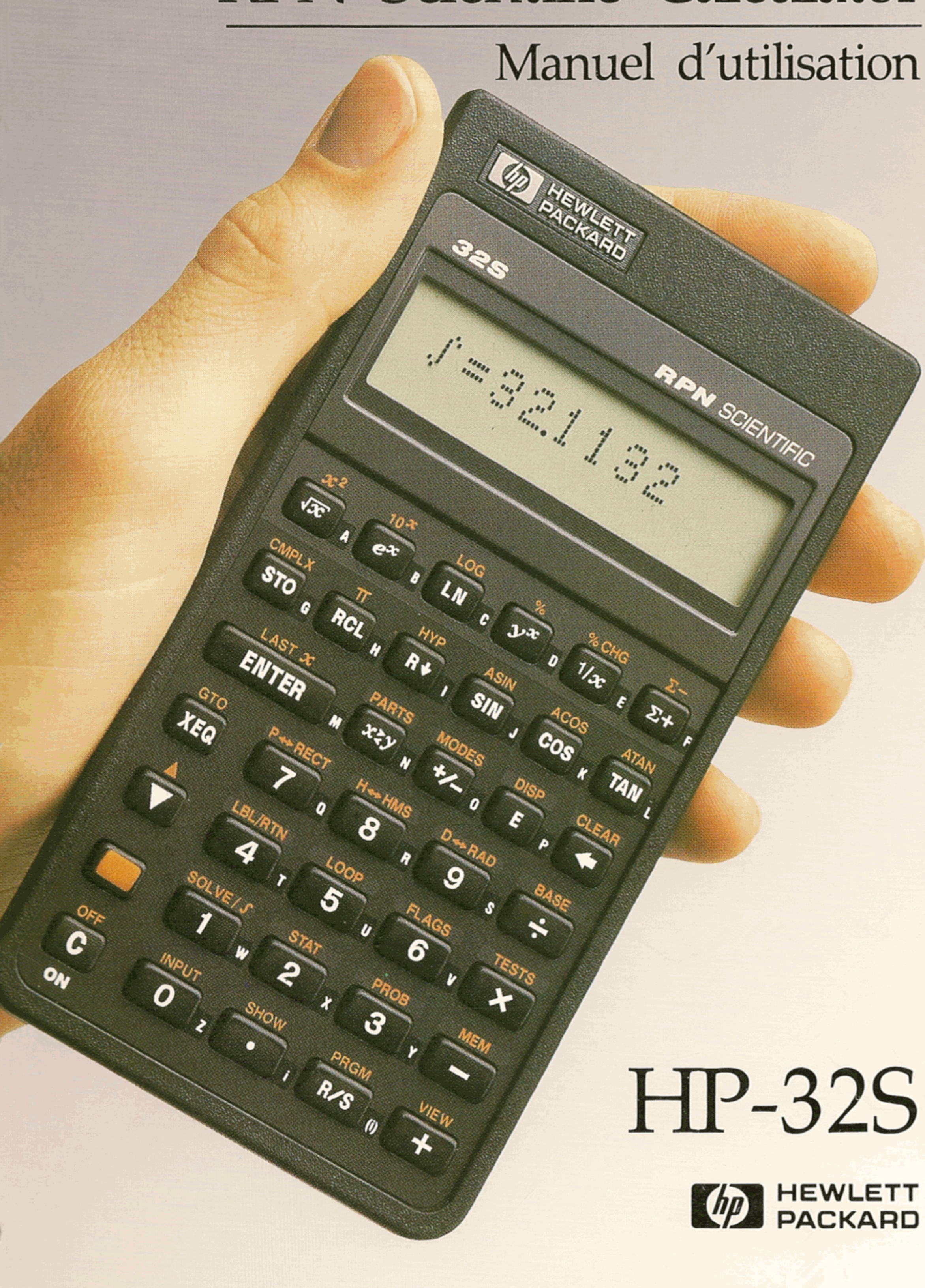
HEWLETT-PACKARD

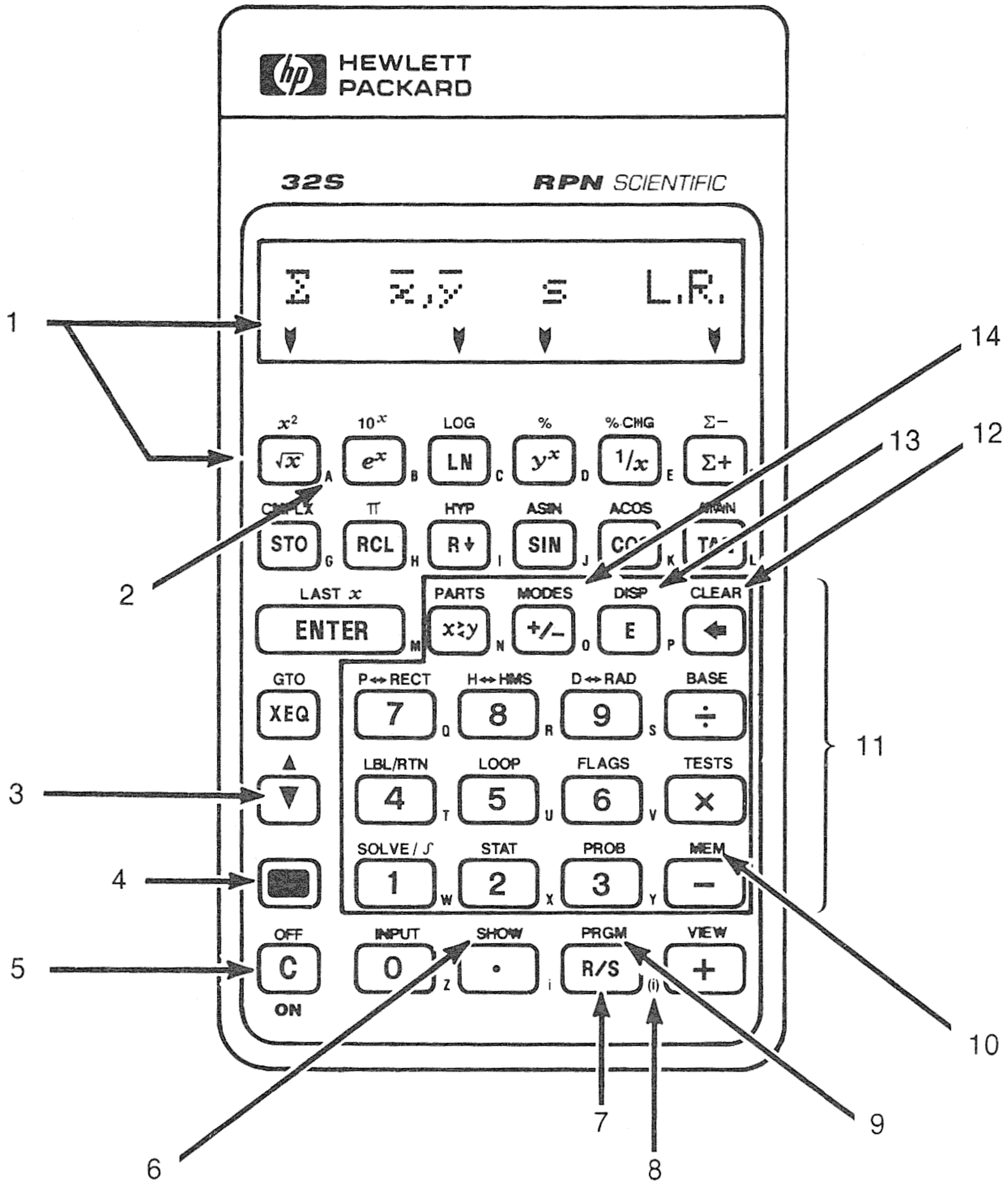
RPN Scientific Calculator

Manuel d'utilisation



HP-32S

HEWLETT PACKARD



1. Menu et touches de menu
2. Touches alphabétiques pour variables et labels
3. Permet de faire défiler les lignes d'un programme ou d'une liste
4. Touche préfixe (« *shift* »)
5. Mise sous tension ; sert aussi à annuler l'affichage, un menu, une saisie de programme
6. Montre toutes les décimales
7. Touche de mise en marche/d'arrêt pour l'exécution des programmes
8. Pour l'adressage indirect via *i*
9. Fait entrer et sortir du mode de saisie de programme
10. Mémoire utilisateur ; stockage de variables et de programmes
11. Touches de menu (encadrées)
12. Efface toutes les zones de la mémoire
13. Formats d'affichage
14. Modes angulaires, choix de la virgule comme séparateur décimal

Un petit effort !

En prenant le temps de remplir cette carte réponse, vous nous aidez à mieux comprendre vos besoins. Lisez d'abord les questions avant de répondre. Merci d'avance.

Modèle numéro (Model number) Date d'achat _____

Nom _____

Adresse _____

Code postal, Ville _____

Age _____ Tél. (_____) _____ Prof. ___ ou privé ___

1. Quel POSTE occupez-vous ? **(Ne cochez qu'une seule case)**

101 ☐ Etudiant
102 ☐ Educateur, chercheur
103 ☐ Cadre
104 ☐ Cadre moyen
105 ☐ Cadre sup.
106 ☐ P.d.g.
107 ☐ Représentant
108 ☐ Technicien
109 ☐ Indépendant, prof. libérale
110 ☐ Retraité
111 ☐ Autre _____

2. Quel est votre DOMAINE D'ACTIVITE ? **(Ne cochez qu'une seule case)**

201 ☐ Mécanique
202 ☐ Génie civil
203 ☐ Electricité/électronique
204 ☐ Chimie
205 ☐ Autre ingénierie ____
206 ☐ Topographie
207 ☐ Informatique
208 ☐ Contrôle de qualité
209 ☐ Achats, planning, gestion des stocks
210 ☐ Comptabilité, audit
211 ☐ Finance, analyse d'investissements
212 ☐ Services administratifs/direction générale
213 ☐ Marketing
214 ☐ Ventes
215 ☐ Services après-vente, maintenance
216 ☐ Autre _____

3. Dans quel SECTEUR travaillez-vous ? **(Ne cochez qu'une seule case)**

301 ☐ Education
302 ☐ Banque, finance
303 ☐ Assurance
304 ☐ Immobilier
305 ☐ Affaires/conseils commerciaux
306 ☐ Conseils techniques
307 ☐ Logiciel, services informatiques
308 ☐ Construction, architecture
309 ☐ Mines, extraction pétrolière, prospection
310 ☐ Chimie, raffinage
311 ☐ Agriculture, élevage, expl. forestière
312 ☐ Industrie et distribution agro-alimentaire
313 ☐ Fabrication de biens industriels
314 ☐ Fabrication de biens de consommation
315 ☐ Transports
316 ☐ Communication, distr. él., gaz, tél.
317 ☐ Services publics, armée
318 ☐ Autre _____

4. Où avez-vous acheté votre calculateur HP ? **(Ne cochez qu'une seule case)**

401 ☐ Magasin de produits informatique
402 ☐ Magasin de fournitures de bureau
403 ☐ Librairie
404 ☐ Grand magasin
406 ☐ Sur catalogue
407 ☐ Vente par correspondance
408 ☐ Papeterie
409 ☐ Acheté par entreprise/école
410 ☐ Directement chez HP
411 ☐ Autre _____

5. Où avez-vous entendu parler de ce modèle la première fois ?

501 ☐ Utilisateur de calculateur HP
502 ☐ Amis, collègues, professeur
503 ☐ Publicité écrite
504 ☐ Articles de presse
505 ☐ Publi-postage
506 ☐ Vendeur
507 ☐ Brochure dans un magasin
508 ☐ Autre _____

Prière
d'affranchir

Hewlett-Packard France
Département Calculateurs de poche
28, rue Jacques Ibert
B.P. 5820
F-75858 Paris Cedex 17
France

Commentaires concernant le manuel d'utilisation

Nous apprécions votre évaluation de ce manuel. Vos commentaires et suggestions nous aident à améliorer nos publications.

Manuel d'utilisation du HP-32S

Date d'impression du manuel (indiquée sur la page titre) _____

Veuillez cercler une réponse pour chacune des affirmations ci-dessous (valeur faible pour indiquer votre désaccord, et valeur élevée lorsque vous êtes d'accord avec l'affirmation en regard). Vous pouvez utiliser l'espace intitulé **Commentaires** pour toute information supplémentaire.

- Le manuel est bien organisé. 1 2 3 4 5
- Les informations sont faciles à trouver. 1 2 3 4 5
- Les informations sont correctes. 1 2 3 4 5
- Les instructions sont compréhensibles. 1 2 3 4 5
- Le manuel contient suffisamment d'exemples. 1 2 3 4 5
- Les exemples sont utiles et bien choisis. 1 2 3 4 5
- La présentation et le format sont attractifs et pratiques. 1 2 3 4 5
- Les illustrations sont claires et utiles. 1 2 3 4 5
- Le manuel est : trop long suffisant trop court.
- Les chapitres et annexes consultés le plus fréquemment sont :

  1 2 3 4 5 6 7 8 9 10 11 12 13 14 A B C D

Commentaires : _____

_____

_____

_____

_____

_____

Nom : _____

Adresse : _____

Ville/Code postal : _____

Profession : _____

Prière
d'affranchir

Hewlett-Packard France
Département Calculateurs de poche
28, rue Jacques Ibert
B.P. 5820
F-75858 Paris Cedex 17
France

HP-32S
RPN Scientific Calculator

Manuel d'utilisation

HEWLETT PACKARD

Edition 1 Mai 1988
N° de référence 00032-90043

Avertissement

1. Les informations contenues dans ce document peuvent faire l'objet de modifications sans préavis.
2. En raison de la complexité des techniques informatiques, ce document est remis au lecteur dans le seul but de faciliter sa compréhension du produit dont il traite. HPF décline en conséquence toute responsabilité pour tout dommage pouvant résulter des informations contenues dans ce document.
3. HPF ne garantit ni la fiabilité ni les conséquences de l'utilisation de ses produits logiciels lorsqu'ils sont utilisés sur des produits dont il n'a pas assuré la fourniture.
4. Les informations contenues dans ce document sont originales. Elles ont été conçues et mises au point par Hewlett-Packard. L'acheteur s'interdit en conséquence, sauf accord préalable et écrit de HPF :
   — de les divulguer ou d'en faciliter la divulgation ;
   — de les copier ou de les reproduire en tout ou en partie par n'importe quel moyen et sous n'importe quelle forme ;
   — de les traduire dans toute autre langue.

© Hewlett-Packard Co. 1987. Tous droits réservés.
Pour des informations plus spécifiques sur la garantie qui couvre ce calculateur, consultez les pages 291 et 295.

Corvallis Division
1000 N.E. Circle Blvd.
Corvallis, OR 97330, U.S.A.

Historique de l'impression

Edition 1 Mai 1988 Mfg. No. 00032-90044

Bienvenue au HP-32S

Votre HP–32S est l'aboutissement de quarante années de recherches et de perfectionnements ; il a été conçu et fabriqué avec soin par Hewlett-Packard. Vous pouvez attendre de nous des accessoires, un service après-vente de niveau mondial et une équipe de spécialistes expérimentés pour vous aider à utiliser le HP–32S au mieux de ses possibilités (voyez à ce sujet la page intérieure de couverture arrière).

La qualité Hewlett-Packard

Nos calculateurs ont été conçus pour être performants, durables et aisés à utiliser.

- Le HP–32S a été testé pour sa résistance aux chocs, aux vibrations, aux substances polluantes (smog, ozone), aux variations extrêmes de la température et de l'humidité.
- Ce calculateur et son manuel ont été conçus et testés pour être faciles à utiliser. Nous avons choisi une reliure à spirale et avons ajouté de nombreux exemples.
- Le clavier est construit de matériaux durables et le lettrage des touches a été fait par moulage ; le clavier est agréable au toucher et les touches confirment la frappe par un léger déclic.
- Les circuits CMOS et l'écran à cristaux liquides permettent de conserver les données, même lorsque le calculateur est éteint, et prolongent en outre la vie des piles.
- Le microprocesseur a été perfectionné pour permettre des calculs rapides et précis ; les calculs utilisent une précision interne de 15 chiffres.
- Les recherches ont permis de minimiser les effets de l'électricité statique (source d'éventuelles pertes de données pour les calculateurs de poche).

Caractéristiques du calculateur

Ce calculateur reflète largement les désirs exprimés par beaucoup d'utilisateurs de calculateurs. Voici ses caractéristiques principales :

- Toutes les fonctions sont disponibles, soit directement sur le clavier, soit dans des menus ; il n'est pas nécessaire de frapper leur nom au clavier.
- Des messages et lignes de programme remplacent les habituels messages d'erreur numériques ; par exemple, `DIVIDE BY 0` (divisez par) remplace `ERR 21`. Le stockage de données est possible dans les variables de *A* à *Z*.
- La logique RPN (*Notation Polonaise Inverse*), qui économise les frappes au clavier et facilite les calculs en chaîne.
- 390 octets de mémoire pour stocker données et programmes.
- Des fonctions de statistiques, de conversions de bases, d'arithmétique complexe, d'intégration et de résolution d'équations à une inconnue.
- Les possiblités étendues de programmation permettent la correction, les labels d'entrées/sorties, les sous-programmes, les boucles de programmes, les instructions conditionnelles, les indicateurs binaires et l'adressage indirect.

Table des matières

Première partie : fonctionnement de base

1 **14** **Introduction**
14 Avant de commencer
14 Mise sous-, et hors tension du calculateur
14 Réglage du contraste de l'affichage
15 Présentation du clavier et de l'affichage
15 Utilisation de la touche préfixe
15 Les touches alphabétiques
15 Espace arrière et effacement
16 Utilisation des menus
19 Sortie des menus
20 Témoins
21 Saisie de nombres
21 Changer le signe des nombres
22 Exposants de dix
23 Saisie de nombres
24 Capacité de la machine
24 Arithmétique
24 Fonctions monadiques
25 Fonctions diadiques
26 Calculs en chaîne
29 Exercices
29 Contrôle du format d'affichage
29 Choix de la virgule comme séparateur décimal
30 Choix du nombre de décimales après la virgule (■ DISP)
31 Comment visualiser la pleine précision de 12 chiffres
32 Messages
33 La mémoire du calculateur
33 Combien reste-t-il de mémoire ?
34 Effacement de la mémoire

2

35 **La pile automatique**
35 Qu'est-ce que la « pile » ?
36 Comment prendre connaissance de la pile (R↓)
37 Echange des registres X et Y dans la pile (x≷y)
38 Arithmétique et pile automatique
39 La touche ENTER
40 La fonction CLEAR
41 Le registre LAST X
42 Correction d'erreurs avec ■ LASTx
43 Récupération de nombres avec ■ LASTx
44 Calculs en chaîne
45 L'ordre des calculs
46 Exercices

3

47 **Comment stocker des données dans des variables**
48 Stockage et rappel de nombres
49 Examen de variables dans le catalogue VAR
50 Effacer des variables
50 Arithmétique sur des variables stockées
50 Arithmétique de stockage
51 Arithmétique de rappel
53 La variable « *i* »

4

54 **Fonctions sur les nombres réels**
55 Fonctions exponentielles et logarithmiques
56 La fonction de puissance (y^x)
56 Trigonométrie
56 Saisie de π
56 Définition du mode angulaire
57 Fonctions trigonométriques
59 Fonctions hyperboliques
59 Fonctions de pourcentage (%, %CHG)
60 Fonctions de conversion
60 Conversions de coordonnées (P↔RECT)
63 Conversions fractionnaires (H↔HMS)
64 Conversions angulaires (D↔RAD)
65 Fonctions de probabilités
67 Parties de nombres
67 Noms des fonctions

Deuxième partie : programmation

5 **Programmation élémentaire** — **70**

Création d'un programme — **71**
- Limites d'un programme (LBL et RTN) — **71**
- Saisie d'un programme (PRGM) — **72**

Exécution d'un programme — **75**
- Exécution d'un programme avec XEQ — **75**
- Test d'un programme — **76**

Entrées et sorties de données — **77**
- Saisies de données dans des variables (INPUT) — **77**
- Affichage de données dans des variables (VIEW) — **79**

Arrêt ou interruption d'un programme — **82**
- Programmation d'un arrêt ou d'une pause (STOP, PSE) — **82**
- Interruption d'un programme en cours d'exécution — **82**
- Arrêts pour erreur — **82**

Correction d'un programme — **83**

Mémoire de programme — **84**
- Visualisation de la mémoire de programme — **84**
- Utilisation de la mémoire — **84**
- Le catalogue de programmes (MEM) — **85**
- Effacement d'un ou de plusieurs programmes — **85**
- Le total de contrôle — **86**

Fonctions non programmables — **87**

Expressions polynomiales et méthode de Horner — **87**

6 **Techniques de programmation** — **90**

Sous-programmes à l'intérieur de programmes — **90**
- Appel de sous-routines (XEQ, RTN) — **91**
- Sous-routines emboîtées — **92**

Branchements (GTO) — **93**

Instructions conditionnelles — **95**
- Tests de comparaison (TESTS) — **96**
- Indicateurs — **97**

Boucles (GTO, LOOP) — **99**
- Boucles conditionnelles (GTO) — **100**
- Boucles avec compteurs (DSE, ISG) — **101**

Adressage indirect de variables et de labels — **103**
- La variable « *i* » — **103**
- L'adresse indirecte (*i*) — **104**
- Contrôle de programme avec (*i*) — **105**

Troisième partie : fonctionnement de haut niveau

7 **110** **Résolution d'une équation à une inconnue**
111 Utilisation de SOLVE
112 Création de programmes pour la fonction SOLVE
113 Exemples d'utilisation de SOLVE
118 Comprendre et contrôler la fonction SOLVE
119 Vérification du résultat
119 Interruption du calcul effectué avec SOLVE
120 Choix d'estimations de départ pour SOLVE
124 Utilisation de SOLVE dans un programme
125 Pour plus d'informations

