, un nom de groupe, etc. et jusqu'à 12 colonnes pour indiquer les mois de l'année.

Attention

- Ce programme utilise des cartes programmes et des cartes données. Utilisez les en couple, en introduisant une carte programme dans la fente 0 et une carte données dans la fente 1.
- Introduire CLEAR ⏎ lorsque vous exécutez ce programme avec une nouvelle carte données.

Exemple:

Les ventes de Monsieur A en janvier et février sont introduites pour chaque produit. Les totaux sont obtenus et puis les données pour mars sont introduites afin de renouveler les totaux.



RUN↵

```
(1FORMAT 2INPUT 3END)
```

Quand on exécute le programme, le menu ci-dessus est tout d'abord affiché.

1 FORMAT .... A choisir pour préparation de format.
2 INPUT ..... A choisir pour introduction ou correction de données.
3 END ..... A choisir pour terminer le programme.

Commençons maintenant la préparation de format correspondant à l'exemple.

1

↵

↵

```
* FORMAT *
<<< NEW CARD >>>
 CARD NAME ?
```

Le nom de la carte (CARD NAME) est demandé. Puisque les données concernent Monsieur A, introduisons A.

A↵

```
NUMBER OF LINES (1-32) ?
```

Le nombre de lignes (NUMBER OF LINES) est demandé. Il nous faut décider quel nombre de lignes nous allons utiliser. Nous avons le choix entre tous les chiffres de 1 à 32. Dans ce cas, nous avons besoin de cinq lignes.

5↵

```
NUMBER OF COLUMNS(1-12)?
```

Le nombre de colonnes (NUMBER OF COLUMNS) – c'est à dire de mois – est ensuite choisi. Trois mois dans notre cas. En choisissant un format à 12 mois il devient possible d'introduire les données d'un mois désiré.

3↵

```
ITEM NAME 1:?
```

Dans la colonne du nom de l'élément (ITEM NAME), on introduit des noms de personne, de produit, de groupe, de district, etc. Dans notre cas, nous introduisons # (numéro.).

#01↵

⋮

#05↵

```
ITEM NAME 2:?
⋮
(1FORMAT 2INPUT 3END)
```

Avec le nom de la carte, les spécifications de ligne et de colonne et le nom de l'élément, la préparation du format est maintenant complète. L'affichage revient au menu. Il reste à introduire les données.

2

↵

↵

```
* INPUT *
<< A'S CARD >>
INPUT (1- 3)?
```

Les données sont introduites pour chaque mois. Dans notre cas, nous commençons par introduire celles du mois de janvier.

1↵ (Janvier est spécifié.)
100↵ (Introduction des données à partir de #01.)
105↵

```
1:#01          ?
2:#02          ?
3:#03          ?
```

Les données sont ensuite introduites en ordre séquentiel

⋮

195↵

2↵ (Février est spécifié.)
110↵ (Introduction des données à partir de # 01.)

⋮

150↵

```
INPUT (1- 3)?
 1:#01          ?
 2:#02          ?
⋮
INPUT (1- 3)?
```

Les données ont été introduites pour janvier et février, mais le mois suivant est demandé car le format comprend aussi mars.

Dans notre exemple, les données pour mars sont introduites comme supplément. Pour arrêter l'introduction de données de la colonne (janvier et février), introduire E (END).

E↵	PRINT OUT Y/N

L'affichage indique ici si l'on utilise une imprimante ou non. Si l'on appuie sur E ↵ après introduction des données, la tabulation verticale et horizontale accomplit automatiquement un calcul interne. Donc, même si l'unité centrale est reliée au FA-20, les totaux sont imprimés. Ici, nous pressons la touche N.

N	RECALL (1- 3)?

Ce menu spécifie le mois pour lequel les résultats du calcul des données doivent être affichés.

1—1 ↵ (Spécifie janvier.)	A<< 1 >>
(Affiche dans l'ordre les données pour janvier.)	#01: 100
	⋮
(Affiche le total de la colonne pour janvier.)	1TOTAL 670
(Affiche le total de la ligne pour janvier.)	#01: 100
	⋮
(Affiche le total général pour janvier et février.)	G. TOTAL 1255
(L'affichage revient au menu.)	(1FORMAT 2INPUT 3END)

Remarque:

Quand les mois spécifiés se terminent au milieu de la période couverte par le format, les totaux de rangées sont affichés pour les mois spécifiés.

Si on spécifie un mois particulier, les données pour ce mois sont affichées en premier, suivies par les totaux de colonne et les totaux de rangée. Finalement, le total général est affiché. Le total général de toutes les données mémorisées jusqu'à présent vient affiché quoiqu'il ait été le móis spécifié.

■ Supplément de données

Choisir 2 sur le menu.

	(1FORMAT 2INPUT 3END)
2	* INPUT *
↵	<< A'S CARD >>
↵	INPUT (1- 3)?

Le mois pour lequel des données doivent être introduites est demandé. Spécifier le mois à ajouter. Par exemple, dans le cas d'un format de 12 mois, INPUT (1 – 12)? est affiché. Spécifier simplement le mois pour lequel des données doivent être ajoutées.

Dans ce cas il s'agit du mois de mars.

3↵	1:#01 ?

Introduire les données pour mars.

80↵ (Introduire les donnés à partir de #01.)	2:#02 ?

Quand l'introduction est terminée.

E↵	PRINT OUT Y/N
N	RECALL (1- 3)?

L'introduction de 3 – 3 ↵ provoque l'affichage des totaux et des données; le total des données est indiqué.

Remarque:

Il arrive parfois que, après avoir enregistré sur une carte le nom de la carte et des données, ces derniéres aient besoin d'être mises à jour. Dans ce cas appuyer sur ⎚ (INPUT) pour modifier les données. Lorsque c'est le format 1 qui a été sélectionné, DATA CLEAR Y/N (effacement des données) est affiché après l'affichage du nom de la carte. Si vous voulez ajouter des données, appuyer sur la touche N. Faire attention de ne pas appuyer sur la touche Y, car toutes les données de la carte seraient effacées.

■ Pour examiner des données

Sélectionner 2 dans le menu.

```
(1FORMAT 2INPUT 3END)
* INPUT *
<< A'S CARD >>
INPUT (1- 3)?
PRINT OUT Y/N
RECALL (1- 3)?
```

2

↵ (Vérification des données introduites)

↵ (Vérification de la carte de Monsieur A)

E ↵ (L'introduction n'est pas effectuée)

N (L'impression n'est pas faite)

Spécifier le mois désiré à l'aide de 0 – 0 ↵.

Presser 1 – 1 ↵ si vous désirez voir les données et le total pour le seul mois de janvier et presser 1 – 3 ↵ si vous désirez voir les données et le total de janvier à mars.

Lorsque tout le travail est fini, sélectionner 3 dans le menu pour indiquer la fin (END).

Variables

Nom des variables	Contenu des variables
A( )	Tableau de données
C$( )	Nom d'élément
Z$( )	Attendre l'introduction
F( )	Total de colonne
G( )	Total de ligne
E$( )	Nom de carte
A0 – D4	Blancs
D5S	Variable de fin (END)
P	Nombre de lignes
U	Nombre de colonnes
DS	Introduction de données
V	Conversion de données en valeurs numériques
I	Compteur
J	Compteur
Q	Démarrage du compteur
T	Arrêt du compteur
K	Compteur
LS	Attente d'introduction de menu
M	Conversion de menu en valeurs numériques