8 **126** **Intégration numérique**
127 Comment utiliser l'intégration (∫FN)
128 Création de programmes pour ∫FN
128 Exemples d'utilisation de ∫FN
131 Précision de l'intégration
132 Choix de la précision
132 Interprétation de la précision
134 Utilisation de l'intégration dans un programme
136 Pour plus d'informations

9 **137** **Opérations sur les nombres complexes**
138 La pile complexe
139 Opérations complexes
142 Utilisation de nombres en notation polaire

10 **144** **Conversions de bases et arithmétique**
146 Arithmétique en bases 2, 8 et 16
147 La représentation de nombres
148 Nombres négatifs
149 Plage de nombres
149 Fenêtres pour les nombres binaires longs
150 Visualisation de nombres partiellement cachés
151 Programmation avec BASE
151 Choix d'un mode de base dans un programme
151 Nombres saisis dans des lignes de programmes

11 **153** **Statistiques**
153 Saisie de données statistiques (Σ+, ■Σ−)
154 Saisie de données à une seule variable
154 Saisie de données à deux variables
155 Correction des erreurs de saisie
156 Calculs statistiques
156 Moyenne et écart-type
158 Régression linéaire
160 Limitations de la précision des données
161 Valeurs de sommation et registres statistiques
161 Statistiques de sommation
162 Registres statistiques dans la mémoire du calculateur

Quatrième partie : Programmes d'application

12 **164** **Programmes mathématiques**
164 Opérations sur les vecteurs
175 Solutions d'équations simultanées—Méthode du déterminant
183 Solutions d'équations simultanées—Méthode d'inversion des matrices
191 Equation du second degré
198 Transformations de coordonnées

13 **204** **Programmes statistiques**
204 Ajustement de courbes
215 Distributions normales et inverses-normales

14 **222** **Programmes divers**
222 Calculs financiers sur des flux constants
229 Conversions d'unités
235 Générateur de nombres premiers

Cinquième partie : annexes et matériel de référence

A **240** **Assistance, piles et service après-vente**

240 Comment obtenir de l'aide et des conseils d'utilisation
240 Réponses à quelques questions fréquemment posées
242 Puissance et piles
242 Indication de la faiblesse des piles
243 Comment installer les piles
245 Environnement
245 Comment déterminer si le calculateur doit être réparé
246 Confirmation du bon fonctionnement du calculateur—le test automatique
248 Garantie
249 Que faire des piles usagées ?
249 Si le calculateur doit être réparé
250 Où faire réparer le calculateur
250 Coût du service après-vente
251 Instructions d'envoi
251 Garantie couvrant les réparations
251 Contrats de service après-vente

B **253** **Mémoire utilisateur et pile des résultats**

253 Gestion de la mémoire du calculateur
254 Remise à zéro du calculateur
255 Effacement de la mémoire
256 Situation de la montée de la pile
257 Opérations inactivant la montée de la pile
257 Opérations sans action sur la montée de la pile
258 Situation du registre LAST X

C **259** **Résolution d'équations : notions approfondies**

259 Comment SOLVE trouve une racine
261 Interprétation des résultats
267 Lorsque SOLVE ne peut trouver de racine
272 Erreur d'arrondi et dépassement négatif de capacité

D

273 **Notions complémentaires sur l'intégration**
273 Comment une intégrale est-elle évaluée ?
274 Situations qui pourraient provoquer des résultats incorrects
279 Situations qui prolongent le temps de calcul

281 **Messages**

286 **Index des fonctions**

299 **Index**

Première partie

Fonctionnement de base

Page 14 1 : Introduction

35 2 : Pile automatique

47 3 : Comment stocker des données dans des variables

54 4 : Fonctions sur les nombres réels

1

Introduction

Avant de commencer

Mise sous-, et hors tension du calculateur

Pour allumer le calculateur, appuyez sur la touche [C]. Notez le vocable ON, imprimé sous la touche.

Pour éteindre le calculateur, appuyez sur ■[OFF]. Pour ce faire, appuyez sur la touche préfixe (*Shift*), (■), et relâchez-la, puis appuyez sur la touche [C] (voyez le mot OFF imprimé au-dessus d'elle). Souvenez-vous que le calculateur possède une *mémoire permanente* ; la mise hors tension n'affecte aucunement les données qu'il contient.

Pour conserver l'énergie, le calculateur s'éteint après environ 10 minutes d'inactivité.

Dans la plupart des cas les piles ont une durée de vie de plus d'un an. Si vous remarquez sur l'écran la présence de l'indicateur de faiblesse de charge des piles (▭), remplacez les piles dès que possible. Consultez l'annexe A pour les détails et les instructions nécessaires.

Réglage du contraste de l'affichage

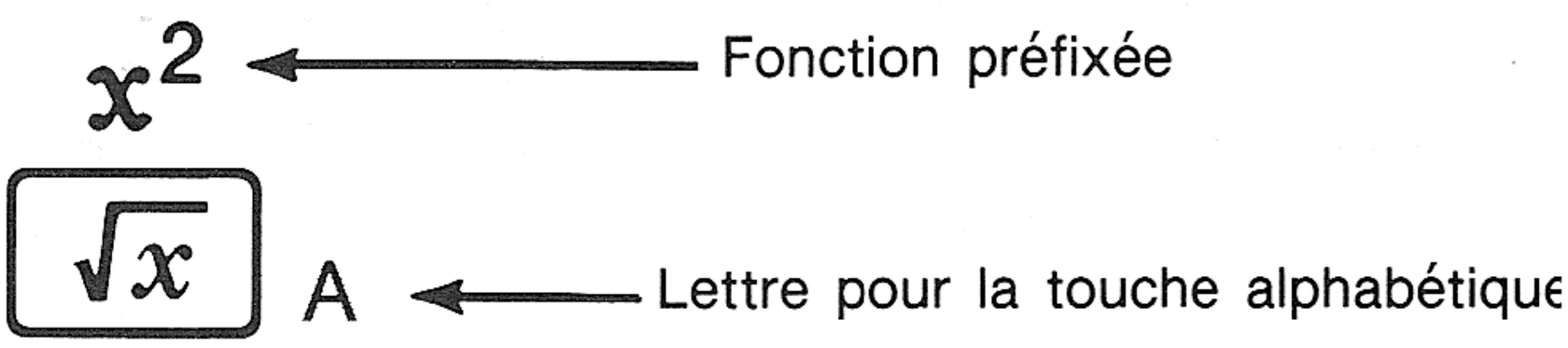
L'intensité du contraste dépend de l'éclairage, de l'angle sous lequel vous regardez l'écran et du réglage du contraste. Pour éclaircir ou assombrir l'affichage, maintenez l'appui sur la touche [C] et appuyez sur [+] ou sur [−].

Présentation du clavier et de l'affichage

Utilisation de la touche préfixe

Chaque touche supporte deux fonctions : l'une est moulée sur elle, l'autre, la fonction préfixée, est imprimée en couleur au-dessus de la touche. Pour obtenir l'exécution de ces fonctions, il faut au préalable appuyer sur la touche préfixe colorée (■). Par exemple, pour éteindre le calculateur, appuyez sur ■ et relâchez la pression, puis appuyez sur [C]. Cette séquence de frappe est notée dans ce manuel comme ceci : ■ [OFF].

L'appui sur la touche ■ provoque l'apparition du témoin (↗), qui reste affiché jusqu'à ce que vous appuyez sur la touche suivante. Pour annuler ↗, appuyez à nouveau sur ■.



Les touches alphabétiques

A côté de la plupart des touches se trouve une lettre, comme indiqué ci-dessus. Chaque fois que vous avez besoin de frapper une lettre–utilisée pour identifier une variable ou un libellé—le témoin **A..Z** apparaît à l'écran pour indiquer que les touches alphabétiques sont « actives » (vous trouverez des informations sur les variables au chapitre 3).

Espace arrière et effacement

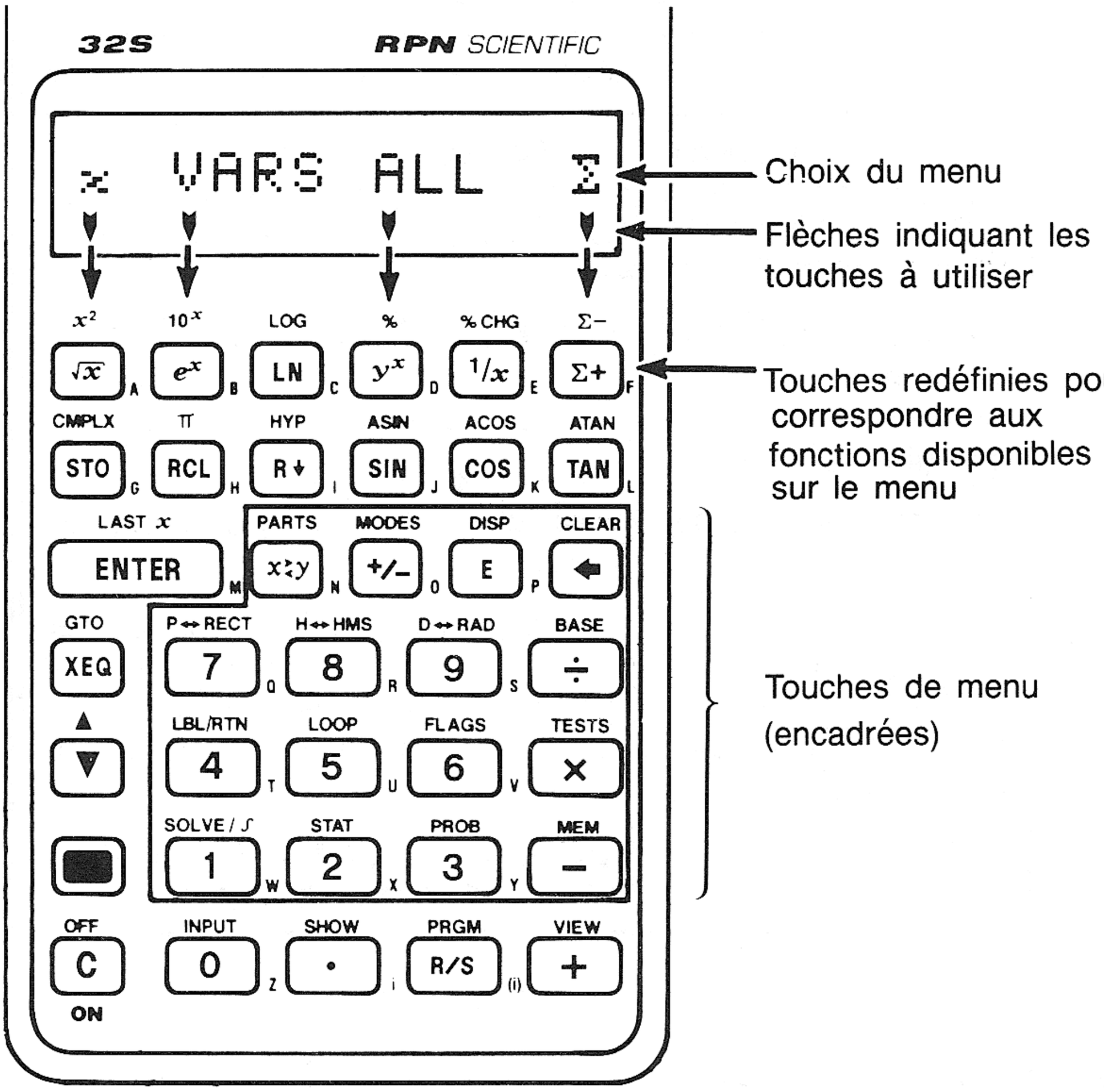
L'une des premières choses que vous devez savoir est comment *effacer ;* comment corriger une frappe, effacer l'affichage et, de manière générale, recommencer à zéro.

Touches d'effacement

Touche	Description
◀	*Espace arrière.* Efface le dernier caractère avant le curseur (_) ou sort du menu en cours. Dans le cas d'un nombre complet (pas de curseur), la touche ◀ efface le nombre. Cette touche efface aussi les messages d'erreur. Pendant la saisie d'un programme, cette touche efface la ligne de programme.
C	*Clear* (effacer) ou *Cancel* (annuler). *Efface* le nombre affiché ou *annule* la situation en cours (ce peut être un menu, un message, un message de sollicitation de saisie, un catalogue ou une entrée de programme).
■ CLEAR	*Le menu CLEAR.* Il vous donne plusieurs options pour effacer des données : {x}, {VARS}, {ALL} et {Σ}. Elles effacent respectivement : le nombre en cours (« *x* »), toutes les variables, toute la mémoire et les données statistiques. Durant la saisie d'un programme, le menu comporte aussi {PGM}, qui efface toute la mémoire de programme.

Utilisation des menus

La puissance du HP-32S ne se limite pas à ce que vous lisez sur le clavier. La moitié des touches « préfixées » sont des *touches de menu,* qui vous offrent d'autres fonctions—ou d'autres options pour d'autres fonctions. Cette « réserve de puissance » est d'accès plus facile que si chaque fonction possédait sa propre touche sur le clavier.



Ces fonctions préfixées sont imprimées sur un fond plus clair sur le calculateur (comme CLEAR) sont des *touches de menu*). L'appui sur ces touches affiche un *menu* sur l'écran—une série de choix.

Les menus du HP-32S

Menu	Description	Au chapitre :
	Fonctions numériques	
PARTS	Fonctions opérant sur des nombres (partie entière, valeur absolue etc.).	4
P↔RECT	Conversions entre coordonnées polaires et rectangulaires.	4
H↔HMS	Conversions entre heure décimale et heure-minutes-secondes.	4
D↔RAD	Conversions entre degrés et radians.	4
BASE	Conversions de bases.	10
SOLVE/∫	Fonctions pour la résolution de racines et l'intégration.	7, 8
STAT	Fonctions statistiques.	11
PROB	Fonctions de probabilités.	4
	Instructions de programmation	
LBL/RTN	Label, « return » (fin) et pause.	5
LOOP	Branchement conditionnel et fonctions de comptage.	6
FLAGS	Fonctions pour armer (*set*), désarmer (*clear*) et tester les indicateurs.	6
TESTS	Tests conditionnels.	6
	Autres fonctions	
MODES	Modes angulaires et séparateur décimal.	4, 1
DISP	Formats d'affichage.	1
CLEAR	Fonctions d'effacement.	1, 3, 5
MEM	Situation de la mémoire : mémoire utilisée par les différents programmes et variables. Catalogues des variables et des programmes.	1

Cherchons, par exemple, la factorielle de 25 :

Touches :	**Affichage :**	**Description :**
25	25_	Affiche le nombre.
■ PROB	Cn,r Pn,r x! R	Affiche le menu de probabilités.
{x!} (la touche 1/x)	1,5511E25	25! est $1\,5511 \times 10^{25}$.

De cette manière, les menus vous aident à exécuter des douzaines de fonctions en vous guidant par des choix successifs. Nul besoin de mémoriser les nombreuses fonctions disponibles sur le HP-32S ou de rechercher un nom parmi d'autres noms imprimés sur le clavier.

Sortie des menus

Lorsque vous exécutez une fonction sur le menu, celui-ci disparaît immédiatement, comme dans l'exemple ci-dessus. Si vous désirez quitter un menu *sans* exécuter de fonction, vous avez trois options :

- Appuyer sur ◄ **sort du menu**, une étape à la fois.

123	123_
■ PROB	Cn,r Pn,r x! R
{R}	RANDOM SEED
◄	Cn,r Pn,r x! R
◄	123,0000

- Le fait d'appuyer sur C **annule le menu.**

123	123_
■ PROB	Cn,r Pn,r x! R
{R}	RANDOM SEED
C	123,0000

■ **Appuyer sur une autre touche de menu** remplace l'ancien menu par le nouveau.

```
123                          123_

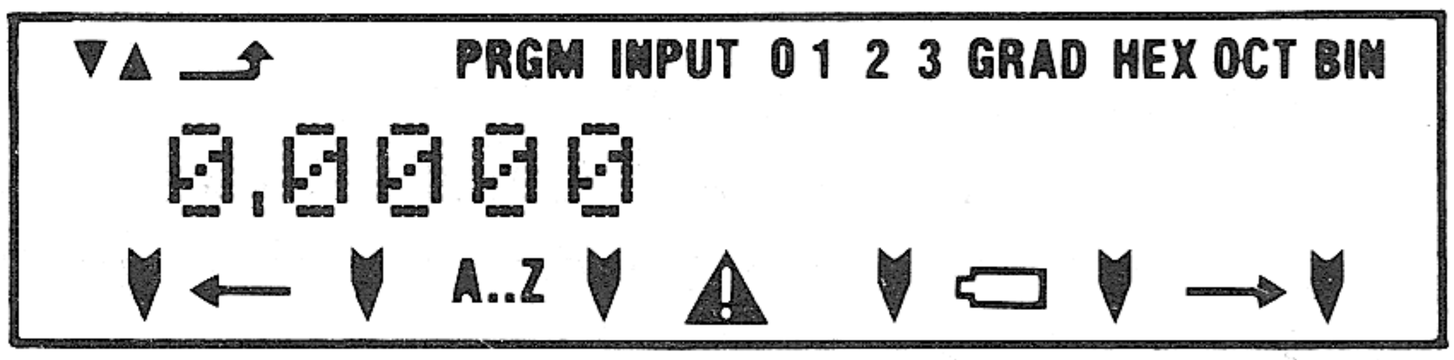
■ PROB                       Cn,r  Pn,r  x!  R

{R}                          RANDOM  SEED

■ CLEAR                      x  VARS  ALL  Σ
```

Témoins

Les symboles montrés ici sont nommés *témoins*. Chacun d'eux a une signification particulière lorsqu'il apparaît sur l'écran.



Témoin	Signification
▼▲	▼ et ■▲ sont actifs et permettent de passer de ligne en ligne dans un programme ou dans une liste (pp. 33, 76).
↗	La touche préfixe (■) est active (page 15).
PRGM	La saisie de programme est active (pp 72, 75).
INPUT	Le programme attend des saisies ; saisissez le nombre et appuyez sur R/S pour reprendre le programme (page 77).
0 1 2 3	Spécifie les indicateurs qui sont armés (page 98).
RAD GRAD	Le mode Radians ou le mode grades est actif (page 57).
HEX OCT BIN	Spécifie la base de nombres active (page 144).
▼	Les touches de la première ligne sont redéfinies en fonction des libellés du menu affiché (page 17).

Témoin	Signification
← →	Il y a d'autres chiffres à gauche où à droite. Utilisez ■ SHOW pour voir le reste d'un nombre décimal ; utilisez les touches de défilement gauche et droit ($\sqrt{x}$, $\Sigma+$) pour voir le reste d'un nombre binaire (p. 150).
A..Z	Les touches alphabétiques sont actives (p. 48).
⚠	Attention ! indique une condition spéciale ou une erreur (pp. 21, 32).
▭	Les piles sont faibles (p. 242).

Saisie de nombres

Vous pouvez frapper un nombre comptant un maximum de 12 chiffres plus un exposant comptant jusqu'à 3 chiffres (maximum $\pm$499). Si vous essayez de saisir un nombre plus important, la saisie s'arrête et le témoin ⚠ apparaît.

Si vous faites une erreur en saisissant un nombre, appuyez sur ⬅ pour revenir en arrière et effacer le dernier chiffre ou appuyez sur la touche C pour effacer tout le nombre.

Changer le signe des nombres

La touche +/– change le signe d'un nombre.

- Pour saisir un nombre négatif, frappez ce nombre puis appuyez sur +/– .
- Pour changer le signe d'un nombre déjà saisi, appuyez simplement sur +/– (si le nombre a un exposant, +/– n'affecte que la mantisse—l'exposant n'est pas modifié).

Exposants de dix

Exposants sur l'affichage. Les nombres avec exposants de dix (comme $4\,2\times10^{-5}$) sont affichés sur l'écran avec la lettre `E` précédant l'exposant (comme dans `4,2000E-5`). Un nombre trop grand ou trop petit pour le format d'affichage sera automatiquement affiché sous forme exponentielle. Par exemple, en format FIX 4 (quatre décimales après la virgule), voyez les effets des frappes suivantes :

Touches :	**Affichage :**	**Description :**
,000062	`0,000062_`	Montre les nombres saisis.
ENTER	`0,0001`	Arrondit le nombre pour accommoder le format d'affichage.
,000042 ENTER	`4,2000E-5`	Utilise la notation scientifique ; autrement, aucun chiffre significatif n'apparaîtrait.

Saisi des exposants de dix. Utilisez E (*exposant*) pour saisir des nombres multipliés par des puissances de dix. Par exemple, la constante de Planck, $6\,6262 \times 10^{-34}$.

1. Frappez la *mantisse* du nombre. Si elle est négative, appuyez sur +/−.

   6,6262 `6,6262_`

2. Appuyez sur E. Remarquez que le curseur vient se placer derrière le `E`.

   E `6,6262E_`

3. Frappez l'exposant (l'exposant le plus grand est ±499). S'il est négatif, appuyez sur +/−.

   34 +/− `6,6262E-34_`

Pour une puissance de 10 sans multiplicateur, comme 10^{34}, appuyez simplement sur E 34. Le calculateur affiche `1E34`.

Autres fonctions d'exposant. Pour *spécifier* un exposant de dix lors de la saisie, utilisez la touche [E]. Pour *calculer* un exposant de dix (cologarithme en base 10), utilisez [■] [10^x] (Cf. chapitre 4). Pour *calculer* le résuiltat de n'importe quel nombre élevé à une puissance (exponentiation), utilisez [y^x] (chapitre 4).

La saisie de chiffres

Lorsque vous saisissez un nombre, le curseur (_) apparaît sur l'écran. Il vous indique l'endroit où viendra s'inscrire le chiffre suivant ; il indique par là également que ce nombre n'est pas achevé, en termes techniques que la *saisie n'est pas terminée.*

Touches :	**Affichage :**	**Description :**
123	123_	La saisie de chiffres n'est pas terminée : le nombre n'est pas complet.

Si vous *exécutez une fonction* pour calculer un *résultat,* le curseur disparaît parce que le nombre est terminé. La saisie de chiffres est terminée.

[√x]	11,0905	Saisie de chiffres terminée.

Le fait d'appuyer sur [ENTER] met également fin à la saisie de chiffres. C'est pourquoi vous devez séparer deux nombres par [ENTER] : il vous faut terminer un nombre avant de pouvoir en saisir un second.

123 [ENTER]	123,0000	Un nombre achevé.
5 [+]	128,0000	Un autre nombre achevé.

Si la saisie de chiffres n'est pas terminée (si le curseur est présent), la [←] revient en arrière pour effacer le dernier chiffre. Si la saisie de chiffres est terminée (si le curseur a disparu), la touche [←] devient semblable à [C] et efface tout le nombre. Essayez !

Capacité de la machine

Le nombre le plus petit pouvant être saisi sur le calculateur est 1×10^{-499} ; le plus grand, $9\ 99999999999 \times 10^{499}$ (affiché sous forme arrondie `1,0000E500`).

- Si un calcul produit un résultat excédant le plus grand nombre possible, alors le nombre $9\ 99999999999 \times 10^{499}$ est affiché à la place, et le message `OVERFLOW` apparaît.
- Si un calcul produit un résultat plus petit que le nombre le plus petit possible, c'est un zéro qui est renvoyé à l'affichage et il n'y a pas de message.

Arithmétique

Lorsque vous appuyez sur une touche de fonction, le calculateur exécute immédiatement la fonction écrite sur la touche. C'est pourquoi toutes les opérandes doivent être présentes *avant* d'appuyer sur la touche de fonction.

Tous les calculs peuvent être réduits à des fonctions monadiques et diadiques.

Fonctions monadiques

Pour utiliser une fonction monadique (comme [1/x], [√x], [■][x²] et [+/−]) :

1. Saisissez le nombre (*il n'est pas nécessaire d'appuyer sur* [ENTER].)
2. Appuyez sur la touche de fonction (pour une fonction préfixée, appuyez sur la touche préfixe d'abord).

Calculez par exemple 1/32 et $\sqrt{148,84}$. Mettez le résultat au carré et changez son signe.

Touches :	**Affichage :**	**Description :**
32	`32_`	Opérande.
[1/x]	`0,0313`	Réciproque.
148,84 [√x]	`12,2000`	Racine carrée.

[■] [x^2]	148,8400	Carré de 12,2.
[+/−]	-148,8400	Négation de 148,84.

Les fonctions monadiques comportent aussi les fonctions trigonométriques, logarithmiques, hyperboliques et les fonctions opérant sur des parties de nombres, qui sont présentées au chapitre 4.

Fonctions diadiques

Pour utiliser une fonction diadique (telle que [+], [−], [×] et [÷]) :

1. Frappez le premier nombre.
2. Appuyez sur [ENTER] pour séparer le premier nombre du second.
3. Frappez le second nombre (n'appuyez *pas* sur [ENTER] cette fois-ci).
4. Appuyez sur la touche de fonction. Pour une fonction préfixée, appuyez d'abord sur la touche préfixée.

L'important est de saisir les deux nombres avant d'exécuter la fonction.

Par exemple :

Pour calculer :	**Appuyez sur :**	**A l'affichage :**
12 + 3	12 [ENTER] 3 [+]	15,0000
12 − 3	12 [ENTER] 3 [−]	9,0000
12 × 3	12 [ENTER] 3 [×]	36,0000
12 ÷ 3	12 [ENTER] 3 [÷]	4,0000

L'ordre de la saisie est, bien sûr, essentiel pour des fonctions non-commutatives telles que [−] et [÷]. Si les nombres ont été saisis dans le mauvais ordre, il est encore possible d'obtenir la bonne réponse sans re-frapper les nombres en appuyant sur [x≷y] pour faire *permuter* l'ordre des nombres. Exécutez alors la fonction projetée (ceci est expliqué en détail au chapitre 2, à la rubrique « Manipulations de la pile ».

Calculs en chaîne

C'est là que la vitesse et la simplicité de calcul du HP–32S sont le plus remarquables. Même durant les calculs les plus longs, *vous ne travaillez jamais qu'avec un ou deux nombres à la fois*—la pile automatique stocke les résultats intermédiaires jusqu'à ce que vous en ayez besoin, et les insère alors dans les calculs.*

- Cette méthode nécessite moins de frappes—moins de gestes—que la logique utilisée par les autres calculateurs et elle s'adapte naturellement à la programmation.
- Le processus de résolution d'un problème et le même que celui que vous suivez lorsque vous faites une opération sur le papier mais ici, c'est le calculateur qui fait le plus gros du travail...

Résolvons, par exemple, $(12 + 3) \times 7$.

Partons des parenthèses. Si vous essayez de résoudre ce problème sur le papier, vous calculerez probablement d'abord le résultat de $(12 + 3)$...

$$(12 + 3) \times 7 =$$

...puis vous multiplierez ce résultat par 7.

$$15 \times 7 = 105$$

Le problème est résolu de la même façon sur le HP–32S, en commençant par l'intérieur des parenthèses :

Touches :	**Affichage :**	**Description :**
12 [ENTER] 3 [+]	15,0000	Calculez le résultat intermédiaire d'abord.

Pas besoin d'appuyer sur [ENTER] pour sauvegrader ce résultat intermédiaire avant de vous en servir. C'est un résultat calculé, il est donc sauvegardé automatiquement.

* Ne vous inquiétez pas de comprendre le fonctionnement de la pile automatique de mémoire des résultats (et de la logique *RPN*). Elle sera expliquée au chapitre 2.

7 [×]	105,0000	L'appui sur la touche de fonction donne la réponse. Ce résultat peut être utilisé dans d'autres calculs.

Etudions ces exemples. Remarquez que vous n'appuyez sur [ENTER] que pour séparer les nombres introduits *séquentiellement,* comme au début d'un problème. Les opérations elles-mêmes ([+], [−] etc.) séparent les autres nombres et sauvegardent les résultats intermédiaires. Le dernier résultat qui a été sauvegardé sera le premier à être récupéré comme requis par le calcul.

Calculez d'abord $\frac{2 + 3}{10}$:

Touches :	**Affichage :**	**Description :**
2 [ENTER] 3 [+] 10 [÷]	0,5000	(2 + 3) ÷ 10.

Calculez maintenant $\frac{2}{3 + 10}$:

Touches :	**Affichage :**	**Description :**
3 [ENTER] 10 [+]	13,0000	Calcule (3 + 10) d'abord.
2 [x≷y] [÷]	0,1538	Place un 2 *avant* 13 pour que la division soit correcte : 2 ÷ 13.

Calcule $\frac{14 + 7 + 3 - 2}{4}$.

Touches :	**Affichage :**	**Description :**
14 [ENTER] 7 [+] 3 [+] 2 [−]	22,0000	Calcule (14 + 7 + 3 − 2) d'abord.
4 [÷]	5,5000	22 ÷ 4.

Calculez à présent $\frac{4}{14 + (7 \times 3) - 2}$.

7 [ENTER] 3 [×]	21,0000	Calcule (7 × 3).
14 [+] 2 [−]	33,0000	Traite les autres nombres.
4 [x≷y]	33,0000	Place le 4 *avant* 33 en prévision de la division.
[÷]	0,1212	Calcule la réponse.

Les problèmes comportant de multiples parenthèses sont résolus de la même manière, en utilisant le stockage automatique des résultats intermédiaires. Par exemple, pour résoudre (3 + 4) × (5 + 6) sur le papier, vous

effectuez d'abord l'opération
placée entre ces parenthèses...

$$(3 + 4) \times (5 + 6)$$

...et ensuite celle qui se trouve
entre celles-ci...

...et multipliez ensuite les deux résultats entre eux.

C'est cette manière de travailler que vous utilisez avec le HP-32S, sauf que vous n'avez pas à inscrire les résultats intermédiaires—le calculateur les a mémorisés pour vous.

Touches :	**Affichage :**	**Description :**
3 [ENTER] 4 [+]	7,0000	Ajoute d'abord (3 + 4).
5 [ENTER] 6 [+]	11,0000	Puis (5 + 6).
[×]	77,0000	Multiplie ensuite les résultats intermédiaires entre eux pour la réponse finale.

Les avantages. Cette méthode de saisie de nombres, nommée *notation polonaise inverse,* n'est pas ambigüe, et par là même ne nécessite pas de parenthèses.

- Vous ne travaillez jamais avec plus de deux nombres à la fois.
- Utilisez [ENTER] pour séparer deux nombres introduits à la suite.
- L'appui sur une touche de fonction exécute immédiatement la fonction.
- Les résultats intermédiaires apparaissent au fur et à mesure de leur calcul, vous pouvez donc vérifier chaque étape au fur et à mesure.
- Les résultats intermédiaires sont automatiquement stockés. Ils apparaissent dès qu'ils sont nécessaires au calcul—le dernier résultat stocké sera le premier à ressortir.
- La logique de calcul est la même que celle qui est suivie lorsque vous calculez à l'aide d'un crayon et d'une feuille de papier.

Exercices

Calculez : $\dfrac{\sqrt{(16{,}3805 \times 5)}}{0{,}05} = 18{,}000$

Solution : 16,3805 [ENTER] 5 [×] [$\sqrt{x}$] ,05 [÷]

Calculez : $\sqrt{[(2 + 3) \times (4 + 5)]} + \sqrt{[(6 + 7) \times (8 + 9)]} = 21{,}5743$

Solution : 2 [ENTER] 3 [+] 4 [ENTER] 5 [+] [×] [$\sqrt{x}$] 6 [ENTER] 7 [+] 8 [ENTER] 9 [+] [×] [$\sqrt{x}$] [+]

Calculez : $(10 - 5) \div [(17 - 12) \times 4] = 0{,}2500$

Solution : 17 [ENTER] 12 [−] 4 [×] 10 [ENTER] 5 [−] [$x \gtrless y$] [÷]

ou bien

10 [ENTER] 5 [−] 17 [ENTER] 12 [−] 4 [×] [÷]

Contrôle du format d'affichage

Choix de la virgule comme séparateur décimal

Pour séparer sur l'écran de votre HP–32S les unités des fractions décimales par une *virgule* plutôt que par un point :

1. Appuyez sur ■ [MODES] pour afficher le menu MODES.

2. Spécifiez la virgule comme séparateur décimal en appuyant sur la dernière touche, à droite du menu, qui se trouve en regard de {.}. Le nombre « un million » apparaît ainsi :

■ 1,000,000.0000.

Choix du nombre de décimales après la virgule (■ DISP)

Tous les nombres sont *stockés de façon interne* avec une précision de 12 chiffres,* mais vous pouvez choisir le nombre de décimales *affichées* en utilisant la fonction ■ DISP. Le nombre est arrondi en fonction du format choisi. Voici les quatre choix du menu DISP :

FX SC EN ALL

Format en virgule fixe ({FX}). Le format FIX affiche un nombre avec un maximum de 11 espaces décimaux. Après le message de sollicitation FIX _, spécifiez le nombre de décimales. Pour obtenir 10 ou 11 décimales, appuyez sur ⊡ 0 ou sur ⊡ 1.

Décimales

123.456,0000

Tout nombre trop grand ou trop petit pour être affiché sous le format que vous aurez choisi est automatiquement affiché en format scientifique.

Format scientifique ({SC}). Le format SCI affiche un nombre en notation scientifique (un seul chiffre avant la virgule et jusqu'à 11 chiffres après la virgule, et jusqu'à trois chiffres dans l'exposant). Après le message de sollicitation SCI _, qui apparaît lorsque vous avez choisi SC sur le menu, spécifiez le nombre de décimales, ⊡ 0 pour 10 ou ⊡ 1 pour 11.

Décimales — Puissance de 10

1,2346E5

Mantisse

* Pendant certains calculs internes complexes, le calculateur utilise une précision de 15 chiffres pour les résultats intermédaires.

Format ingénieur ({ENG}). Le format ENG (*engineering*) affiche un nombre de manière similiare à la notation scientifique, mais l'exposant est un multiple de trois et il peut être de un, deux ou trois chiffres avant la virgule. Ceci est particulièrement utile lors de calculs d'ingénierie utilisant des unités définies en multiples de 10^3 (telles que les unités micro-, milli- et kilo-).

Après le message de sollicitation ENG _, spécifiez le nombre de chiffres désiré *après* le premier chiffre significant (la partie entière sera toujours inférieure à 1 000.) Pour 10 ou 11 chiffres, appuyez sur [·] 0 ou sur [·] 1.

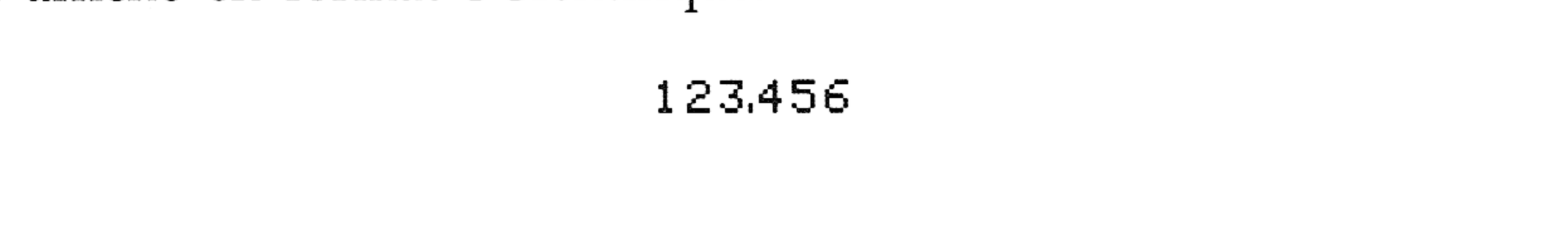


Format ALL ({ALL}). Le format ALL affiche un nombre de manière aussi précise que possible (avec un maximum de 12 chiffres). Si tous les chiffres ne peuvent prendre place sur l'affichage, le nombre est affiché en format SCIentifique.

123.456

Comment visualiser la pleine précision de 12 chiffres

Vous vous en souvenez, le nombre de décimales affichées n'affecte pas les calculs internes, qui utilisent la pleine précision de 12 chiffres.



Pour afficher temporairement un nombre en pleine précision, appuyez sur ■ [SHOW]. Ceci vous montre *la mantisse* du nombre tant que vous maintenez la pression sur la touche [SHOW].

Touches :	Affichage :	Description :
■ DISP {FX} 4		Demande l'affichage à quatre décimales.
45 ENTER 1,3 ×	58,5000	Quatre décimales sont affichées.
■ DISP {SC} 2	5,85E1	Format scientifique : deux décimales et un exposant.
■ DISP {ALL}	58,5	Tous les chiffres significatifs sont là, les zéros sont omis.
■ DISP {FX} 4	58,5000	Quatre décimales, pas d'exposant.
1/x	0,0171	
■ SHOW (hold)	170940170940	Montre temporairement la pleine précision.

Messages

Le calculateur répond à certaines conditions ou frappes en affichant un message. Le symbole ⚠ apparaît pour attirer l'attention.

- Pour effacer un message, appuyez sur C ou sur ◄.
- Pour effacer le message *et* effectuer une autre tâche, appuyez sur une autre touche.

Si aucun message n'apparaît, mais que le témoin ⚠ apparaît, vous avez probablement appuyé sur une touche *inactive*, c'est-à-dire une touche qui n'a aucune signification à ce moment, par exemple la touche 3 en mode binaire.

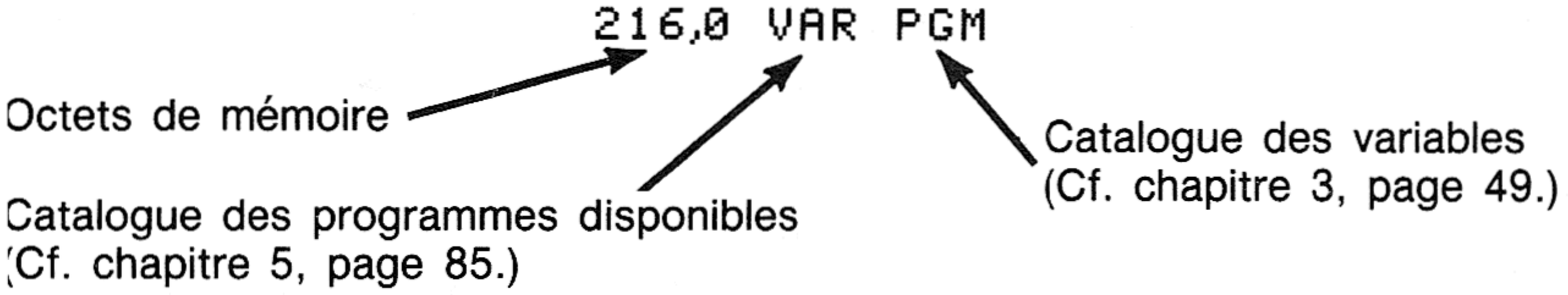
Tous les messages affichés sont expliqués dans la liste des messages, qui se trouve en page 281.

La mémoire du calculateur

Il y a 390 octets de mémoire utilisateur disponibles dans le HP-32S, qui vous sont accessibles pour n'importe quelle combinaison de données (variables ou lignes de programme). Les besoins en mémoire pour les diverses activités sont listées à la rubrique « Gestion de la mémoire », à l'annexe B.

Combien reste-t-il de mémoire ?

Le fait d'appuyer sur ■ MEM vous montre la quantité de mémoire restante.



1. Pour entrer dans le catalogue de variables, appuyez sur {PGM}.
2. Pour consulter le catalogue, appuyez sur ▼ ou sur ■ ▲.
3. Pour supprimer une variable ou un programme, appuyez sur ■ CLEAR en le visualisant dans le catalogue.
4. Pour sortir du catalogue, appuyez sur C.

Effacement de la mémoire

Il provoque l'effacement de tous les nombres, de toutes les lignes de programme stockés. Il n'a pas d'influence sur les réglages (choix de modes et de formats). Pour effacer réglages *et* données, consultez « Effacement de la mémoire », à l'annexe B.

Pour effacer toute la mémoire :

1. Appuyez sur ■ CLEAR {ALL}. Vous lirez alors le message de confirmation CLR ALL? Y N, qui prévient l'effacement involontaire de la mémoire.
2. Appuyez sur {Y} (*yes*).

2

La pile automatique

Ce chapitre explique comment les calculs se font dans la pile automatique de mémoire et pourquoi cette méthode minimise le nombre de frappes pour les calculs complexes. *Notez bien qu'il n'est pas nécessaire d'avoir lu et compris ces notions pour pouvoir se servir du calculateur.* Cependant, vous gagnerez une plus grande compréhension de son fonctionnement après la lecture de ce chapitre, particulièrement en ce qui concerne la programmation, et vous vous en servirez probablement avec plus de satisfaction.

Dans la deuxième partie du manuel, « Programmation », vous verrez que la disposition de la pile facilite beaucoup la manipulation des données pour les programmes.

Qu'est-ce que la « pile » ?

C'est le *stockage automatique des résultats intermédiaires* qui fait que le HP–32S facilite grandement le traitement des calculs complexes, et cela sans utilisation de parenthèses. La clef de ce stockage automatique est *la pile automatique de mémoire, en notation polonaise inverse.**

La pile de mémoire consiste en quatre emplacements de stockages, nommés *registres de stockage*, « empilés » les uns au-dessus de autres. Ils constituent une zone de travail pour les calculs. Ces registres—qui sont désignés par les lettres **X**, **Y**, **Z** et **T**—stockent et manipulent quatre nombre en cours. Le plus « ancien » de ces nombres est celui qui se trouve dans le registre **T** (*Angl. top, dessus*).

* La logique de fonctionnement qu'utilise Hewlett-Packard se fonde sur la *Notation polonaise* développée par le logicien polonais Jean Łukasiewicz (1878—1956). Alors que la notation algébrique traditionnelle place les opérateurs *entre* les nombres ou variables, la notation inventée par Łukasiewicz les place *avant* les nombres ou variables. D'où le terme notation polonaise *inverse*, en anglais *Reverse Polish Notation*, ou *RPN*.

T	0,0000	Nombre le plus « ancien »
Z	0,0000	
Y	0,0000	
X	0,0000	Nombre affiché

Le nombre le plus « récent » est placé dans le registre X : *c'est le nombre que vous voyez affiché sur l'écran.*

En programmation, la pile est utilisée pour effectuer les calculs, pour stocker temporairement les résultats intermédiaires, pour passer les valeurs stockées (variables) parmi les programmes et sous-programmes, pour accepter les saisies et pour fournir les résultats.

Le registre X est affiché. Le registre X est celui que vous lisez, *sauf* lorsqu'un menu, un message ou une ligne de programme occupent l'écran. Vous aurez sans doutre remarqué que plusieurs noms de fonctions comprennent un *x* ou un *y*. Ces lettres font référence aux *registres* X et Y. Par exemple, [■] [10ˣ] élève 10 à la puissance du nombre placé dans le registre des X, le nombre affiché à l'écran.