Liste de programme

```
P0
 10 CLS :PRINT "(1F
    ORMAT 2INPUT 3E
    ND)";
 20 L$=INKEY$:IF L$
    ="" THEN 20
 30 CLS :M=VAL(L$)
 40 IF M=0 THEN 10
    ELSE IF M>3 THE
    N 10
 50 IF M=1 THEN GOT
    O PROG 1
 70 IF M=2 THEN GOT
    O PROG 2
 80 IF M=3 THEN PRI
    NT "*** END OF
    JOB ***":END
P1
 10 PRINT "* FORMAT
     *"
 20 IF E$="" THEN P
    RINT "<<< NEW C
    ARD >>>": GOTO
    70
 30 PRINT "<< ";E$;
    CHR$(39);"S CAR
    D >>"
 40 PRINT "DATA CLE
    AR  Y/N  ";
 50 Z$=INKEY$:IF Z$
    ="" THEN 50 ELS
    E CLS
 60 IF Z$<>"Y" THEN
     GOTO PROG 0
 70 CLEAR :DIM A(32
    ,11),C$(32)*7
 80 A0=0:A1=0:A2=0:
    A3=0:A4=0:A5=0:
    A6=0:A7=0:A8=0:
    A9=0
 90 B0=0:B1=0:B2=0:
    B3=0:B4=0:B5=0:
    B6=0:B7=0:B8=0:
    B9=0
100 C0=0:C1=0:C2=0:
    C3=0:C4=0:C5=0:
    C6=0:C7=0:C8=0:
    C9=0
110 D0=0:D1=0:D2=0:
    D3=0:D4=0
120 DIM F(32),G(12)
130 D5$="END"
140 INPUT " CARD NA
    ME ";E$
150 PRINT "NUMBER O
    F LINES (1-32)
    ";
155 INPUT P
160 PRINT "NUMBER O
    F COLUMNS(1-12)
    ";
165 INPUT U
170 CLS :FOR I=1 TO
     P
180 PRINT "ITEM NAM
    E";I;":";:INPUT
     C$(I)
190 NEXT I
200 GOTO PROG 0
P2
 10 PRINT "* INPUT
    *"
 20 PRINT "<< ";E$;
    CHR$(39);"S CAR
    D >>"
 30 PRINT "INPUT (1
    -";U;")";:INPUT
     D$:N=VAL(D$)
 40 IF D$="E" THEN
    100
 50 IF N<1 THEN 30
    ELSE IF N>U THE
    N 30 ELSE N=N-1
 60 FOR I=1 TO P
 70 PRINT I;":";C$(
    I);:LOCATE 15:I
    NPUT D$:A(I,N)=
    VAL(D$)
 80 IF D$="E" THEN
    I=P
 90 NEXT I: GOTO 30
100 PRINT "PRINT OU
    T Y/N ";
110 Z$=INKEY$:IF Z$
    ="" THEN 110
120 CLS :PRINT "REC
    ALL (1-";U;")";
    :INPUT D$
130 IF D$="E" THEN
    GOTO PROG 0
140 IF Z$="Y" THEN
    PRINT ON
150 V=LEN(D$)
160 IF V<3 THEN 120
     ELSE IF V>5 TH
    EN 120
170 IF V=4 THEN Q=2
     ELSE IF V=5 TH
    EN Q=2 ELSE Q=1
180 S=VAL(LEFT$(D$,
    2)):T=VAL(RIGHT
    $(D$,Q)):IF T>U
     THEN 120
190 FOR J=S TO T
200 CLS :PRINT E$;"
    <<";J;" >>";:O=
    50: GOSUB 500
210 FOR I=1 TO P
220 CLS :PRINT C$(I
    );":";A(I,J-1);
    : GOSUB 500
```

```
230 NEXT I:NEXT J
240 FOR J=S TO T
250 G(0)=0
260 FOR I=1 TO P
270 G(0)=G(0)+A(I,J
    -1)
280 NEXT I
290 G(J)=G(0)
300 CLS :PRINT J;"T
    OTAL";G(J)::O=2
    00: GOSUB 500
310 NEXT J
320 A1=0
330 FOR J=1 TO U
340 FOR I=1 TO P
350 A1=A1+A(I,J-1)
360 NEXT I
370 NEXT J
380 FOR J=1 TO P:F(
    0)=0
390 FOR I=1 TO T
400 F(0)=F(0)+A(J,I
    -1)
410 NEXT I
420 CLS :PRINT C$(J
    );":";F(0);: GO
    SUB 500
430 NEXT J
440 CLS :PRINT "G.
    TOTAL";A1;: GOS
    UB 500
450 CLS :PRINT OFF
460 GOTO PROG 0
470 NEXT J
480 CLS :PRINT OFF
490 GOTO PROG 0
500 IF Z$="Y" THEN
    PRINT :RETURN
510 FOR K=1 TO O:NE
    XT K:RETURN
```


6-3 Aide memoire(memo card)

Monsieur A se porte très bien, et pourtant il va tous les jours à l'hôpital. Il n' y a à cela rien d'étonnant puisqu'il est représentant de commerce pour les Laboratoires X. Aujourd'hui, il doit faire la tournée de cinq hôpitaux. C'est là que le FX-750P et ses cartes RAM entrent en jeu. Avant de partir en tournée, il étudie soigneusement ses méthodes d'approche en appelant la dernière liste de commandes et les ventes du mois précédent grâce à la carte RAM dans laquelle ces données sont stockées.

50 aide mémoire peuvent être utilisés, comprenant jusqu'à 30 caractères chacun. Ces aide mémoire électroniques peuvent être manipulés sans difficulté dans une voiture ou chez le client.

Exemple

Hôpital N

No. de l'article (nom de l'article)	Quantité commandée
001	2 douz.
005	25 douz.
090	4 douz.
101	20 douz.
Ventes du mois précédent	$7.000

Hôpital O

No. de l'article (nom de l'article)	Quantité (commandée)
003	5 douz.
005	30 douz.
081	2 douz.
101	15 douz.
Ventes du mois précédent	$8.900

RUN↵

↵

```
** MEMO CARD **
INITIAL (Y/N)?
```

Le premier message est une interrogation: il demande s'il y a une introduction initiale de données ou s'il y a continuité dans l'utilisation des données (faire très attention lorsqu'on répond à cette question, car toutes les données introduites auparavant seront effacées si l'on appuie sur la touche [Y]).

Y↵

```
SELECT (A,E,D,S,L,P)?
```

A . . . . . Introduction et addition de données (ADD)
E . . . . . Correction de données (EDIT)
D . . . . . Effacement de données (DELETE)

S . . . . . Recherche de données (SEARCH)
L . . . . . Affichage de données dans l'ordre d'introduction (LIST)
P . . . . . Mise en marche/arrêt de l'imprimante (PRINT)

Nous sélectionnerons A pour introduire les données.

A↵	DATA # 1=?

La première entrée de données est la liste des commandes de l'hôpital N.

N–001–2D↵	DATA # 2=?
N–005–25D↵	DATA # 3=?
N–009–4D↵	DATA # 4=?
N–101–20D↵	DATA # 5=?
N–$7,000↵	DATA # 6=?

Ensuite, introduire les données pour l'hôpital O.

O–003–5D↵	DATA # 7=?
⋮	⋮
O–$8,900↵	DATA # 11=?

Une fois que toutes les données ont été introduites, introduire END (fin).

END↵	SELECT (A,E,D,S,L,P)?

Nous allons maintenant rappeler la liste pour l'hôpital O.
Vu que c'est une opération de recherche, nous sélectionnerons S.

S↵	SEARCH DATA=?

Pour exécuter une recherche, il faut introduire la première lettre de la donnée.
C'est donc O que nous introduirons pour l'hôpital O.

O↵	SEARCHING...

	# 6=O-003-5D
↵	# 7=O-005-30D
↵	# 8=O-081=2D
↵	# 9=O-101=15D
↵	# 10=O-$8,900
↵	SELECT (A,E,D,S,L,P)?

Passons maintenant aux corrections de données.

E↵	EDIT #=?

L'affichage demande quel est le numéro de la donnée à corriger. Si vous n'êtes pas sûr de la réponse, vérifiez en appuyant sur L ↵ pour afficher la liste de données. Dans le cas présent, nous allons modifier le numéro d'article de la sixième donnée en 0,04.

6↵	O-003-5D

S'il faut corriger la donnée, appuyer sur la touche ↵ .

↵	UP-TO-DATE=?

L'ordinateur demande l'introduction de la donnée mise à jour (up-to-date).

O–004–5D↵	SELECT (A,E,D,S,L,P)?

Effaçons maintenant une donnée.

D↵	DELETE #=?

L'ordinateur demande le numéro de la donnée à effacer (delete). Effaçons la huitième donnée.

8↵	DATA=O-081-2D

Demande de confirmation d'effacement.

↵

```
OK?
```

En cas de confirmation (OK), appuyer sur la touche [Y].

Y ↵

```
** WAIT **
SEARCH (A,E,D,S,L,P)?
```

La huitième donnée sera effacée et la neuvième viendra se placer en huitième position. Toutes les données qui suivent remonteront elles-aussi d'une position. Toutefois on ne peut pas faire d'insertion entre deux données.
(Puisque la recherche de donnée est exécutée avec la première lettre, la donnée demandée peut être affichée, même si elle est introduite au hasard.)