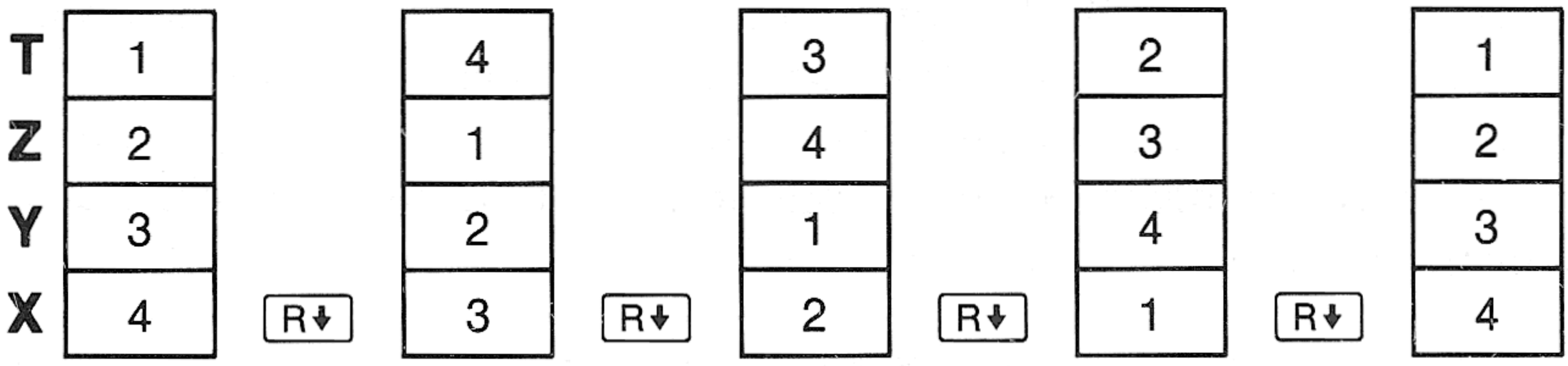
Deux méthodes pour annuler : [■] [CLEAR] {⇐} et [C]. Le fait d'appuyer sur [■] [CLEAR] {⇐} efface *toujours* le registre X, et il est utilisé pour programmer cette instruction. Par contre, la touche [C] est dépendante du contexte. Elle efface ou annule l'affichage en cours, selon la situation à ce moment-là : elle n'agit de la même façon que [■] [CLEAR] {⇐} que lorsque le registre X est affiché.* Elle *annule* les autres affichages : menus, nombres étiquetés d'un indicateur, messages et saisies de programme.

Comment prendre connaissance de la pile ([R↓])

La touche [R↓] (*roll down,* faire défiler vers le bas) permet de faire défiler tout le contenu de la pile, registre par registre. Vous « verrez passer » chaque nombre lorsqu'il entrera dans le registre.

* [◀] agit également comme [■] [CLEAR] {⇐} lorsque le registre X est affiché *et* que la saisie de chiffres est achevée (lorsqu'il n'y a plus de curseur sur l'écran).

Supposons que les nombres 1, 2, 3, 4 se trouvent sur la pile (appuyez sur 1 [ENTER] 2 [ENTER] 3 [ENTER] 4 pour les y mettre). Le fait d'appuyer sur la touche [R↓] 4 fois fait défiler tous les nombres et revient à la situation de départ :



Le contenu du registre X fait le tour et passe dans le registre T (*top*). C'est le *contenu* des registres qui se déplace ; les registres eux-mêmes conservent leur position, et le registre X est toujours celui qui est affiché.

Echange des registres X et Y dans la pile ([x≷y])

Une autre touche qui manipule la pile est [x≷y] (*permuter x et y*). Elle ne concerne que les registres X et Y, sans affecter le reste de la pile. Si bien sûr vous appuyez deux fois sur [x≷y], vous rétablirez l'ordre primitif.

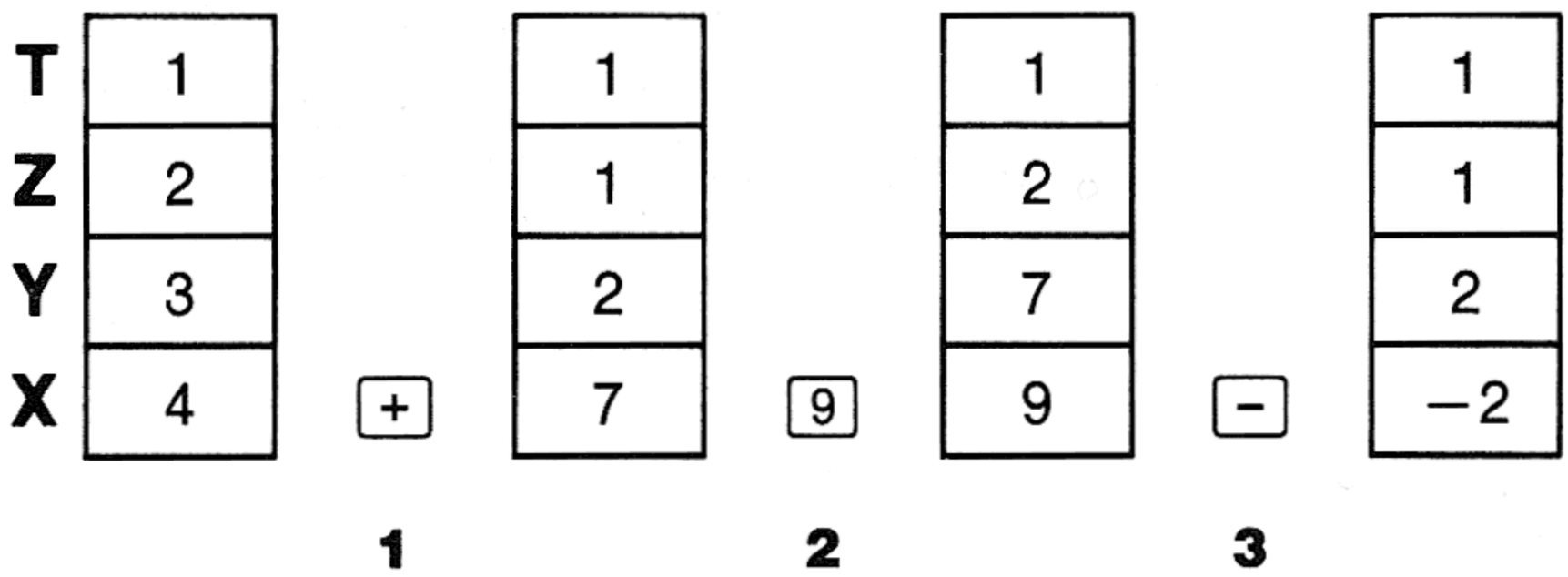
La fonction [x≷y] est utilisée pour deux raisons :

- Visualiser *y*, puis revenir au registre X (on appuie alors deux fois sur [x≷y]). Certaines fonctions donnent en effet deux résultats : l'un est donné dans le registre des X, l'autre dans le registre Y. La fonction {y,x→θ,r} convertit les coordonnées rectangulaires dans les registres X et Y en coordonnées polaires dans les registres X et Y.
- Echanger l'ordre des nombres dans un calcul. Par exemple, pour calculer 9 ÷ (13 × 8) , appuyez sur 13 [ENTER] 8 [×] 9 [x≷y] [÷].

Arithmétique et pile automatique

Le contenu de la pile monte et descend automatiquement au fur et à mesure de l'introduction de nouveaux nombres dans le registre X (*monter la pile*) et que des opérateurs combinent deux nombres pour en produire un nouveau (*descendre la pile*) dans le registre X. Supposons que la pile contienne les nombres 1, 2, 3 et 4. Examinons les montées et descentes de la pile qui se produisent lors du calcul suivant :

$$3 + 4 - 9:$$



1 La pile « descend » son contenu (et le registre du haut, T, reproduit son contenu).

2 La pile « remonte » son contenu (le contenu du registre du haut, T, est perdu).

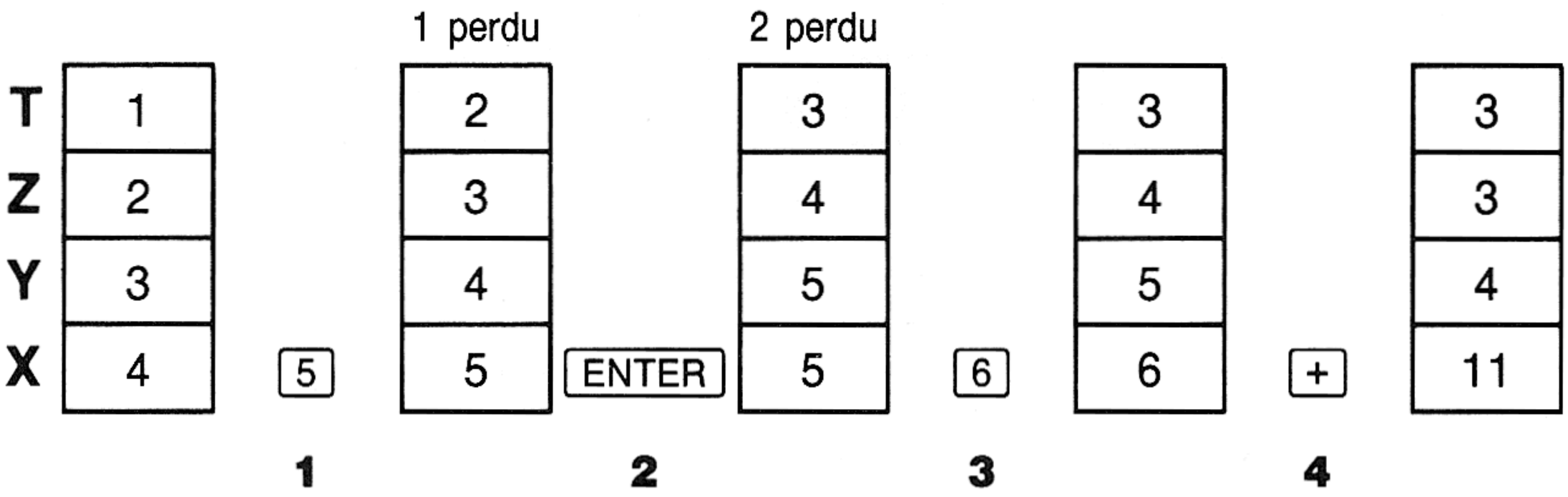
3 La pile redescend.

- Lorsque la pile monte, elle repousse le contenu du registre supérieur, et ce contenu est perdu. C'est la limite de capacité de la pile : elle ne peut contenir que quatre nombres.
- Lorsque la pile descend, elle reproduit le contenu du registre T.
- A cause du mouvement automatique de la pile, vous n'avez *pas* besoin d'effacer l'écran avant de commencer un nouveau calcul.
- La plupart des calculs préparent la pile pour une « ascension » *lorsque le nombre suivant pénètre dans le registre* X. Vous trouverez à l'annexe B la liste des fonctions qui font remonter la pile.

La touche ENTER

Vous savez déjà que [ENTER] sépare deux nombres qui ont été frappés l'un après l'autre. Comment cette séparation est-elle reflétée dans la pile ? Supposons cette fois encore que la pile soit emplie par les nombres 1, 2, 3 et 4. Saisissez maintenant deux nombres et ajoutez-les :

$5 + 6$:



1 Remonte la pile.

2 Remonte la pile et reproduit le registre X.

3 Ne remonte *pas* la pile.

4 Descend la pile et reproduit le contenu du registre T.

[ENTER] reproduit le contenu du registre X dans le registre Y. Le nombre suivant qui sera saisi (ou rappelé) *écrase* (remplace) le premier nombre laissé dans le registre X. L'effet obtenu est la séparation de deux nombres saisis à la suite l'un de l'autre.

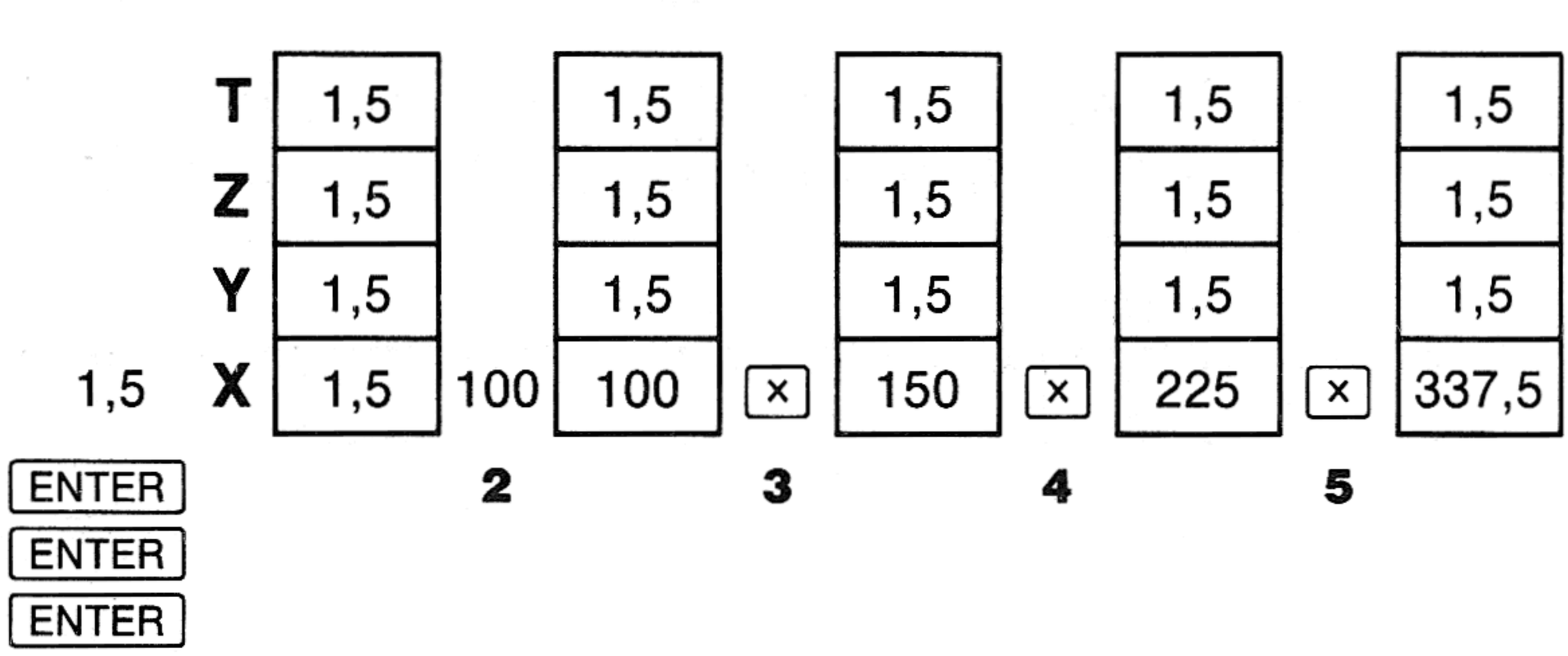
L'effet « duplicateur » de la touche [ENTER] peut être utilisé pour effacer rapidement la pile : appuyez sur 0 [ENTER] [ENTER] [ENTER]. Tous les registres contiennent maintenant zéro. Mais il n'est pas *nécessaire* d'effacer la pile avant de commencer de nouveaux calculs.

Utiliser le nombre deux fois en suivant. La caractéristique de duplication de la touche [ENTER] peut être utilisée à d'autres fins. Pour ajouter un nombre à lui-même, appuyez sur [ENTER] [+].

Remplir la pile d'une constante. L'effet duplicateur de [ENTER] associé à celui de la « descente » de la pile (de T à Z) permet d'emplir la pile d'une constante numérique pour les calculs.

Exemple : constante de croissance. Une culture bactérienne présente un taux de croissance de 50 %. Quelle sera la taille d'une population de 100 après trois jours ?

Reproduit le contenu de T



1 Emplit la pile avec le taux de croissance.

2 Saisie de la population de départ.

3 Calcul de la population après 1 jour.

4 Calcul de la population après 2 jours.

5 Calcul de la population après 3 jours.

La fonction CLEAR x

Le fait d'effacer l'écran (le registre X) y place un zéro. Le numéro que vous introduirez ensuite (ou que vous rappellerez) *écrasera* ce zéro.

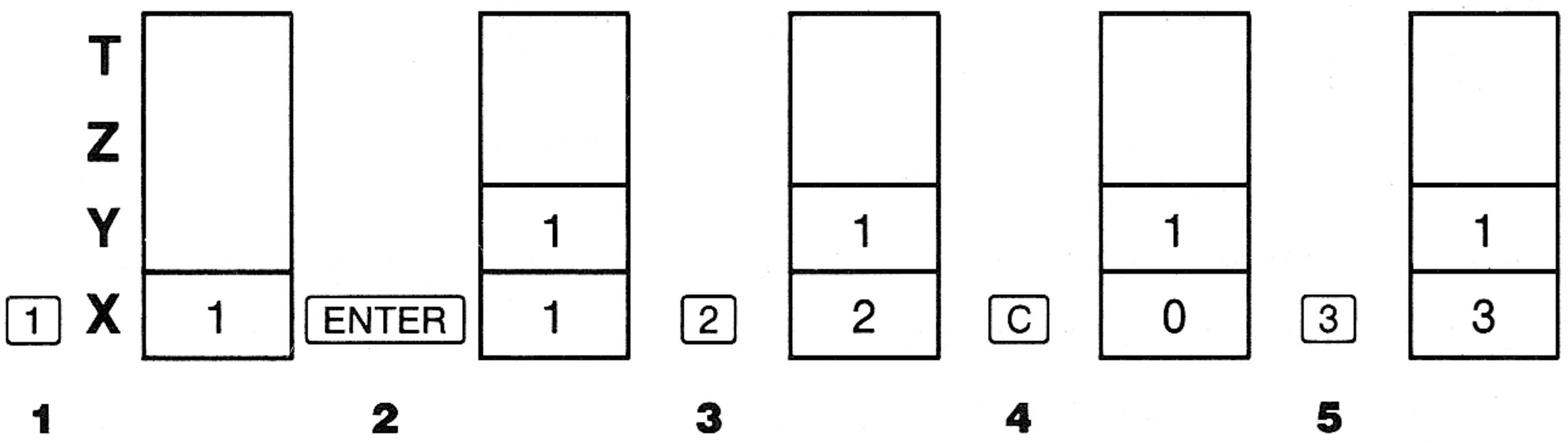
Il y a trois manières d'effacer le nombre placé dans le registre X, c'est-à-dire, d'exécuter *clear x* :

- Appuyez sur [C].
- Appuyez sur [←].
- Appuyez sur [■] [CLEAR] {×}. Ces frappes sont principalement utilisées lors de la saisie de programmes.

Il y a cependant des exceptions :

- Pendant la saisie de programme, [←] supprime les lignes de programme et [C] annule les saisies de programme.
- Pendant la saisie de chiffres, [←] revient d'un espace en arrière et efface un nombre chiffre par chiffre.
- Si l'écran affiche un nombre ayant reçu un *label* (par exemple, A=2,0000), le fait d'appuyer sur [C] ou sur [←] annule l'écran et affiche le registre X.

Si par exemple vous vouliez saisir 1 et 3 mais que, par erreur, vous avez saisi 1 et 2, voici comment procéder :



1 Remonte la pile.

2 Monte la pile et reproduit le registre X.

3 Ecrase le contenu du registre X.

4 Efface le contenu de *x* en le remplaçant par zéro.

5 Ecrase le contenu dex (remplace le zéro).

Le registre LAST X

Il est en quelque sorte une extension de la pile ; il contient le nombre qui *était* dans le registre X avant que la dernière fonction ait été exécutée (une fonction numérique est un opérateur qui produit un résultat à partir d'un ou de plusieurs nombres, comme [√x]). Le fait d'appuyer sur ■ [LASTx] renvoie cette valeur dans le registre X. Cette faculté de récupérer le « dernier X » est utilisée de deux façons : la correction d'erreurs et la réutilisatioin d'un nombre lors d'un calcul.

Voyez, à l'annexe B, une liste complète des fonctions qui sauvegardent *x* dans le registre LAST X.

Correction d'erreurs avec ■ LASTx

Fonction monadique incorrecte. S'il vous arrive d'effectuer une fonction monadique incorrecte, utilisez ■ LASTx pour récupérer le nombre de façon à exécuter la fonction correcte (et si vous désirez éliminer le résultat erroné de la pile, appuyez sur C *d'abord*).

Puisque ■ % et ■ %CHG ne causent pas de « descente » de la pile, vous pouvez rétablir la situation de la même manière que dans le cas des fonctions monadiques.

Exemple. Supposons que vous veniez de terminer ln 4,7839 × (3,879 × 10^5) et que vous désiriez trouver sa racine carrée, mais que vous ayez appuyé par erreur sur e^x. Nul besoin de recommencer. Pour trouver le résultat correct, appuyez sur ■ LASTx $\sqrt{x}$.

Erreurs impliquant des fonctions diadiques. Si vous faites une erreur lors de l'exécution d'une fonction à deux nombres (+, −, ×, ÷ ou y^x), vous pourrez la corriger en utilisant ■ LASTx et l'*inverse* de cette fonction diadique (− ou +, ÷ ou ×, 1/x ou y^x) :

1. Appuyez sur ■ LASTx pour récupérer le second nombre (*x* juste avant l'opération).
2. Exécutez la fonction inverse. Ceci vous renvoie le nombre qui était le premier à l'origine. Le second nombre est toujours dans le registre LAST X. Ensuite :
   - Si vous exécutez une *fonction* erronnée, appuyez encore sur ■ LASTx pour rétablir le contenu original de la pile. Puis exécutez la fonction correcte.
   - Si vous avez introduit un *second nombre* incorrect, frappez le nombre correct et exécutez la fonction.

Si vous avez utilisé un *premier nombre* erroné, frappez le premier nombre correct, appuyez sur ■ LASTx pour récupérer le second nombre et exécutez à nouveau la fonction (appuyez d'abord sur C si vous désirez effacer le résultat incorrect de la pile).

Exemple. Supposons que vous fassiez une erreur lors du calcul suivant :

$$16 \times 19 = 304.$$

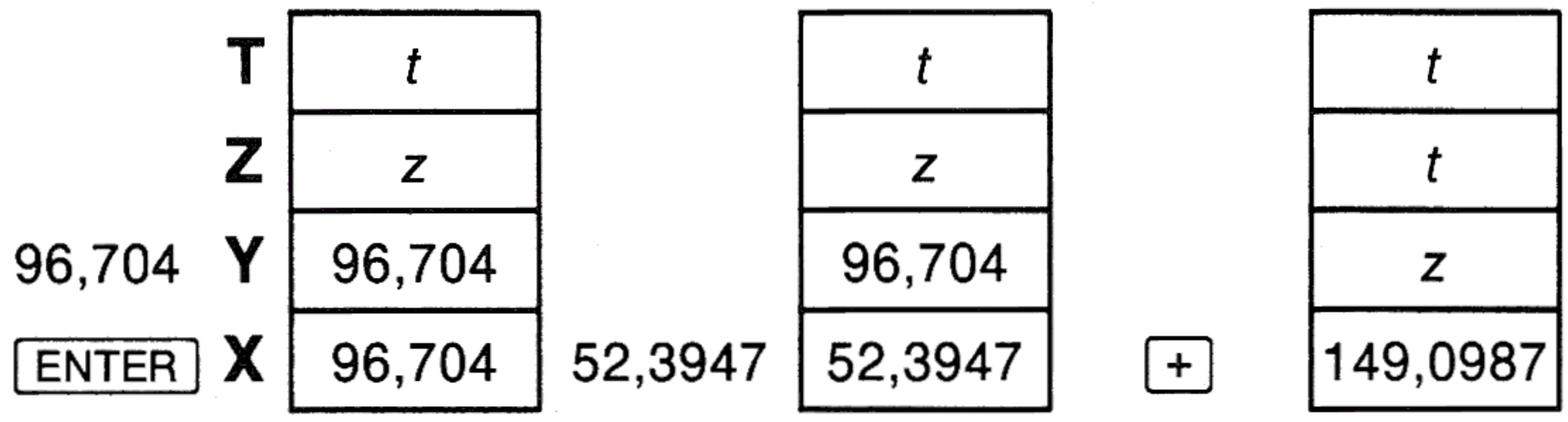
Vous pouvez avoir fait trois sortes d'erreurs :

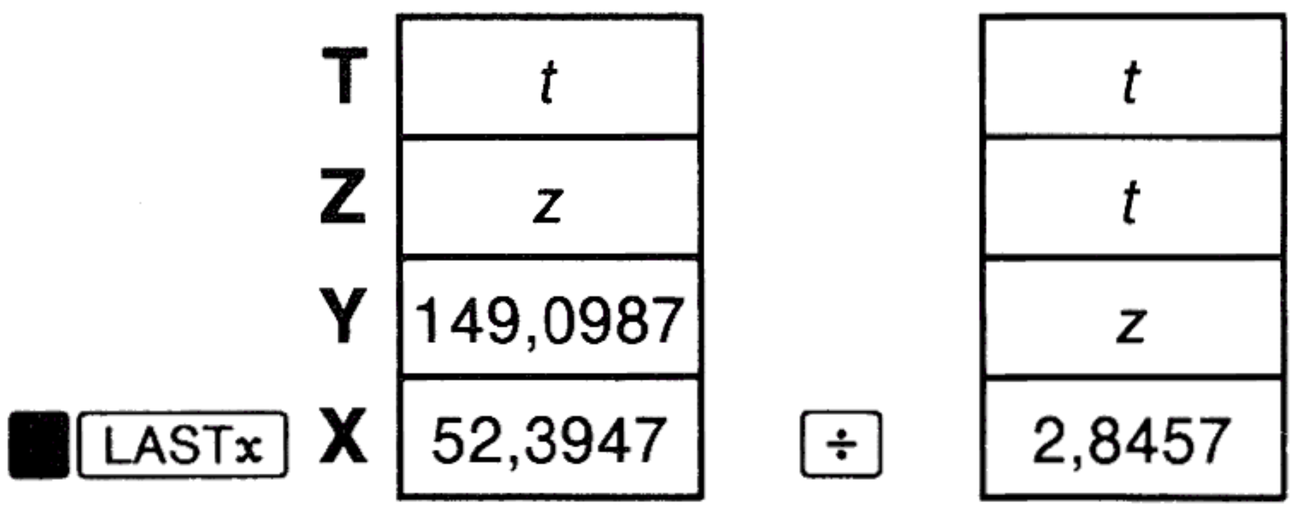
Calcul erroné	Erreur	Correction
16 ENTER 19 −	Fonction incorrecte.	■ LASTx + ■ LASTx ×
15 ENTER 19 ×	Le premier nombre est incorrect.	16 ■ LASTx ×
16 ENTER 18 ×	Le second nombre est incorrect.	■ LASTx ÷ 19 ×

Récupération de nombres avec ■ LASTx

Vous pouvez utiliser ■ LASTx pour récupérer un nombre (tel qu'une constante) dans un calcul. Souvenez-vous d'introduire la constante en second lieu, juste avant d'exécuter la foncion arithmétique, de façon à ce que la constante soit le dernier nombre à passer dans le registre *X*, et puisse ainsi être récupérée par ■ LASTx.

Exemple. Calculez $\frac{96{,}704 + 52{,}3947}{52{,}3947}$.





Touches :	Affichage :	Description :
96,704 [ENTER]	96,7040	
52,3947 [+]	149,0987	Résultat intermédiaire.
■ [LASTx]	52,3947	Rétablit l'affichage présent avant [+].
[÷]	2,8457	Résultat final.

Exemple. Deux voisins stellaires de la terre sont Rigel du Centaure (à 4,3 années-lumières de distance) et Sirius (à 8,7 années-lumières). Connaissant la vitesse de la lumière, *c*, ($9{,}5 \times 10^{15}$ mètres par an) convertissez la distance spérant la terre de ces étoiles en mètres.

$$\text{A Rigel du Centaure} = 4{,}3 \text{ an.} \times (9{,}5 \times 10^{15} \text{ m/an.})$$

$$\text{A Sirius} = 8{,}7 \text{ an.} \times (9{,}5 \times 10^{15} \text{ m/an.})$$

Touches :	Affichage :	Description :
4,3 [ENTER]	4,3000	Distance de la terre à R. du Centaure, en années-lumière.
9,5 [E] 15	9,5E15_	Vitesse de la lumière, *c*.
[x]	4,085E16	Distance à R. du Centaure.
8,7 ■ [LASTx]	9,5000E15	Récupère *c*.
[x]	8,2650E16	Donne la distance de la terre à Sirius.

Calculs en chaîne

La montée et la descente automatique de la pile vous permet de conserver les résultats intermédiaires sans les stocker ou les saisir à nouveau, et sans utiliser de parenthèses. Ceci est un des avantages de la notation polonaise inverse sur les autres méthodes de traitement des données.

L'ordre des calculs

Au premier chapitre nous recommendions de résoudre les calculs en chaîne en partant de l'intérieur des parenthèses ; vous pouvez cependant également choisir de les résoudre en procédant de la gauche vers la droite.

Par exemple, au chapitre 1 vous avez calculé :

$$4 \div [14 + (7 \times 3) - 2]$$

en commençant par l'intérieur des parenthèses (7 × 3) et en progressant vers l'extérieur, comme vous l'auriez fait lors d'un calcul à main levée. Les frappes étaient les suivantes :

7 [ENTER] 3 [×] 14 [+] 2 [−] 4 [x≷y] [÷].

En procédant de la gauche vers la droite, la solution serait :

4 [ENTER] 14 [ENTER] 7 [ENTER] 3 [×] [+] 2 [−] [÷],

ce qui ajoute une frappe au clavier. Remarquez que le résultat intermédiaire est toujours celui qui se trouve au plus profond de la parenthèse (7 × 3). L'avantage de la méthode « de gauche à droite » est que vous n'avez pas à utiliser [x≷y] pour repositionner les opérandes pour les fonctions non-commutatives ([−] et [÷]).

La première méthode (en commençant par la partie la plus interne de la parenthèse) est préférable parce que :

- Elle nécessite moins de frappes.
- Elle nécessite moins de registres de stockage dans la pile.

Lors de l'utilisation de la méthode « de gauche à droite », assurez-vous que moins de quatre résultats intermédiaires seront nécessires à la fois, puisque c'est la capacité maximale de la pile. Lors de la résolution de cet exemple par la méthode de gauche à droite, tous les registres de la pile étaient nécessaires à un certain point :

$$4 \div [14 + (7 \times 3) - 2]$$

Touches :	Affichage :	Description :
4 [ENTER]		Sauvegarde 4 et 14,
14 [ENTER]	14,0000	nombres intermédiaires, dans la pile.
7 [ENTER] 3	3_	A ce stade, la pile est pleine de nombres.
[×]	21,0000	Résultat intermédiaire.
[+]	35,0000	Résultat intermédiaire.
2 [−]	33,0000	Résultat intermédiaire.
[÷]	0,1212	Résultat final.

Exercices

Voici quelques autres problèmes qui vous permettront de pratiquer la notation polonaise :

Calculez : $(14 + 12) \times (18 - 12) \div (9 - 7) = 78{,}0000$
Solution : 14 [ENTER] 12 [+] 18 [ENTER] 12 [−] [×] 9 [ENTER] 7 [−] [÷]

Calculez : $23^2 - (13 \times 9) + 1/7 = 412{,}1429$
Solution : 23 ■ [x^2] 13 [ENTER] 9 [×] [−] 7 [1/x] [+]

Calculez : $\sqrt{(5{,}4 \times 0{,}8) \div (12{,}5 - 0{,}7^3)} = 0{,}5961$
Solution : 5,4 [ENTER] ,8 [×] ,7 [ENTER] 3 [y^x] 12,5 [$x \gtrless y$] [−] [÷] [$\sqrt{x}$]
ou
5,4 [ENTER] ,8 [×] 12,5 [ENTER] ,7 [ENTER] 3 [y^x] [−] [÷] [$\sqrt{x}$]

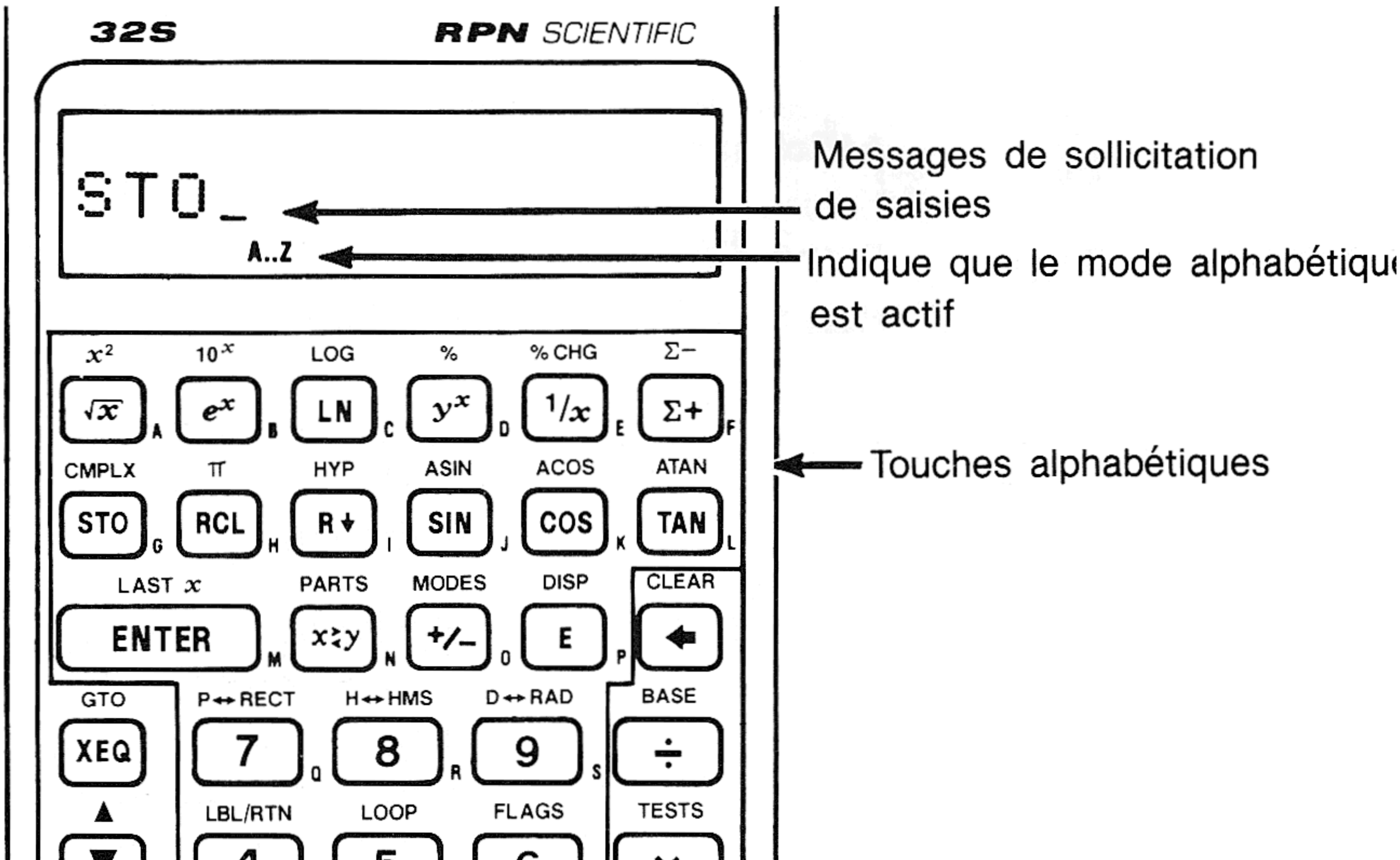
Calculez : $\sqrt{\dfrac{8{,}33 \times (4 - 5{,}2) \div [(8{,}33 - 7{,}46) \times 0{,}32]}{4{,}3 \times (3{,}15 - 2{,}75) - (1{,}71) \times (2{,}01)}} = 4{,}5728$

Solution : 4 [ENTER] 5,2 [−] 8,33 [×] ■ [LASTx] 7,46 [−] 0,32 [×] [÷]
3,15 [ENTER] 2,75 [−] 4,3 [×] 1,71 [ENTER] 2,01 [×] [−] [÷] [$\sqrt{x}$]

3

Comment stocker des données dans des variables

Le HP-32S possède 390 octets de *mémoire utilisateur* : vous pouvez y stocker des nombres ou des lignes de programme. Les nombres sont stockés dans des *variables* ; chacune d'entre elles est identifiée par une letre de *A* à Z. Par exemple *S* serait le solde d'un compte bancaire, *C* la vitesse de la lumière.*



* Notez que les variables *X*, *Y*, *Z* et *T* se trouvent en une zone de mémoire *différente* de celle des registres X, Y, Z et T.

Chaque lettre est associée à une touche et à une variable uniques. Les touches alphabétiques sont automatiquement activées lorsqu'elles deviennent nécessaires, et le témoin **A..Z** le confirme.

Stockage et rappel de nombres

Les nombres sont stockés et rappelés des variables par les touches [STO] (*store,* stocker) et [RCL] (*recall,* rappeler).

Pour sauvegarder une copie d'un nombre se trouvant à l'écran (registre X) dans une variable : appuyez sur [STO] *touche alphabétique*.

Pour rappeler une copie d'un nombre d'une variable sur l'écran : appuyez sur [RCL] *touche alphabétique*.

Exemple : stockage de nombres. Stockez le *nombre d'Avogadro* (approximativement $6{,}0225 \times 10^{23}$) dans la variable *A*.

Touches :	**Affichage :**	**Description :**
6,0225 [E] 23	6,0225E23_	
[STO]	STO _	Sollicite une variable.
A (touche [√x])	STO A	Affiche la fonction tant que l'appui est maintenu.
	6,0225E23	Stocke une copie du nombre d'Avogadro dans *A*. Ceci met également fin à la saisie (le curseur a disparu).
[C]	0,0000	Efface le nombre de l'affichage.
[RCL]	RCL _	Sollicite le nom de la variable.
A	6,0225E23	Copie le nombre d'Avogadro de *A* sur l'affichage.

Visualiser une variable sans la rappeler. La fonction ■ VIEW montre le contenu d'une variable sans placer le nombre correspondant dans le registre X. L'affichage indique la variable et le nombre stocké :

```
A=1234,5678
```

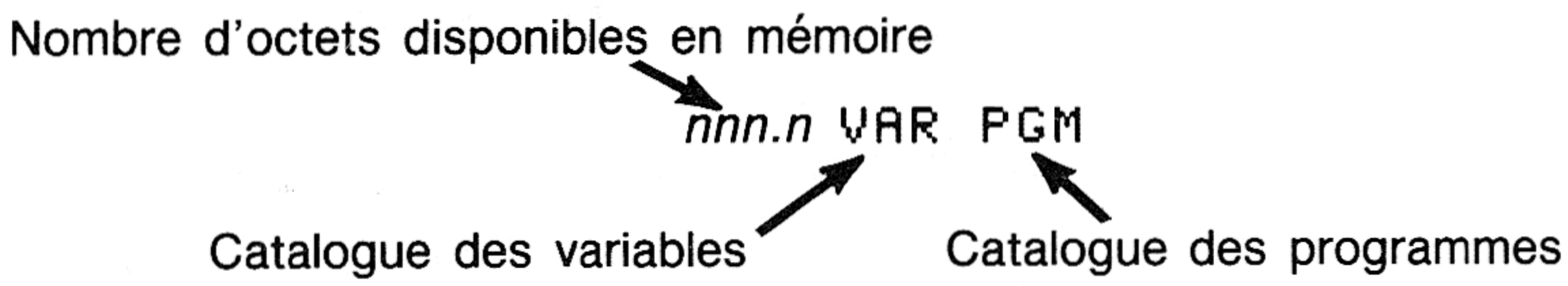
Si le nombre est trop grand pour tenir sur l'écran avec la variable, il est arrondi et les chiffres placés à l'extrême droite sont éliminés (mais un exposant est affiché complètement). Pour visualiser la mantisse en entier, appuyez sur ■ SHOW.

■ VIEW est le plus souvent utilisée en programmation.

Pour annuler l'affichage VIEW, appuyez sur ◄ ou sur C.

Examen de variables dans le catalogue VAR

La fonction ■ MEM (*mémoire*) renseigne sur l'état de la mémoire :



Pour examiner la valeur de toute variable non-nulle :

1. Appuyez sur ■ MEM {VAR}.
2. Appuyez sur ▼ ou sur ■ ▲ pour déplacer la liste et afficher la variable désirée (notez la présence du témoin ▼▲, indiquant que les touches-flèches sont actives).

   Pour visualiser tous les nombres significatifs d'un nombre affiché dans le catalogue {VAR}, appuyez sur ■ SHOW (s'il s'agit d'un nombre binaire de plus de 12 chiffres, utilisez les touches √x et Σ+ pour lire le reste).
3. Pour recopier une variable affichée du catalogue dans le registre X, appuyez sur ENTER.
4. Pour effacer une variable, appuyez sur ■ CLEAR lorsqu'elle est affichée dans le catalogue.
5. Appuyez sur C pour annuler le catalogue ou ◄ pour revenir au menu.

Effacer des variables

Les valeurs des variables sont conservées dans la *mémoire permanente* du calculateur jusqu'à ce que vous les remplaciez ou les effaciez. Le fait d'*effacer* une variable place un zéro à cet endroit ; cette valeur n'occupe pas d'espace-mémoire.

Pour effacer une seule variable : stockez-y un zéro.

Pour effacer plusieurs variables :

1. Appuyez sur ■ [MEM] {VAR} et utilisez [▼] ou ■ [▲] pour effacer la variable.
2. Appuyez sur ■ [CLEAR].
3. Appuyez sur [C] pour effacer le catalogue, ou sur [←] pour revenir en arrière.

Pour effacer toutes les variables d'un coup : ■ [CLEAR] {VARS}.

Arithmétique sur des variables stockées

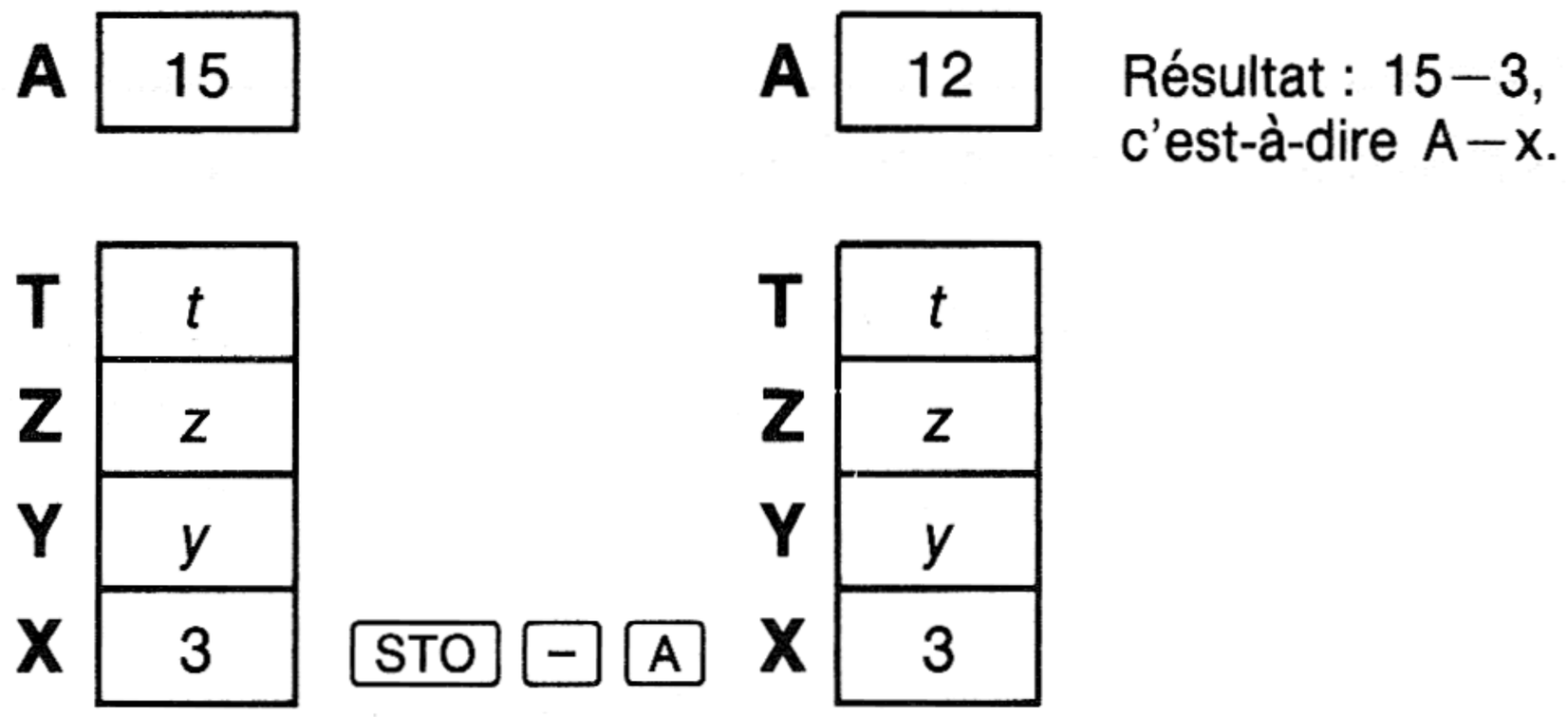
L'arithmétique de stockage et *l'arithmétique de rappel* vous permettent d'effectuer des calculs sur un nombre stocké dans une variable *sans véritablement le rappeler dans la pile*. Le calcul utilise un nombre pris sur le registre X et un nombre stocké dans la variable spécifiée.