Variables

Nom de variable	Contenus de variable
C$	INITIAL (Y/N)
C$	SELECT (A, E, D, S, L, P)
D$ ( )	Données
D$ ( )	Variable de stockage de données
N	Nombre de données
P	Donnée No.
PP	Variable de contrôle d'impression
R	Variable d'effacement de données
D$ ( )	Données de recherche
I	Variable de boucle FOR ~ NEXT
Y	Variable de discrimination Y/N

Liste de programme

```
  5 PRINT "     **
    MEMO CARD **"
 10 INPUT "INITIAL
    (Y/N)";C$:IF C$
    ="N" THEN 30
 20 CLEAR :N=50:DIM
     D$(N)*30:PP=0
 30 INPUT "SELECT (
    A,E,D,S,L,P)";C
    $
 40 IF C$="A" THEN
    100
 50 IF C$="E" THEN
    170
 60 IF C$="D" THEN
    230
 70 IF C$="S" THEN
    360
 80 IF C$="L" THEN
    440
 90 GOTO 30
100 REM ADD
110 P=P+1:IF P<=N T
    HEN 130
120 BEEP :PRINT "SO
    RRY! NO ROOM!":
    P=P-1: GOTO 30
130 PRINT "DATA #";
    P;"=";
140 INPUT D$(0):IF
    D$(0)="END" THE
    N P=P-1: GOTO 3
    0
150 D$(P)=D$(0)
160 GOTO 110
170 REM EDIT
180 INPUT "EDIT #="
    ;R
190 IF R>P THEN BEE
    P : GOTO 180
200 PRINT D$(R)
210 INPUT "UP-TO-DA
    TE=";D$(R)
220 GOTO 30
230 REM DELETE
240 INPUT "DELETE #
    =";R
250 IF R>P THEN BEE
    P : GOTO 240
260 PRINT "DATA=";D
    $(R)
270 INPUT "OK";Y$
280 IF Y$<>"Y" THEN
     30
290 PRINT "     **
    WAIT **";
300 IF R=N THEN 340
310 FOR I=R+1 TO P
320 D$(I-1)=D$(I)
330 NEXT I
340 P=P-1:CLS
350 GOTO 30
360 REM SEARCH
370 INPUT "SEARCH D
    ATA=";D$(0)
380 O=LEN(D$(0)):PR
    INT "    SEARCH
    ING...";
390 FOR I=1 TO P
400 IF LEFT$(D$(I),
    O)<>D$(0) THEN
    420
410 CLS :PRINT "#";
    I;"=";D$(I):PRI
    NT "SEARCHING..
    .";
420 NEXT I:CLS
430 GOTO 30
440 FOR I=1 TO P
450 PRINT "DATA#";I
    ;"=";D$(I)
460 NEXT I
470 GOTO 30
480 REM PRINTER
490 IF PP=1 THEN PP
    =0:PRINT OFF :
    GOTO 510
500 IF PP=0 THEN PP
    =1:PRINT ON
510 GOTO 30
```

6-4 Contrôle de la méthode d'appel et de l'efficacité d'un démarchage par correspondance

Le démarchage par correspondance est très souvent utilisée pour promouvoir la vente d'un produit, mais on s'intéresse surtout de la réaction qu'il entraîne. Ce programme a pour but d'établir l'efficacité de la diffusion d'un nouveau prospectus, préparé et adressé avec un nouvel appel, différent du précédent; il se base pour cela sur le nombre de commandes reçues. La méthode qu'utilise ce programme est celle du contrôle de pourcentage (contrôle unilatéral), qui est l'une des méthodes utilisées en statistique (elle ne vaut que si les commandes excèdent les pourcentages précédents).

Exemple:
Pourcentage précédent de commande (nombre de commandes/nombre de prospectus adressés) = 2,8%
Nombre de prospectus adressés cette fois ci = 6000
Nombre de commandes reçues cette fois ci = 186
Il s'avère donc que cette distribution de prospectus a amené un pourcentage de commandes de 3,1%. Mais que donnerait un contrôle de pourcentage?

```
RUN↵                **DIRECT MAIL**
↵                   BEFORE(%)?
```

Introduire le pourcentage précédent de commandes.

```
2.8↵                SAMPLE SIZE=?
```

Introduire 6000 pour les prospectus adressés cette fois ci.

```
6000↵               ORDERS RECEIVED=?
```

Introduire le nombre de commandes reçues cette fois ci.

```
186↵                RELIABILITY (90,95,99%)?
```

Introduire 90%, 95% ou 99% pour la fiabilité de la méthode d'appel du démarchage par correspondance.

```
90↵                 BEFORE(%)= 2.8
```


Les résultats du contrôle des données introduites pour une fiabilité de 90% seront affichés.

```
↵                   SAMPLE SIZE= 6000
↵                   ORDERS RECEIVED (%)= 3.1
↵                   RELIABILITY= 90%
↵                   CRITERION 1.408590425
↵                   EFFECTIVE!
```

Pour une fiabilité de 90%, l'ordinateur démontre que l'appel du nouveau démarchage est efficace.
Qu'est ce que cela donnera maintenant avec une fiabilité de 95%?

```
↵                   NEXT RELIABILITY (Y/N)?
Y↵                  RELIABILITY (90,95,99%)?
95↵                 BEFORE(%)= 2.8
↵                   SAMPLE SIZE= 6000
↵                   ORDERS RECEIVED (%)= 3.1
↵                   RELIABILITY= 95%
↵                   CRITERION 1.408590425
↵                   INEFFECTIVE!
```

L'ordinateur a calculé que pour une fiabilité de 95% l'appel n'est pas efficace.

En procédant de la même manière avec une fiabilité de 99%, nous obtiendrons les résultats suivants.

Critère Z = 1,409 (1,408590425)

Sera efficace pour une fiabilité de 90% puisque

$Z = 1{,}409 > Z_{0,1} = 1{,}282.$

Ne sera pas efficace pour une fiabilité de 95% puisque

$Z = 1{,}409 < Z_{0,05} = 1{,}645.$

Ne sera pas efficace pour une fiabilité de 99% puisque

$Z = 1{,}409 < Z_{0,01} = 2{,}326.$

($Z_{0,1}$, $Z_{0,05}$ et $Z_{0,01}$ sont basés sur un tableau de distribution normal.)

Variables

Nom des variables	Contenu des variables
A	Sélection de la fiabilité
C( )	Critère
D	Indicateur de critère
F	Quantité de critère
E$	Changement ou non
N	Nombre de prospectus adressés
P	Pourcentage de commandes reçues jusqu'à maintenant
PE$	%
X	Commandes reçues cette fois-ci

Liste de programme

```
  5 PRINT "    **DI
    RECT MAIL**"
 10 CLEAR :DIM C(3)
    :PE$=CHR$(37)
 20 C(1)=1.282:C(2)
    =1.645:C(3)=2.3
    26
 30 INPUT "BEFORE("
    +PE$+")";P:P=P/
    100
 40 INPUT "SAMPLE S
    IZE=";N:IF N<=0
     THEN 40
 50 INPUT "ORDERS R
    ECEIVED=";X:X=X
    /N
 60 PRINT "RELIABIL
    ITY (90,95,99"+
    PE$+")";:INPUT
    A
 70 IF A=90 THEN D=
    1: GOTO 110
 80 IF A=95 THEN D=
    2: GOTO 110
 90 IF A=99 THEN D=
    3: GOTO 110
100 GOTO 60
110 F=(X-P)/SQR(P*(
    1-P)/N)
120 PRINT "BEFORE("
    ;PE$;")=";P*100
130 PRINT "SAMPLE S
    IZE=";N
140 PRINT "ORDERS R
    ECEIVED (";PE$;
    ")=";X*100
150 PRINT "RELIABIL
    ITY=";A;PE$
160 PRINT "CRITERIO
    N";F
170 IF F<-C(D) THEN
     BEEP :PRINT "E
    FFECTIVE!": GOT
    O 200
180 IF F>C(D) THEN
    BEEP :PRINT "EF
    FECTIVE!": GOTO
     200
190 BEEP :BEEP :PRI
    NT "INEFFECTIVE
    !"
200 PRINT "NEXT REL
    IABILITY (Y/N)"
    ;:INPUT E$
210 IF E$="Y" THEN
    60
220 END
```

6-5 Jeu: le monstre de la route

Vous rentrez tranquillement chez vous quand soudain vous apercevez un monstre tapi au bord du chemin. Et il n'y a qu'une route qui mène à votre maison! Cependant le monstre a ses points faibles: il n'a pas bonne vue et n'attrapera que ce qui bouge. Tant que vous ne bougez pas, vous n'avez rien à craindre.
craindre.
Suivez ce conseil et rentrez bien chez vous. Bonne chance!

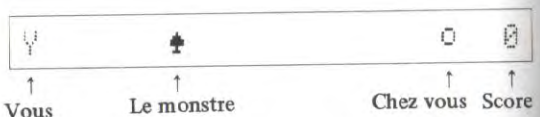
Caractères qui apparaissent

Y (vous) . . . . . Actionnez la touche [1] (ralentissement) et la touche [3] (accélération) et essayez de rentrer chez vous sans vous laisser attraper par le monstre.

♠ (le monstre) . . . . Apparaît et disparaît, se déplace au hasard sur l'écran.

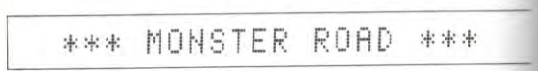
o (chez vous) . . . . 10 points vous seront ajoutés à chaque fois que vous parviendrez à rentrer chez vous sain et sauf.

Affichage du jeu



Règle du jeu

RUN↵



Le jeu commence après l'affichage du titre.
Dirigez vous vers votre maison tout en surveillant les mouvements du monstre.

[3] (accélération)



Le monstre approche. Ralentissez en appuyant sur la touche [1] et surveillez attentivement le monstre.