Arithmétique de stockage

L'arithmétique de stockage utilise [STO] [+], [STO] [−], [STO] [×] ou [STO] [÷] pour effectuer des calculs arithmétiques dans la variable elle-même et pour y stocker le résultat. Elle utilise la valeur placée dans le registre X et ne modifie pas la pile.

La nouvelle valeur d'une variable = Valeur précédente $\{+, -, \times, \div\}$ x

Supposons par exemple que vous désirez remplacer la valeur placée dans *A* (15) par le nombre se trouvant dsans le registre X (3, affiché). Appuyez sur [STO] [−] A. Maintenant $A = 12$, et 3 est toujours affiché.



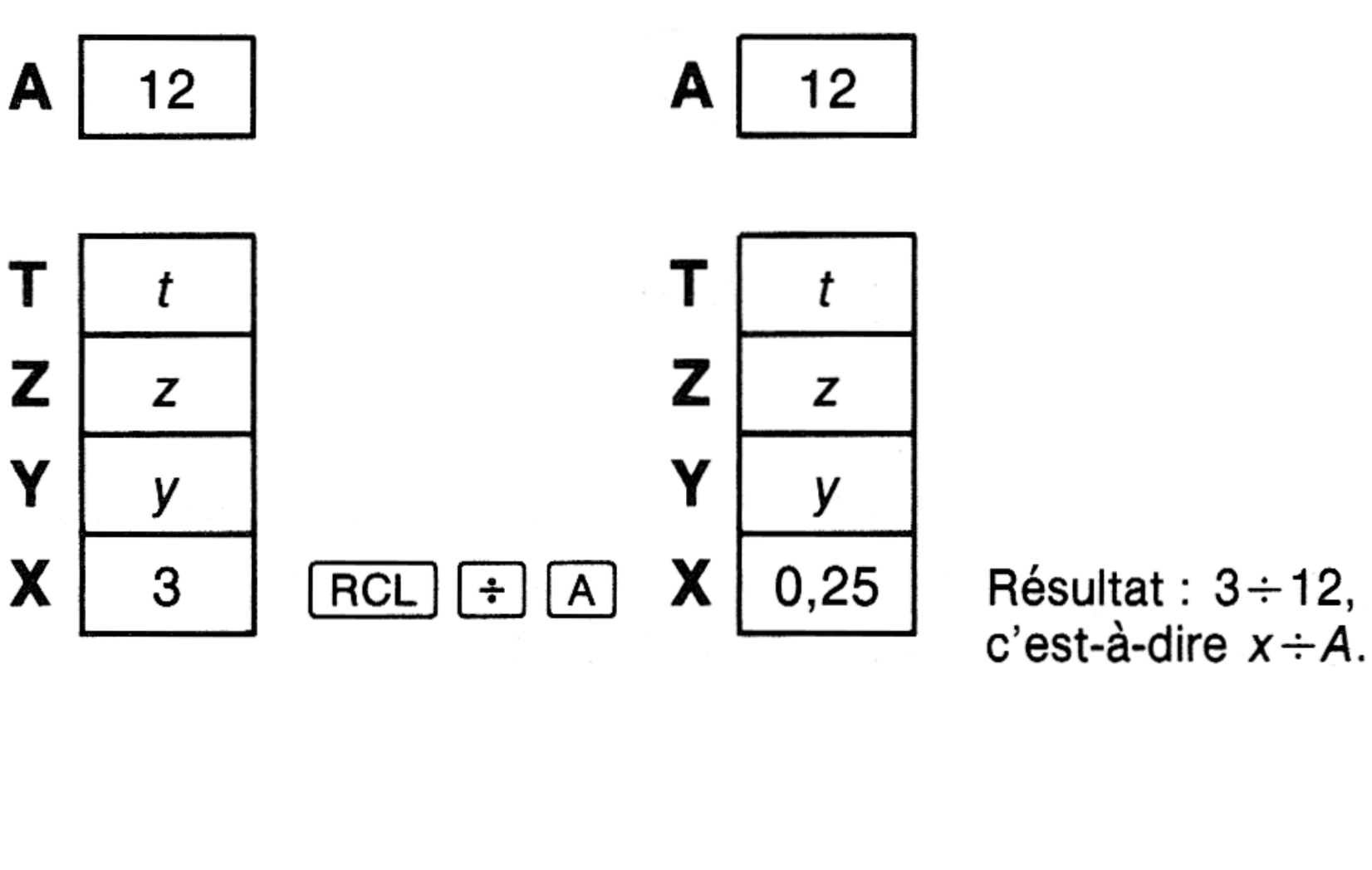
Résultat : 15−3, c'est-à-dire A−x.

Arithmétique de rappel

Elle utilise RCL +, RCL −, RCL × ou RCL ÷ pour effectuer des calculs arithmétiques dans le registre X en utilisant un nombre rappelé de la mémoire et en laissant le résultat sur l'affichage. Seul le registre X est modifié ; les autres registres de stockage ne sont pas concernés.

$$\text{Nouvel } x = x \text{ précédent } \{+,\ -,\ \times,\ \div\} \text{ variable}$$

Supposons par exemple que vous désirez remplacer le nombre dans le registre X (3, nombre affiché) par la valeur stockée dans *A* (12). Appuyez sur RCL ÷ A. Maintenant, $x = 0{,}25$, alors que la valeur 12 est toujours stockée dans *A*.*



Résultat : 3÷12, c'est-à-dire *x*÷*A*.

* L'arithmétique de rappel épargne l'espace mémoire dans les programmes. Le fait d'utiliser RCL + A (une seule instruction) utilise la moitié d'espace de l'opération RCL A, + (deux instructions).

Quelques exemples. Supposez que les variables *D*, *E* et *F* contiennent les valeurs 1, 2 et 3. Utilisez l'arithmétique de stockage pour ajouter 1 à chacune de ces variables.

Touches :	Affichage :	Description :
1 STO D		Stocke les valeurs dans les variables.
2 STO E		
3 STO F	3,0000	
1 STO + D		Ajoute 1 à *D*, à *E* et à *F*.
STO + E		
STO + F	1,0000	
■ VIEW D	D=2,0000	Affiche la valeur en cours de *D*.
■ VIEW E	E=3,0000	
■ VIEW F	F=4,0000	
←	1,0000	Efface l'affichage VIEW ; vous retrouvez le registre X.

Supposons que les variables *D*, *E* et *F* contiennent les valeurs 2, 3 et 4 qui proviennent de l'exemple précédent. Divisez 3 par *D*, multipliez-le par *E* et ajoutez *F* au résultat.

Touches :	Affichage :	Description :
3 RCL ÷ D	1,5000	Calcule $3 \div D$.
RCL × E	4,5000	$3 \div D \times E$.
RCL + F	8,5000	$3 \div D \times E + F$.

La variable « *i* »

Il existe une 27e variable—la variable *i* (le [i] se trouve « sous » la touche [·]). Bien qu'il stocke des nombres, comme les autres variables, [i] permet aussi de *référer à d'autres variables* (par la fonction [(i)] ; c'est là une technique nommée *adressage indirect*). Vous trouverez une description de la technique d'adressage indirect au chapitre 6, sous la rubrique « Adressage indirect de variables et de labels ».

4

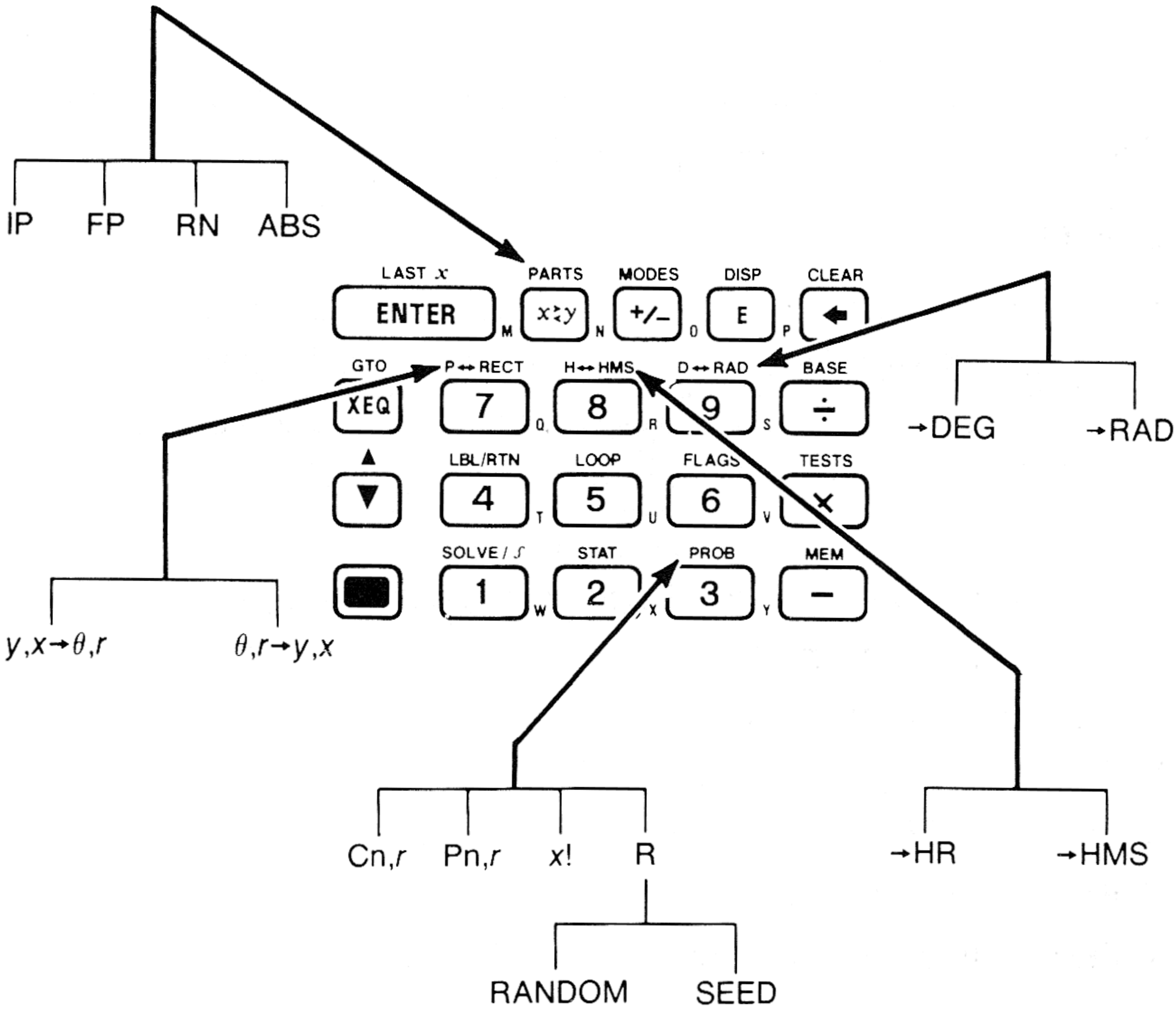
Fonctions sur les nombres réels

Ce chapitre traite de la plupart des fonctions du calculateur qui travaillent sur les nombres réels, y compris certaines fonctions numériques prévues pour la programmation (telles la fonction *valeur absolue*) :

- Fonctions exponentielles et logarithmiques.
- Fonctions trigonométriques.
- Fonctions hyperboliques.
- Fonctions de pourcentage.
- Fonctions de conversion pour coordonnées, angles et fractions.
- Fonctions de probabilités.
- Parties de nombres (fonctions modifiant des nombres).

Les fonctions et le calcul arithmétiques étaient traités au chapitres 1 et 2. Les opérations numériques de haut niveau (recherche de la racine, intégration, nombres complexes, conversions de bases et statistiques) se trouvent en troisième partie de ce manuel.

La plupart des fonctions numériques apparaissent sur les touches des deux rangées supérieures du clavier du calculateur. Le reste se trouve dans l'un de ces menus :



Fonctions exponentielles et logarithmiques

Placez le nombre sur l'affichage, puis exécutez la fonction. Nul besoin d'appuyer sur [ENTER].

Pour calculer :	Appuyer sur :
Logarithme naturel (base *e*)	[LN]
Logarithme commun (en base 10)	■ [LOG]
Exponentiel naturel	[e^x]
Exponentiel commun (cologarithme)	■ [10^x]

La fonction de puissance (y^x)

Pour calculer un nombre y elevé à une puissance x, frappez y [ENTER] x [y^x].

Si $y > 0$, x peut être n'importe quel nombre rationnel. Si $y < 0$, x doit être entier. Si $y = 0$, x doit être positif.

Par exemple :

Pour calculer :	Appuyer sur :	Résultat :
15^2	15 ■ [x^2]	225,0000
$2^{-1,4}$	2 [ENTER] 1,4 [+/−] [y^x]	0,3789
$(-1,4)^3$	1,4 [+/−] [ENTER] 3 [y^x]	−2,7440
$\sqrt[3]{2}$ or $2^{1/3}$	2 [ENTER] 3 [1/x] [y^x]	1,2599

Trigonométrie

Saisie de π

Appuyez sur ■ [π] Pour placer les premiers 12 chiffres de π dans le registre X (le nombre sera affiché en fonction du format d'affichage choisi). Ceci étant une fonction, nul besoin d'appuyer sur [ENTER].

Notez bien que le calculateur ne peut représenter *exactement* π, puisque π est un nombre irrationnel.

Définition du mode angulaire

Le mode angulaire spécifie l'unité de mesure qui sera utilisée pour les angles utilisés dans les fonctions trigonométriques. *Ce mode ne convertit pas les nombres déjà présents* (voyez « Fonctions de conversion », plus loin dans ce chapitre).

360 degrés = 2π radians = 400 degrés

Pour choisir le mode angulaire, appuyez sur ■ MODES. Ensuite, choisissez une option.

Option	Description	Témoin
{DG}	Définit le mode degrés (DEG). Utilise des fractions décimales et non des minutes et des secondes.	aucun
{RD}	Définit le mode Radians (RAD).	**RAD**
{GR}	Définit le mode Grades (GRAD).	**GRAD**

Fonctions trigonométriques

x étant affiché :

Pour calculer :	Appuyer sur :
Le sinus de *x*.	SIN
Le cosinus de *x*.	COS
La tangente de *x*.	TAN
L'arc sinus de *x*.	■ ASIN
L'arc cosinus de *x*.	■ ACOS
L'arc tangente de *x*.	■ ATAN

Remarque Les calculs qui impliquent π, nombre irrationnel, ne peuvent être exprimés *exactement* à cause de la précision du calculateur, limitée à 12 chiffres. Ceci se remarque particulièrement en trigonométrie ; par exemple, le calcul du sinus de π n'est pas zéro mais $-2{,}0676 \times 10^{-13}$, un très petit nombre proche de zéro.

Exemple. Montrez que le cosinus de $(5/7)\pi$ radians et le cosinus de 128,57 ° sont identiques.*

Touches :	**Affichage :**	**Description :**
■ MODES {RD}		Définit le mode Radians ; le témoin RAD apparaît.
5 ENTER 7 ÷ ■ π ×		Cos $(5/7)\pi$.
COS	-0,6235	
■ MODES {DG}		Passe au mode Degrés (pas de témoin cette fois) et calcule le cosinus de 128,57 °, qui est le même que le cosinus de $(5/7)\pi$.
128,57 COS	-0,6235	

Note de programmation. Les équations utilisant les fonctions trigonométriques inverses pour déterminer un angle θ ont souvent l'aspect suivant :

$$\theta = \arctan\ (y/x).$$

Si $x = 0$, $y \div x$ est indéfini, ce qui résulte en l'erreur DIVIDE BY 0 (division par zéro). Dans un programme, il serait donc plus prudent de déterminer θ par une conversion *rectangulaire en polaire,* qui convertit x, y en r, θ. Voyez « Conversions de coordonnées », plus loin dans ce chapitre.

* En fait, ces résultats calculés ne sont les mêmes que pour quatre chiffres significatifs, à cause de la représentation approximative de π (appuyez sur ■ SHOW pour visualiser plus de chiffres).

Fonctions hyperboliques

x étant affiché :

Pour calculer :	Appuyer sur :
Sinus hyperbolique de x (SINH).	■ HYP SIN
Cosinus hyperbolique de x (COSH).	■ HYP COS
Tangente hyperbolique de x (TANH).	■ HYP TAN
Arc sinus hyperbolique de x (ASINH).	■ HYP ■ ASIN
Arc cosinus hyperbolique de x (ACOSH).	■ HYP ■ ACOS
Arc tangent hyperbolique de x (ATANH).	■ HYP ■ ATAN

Fonctions de pourcentage (%, %CHG)

Les fonctions de pourcentage sont particulières (comparées à [×] et à [÷]) en ceci qu'elles préservent la valeur du nombre de base (dans le registre Y) lorsqu'elles renvoient le résultat du calcul de pourcentage (dans le registre X). Vous pouvez alors exécuter les calculs suivants en utilisant à la fois le nombre de base et le résultat, sans saisir à nouveau le nombre de base.

Pour calculer :	Frappez :
x% de y	y ENTER x ■ %
Pourcentage de changement de y à x. ($y \neq 0$)	y ENTER x ■ %CHG

Exemple. Une taxe de 6 % est appliquée à un article coûtant 15 francs 76 centimes. Utilisez FIX 2 pour arrondir le résultat comme il convient.

Touches :	Affichage :	Description :
■ DISP {FX} 2		Réduit l'affichage à deux décimales.

15,76 ENTER 6 ■ %	0,95	Calcule la taxe de 6 %.
+	16,71	Coût total (prix de base + taxe).

Supposez que l'article que vous avez acheté 15,76 frs coûtait 16,12 frs l'an dernier. Quel est le pourcentage de changement par rapport au prix de l'an dernier ?

Touches :	**Affichage :**	**Description :**
16,12 ENTER 15,76 ■ %CHG	-2,23	Cette année, le prix a baissé d'environ 2,2 % par rapport à l'an dernier.
■ DISP {FX} 4	-2,2333	Rétablit l'affichage de 4 décimales après la virgule.

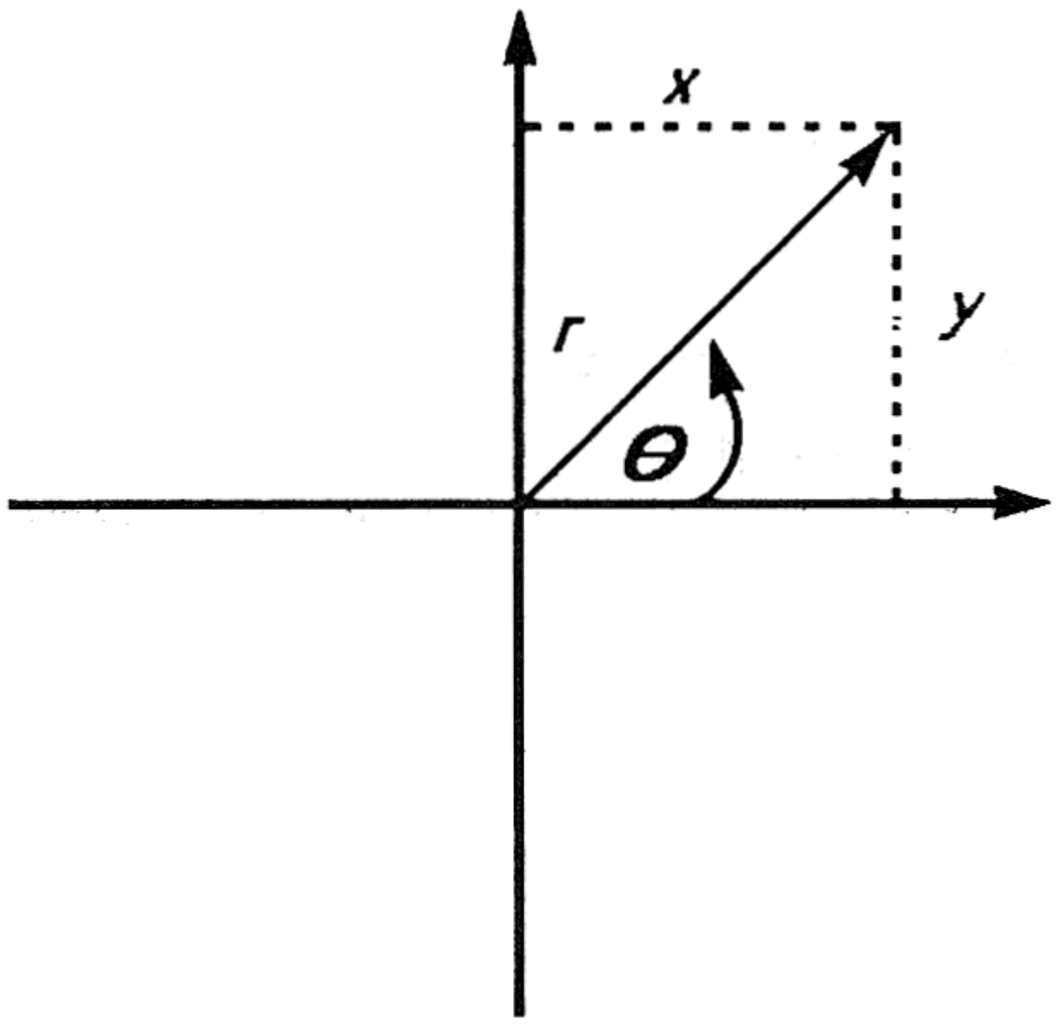
Notez que l'*ordre* des deux nombres est important pour ce calcul. L'ordre détermine si le pourcentage de changement est positif ou négatif.