[1] (ralenti)



Tant que vous ne bougerez pas, vous n'aurez pas à craindre d'être attrapé par le monstre mais, si le monstre se rend compte que vous bougez, il vous attaquera implacablement. Le monstre est venu derrière vous pendant que vous étiez immobile. Dépêchez vous de rentrer chez vous. Bonne chance!

[3]



Encore un peu.

[3]



Vous avez échappé au monstre et êtes arrivé chez vous.

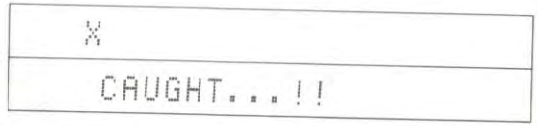


Votre score est affiché et le jeu suivant commence.

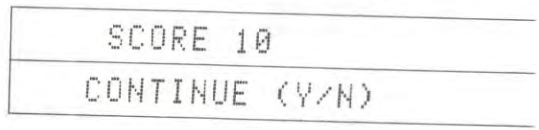
[3]



Cette fois-ci vous avez manqué de chance et vous êtres fait attraper.



Le score que vous avez obtenu jusqu'à maintenant sera affiché avec accompagnement d'un "bip" sonore.
L'ordinateur vous demande si vous souhaitez jouer encore une fois.
Si oui, appuyez sur la touche [Y] , si non, appuyez sur la touche [N] .



Variable

Nom des variables	Contenu de variable
X	Votre position
Y	Position du monstre
Z	Score
P	Vitesse
A	Votre position précédente
B	Position précédente du monstre
C	Contrôle d'introduction
D$	Introduction

Liste de programme

```
10 CLS :WAIT 60:PR
   INT "  *** MONS
   TER ROAD ***":Z
   =-10
20 CLS :Z=Z+10:LOC
   ATE 20:PRINT CH
   R$(237);USING"#
   ##";Z;:BEEP
30 A=0:B=0:P=0:X=0
   :Y=INT(RND*15+3
   )
40 LOCATE A:PRINT
   " ";:LOCATE B:P
   RINT " ";
50 LOCATE X:PRINT
   "Y";:LOCATE Y:P
   RINT CHR$(232);
60 A=X:B=Y:IF X=Y
   THEN IF P>0 THE
   N 200
70 FOR C=0 TO 5:D$
   =INKEY$:IF D$="
   " THEN NEXT C
80 IF D$="3" THEN
   P=P+1:IF P=4 TH
   EN P=3
90 IF D$="1" THEN
   P=P-1:IF P=-1 T
   HEN P=0
100 X=X+P:IF X>19 T
   HEN 20
110 Y=Y+SGN(A-Y-.1)
   *SGN(RND-.1)*IN
   T(RND*4+1)
120 IF Y<1 THEN Y=1
    ELSE IF Y>18 T
   HEN Y=18
130 GOTO 40
200 BEEP 1:BEEP :LO
   CATE X:PRINT "X
   ":BEEP
210 PRINT :PRINT "
       CAUGHT...!!"
220 PRINT "    SCOR
   E";Z
230 PRINT "   CONTI
   NUE (Y/N)";
240 D$=INKEY$:IF D$
   ="" THEN 240
250 IF D$="Y" THEN
   10 ELSE END
```


CHAPITRE 7

LISTE DES FONCTIONS ET NORMES

7-1 FONCTIONS NUMERIQUES

Fonction	Expression mathématique	Format	Signification
Fonctions trigonométriques	sin	SIN (expression numérique) *** A partir d'ici X.**	$-5400° < X < 5400°$ (30π rad, 6000 gra)
	cos	COS (X)	$-5400° < X < 5400°$ (30π rad, 6000 gra)
	tan	TAN (X)	$-5400° < X < 5400°$ (30π rad, 6000 gra) (à l'exclusion des cas où $\|X\| = (2n-1) \times 90°$) *** n est un nombre entier.**
Fonctions trigonométriques inverses	$\sin^{-1}$	ASN (X)	$\|X\| \leqq 1, -90° \leqq$ ASN (X) $\leqq 90°$
	$\cos^{-1}$	ACS (X)	$\|X\| \leqq 1, 0° \leqq$ ACS (X) $\leqq 180°$
	$\tan^{-1}$	ATN (X)	$\|X\| < 10^{100}, -90° \leqq$ ATN (X) $\leqq 90°$
Fonctions hyperboliques	sin h	HYPSIN (X)	$\|X\| \leqq 230$
	cos h	HYPCOS (X)	$\|X\| \leqq 230$
	tan h	HYPTAN (X)	$\|X\| \leqq 10^{100}$, $-1 \leqq$ HYP TAN(X) $\leqq 1$
Fonctions hyperboliques inverses	$\sin h^{-1}$	HYPASN (X)	$\|X\| < 5 \times 10^{99}$
	$\cos h^{-1}$	HYPACS (X)	$1 \leqq X < 5 \times 10^{99}$
	$\tan h^{-1}$	HYPATN (X)	$\|X\| < 1$
Fonction exponentielle	e^x	EXP (X)	$-227 \leqq X \leqq 230$
Logarithme naturel	$\log_e x$	LOG (X)	$X>0$ la précision sera de ±1 au 8ème chiffre quand $1 - 10^{-6} < X < 1 + 10^{-6}$
Logarithme ordinaire	$\log_{10} x$	LGT (X)	$X > 0$ la précision sera de ±1 au 8ème chiffre quand $1 - 10^{-6} < X < 1 + 10^{-6}$
Racine carrée	$\sqrt{\ }$	SQR (X)	$X \geqq 0$
Valeur absolue	$\|x\|$	ABS (X)	Donne la valeur absolue de X.
Signe		SGN (X)	Donne {-1 quand $X < 0$, 0 quand $X = 0$, 1 quand $X > 0$.}
Nombre entier		INT (X)	Fonction de gauss: donne le plus grand nombre entier qui ne dépasse pas la valeur de X.

Fonction	Expression mathématique	Format	Signification
Fraction		FRAC (X)	Donne la fraction de X.
Arrondi		ROUND(x,y[,f][,{+ / -}])	Arrondit la valeur de x à la position 10^y.
		x,y: X f: 0 ou 1	{f est omis: / $f = 0$: } spécifie la position de 10^y $f = 1$: Spécifie la $y$ème position d'un nombre significatif
			{Signe omis: arrondit au plus près +: arrondit au nombre supérieur −: écarte
Degrés/minutes /secondes	Nombres sexagésimaux → décimaux	DEG (d[,m[,s]]) d, m, s: X	Donne des valeurs converties en décimales
Constante circulaire	π	PI	Donne une valeur approximative de π à 3,1415926536
Nombre aléatoire		RND	0 < Nombre aléatoire < 1 Un nombre aléatoire à 10 chiffres est constitué.

(X) → Expression numérique

* Les parenthèses ne peuvent pas être omises pour ROUND et DEG, elles peuvent par contre être omises dans les autres cas où des valeurs numériques ou des variables sont utilisées comme (expression numérique).

7-2 FONCTIONS STATISTIQUES

Format		Fonction
CNT*	n	Nombre de données statistiques traitées
SUMY*	Σy	Total des données y
SUMX*	Σx	Total des données x
SUMXY*	Σxy	Somme des produits des données x par les données y
SUMX2*	Σx^2	Somme des carrés des données x
SUMY2*	Σy^2	Somme des carrés des données y
SDX	$x\sigma_{n-1}$	Valeur de l'écart type de l'échantillon de la donnée x $\sqrt{\frac{n\Sigma x^2-(\Sigma x)^2}{n(n-1)}}$
SDY	$y\sigma_{n-1}$	Valeur de l'écart type de l'échantillon de la donnée y $\sqrt{\frac{n\Sigma y^2-(\Sigma y)^2}{n(n-1)}}$
SDXN	$x\sigma_n$	Valeur de l'écart type démographique de la donnée x $\sqrt{\frac{n\Sigma x^2-(\Sigma x)^2}{n^2}}$
SDYN	$y\sigma_n$	Valeur de l'écart type démographique de la donnée y $\sqrt{\frac{n\Sigma y^2-(\Sigma y)^2}{n^2}}$
LRA	A	Limite de la constante de régression linéaire $\frac{\Sigma y-\mathrm{LRB}\cdot\Sigma x}{n}$
LRB	B	Coefficient de régression linéaire $\frac{n\cdot\Sigma xy-\Sigma x\cdot\Sigma y}{n\cdot\Sigma x^2-(\Sigma x)^2}$
COR	r	Coefficient de corrélation $\frac{n\Sigma xy-\Sigma x\cdot\Sigma y}{\sqrt{\{n\Sigma x^2-(\Sigma x)^2\}\{n\Sigma y^2-(\Sigma y)^2\}}}$
EOX (X)	$\hat{x}$	Valeur estimée (Valeur de x estimée à partir de la valeur de y) $\mathrm{EOX}(y_n)=\frac{y_n-\mathrm{LRA}}{\mathrm{LRB}}$
EOY (X)	$\hat{y}$	Valeur estimée (Valeur de y estimée à partir de la valeur de x) $\mathrm{EOY}(x_n)=\mathrm{LRA}+x_n\cdot\mathrm{LRB}$

(X) → Expression numérique

* Sur introduction de STAT LIST ↵ les statistiques de base dont le nom est suivi du signe ★ sont affichées en ordre séquentiel avec leur valeur.
Appuyez sur ↵ ou ENTER si vous désirez interrompre l'affichage, appuyez à nouveau sur la même touche si vous voulez rétablir l'affichage. Pour les sorties imprimées, introduisez STAT LLIST ↵.

7-3 FONCTIONS DE CARACTERES

Num	Format	Fonction
ASC	ASC(X$)	Donne le code ASCII du premier caractère de l'expression de caractères.
CHR$	CHR$(I)	Donne le caractère correspondant au code ASCII par l'intermédiaire d'une expression numérique. $0 \leq I < 256$
DMS$	DMS$(I)	Convertit la décimale en sexagésimale. $\|I\| < 10^{100}$
HEX$	HEX$(I)	Donne la chaîne de caractères de l'hexadécimale obtenue par conversion d'une décimale. $-32769 < I < 65536$
INKEY$	INKEY$	Permet l'entrée d'un caractère à partir du clavier.
LEN	LEN(X$)	Donne le nombre des caractères contenus dans une expression de caractères. $0 \leq X\$ \leq 79$
VAL	VAL(X$)	Convertit une expression de caractères en valeur numérique.
LEFT$	LEFT$(X$, I)	Recherche le nombre de caractères spécifiés par une expression numérique à partir de la gauche d'une expression de caractères. $0 \leq I < 256$
MID$	MID$(X$, I, J)	Recherche le nombre de caractères spécifiés par l'expression numérique No. 2 à partir de la position spécifiée par l'expression numérique No. 1.
RIGHT$	RIGHT$(X$, I)	Recherche le nombre de caractères spécifié par une expression numérique à partir de la droite d'une expression de caractères. $0 \leq I < 256$
STR$	STR$ (I)	**Convertit une expression numérique en chaîne de caractères.**

X$ → Expression de caractères
I → Expression numérique ou expression numérique 1.
J → Expression numérique 2.

7-4 CONSTANTES SCIENTIFIQUES

Opération	Nom	Symbole	Valeur numérique	Unité
F 0	Vitesse d'accélération d'un corps en chute libre	g	9.80665	ms^{-2}
F 1	Vitesse de la lumière (dans l'espace)	c	299792458	ms^{-1}
F 2	Constante de Plank	h	6.626176×10^{-34}	Js
F 3	Constante de la gravitation	G	6.672×10^{-11}	Nm^2kg^{-2}
F 4	Charge élémentaire	e	$1.6021892 \times 10^{-19}$	C
F 5	Masse de l'électron	me	9.109534×10^{-31}	kg
F 6	Masse atomique	u	$1.6605655 \times 10^{-27}$	kg
F 7	Constante d'Avogadro	N_A	6.022045×10^{23}	mol^{-1}
F 8	Constante de Boltzmann	k	1.380662×10^{-23}	JK^{-1}
F 9	Volume de 1 kg-mole sous STP.	V_m	0.02241383	m^3mol^{-1}

7-5 COMMANDES MANUELLES

Commande	Exemple d'utilisation	Fonction
CONT	CONT 0	Réexécute un programme arrêté à l'aide de l'instruction STOP ou de la touche STOP.
DELETE	DELETE 20	Efface uniquement la ligne 20.
	DELETE 200 –	Efface à partir de la ligne 200.
	DELETE – 80	Efface de la première ligne à la ligne 80.
	DELETE 50 – 100	Efface de la ligne 50 à la ligne 100.
EDIT	EDIT	Affiche la première ligne du programme et met en mode de correction (EDIT).
	EDIT 50	Affiche la ligne 50 et met en mode de correction (EDIT).
LIST	LIST	Affiche les contenus du programme dans la zone de programme actuellement spécifiée.
	LIST 20	Affiche la ligne 20.
	LIST 200 –	Affiche à partir de la ligne 200.
	LIST – 80	Affiche de la première ligne à la ligne 80.
	LIST 50 – 100	Affiche de la ligne 50 à la ligne 100.
	LIST ALL	Affiche les contenus des programmes dans toutes les zones de programme.
	LIST V	Affiche les noms des variable enregistrées et les noms des variables de tableau déclarées.
LLIST	LLIST	Imprime les contenus des programmes dans la zone de programme actuellement spécifiée.
	LLIST 20	Imprime la ligne 20.
	LLIST 200 –	Imprime à partir de la ligne 200.
	LLIST – 80	Imprime de la première ligne à la ligne 80.
	LLIST 50 – 100	Imprime de la ligne 50 à la ligne 100.
	LLIST ALL	Imprime les contenus des programmes dans toutes les zones de programme.
	LLIST V	Imprime les noms des variables enregistrées et les noms des variables de tableau déclarées.
LOAD	LOAD	Charge le programme d'une cassette dans la zone de programme actuellement spécifiée.
	LOAD ALL	Charge les programmes d'une cassette dans toutes les zones de programme.
	LOAD,A	Charge un programme en format de code ASCII dans la zone de programme actuellement spécifiée.
	LOAD,M	Mélange le programme dans la zone de programme actuellement spécifiée avec le programme introduit avec format de code ASCII.

Commande	Exemple d'utilisation	Fonction
LOAD	LOAD "ABC" LOAD ALL "CASIO" LOAD "TEST" A LOAD "TEST",M	Accomplit les mêmes fonctions que le format ci-dessus en liaison avec le programme et avec un nom particulier.
NEW	NEW	Efface les programmes dans la zone de programme actuellement spécifiée.
	NEW ALL	Efface les programmes dans toutes les zones de programme et toutes les variables.
PASS	PASS "ABC"	Affecte/libère un mot de passe.
PROG	PROG 3	Spécifie P3 comme zone de programme à utiliser.
RUN	RUN	Démarre l'exécution à partir de la première ligne du programme dans la zone de programme actuellement spécifiée.
	RUN 1000	Démarre l'exécution du programme à partir de la ligne 1000.
SAVE	SAVE	Sauvegarde le programme des zones de programme actuellement spécifiées sur une cassette.
	SAVE ALL	Sauvegarde tous les programmes de toutes les zones de programme sur une cassette.
	SAVE,A	Enregistre le programme dans la zone de programme actuellement spécifiée sur une cassette avec le format de code ASCII.
	SAVE "ABC" SAVE ALL "CASIO" SAVE "TEST",A	Stocke le programme avec son nom de fichier sur une cassette.
SYSTEM	SYSTEM	Affiche la situation de la zone de programme et le nombre d'octets qui restent pour la zone de l'utilisateur.
	SYSTEM P	Affiche le nombre d'octets utilisés pour un programme et le nombre d'octets pour la zone de gestion de programme.
	SYSTEM V	Affiche le nombre d'octets utilisés pour des variables et aussi le nombre d'octets pour une zone de gestion de variable.
VERIFY	VERIFY	Contrôle l'état de sauvegarde des programmes et des données sur une cassette.
	VERIFY "ABC"	Accomplit la même opération que ci-dessus pour un programme ayant un nom de fichier spécifié.

7-6 COMMANDES DE PROGRAMME

Les commandes marquées d'une astérique * peuvent fonctionner aussi en commandes manuelles.

Commande	Exemple d'utilisation	Fonction
ANGLE*	ANGLE 0	Sélectionne les degrés comme unités d'angle.
	ANGLE 1	Sélectionne les radians comme unités d'angle.
	ANGLE 2	Sélectionne les grades comme unités d'angle.
BEEP*	BEEP	Produit un "bip" sonore de tonalité basse.
	BEEP 0	Voir BEEP
	BEEP 1	Produit un "bip" sonore de tonalité haute.
CHAIN*	CHAIN	Charge le programme d'une cassette, qui apparait en premier, et l'exécute.
	CHAIN "ABC"	Charge le programme d'une cassette ayant le nom de fichier spécifié, et l'exécute.
CLEAR	CLEAR*	Efface toutes les variables
CLS	CLS	Efface l'affichage et envoie le curseur à l'extrémité gauche de l'écran.
DATA*	DATA 1,2,3	Stocke les données à lire avec l'instruction READ.
DIM*	DIM A(3)	Déclare les tableaux numériques à une dimension et en simple précision.
	DIM B(2,3)	Déclare les tableaux numériques à deux dimensions et en simple précision.
	DIM C!(4)	Déclare les tableaux numériques à une dimension et en demie-précision.
	DIM D!(3,4)	Déclare les tableaux numériques à deux dimensions et en demie-précision.
	DIM G$(2)*3	Déclare un tableau de caractères à une dimension et spécifie 3 comme longueur de caractères.
	DIM H$(4,5)*6	Déclare un tableau de caractères à deux dimensions et spécifie 6 comme longueur de caractères.
END	END	Termine l'exécution d'un programme.
ERASE*	ERASE D	Annule la définition de la variable enregistrée D et de la variable de tableau D.
FOR ~ TO ~ STEP ~ NEXT	FOR I = 5 TO 20 STEP 0,5 ~ NEXT I	Change de manière répétitive les variables de contrôle entre l'instruction FOR et l'instruction NEXT en les faisant passer de la valeur initiale à la valeur ayant les incréments spécifiés.
GET*	GET A	Introduit les données stockées sur une cassette et dans une variable à l'aide de la commande PUT.
	GET "TEST" A	Introduit les données ayant le nom de fichier "TEST" dans la variable A.
GOSUB	GOSUB 1000	Saute à la ligne 1000.
	GOSUB PROG 9	Saute à la zone de programme 9, préparée pour un sous-programme.