Fonctions de conversion

Il y a trois types de conversions : coordonnées (polaires/rectangulaires), angles (degrés/radians) et fractions (décimales/minutes-secondes).

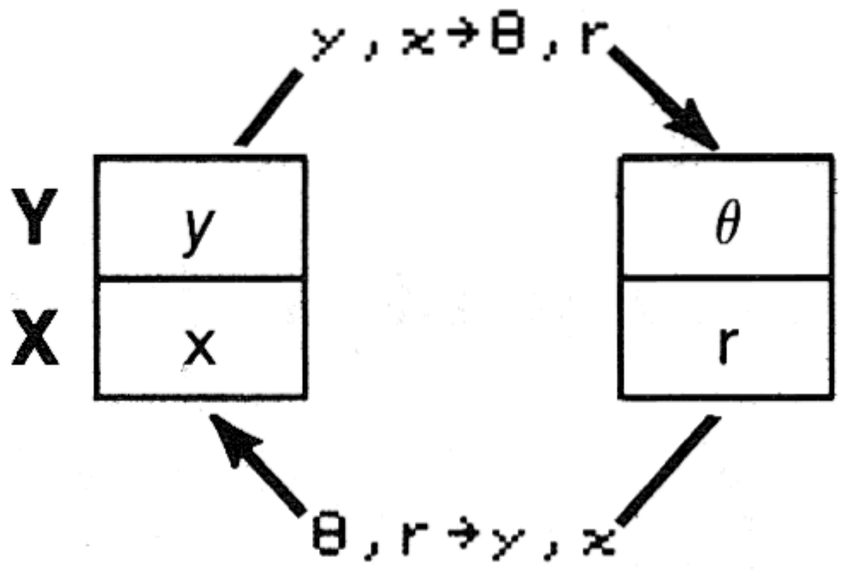
Conversions de coordonnées (P↔RECT)

Les coordonnées rectangulaires (x, y) et polaires (r, θ) sont mesurées comme indiqué dans l'illustration de la page suivante. Les fonctions du menu P↔RECT (*polaires/rectangulaires* et vice-versa) convertissent de l'un à l'autre. L'angle θ utilise les unités déterminées par le mode angulaire en cours à ce moment. Un résultat calculé pour θ se situera entre -180 ° et 180 °, entre $-\pi$ et π radians ou entre -200 et 200 grades.

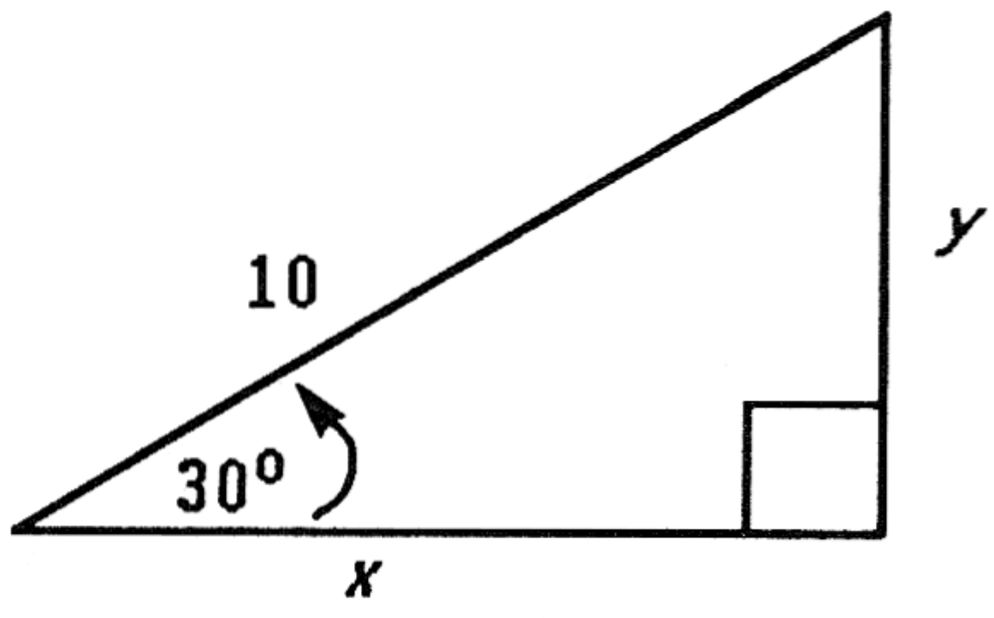


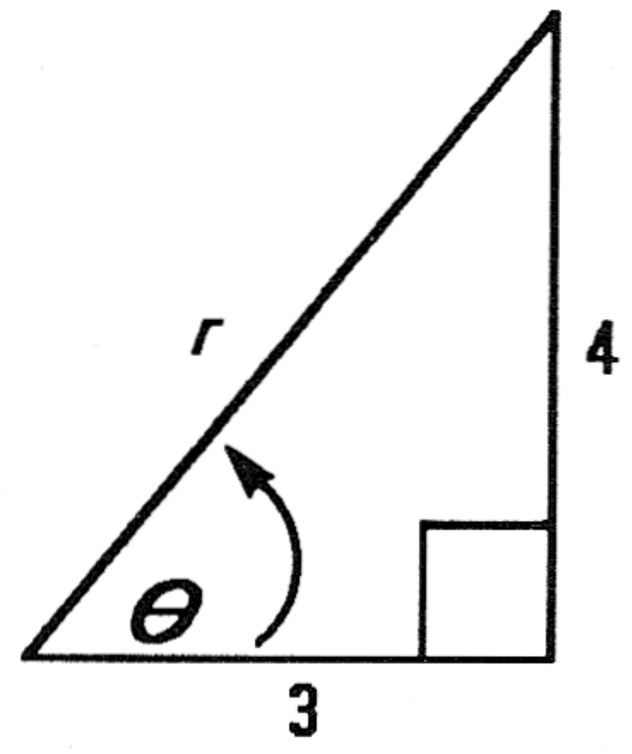
Pour convertir entre coordonnées rectangulaires (*x*, *y*) et coordonnées polaires (*r*, θ) :

1. Entrez d'abord les coordonnées que vous désirez convertir (en format rectangulaire ou polaire). L'ordre est *y* ENTER *x* ou θ ENTER *r*.
2. Appuyez sur ■ P↔RECT.
3. Exécutez la conversion désirée : {y,x→θ,r} (*rectangulaires en polaires*) ou {θ,r→y,x} (*polaires en rectangulaires*). Les coordonnées converties occupent ensuitre les registres X et Y.
4. L'affichage montrera soit *r* (résultat polaire), soit *x* (résultat rectangulaire). Appuyez sur x≷y pour voir θ ou *y*.



Exemple : conversion polaires en rectangulaires. Trouvez *x* et *y* dans le triangle rectangle de gauche de l'illustration de la page suivante, et *r* et θ dans le triangle rectangle de droite.

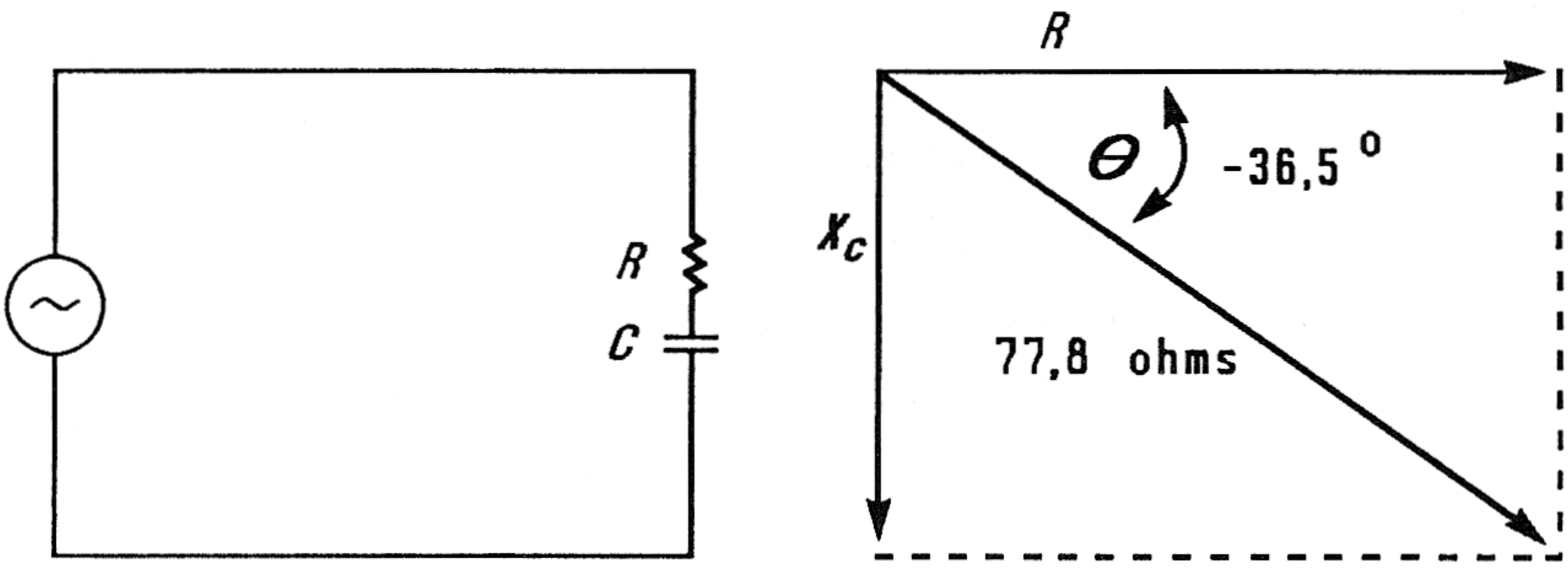




Touches :	**Affichage :**	**Description :**
■ MODES {DG}		Définit le mode degrés.
30 ENTER 10		Calcule x.
■ P↔RECT {θ,r→y,x}	8,6603	
x≷y	5,0000	Affiche y.
4 ENTER 3		Calcule l'hypoténuse (r).
■ P↔RECT {y,x→θ,r}	5,0000	
x≷y	53,1301	Affiche θ.

Exemple : conversion avec vecteurs. Un ingénieur détermine que, dans le circuit RC illustré en haut et à gauche de la page suivante, l'impédance est de 77,8 ohms et que le voltage suit le courant avec un retard de 36,5 °. Quels sont les valeurs de la résistance, R et de la réactance capacitive, X_C, dans le circuit ?

Utilisez un diagramme et tracez des vecteurs comme indiqué, l'impédance étant égale au module r, et le voltage égal à l'argument θ (l'angle), en degrés. Lorsque les valeurs sont converties en coordonnées rectangulaires, la valeur d'abcisse x, est R, en ohms, alors que la valeur de l'ordonnée y est X_C, en ohms.



Touches :	Affichage :	Description :
■ MODES {DG}		Définit le mode degrés.
36,5 +/– ENTER	-36,5000	Saisit θ, degrés de déphasage.
77,8	77,8_	Saisit r, ohms d'impédance totale.
■ P↔RECT {θ,r→y,x}	62,5401	Calcule x, résistance en ohms, R.
x≷y	-46,2772	Affiche y, réactance, en ohms, X_c.

Pour des opérations plus sophistiquées sur les vecteurs (addition, soustraction, produit vectoriel et produit scalaire), faites référence au programme « Opérations sur les vecteurs » au chapitre 12 (« Programmes mathématiques »).

Conversions fractionnaires (H↔HMS)

Les valeurs de temps (en heures, *H*) ou d'angles (en degrés, *D*) peuvent être converties entre la forme fraction décimale (*H.h* ou *D.d*) et minutes-secondes (*H.MMSSss* ou *D.MMSSss*) en utilisant le menu H↔HMS (*heures/heures-minutes-secondes*).

Conversion entre fractions décimales et minutes-secondes :

1. Introduisez le temps ou l'angle que vous désirez convertir (en forme décimale ou minutes-secondes).
2. Appuyez sur ■ [H↔HMS].
3. Choisissez {→HR} (*heures-minutes-secondes en heures*) ou {→HMS} (*heures en heures-minutes-secondes*). Le résultat est affiché.

Exemple : conversion de formats de temps. Quelle est la valeur heures-minutes-secondes dans 1/7 d'heure ? Utilisez FIX 6 comme format d'affichage.

Touches :	**Affichage :**	**Description :**
■ [DISP] {FX} 6		
7 [1/x]	0,142857	Un septième sous forme décimale.
■ [H↔HMS] {→HMS}	0,083429	Egale 8 minutes et 34,29 secondes.
■ [DISP] {FX} 4		Rétablit le format FIX 4.

Conversions angulaires (D↔RAD)

Le menu D↔RAD (*degrés/radians*) fonctionne de manière indépendante du mode angulaire. Lors de la conversion en radians, le nombre du registre X est supposé être fourni en degrés. De même, lors de la conversion en degrés, le nombre du registre X est supposé être en radians.

Conversion d'un angle entre degrés et radians :

1. Introduisez l'angle (en degrés décimaux ou en radians).
2. Appuyez sur ■ [D↔RAD].
3. Choisissez {→DEG} (*radians en degrés*) ou {→RAD} (*degrés en radians)*. Le résultat est affiché.

Fonctions de probabilités

Le menu PROB (*probabilités*) comporte des fonctions calculant la factorielle, les combinaisons et permutations, et permet d'obtenir des nombres aléatoires.

Le menu PROB

Libellé de menu	Description
{Cn,r}	*Combinaisons.* Saisissez d'abord *n*, puis *r.* (uniquement des entiers non négatifs). Calculez le nombre d'*ensembles* de *n* articles pris *r* à la fois. Aucun article n'est présent plus d'une fois dans un ensemble et différents ordres des mêmes articles *r* ne sont *pas* comptés séparément.
{Pn,r}	*Permutations.* Saisissez d'abord *n* , puis *r.* (uniquement des entiers non négatifs). Calcule le nombre de *combinaisons* posibles de *n* articles pris *r* à la fois. Aucun article ne se présente plus d'une fois dans une combinaison, et différentes combinaisons du même *r* sont comptées séparément.
{x!}	*Factorielle* et *Gamma*. Calcule la factorielle du nombre entier positif affiché ($0 \leqslant x \leqslant 253$). Pour calculer la fonction gamma de *a*, $\Gamma(a)$, frappez $(a - 1)$ et appuyez sur ■ PROB {x!} . La fonction {x!} calcule $\Gamma(x + 1)$. La valeur de *x* ne peut être un entier négatif.
{R}	*Générateur de nombres aléatoires*. Ce menu a deux sous-options. Le fait d'appuyer sur {RANDOM} génère un nombre aléatoire dans la plage de $0 \leqslant x < 1$.* Le fait d'appuyer sur {SEED} (valeur de départ) lance une nouvelle séquence de nombre aléatoire avec le nombre qui se trouve dans le registre X.

* Le générateur de nombres aléatoires du HP-32S renvoie un nombre qui fait partie d'une séquence de nombres pseudo-aléatoires distribuée uniformément. Il passe le test spectral (D. Knuth, *Seminumerical Algorithms,* vol. 2. Londres : Addison Wesley, édit., 1981).

{RANDOM} utilise une valeur de départ pour générer un nombre aléatoire. Chaque nombre aléatoire devient la valeur de départ pour le suivant. Vous pouvez stocker une nouvelle valeur de départ avec la fonction {SEED} . Si la mémoire est effacée, cette valeur est bien sûr remise à zéro.

Exemple : combinaisons de personnes. Une entreprise qui emploie 14 femmes et 10 hommes est en train de former un comité de 6 personnes. Quelles sont les différentes combinaisons possibles ?

Touches :	Affichage :	Description :
24 [ENTER] 6	6_	Vingt-quatre personnes groupées par six.
■ [PROB]	Cn,r Pn,r x! R	Menu de probabilités.
{Cn,r}	134.596,0000	Nombre total de combinaisons possibles.

Si les employés sont choisis de manière aléatoire, quelle est la probabilité que le comité soit composé de six femmes ? Pour établir la *probabilité* d'un évènement, divisez le nombre de combinaisons concernant cet évènement par le *nombre total* de combinaisons.

Touches :	Affichage :	Description :
14 [ENTER] 6	6_	Quatorze femmes regroupées par 6.
■ [PROB] {Cn,r}	3.003,0000	Nombre de combinaisons de six femmes dans le comité.
[x≷y]	134.596,0000	Amène le nombre total de combinaisons dans le registre X.
[÷]	0,0223	Divise les combinaisons d'employés féminins par le nombre total de combinaisons pour déterminer la probabilité que l'une des combinaisons comprenne tous les employés femmes de la société.

Parties de nombres

Les fonctions du menu PARTS modifient le nombre du registre X de manière simple. Elles sont utilisées en programmation.

Le menu PARTS

Libellé de menu	Description
{IP}	*Partie entière (Integer part).* Retire la partie entière de *x* et la remplace par des zéros. Par exemple, la partie entière de 14,2300 est 14,0000.
{FP}	*Partie fractionnaire (Fractional part).* Retire la partie entière de *x* et la remplace par des zéros. Par exemple, la partie fractionnaire de 14,2300 est 0,2300.
{RN}	*Round, Arrondi* (RND). Arrondit *x de manière interne* au nombre de chiffres spécifiés par le format d'affichage (s'il n'est pas arrondi, le nombre interne est représenté par 12 chiffres).
{ABS}	*Valeur absolue.* Remplace *x* par sa valeur absolue.

Noms des fonctions

Vous avez peut-être déjà remarqué que le nom de la fonction apparaît à l'affichage lorsque vous appuyez sur la touche qui la déclenche et que vous maintenez votre pression sur cette touche ; il disparaît dès que vous la relâchez. Par exemple, lorsque vous appuyez sur [√x], vous lisez SQRT sur l'affichage. « SQRT » est le nom de la fonction tel qu'il apparaîtra dans un programme et le nom sous lequel la fonction se trouve dans l'index des fonctions.

Deuxième partie

Programmation

Page 70 5 : Programmation élémentaire

90 6 : Techniques de programmation

5

Programmation élémentaire

La première partie de ce manuel vous a présenté les fonctions et les opérations que vous pouvez utiliser *manuellement,* c'est-à-dire par une frappe du clavier. Un *programme* vous permet de répéter les opérations et les calculs sans répéter les frappes sur le clavier. Dans ce chapitre nous allons vous montrer comment programmer une série d'opérations pour qu'elles se produisent automatiquement. Dans le chapitre suivant, « Techniques de programmation », vous apprendrez comment l'utilisation de sous-programmes et d'instructions conditionelles.

Introduction : un exemple de programmation élémentaire. Pour trouver la surface d'un cercle de 5 de rayon, vous utiliseriez normalement la formule $A = \pi r^2$ et vous appuieriez sur :

5 [x²] [π] [×]

pour obtenir le résultat, 78,5398.

Mais si vous désirez connaître la surface d'une série de cercles ? Plutôt que de répéter la même séquence de frappe, vous pouvez placer ces frappes dans le programme suivant :

```
001  x²
002  π
003  ×
```

Ce programme très simple suppose que la valeur du rayon se trouve dans le registre X (l'affichage) lorsque le programme commence son exécution. Il calcule la surface et la place dans le registre X.

Pour introduire ce programme en mémoire de programme, procédez comme ceci :

Touches :	**Affichage :**	**Description**
■ PRGM		Ceci remet le pointeur à l'état initial.
■ GTO · ·	PRGM TOP	
■ x^2 ■ π ×	001 x^2 002 π 003 ×	
■ PRGM		

Essayons maintenant de trouver la surface d'un cercle de rayon 5.

Touches :	**Affichage :**	**Description :**
■ GTO · ·		Ceci remet le programme à son début.
5 R/S	78,5398	Voici la réponse.

Création d'un programme

Nous continuerons à utiliser ce programme pour le calcul de la surface du cercle pour illustrer les concepts et les méthodes de programmation.

Limites des programmes (LBL et RTN)

Si vous désirez que plus d'un programme soit logé en mémoire de programme, il est nécessaire de marquer le début du programme d'un *label* (tel que A01 LBL A) et un *retour* (RTN, *return*) pour en marquer la fin (par exemple A05 RTN). Notez que les nombres de ligne acquièrent un A pour s'accorder avec leur label.

Labels de programme. Les programmes et les segments de programmes (nommés *sous-programmes*) doivent commencer par un label. Pour enregistre un label, appuyez sur :

■ LBL/RTN {LBL} *touche alphabétique*

Le label est utilisé comme identification d'un programme ou d'un sous-programme spécifique. Il consiste en une simple lettre, choisie de A à Z. Les touches alphabétiques sont utilisées de la même façon que pour les variables (Cf. chapitre 3). Vous ne pouvez assigner le même label plus d'une seule fois (si vous le faites, vous provoquerez l'apparition du message `DUPLICAT. LBL`), mais il est permis à un label d'utiliser la même lettre qu'une variable.

Il est possible d'avoir un programme sans label en mémoire. Mais les autres programmes ont besoin d'un label pour pouvoir être distingués.

Numéros des lignes de programme. Les numéros des lignes sont précédés de la lettre du label, par exemple `A01`. Si le sous-programme correspondant à un label comporte plus de 99 lignes, le numéro de ligne apparaît avec un point décimal au lieu du chiffre de gauche, comme dans `A.01` pour la ligne 101 de A. S'il compte plus de 199 lignes, le numéro de ligne utilise une virgule, comme dans `A,01` pour la ligne 201.