Commande	Exemple d'utilisation	Fonction
RETURN	RETURN	Revient à la ligne située juste après l'instruction GOSUB.
GOTO	GOTO 150	Saut inconditionnel à la ligne 150.
	GOTO PROG 5	Saut inconditionnel à la zone de programme P5.
IF expression conditionnelle THEN ~ ELSE ~	IF X > 10 THEN 10 ELSE 80	Exécute les instructions qui suivent THEN quand l'expression conditionnelle est oui (YES). Exécute les instructions qui suivent ELSE si la réponse n'est pas juste.
INPUT	INPUT X	Affiche ? et attend l'introduction de la variable X.
	INPUT "Message", X	Affiche un message et attend l'introduction de la variable X.
	INPUT "TEST"; X	Affiche un message et ? et attend l'introduction de la variable X.
LET	LET A = B	Affecte la valeur de droite du même signe que la variable située à gauche.
LOCATE	LOCATE 5	Spécifie la position du curseur.
ON ~ GOSUB	ON X GOSUB 200, 300	Saute à un sous-programme en fonction de la valeur de la variable X.
ON ~ GOTO	ON A GOTO 100, 110	Saut inconditionnel en fonction de la valeur de la variable A.
PRINT*	PRINT X,Y	Affiche les valeurs de X et Y.
	PRINT X;Y	Affiche successivement les valeurs de X et Y.
LPRINT*	LPRINT X,Y	Imprime les valeurs de X et Y.
	LPRINT X;Y	Imprime successivement les valeurs de X et Y.
PRINT ON*	PRINT ON	Met en mode d'impression.
PRINT OFF*	PRINT OFF	Libère le mode d'impression.
PUT*	PUT A	Enregistre la valeur affectée à la variable A sur une cassette.
	PUT "DATA" A	Enregistre la valeur affectée à la variable avec un nom de fichier spécifié.
READ	READ X	Introduit les données stockées par une instruction DATA dans une variable spécifiée.
REM	REM ***	Fournit un commentaire à un programme.
RESTORE	RESTORE	Spécifie que la première instruction DATA doit être lue lorsqu'on exécute l'instruction READ suivante.
	RESTORE 100	Spécifie que l'instruction DATA de la ligne 100 doit être lue lorsqu'on exécute l'instruction READ suivante.

Commande	Exemple d'utilisation	Fonction
STAT*	STAT X,Y;N	Introduit des données statistiques et le nombre de ces données.
	STAT CLEAR	Initialisation des registres contenant des statistiques de base.
	STAT LIST	Affiche les statistiques de base.
	STAT LLIST	Imprime les statistiques de base.
STOP	STOP	Arrête l'exécution d'un programme.
TRON*	TRON	Met en mode d'analyse.
TROFF*	TROFF	libère le mode d'analyse.
WAIT	WAIT	Spécifie un temps de pause pendant l'exécution d'une instruction PRINT.

7-7 FONCTIONS DE CONTRÔLE

Num	Format	Fonction
TAB	TAB (expression numérique)	Imprime un nombre spécifié d'espaces.
USING	USING	Spécifie les formats de sortie des instructions PRINT et LPRINT.

7-8 MESSAGES D'ERREUR

Message d'erreur	Erreur	Mesure corrective
Erreur BS (indice inférieur incorrect)	1) La valeur de l'indice d'une variable de tableau est négative ou plus grande ou égale à 256. Exemple: DIM A (256) 2) La valeur numérique spécifiée sort de la plage de l'argument. Exemple: l'instruction LOCATE dépasse les limites de $0 \leqq X < 24$	1) Changer l'indice en choisissant une valeur comprise dans les limites spécifiées. Si l'indice est une variable, vérifier le programme correspondant. 2) Respécifier l'indice inférieur de telle sorte qu'il soit dans la plage spécifiée de l'argument.
Erreur BV (Débordement de tampon)	1) Un dépassement s'est produit dans les tampons d'entrée et de sortie.	1) Exprimer l'opération et le programme en 79 caractères ou moins par ligne.
Erreur DA (Introduction sans donnée)	1) Une instruction READ ou une instruction GET a été exècutée alors qu'il n'y avait pas de données à lire.	1) Contrôler le rapport entre les instructions READ et DATA. Créer dans l'instruction des données à affecter à la variable.
Erreur DD (Définition en double)	1) Des tableau de même nom mais d'indices inférieurs différents ont été définis en double. Exemple: Les déclarations de variable de tableau suivantes avec une instruction DIM entraîne une erreur DD. DIM A(1), A(2, 3) DIM A!(1), A!(2, 3) DIM A$(1), A$(2, 3) DIM A$(1)*20, A$(2,3)*20	1) Contrôler la variable de la ligne dans laquelle l'erreur s'est produite. Contrôler aussi les indices inférieurs des tableaux ayant le même nom. Si ces indices sont différents, changer un des noms de tableau et réorganiser le programme.
Erreur FC (Appel illégal de fonction)	1) On a essayé d'exécuter l'une des commandes manuelles suivantes dans un programme: CONT, PASS, RUN, EDIT et DELETE. 2) On a essayé d'exécuter manuellement l'une des commandes programmées suivantes: END, LET, REM, STOP, LOCATE, GOTO, GOSUB, RETURN, INPUT, DATA, READ, RESTORE, FOR ~ NEXT, IF ~ THEN ~ ELESE.	1) Annuler la commande incorrecte et réorganiser le programme. 2) Exécuter la commande, avec un numéro de ligne rattaché dans un programme.

Message d'erreur	Erreur	Mesure corrective
Erreur FC (Appel illégal de fonction)	3) Une commande CONT a été employée alors que l'exécution du programme ne pouvait pas continuer.	3) Appuyer sur la touche BRK et réexécuter le programme à partir du début. Si vous savez à quelle ligne le programme est interrompu, réexécuter à partir de la ligne suivante en introduisant RUN numéro de ligne ↵.
Erreur FO (NEXT sans FOR)	1) Il n'y a pas d'instruction FOR correspondant à une instruction NEXT.	1) Contrôler la structure d'emboîtement.
Erreur GS (RETURN sans GOSUB)	1) Il y a une instruction RETURN qui ne correspond à aucune instruction GOSUB.	1) Contrôler la structure d'emboîtement de l'instruction GOSUB. Séparer le programme principal et le sous-programme.
Erreur MA (Erreur mathématique)	1) Une opération arithmétique ou une opération de fonction numérique est incertaine ou impossible. Exemple: division par 0.	1) Contrôler l'expression numérique de la ligne à laquelle l'erreur s'est produite. Contrôler aussi la valeur de la variable associée.
Erreur NO (Débordement d'emboîtement)	1) Les niveaux d'emboîtement dépassent la plage specifiée. Exemple: GOSUB/RETURN 12 niveaux maximum FOR ~ NEXT 6 niveaux maximum	1) Contrôler la structure d'emboîtement et amener le nombre d'emboîtements dans la plage tolérée.
Erreur NR (Dispositifs pas prêt)	1) Les dispositifs d'E/S ne sont pas correctement connectés. Exemple: Aucune imprimante n'est connectée.	1) S'assurer que les dispositifs périphériques sont correctement connectés. S'assurer aussi que le FA-20 est suffisamment alimenté et que le papier n'est pas bloqué dans l'imprimante.
Erreur OM (Dépassement de mémoire)	1) La capacité de la carte RAM est insuffisante. Exemple: Une déclaration de tableau a provoqué un dépassement de mémoire.	1) Effacer les programmes inutiles. Augmenter la capacité mémoire avec la carte RAM. Contrôler le nombre d'octets restant à l'aide de la commande SYSTEM.
Erreur OV (Erreur de débordement)	1) Le résultat d'une opération ou la valeur numérique introduite dépasse $\|X\| < 10^{100}$.	1) Contrôler l'expression numérique de la ligne à laquelle l'erreur s'est produite. Insérer une instruction PRINT dans le programme pour contrôler la valeur de la variable.

Message d'erreur	Erreur	Mesure corrective
Erreur PR (Erreur protégée)	1) On a essayé d'exécuter une des commandes suivantes, alors qu'elles ne peuvent pas être exécutées dans un programme ayant un mot de passe: DELETE, LIST, LLIST, NEW et EDIT. 2) Un mot de passe différent a été introduit. 3) On a essayé de charger un programme dont le mot de passe est différent de celui du FX-750P.	1) Liberer le blocage en ré-insérant le mot de passe et exécuter le programme. 2) Introduire le mot de passe correct. 3) Annuler le mot de passe du FX-750P avant le chargement. Dans ce cas le mot de passe du programme devient le mot de passe du FX-750P.
Erreur RW (Erreur d'écriture/lecture)	1) Une erreur de parité s'est produite pendant l'exécution d'une commande LOAD ou VERIFY.	1) Sauvegarder à nouveau le programme à l'aide d'une commande SAVE.
Erreur SN (Erreur de syntaxe)	1) Il y a une erreur dans le format de commande. 2) Un numéro de ligne comprend une partie fractionnaire. 3) Un tableau à plus de deux dimensions a été déclaré.	1) Employer la commande EDIT pour corriger la ligne à laquelle l'erreur s'est produite. 2) Corriger le numéro de ligne. 3) Amener le tableau déclaré à un nombre de dimensions ne dépassant pas deux.
Erreur SO (Erreur de dépassement d'emboîtement)	1) L'emboîtement d'une valeur numérique est supérieur à huit niveaux. 2) L'emboîtement d'opérateurs dépasse 20 niveaux. 3) L'emboîtement de caractères dépasse dix niveaux.	1) et 2) Simplifier ou diviser l'expression numérique afin que le niveau d'emboîtement soit amené dans la plage spécifiée. 3) Simplifier ou diviser l'expression de caractères afin que le niveau d'emboîtement soit amené dans la plage spécifiée.
Erreur ST (Erreur de chaîne)	1) On a essayé d'affecter à une variable de caractères, une chaîne de caractères dépassant la longueur autorisée pour une variable de caractères. Les longueurs tolérées de variables de caractères sont les suivantes: Variables à caractères fixe: Max. 7 caractères Variable à caractères enregistrée: Max. 16 caractères Variable de tableau à caractères: Max. 79 caractères	1) Remplacer par une variable contenant plus de caractères. Diminuer le nombre de caractères de la chaîne de caractères à affecter à la variable. Faire attention quand on relie des chaînes de caractères.

Message d'erreur	Erreur	Mesure corrective
Erreur TM (Désaccord de type)	1) Dans une instruction d'affectation, les côtés gauche et droit ont un type de variable différent. 2) Lors d'une affectation, les types d'argument ne correspondent pas.	1) Unifier les côtés gauche et droit de l'instruction d'affectation; ils doivent être soit des variables numériques soit des variables à caractères. 2) Affecter une valeur numérique à une variable numérique et une chaîne de caractères à une variable de caractères.
Erreur UL (Numéro de ligne indéfini)	1) Le numéro de ligne spécifié dans une instruction IF ~ THEN, GOTO ou GOSUB n'existe pas. 2) Il n'y a pas de programme dans la zone de programme spécifiée par une instruction GOTO ou GOSUB.	1) Créer un numéro de ligne auquel un saut doit être fait ou changer le numéro de ligne spécifié auquel un saut doit être fait. 2) Créer une zone de programme pour la destination de branchement ou changer la destination de branchement.
Erreur UV (Variable non définie)	1) Une variable de tableau a été employée sans qu'elle soit déclarée par une instruction DIM.	1) Déclarer le tableau par une instruction DIM au début du programme.
Erreur VA (Erreur de variable)	1) On a essayé d'enregistrer plus de 41 variables.	1) On peut employer jusqu'à 40 variables enregistées et variables de tableau. Contrôler les noms de variable à l'aide de LIST V et supprimer les noms de variable inutiles à l'aide de CLEAR ou ERASE.

7-9 TABLEAU DES CODES DE CARACTERES

(Hexa-déci-male)	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	16	(Espace) 32	0 48	@ 64	P 80	96	112	128	144	(Espace) 160	ー 176	タ 192	ミ 208	224	240
1	1	(DEL) 17	! 33	1 49	A 65	Q 81	97	113	129	145	161	ア 177	チ 193	ム 209	225	241
2	(LINE TOP) 2	(INS) 18	" 34	2 50	B 66	R 82	98	114	130	146	162	イ 178	ツ 194	メ 210	226	242
3	3	19	# 35	3 51	C 67	S 83	99	115	131	147	163	ウ 179	テ 195	モ 211	227	243
4	4	20	$ 36	4 52	D 68	T 84	100	116	132	148	164	エ 180	ト 196	ヤ 212	228	244
5	5	21	% 37	5 53	E 69	U 85	101	117	133	149	165	オ 181	ナ 197	ユ 213	229	245
6	(LINE END) 6	22	& 38	6 54	F 70	V 86	102	118	134	150	ヲ 166	カ 182	ニ 198	ヨ 214	230	246
7	7	(ENTER) 23	' 39	7 55	G 71	W 87	103	119	135	151	ア 167	キ 183	ヌ 199	ラ 215	231	247
8	8	(▲) 24	(40	8 56	H 72	X 88	104	120	136	152	イ 168	ク 184	ネ 200	リ 216	♠ 232	248
9	9	25	) 41	9 57	I 73	Y 89	105	121	137	153	ウ 169	ケ 185	ノ 201	ル 217	♥ 233	249
A	10	26	* 42	: 58	J 74	Z 90	106	122	138	154	エ 170	コ 186	ハ 202	レ 218	◆ 234	250
B	11	27	+ 43	; 59	K 75	[91	107	123	139	155	オ 171	サ 187	ヒ 203	ロ 219	♣ 235	251
C	(CLS) 12	(→) 28	, 44	< 60	L 76	¥ 92	108	124	140	156	ヤ 172	シ 188	フ 204	ワ 220	● 236	252
D	(CR) 13	(←) 29	− 45	= 61	M 77	] 93	109	125	141	157	ユ 173	ス 189	ヘ 205	ン 221	○ 237	253
E	14	30	. 46	> 62	N 78	^ 94	110	126	142	158	ヨ 174	セ 190	ホ 206	゛ 222	238	254
F	15	31	/ 47	? 63	O 79	― 95	111	127	143	159	ツ 175	ソ 191	マ 207	° 223	239	255

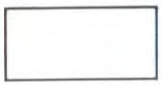
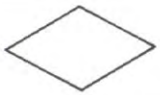
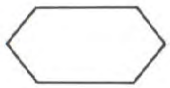
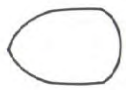
* Les caractères entre parenthèses sont des codes de contrôle et ne sont pas affichés.

7-10 LISTE DES MOTS RESERVES

Les mots ci-dessous sont appelés des mots réservés. Il servent à désigner des commandes et des fonctions. Les mots suivis du symbole * ne peuvent pas être utilisés pour des variables enregistrées.

&H	ERASE	LRA	SIN
ABS	EXP	LRB	SQR
ACS	FOR	MID$	STAT
ANGLE	FRAC	NEW	STEP
ASC	GET	NEXT	STOP
ASN	GOSUB	OFF	STR$
ATN	GOTO	ON*	SUMX
BEEP	HEX$	PASS	SUMX 2
CHAIN	HYP	PI*	SUMX Y
CHR$	IF*	PRINT	SUMY
CLEAR	INKEY$	PROG	SUMY 2
CLS	INPUT	PUT	SYSTEM
CNT	INT	READ	SYSTEM P
CONT	LEFT$	RESTORE	SYSTEM V
COR	LEN	RETURN	TAB
COS	LET	RIGHT$	TAN
DATA	LGT	RND	THEN
DEG	LIST	ROUND	TO*
DELETE	LIST V	RUN	TROFF
DMS$	LLIST	SAVE	TRON
EDIT	LLIST V	SDX	USING
ELSE	LOAD	SDXN	VAL
END	LOCATE	SDY	VERIFY
EOX	LOG	SDYN	WAIT
EOY	LPRINT	SGN	

7-11 LISTE DES SYMBOLES DES ORGANIGRAMMES

Symbole	Signification	Détails
	**Traitement**	Indique toutes sortes de types de fonctions de traitement.
	**Décision**	Décide de l'exécution de programme à brancher.
	**Document**	Indique une sortie sur imprimante.
	**Limite**	Indique le début, la fin ou l'arrêt d'un traitement.
	**Entrée manuelle**	Indique des introductions à partir du clavier.
	**Préparation**	Préparations pour l'établissement d'une valeur initiale, etc.
	**Bande magnétique**	Pour les dispositifs de mémoire externe tels qu'un magnétophone.
	**Affichage**	Indique l'affichage de messages ou de données.
	**Opération prédéterminée**	Indique des opérations prédéterminées comme les sous-programmes.

Symbole	Signification	Détails
	**Entrée/sortie**	Indique une entrée/sortie générale pour laquelle le support d'E/S n'est pas particulièrement spécifié.
	**Connecteur**	Indique une sortie vers un autre endroit ou une entrée à partir d'un autre endroit de l'organigramme.
	Connexion	Connecte les symboles mentionnés ci-dessus.

CARACTERISTIQUES

Type:
FX-750P

Fonctions fondamentales de calcul:
Nombres négatifs, exponentiels, addition/soustraction/multiplication/division entre parenthèses (avec fonction de jugement d'ordre de priorité, conforme à la vraie logique algébrique).

Fonctions incorporées:
Fonctions trigonométriques et trigonométriques inverses (unités d'angles en degrés, radians et grades), fonctions hyperboliques et hyperboliques inverses, fonctions logarithmiques et exponentielles, racines carrées, puissances, nombres entiers, fractions, valeurs absolues, signes, conversions des décimales en sexagésimales, conversions des décimales en hexadécimales, arrondissage, π et nombres aléatoires.

Fonctions de calcul statistique:
Ecart type = nombre de données, somme, somme des carrés, écart type (2 types)
Régression linéaire = nombre de données, somme de x, somme de y, somme de x^2, somme de y^2, somme des produits, écart type de x (2 types), écart type de y (2 types), terme constant, coefficient de régression, coefficient de corrélation, valeur estimée de x et valeur estimée de y.

Commandes:
CONT, DELETE, EDIT, LIST, LLIST, LOAD, NEW, PASS, PROG, RUN, SAVE, SYSTEM, VERIFY, ANGLE, BEEP, CHAIN, CLEAR, CLS, DIM, END, ERASE, FOR ~ TO ~ STEP, NEXT, GET, GOSUB, RETURN, GOTO, IF ~ THEN ~ ELSE, INPUT, LET, LOCATE, ON ~ GOSUB, ON ~ GOTO, PRINT, LPRINT, PRINT ON, PRINT OFF, PUT, READ, RESTORE, DATA, REM, STAT, STAT CLEAR, STAT LIST, STAT LLIST, STOP, TRON, TROFF, WAIT

Fonctions de caractères:
ASC, CHR$, VAL, STR$, DMS$, HEX$, LEFT$, RIGHT$, MID$, LEN, INKEY$

Fonctions d'affichage:
TAB, USING

Plage de calcul:
$\pm 1 \times 10^{-99} \sim \pm 9{,}999999999 \times 10^{-99}$. L'opération interne utilise une mantisse à 12 chiffres.

Langage de programme:
BASIC

Capacité RAM:
dans le cas où une carte RAM de 4 kilo-octets est utilisée,
zone de l'utilisateur: 2800 octets
zone de système: 1088 octets
zone de variable fixe: 208 octets

Capacité ROM (mémoire morte):
environ 27 kilo-octets

Nombre de zones de programme:
10 maximum (P0 – P9)

Nombre d'emboîtements:
sous-programmes: 12 niveaux
boucle FOR ~ NEXT: 6 niveaux
valeurs numériques: 8 niveaux
opérateurs: 20 niveaux

Affichage:
affiche avec une mantisse de 10 chiffres (signe moins non compris) ou 10 chiffres pour la mantisse + 2 chiffres pour l'exposant. Affiche aussi les symboles WAIT, DEG, RAD, GRA, TRON, PRTON et STOP.

Eléments d'affichage:
24 chiffres, affichage à cristaux liquides à impression par points.

Composant principal:
C-MOS-VLSI, etc.

Alimentation:
unité centrale . . . 2 piles au lithium (CR2032)
carte RAM . . . 1 pile au lithium (CR2016)

Consommation:
unité centrale . . . 0,03 W

Longévité des piles:
unité centrale . . . environ 120 heures
avec le FA-20 connecté . . . environ 60 heures
pile de carte RAM (gardée à part)
RC-2 . . . environ 2 ans
RC-4 . . . environ 1 an

Arrêt automatique:
environ 6 minutes après la dernière opération.

Plage de température ambiante:
0°C à 40°C

Dimensions et poids:
Unité centrale – 16 mm H x 186 mm L x 82 mm P, poids 226 g (piles comprises)
Carte RAM – 3,8 mm H x 60 mm L x 50 mm P, poids 26 g (une pile comprise).
pile)

INDEX

A

&H 281
ABS 270
ACS 113, 266
Affectation 44, 73
ANGLE 114, 222
ASC 169, 253
ASN 113, 266
ATN 113, 266

B

BEEP 203, 223
Boucle 231

C

Calculs manuels et corrections d'entrées 42
Calculs statistiques 55
Carte RAM 4
Enlèvement de la carte RAM 9
Mise en place de la carte RAM dans l'ordinateur 7
Remplacement de la pile de la carte RAM 10
CHAIN 224
Cha îne vide 215
CHR$ 145, 253
CLEAR 139, 225
CLS 174, 226
CNT 133, 275
Code ASCII 253
Commentaire 179, 247
Constantes scientifiques 61
CONT 207, 249
Contrôle de l'impression 88
COR 134, 279
Correction du programme 81
COS 111, 265
Coupure automatique de l'alimentation 23
Curseur 25

D

DATA 245
DEG (degrés) 24, 222
DELETE 209
Demie-précision 78
DIM 141, 227
DMS$ 256

E

EDIT 210
Emboîtement 231, 233
END 194, 229
Enlèvement de la carte RAM 9
EOX 134, 279
EOY 134, 280
ERASE 229
EXP 269
Expression conditionnelle 103, 235

F

Fonctions de caractères 53
Fonctions numériques 46, 47
FOR ~ TO ~ STEP/NEXT 149, 230
FRAC 271

G

GET 232
GOSUB 190, 233
GOTO 101, 234
GRADE (Grades 24, 222
Guillemets 40, 83

H

HEX$ 257
HYPASN, HYPACS, HYPATN 268
HYPSIN, HYPCOS, HYPTAN 267

I

IF ~ THEN ~ ELSE 103, 235
INKY$ 167, 261
INPUT 98, 236
INT 201, 270

L

LEFT$ 180, 258
LEN 261
LET 119, 237
LGT 269
LIST, LLIST 211
LOAD (ALL) 212
LOCATE 238
LOG 195, 269
LPRINT 241
LRA, LRB 132, 278

M

Message 98
MESSAGES D'ERREUR 324
Messages d'erreur consécutifs à l'insertion ou à l'enlèvement d'une carte RAM 21
Méthodes de calcul 39
MID$ 181, 260
Mise en place de la carte RAM dans l'ordinateur 7
Mode décalage 27
Mode de fonction 28
Mode direct 26
Modification des entrées (correction, suppression, insertion) 43
MOTS RESERVES 329

N

NEW (ALL) 215
Nom de la variable de contrôle 230
Nombre aléatoire 200, 274
Nombre d'octets selon la combinaison des cartes RAM 14
Nombre de chiffres d'entrée/sortie et de chiffes de calcul 41

O

ON ~ GOSUB 239
ON ~ GOTO 240
Opérateur de relations 38
Option 31
Ordre de priopité des calculs 38

P

PASS 216
PI 274
Point virgule 83, 97
PRINT 241
PRINT ON, PRINT OFF 243
PROG 217
PUT 244

R

RADIAN (Radians) 24, 222
READ 160 245
Réglage du contraste 25
REM 179, 247
Remplacement de la pile de la carte RAM 10
RESTORE 245
RETURN 190, 233
RIGHT$ 182, 259

RND 200, 274
ROUND 146, 272
RUN 82, 217

S

SAVE (ALL) 33, 218
SDX, SDY 126, 277
SDXN, SDYN 126, 277
SGN 271
Simple décision 78
SIN 111, 265
Sous-programme 191
SQR 268
STAT 248
STAT CLEAR 249
STAT LIST 219
STAT LLIST 219
Stockage et chargement de données 206
Stockage et chargement de programme 33
STOP 249
STR$ 165, 255
SUMX, SUMY 124, 275
SUMX2, SUMY2 124, 276
SUMXY 276
SYSTEM 13, 220

T

TAB 262
Tableau à deux dimensions 142
Tableau à une dimension 142
TABLEAU DES CODES DE CARACTERES 328
TAN 111, 265
Topographie mémoire 13
Touche de retour 30, 45
Touche d'introduction 30, 45
Touches d'édition 28
Touches spéciales 28
TRON, TROFF 251

U

USING 115, 263

V

VAL 163, 254
Variables 74
Variables à caractères 77
Variables de tableau 75
Variables fixes 77
Variables marquées 77
Variables numériques 77
Vérification des programmes d'introduction 81
VERITY 221

W

WAIT 203, 252

Z

Zone de gestion de programme 12, 18
Zone de gestion de variable 12, 18
Zone de l'utilisateur 12, 18
Zone de système 12, 13
Zone de système ordinaire 13, 14
Zone de variable fixe 13

CASIO®

094B M Ⓐ Imprimé au Japon