« Retours » de programmes. Les programmes et les sous-programmes doivent se terminer par une instruction de « retour ». Voici la séquence de frappe :

■ LBL/RTN {RTN}

Lorsqu'un programme s'achève, la dernière instruction RTN remet le pointeur de programme au début du programme (`PRGM TOP`).

Saisie de programme (PRGM)

Le fait d'appuyer sur ■ PRGM fait entrer et sortir le calculateur du mode de programme (le témoin **PRGM** apparaît et disparaît). Les frappes de touches effectuées pendant la programmation sont stockées en tant que lignes de programme en mémoire. Chaque nombre ou instruction occupe une ligne de programme et il n'y a pas de limite (autre que la capacité de la mémoire) au nombres de lignes que peut compter un programme.

Pour introduire un programme en mémoire :

1. Appuyez sur ■ PRGM pour la saisie du programme.

2. Appuyez sur ■ [GTO] [·][·] pour afficher `PRGM TOP`. Ceci place le *pointeur de programme* en un endroit connu, placé avant tout autre programme. Les lignes insérées le sont *avant* toutes les autres.

Si vous n'avez pas besoin d'autres programmes qui pourraient se trouver en mémoire, effacez la mémoire de programme en appuyant sur ■ [CLEAR] {PGM}. Pour confirmer que vous désirez effacer *tous* ces programmes, appuyez sur « oui », {Y} après le message `CL PGMS? Y N`.

3. Donnez un *label* au programme—n'importe quelle lettre de l'alphabet. Appuyez sur ■ [LBL/RTN] {LBL} *lettre*. Choisissez une lettre qui vous fera souvenir du programme, comme « s » pour « surface ».

4. Pour enregistrer des opérations du calculateur sous forme d'instructions de programme, appuyez sur les mêmes touches que si vous faisiez l'opération manuellement.

5. Terminez le programme par une instruction « *return* », qui replace le pointeur en haut du programme, `PRGM TOP`, après son exécution. Appuyez sur ■ [LBL/RTN] {RTN}.

6. Appuyez sur [C] (ou sur ■ [PRGM]) pour sortir de la saisie de programme.

Les nombres dans les lignes du programme seront affichés tels que vous les avez saisis, en utilisant le format ALL ou SCI (si les chiffres d'un nombre sont cachés,appuyez sur ■ [SHOW] pour les voir).

Entrées et sorties de données. Pour les programmes qui nécessitent plus d'une introduction de données ou qui produisent plus d'un résultat, il existe des instructions de programme qui sollicitent une variable spécifique (INPUT) et affichent une variable affectée d'un libellé (VIEW). Ces instructions sont traitées dans ce chapitre sous la rubrique « Entrées et sorties de données ».

[C], [◄] et ■ [CLEAR] {x} en saisie de programme. Notez ces fonctions spéciales, en mode de saise de programme :

- [C] annule toujours une saisie de programme, mais n'efface pas un nombre pour le mettre à zéro.
- [◄] supprime la ligne de programme en cours. Cette touche recule d'un espace si un chiffre est saisi (si le curseur est présent).
- Pour *programmer* une fonction qui efface le registre X, utilisez ■ [CLEAR] {x}.

Noms de fonctions dans les programmes. Le nom d'une fonction utilisée dans une ligne de programme n'est pas nécessairement le même que celui qui figure sur la touche du clavier ou dans son menu. Il représente en général une abréviation plus courte que celle utilisée sur la touche du clavier ou dans un libellé de menu. Le nom plus complet apparaît brièvement sur l'affichage lorsque la fonction est exécutée—si vous gardez quelque peu la pression sur la touche.

Exemple : saisie d'un programme identifié par un label. La séquence de touches qui suit détruit le programme précédent (la surface du cercle), et en saisit un nouveau, comportant une instruction de libellé et de retour. Si vous faites une erreur de saisie, appuyez sur ◀ pour supprimer la ligne de programme en cours, puis refaites votre saisie.

Touches :	Affichage :	Description :
■ PRGM		Active la saisie de programme (**PRGM** apparaît).
■ CLEAR {PGM} {Y}	PRGM TOP	Efface toute la mémoire de programme.
■ LBL/RTN {LBL} S	A01 LBL S	Donne le nom « S » à ce programme (*s* comme *surface*).*
■ x^2	A02 x^2	Saisit les trois lignes de programme.
■ π	A03 π	
×	A04 ×	
■ LBL/RTN {RTN}	A05 RTN	Termine le programme.
C		Termine la saisie de programme (le témoin **PRGM** disparaît).

* Si le message DUPLICAT, LBL apparaît, utilisez une autre lettre que « s » ou effacez le programme S existant.

Exécution d'un programme

Pour pouvoir *exécuter* un programme, il faut que le mode de saisie de programme ne soit pas actif (aucun numéro de ligne de programme n'est affiché, ni le témoin **PRGM**). Appuyez sur [C] et vous sortirez du mode de saisie de programme.

Exécution d'un programme avec XEQ

Appuyez sur [XEQ] *label* pour exécuter le programme qui correspond à ce label.* Le témoin **PRGM** clignote sur l'écran pendant l'exécution.

Si nécessaire, introduisez les données avant d'exécuter le programme.

Exemple. Exécutez le programme portant le label S pour trouver les surfaces de cercles de différents rayons, respectivement 5, 2,5 et 2π. Souvenez-vous d'introduire le rayon avant d'exécuter S.

Touches :	**Affichage :**	**Description :**
5 [XEQ] S	78,5398	Introduit le rayon, puis lance le programme S. Le résultat s'affiche.
2,5 [XEQ] S	19,6350	Calcule la surface du second cercle.
2 [■] [π] [×] [XEQ] S	124,0251	Calcule la surface du troisième cercle.

* S'il n'y a qu'un seul programme en mémoire, vous pouvez également l'exécuter en appuy ant sur [■] [GTO] [·] [·] [R/S] (*run/stop*).

Test d'un programme

Si vous savez qu'il existe une erreur dans un programme, mais que vous n'êtes pas absolument sûr de l'endroit où elle se trouve, vous pouvez exécuter le programme pas-à-pas pour le tester. Vous pouvez aussi tester un programme avant de l'exécuter, s'il est particulièrement long ou difficile. L'exécution pas-à-pas vous montre le résultat de chaque ligne de programme.

1. Pour l'exécution normale, asurez-vous que la saisie de programme n'est pas active (le témoin **PRGM** n'est *pas* affiché).
2. Appuyez sur [■] [GTO] *label* pour replacer le pointeur du programme en début de programme (au niveau de l'instruction LBL). L'instruction *go to* place le pointeur sans lancer l'exécution (si le programme est le premier ou le seul en mémoire, vous pouvez appuyer sur [■] [GTO] [·] [·] pour vous placer à son début).
3. Appuyez sur [▼] et maintenez la pression. La ligne de programme en cours s'affiche. Lorsque vous relâchez [▼], la ligne de programme est exécutée. Le résultat de l'exécution est alors affiché (il se trouve dans le registre X).

   Pour passer à la *ligne précédente,* appuyez sur [■] [▲]. L'exécution ne se fait pas.
4. Le pointeur de programme passe à la ligne suivante. Répétez le point 3 jusqu'à ce que vous trouviez une erreur (un résultat incorrect) ou que vous arriviez à la fin du programme.

Si le mode de saisie de programme est actif, [▼] (ou [■] [▲]) change la place du pointeur de programme, sans exécution des lignes du programme. Si vous maintenez la pression sur une touche-flèche en mode de saisie de programme, toutes les lignes de programme défilent.

Exemple : test d'un programme. Exécutez le programme S pas-à-pas. Utilisez un rayon de 5 comme valeur-test. Vérifiez avant de commencer que la saisie de programme n'est pas active.

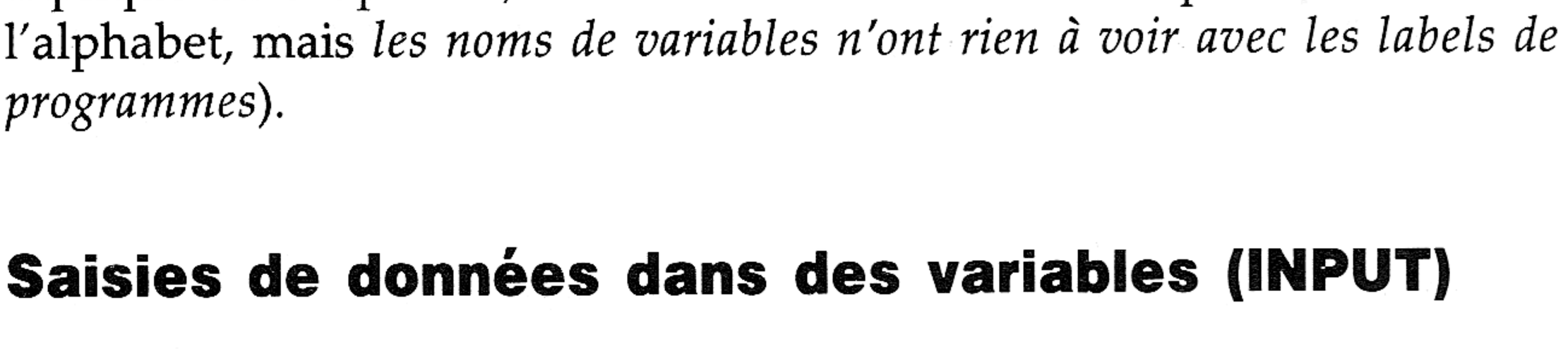
Touches :	Affichage :	Description :
5	5_	
■ GTO S	5,0000	Amène le compteur sur le programme S.
▼ *(maintenez)*	A01 LBL S	
(relâchez)	5,0000	
▼ *(maintenez)*	A02 x^2	Met la saisie au carré.
(relâchez)	25,0000	
▼ *(maintenez)*	A03 π	Valeur de π.
(relâchez)	3,1416	
▼ *(maintenez)*	A04 ×	25π.
(relâchez)	78,5398	
▼ *(maintenez)*	A05 RTN	Fin du programme ; le résultat est correct.
(relâchez)	78,5398	

Entrées et sorties de données

Les *variables* du calculateur sont utilisées pour stocker les saisies de données, les résultats intermédiaires et les résultats finaux (comme expliqué au chapitre 3, les variables sont identifiées par une lettre de l'alphabet, mais *les noms de variables n'ont rien à voir avec les labels de programmes*).

Saisies de données dans des variables (INPUT)

L'instruction INPUT (■ INPUT *variable*) arrête un programme en cours d'exécution et affiche un message de sollicitation pour la variable donnée. Cet affichage comporte la valeur existante de la variable, par exemple



Appuyez sur [R/S] (*run/stop*) ou sur [▼] pour reprendre le programme. La valeur que vous avez introduite écrase le contenu du registre X *et* est stockée dans la variable. Si vous n'avez pas changé la valeur affichée, elle est conservée dans le registre X.

Le programme de la surface du cercle avec l'instruction INPUT a l'aspect suivant :

```
S01 LBL S
S02 INPUT R
S03 x²
S04 π
S05 ×
S06 RTN
```

Utilisation de INPUT dans un programme.

1. Décidez des valeurs dont vous avez besoin et assignez-leur des noms (dans notre exemple, le rayon, *R*).
2. Au début du programme, insérez une instruction INPUT pour chacune des variables dont vous aurez besoin. Plus loin dans le programme, lorsque vous écrirez la partie du calcul qui nécessite une certaine valeur, insérez une instruction [RCL] *variable* pour ramener cette valeur dans la pile.*

Par exemple : voyez le programme « Calcul sur des flux constants », en page 222 de la quatrième partie. La première chose que fait le programme F est d'obtenir les saisies nécessaires pour remplir les variables *N, I, S, P* et *F* (lignes T02 à T06).

* Puisque l'instruction INPUT laisse également la valeur que vous venez de saisir dans le registre X, vous n'êtes pas *obligé* de rappeler la variable à un autre moment—vous pouvez l'introduire au clavier et l'utiliser lorsque vous en aurez besoin. Vous économiserez peut-être un peu d'espace mémoire de cette façon. Cependant, pour un long programme, il est plus simple de saisir toutes les données d'emblée et de les rappeler ensuite individuellement au fur et à mesure des besoins.

  Souvenez-vous aussi que l'utilisateur d'un programme peut effectuer des calculs alors que le programme est arrêté, en attente de saisies. Ceci peut modifier le contenu de la pile, ce qui modifiera le calcul suivant que le programme effectuera. Un programme ne devrait donc jamais supposer que le contenu des registres X, Y et Z seront les mêmes avant et après une instruction INPUT. Si vous obtenez toutes les données en début de programme et les rappelez ensuite lorsqu'elles sont nécessaires au calcul, vous évitez que le contenu de la pile soit modifié juste avant un calcul.

Effet de INPUT lors de l'exécution. Le programme s'arrête à chaque instruction INPUT, le témoin **INPUT** apparaît et le calculateur sollicite une variable, dans notre exemple `R?0,0000`. La valeur affichée (présente dans le registre *X*) sera le contenu de *R*.

- **Pour changer ce nombre,** frappez le nouveau nombre et appuyez sur la touche [R/S].* S'il est nécessaire de calculer ce nombre, vous pouvez le faire avant d'appuyer sur [R/S].
- **Pour le laisser inchangé,** appuyez simplement sur [R/S].
- **Pour utiliser le nombre affiché dans des calculs,** appuyez sur [ENTER] avant de frapper un autre nombre.
- **Pour faire disparaître le témoin INPUT,** appuyez sur [C].† La valeur en cours de la variable reste dans le registre X. Si vous appuyez sur [R/S] pour reprendre le programme, le témoin réapparaît.
- **Pour effacer les chiffres cachés par le message de sollicitation,** appuyez sur ■[SHOW] (si c'est un nombre binaire de plus de 12 chiffres, utilisez [√x] et [Σ+] pour voir le reste).

Affichage des données des variables (VIEW)

L'instruction programmée VIEW (■[VIEW] *variable*) arrête un programme en cours d'exécution et affiche et identifie le contenu d'une variable donnée, comme par exemple

```
A=78,5398
```

Ceci est seulement un *affichage* et n'affecte aucunement le registre X.

- Le fait d'appuyer sur [ENTER] copie ce nombre dans le registre X.
- Si le nombre compte plus de 10 chiffres, le fait d'appuyer sur ■[SHOW] affiche le nombre en sa totalité (si c'est un nombre binaire de plus de 12 chiffres, utilisez [√x] et [Σ+] pour voir le reste).
- Le fait d'appuyer sur [C] (ou sur [←]) efface l'affichage VIEW et montre le registre X.

* Ce nombre « écrase » l'ancienne valeur dans le registre X.

† Si vous appuyez sur [C] durant la saisie, le nombre est réduit à zéro. Appuyez à nouveau sur [C] pour annuler le témoin INPUT.

- Le fait d'appuyer sur ■ CLEAR efface le contenu de la variable affichée.

Appuyez sur R/S pour continuer le programme.

Par exemple : Voyez le programme écrit pour « Solutions d'équations simultanées—Méthode du déterminant », en page 175, 4e partie. Les lignes S24 à S29, à la fin du sous-programme S, affichent les résultats pour *X*, *Y* et Z. Notez aussi que chaque expression VIEW de ce programme—comme dans tous les programmes d'application—est précédée d'une instruction RCL. Celle-ci n'est pas essentielle, mais souhaitable parce qu'elle amène la variable « vue » dans le registre *X*, la rendant accessible aux calculs manuels (le fait d'appuyer sur ENTER pendant la visualisation d'un affichage VIEW aurait le même effet).

Exemple : saisie (par INPUT) et visualisation (par VIEW) dans un programme. Ecrit une équation pour trouver la surface et le volume d'un cylinde, connaissant son rayon et sa hauteur. Nommez le programme C (comme *cylindre*) et utilisez les variables *S* (surface) *V* (volume), *R* (rayon) et *H* (hauteur). Voici les formules :

$$V = \pi R^2 H$$
$$S = 2\pi R^2 + 2\pi RH = 2\ (\pi R^2 + \pi RH).$$

Touches :	Affichage :	Description :
■ PRGM		Entrée en mode de
■ GTO · ·	PRGM TOP	programmation ; le pointeur se place au début de la mémoire.
■ LBL/RTN {LBL} C		Label du programme.
(C est la touche LN)	C01 LBL C	
■ INPUT R	C02 INPUT R	Instructions qui
■ INPUT H	C03 INPUT H	sollicitent les valeurs
RCL R	C04 RCL R	du rayon et de la
■ x^2	C05 x^2	hauteur, pour calculer
RCL × H	C06 RCL× H	le volume et le stocker
■ π	C07 π	dans *V*.
×	C08 ×	
STO V	C09 STO V	
RCL ÷ H	C10 RCL÷ H	Convertit $\pi R^2 H$ en πR^2.

RCL R	C11 RCL R	
RCL × H	C12 RCL× H	
■ π	C13 π	
×	C14 ×	Calcule πRH.
+	C15 +	Calcule $(\pi R^2 + \pi RH)$.
2	C16 2_	Calcule 2 $(\pi R^2 + \pi RH)$
×	C17 ×	et stocke la surface
STO S	C18 STO S	dans *S*.
■ VIEW V	C19 VIEW V	Ceci affichera le volu-
■ VIEW S	C20 VIEW S	me et la surface.
■ LBL/RTN {RTN}	C21 RTN	Fin du programme.
C		Annule la saisie de programme.
■ MEM {PGM}	LBL C 031,5	Vérifie l'utilisation de
■ SHOW (*maintenue*)	CHKSUM=4602	la mémoire et vérifie par total de contrôle. Un total différent signifie que le programme n'a pas été saisi comme indiqué.

Trouvez maintenant le volume et la surface d'un cylindre d'un rayon de 2,5 cm et d'une hauteur de 8 cm.

Touches :	**Affichage :**	**Description :**
XEQ C (C est la touche LN)	R?0,0000	Commence l'exécution de C ; sollicite une valeur pour *R* (il affiche la valeur qui se trouve dans *R* à ce moment).
2,5 R/S	H?0,0000	Sollicite une valeur pour *H* (et affiche la valeur présente dans *H*).
8 R/S	V=157,0796	Donne le résultat, le volume en cm^3.
R/S	S=164,9336	Donne la surface en cm^2.

Arrêt ou interruption de programme

Programmation d'un arrêt ou d'une pause (STOP, PSE)

- Le fait d'appuyer sur [R/S] (*run/stop*) pendant la saisie de programme insère une instruction STOP ; ceci arrêtera un programme en cours d'exécution jusqu'à ce que l'on appuie à nouveau sur [R/S]. STOP peut être utilisé plutôt que RTN pour arrêter un programme sans ramener le pointeur en haut de la mémoire de programme.
- Le fait d'appuyer sur ■ [LBL/RTN] {PSE} pendant la saisie de programme insère une instruction PSE (*pause*). Ceci suspend l'exécution du programme pendant environ une seconde et affiche le contenu du registre X.

Interruption d'un programme en cours d'exécution

Vous pouvez interrompre un programme à tout moment en appuyant sur [C] ou sur [R/S]. Le programme termine l'instruction en cours avant de s'arrêter. Appuyez sur [R/S] (*run/stop*) pour reprendre l'exécution.

Si vous interrompez un programme et appuyez sur [XEQ], sur [GTO], ou sur {RTN}, vous serez *incapable* de reprendre l'exécution avec [R/S]. Recommencez l'exécution ([XEQ] *label*).

Arrêts causés par des erreurs

Si le programme rencontre une erreur pendant son exécution, celle-ci s'arrête et un message d'erreur apparaît (vous en trouverez une liste en fin de manuel, juste avant l'index).

Pour visualiser la ligne du programme qui contient l'erreur, appuyez sur ■ [PRGM]. Le programme se sera arrêté à cet endroit précis (il se peut par exemple que la cause de l'erreur soit une fonction ÷ qui a provoqué une division par zéro).

Correction d'un programme

Un programme présent en mémoire de programme peut être modifié par insertion ou suppression de ses lignes. Même si une ligne de programme ne nécessite qu'un changement minime, il faut la supprimer et en insérer une nouvelle qui la remplace.

Pour supprimer une ligne de programme :

1. Choisissez le programme ou sous-programme concerné ([■] [GTO] *label*), activez la saisie de programme ([■] [PRGM]) et appuyez sur [▼] ou sur [■] [▲] pour trouver la ligne qui doit être modifiée. Maintenez la pression sur la flèche pour continuer le défilement (si vous connaissez le numéro de la ligne qu'il faut corriger, appuyez sur [■] [GTO] [·] *label nn* pour y mettre le pointeur de programme).
2. Supprimez la ligne en appuyant sur [◄]. Le pointeur passe à la ligne *précédente* (si vous supprimez plus d'une ligne, commencez par la *dernière* ligne du groupe).
3. Frappez la nouvelle instruction, s'il y a lieu.
4. Sortez du mode de saisie de programme (par [C] ou par [■] [PRGM]).

Pour insérer une ligne de programme : trouvez et affichez la ligne de programme qui se trouve *avant* l'endroit où vious désirez insérer une ligne. Frappez la nouvelle instruction ; elle s'insère *après* la ligne affichée à ce moment-là.

Si vous désirez par exemple insérer une nouvelle ligne entre les lignes A04 et A05, vous afficherez d'abord la ligne A04, puis frapperez la ou les instructions. Les lignes suivantes, à commencer par la ligne qui s'appelait avant cela A05, sont *descendues* et renumérotées comme il convient.

Mémoire de programme

Examen de la mémoire de programme

Le fait d'appuyer sur ■ PRGM fait entrer et sortir le calculateur du mode de saisie de programme (le témoin **PRGM** apparaît, les lignes du programme s'affichent).

La mémoire de programme commence au début (PRGM TOP). La liste des lignes de programme est circulaire, vous pouvez donc passer de la fin au début et inversément. Lorsque la saisie de programme est active, il y a trois façons de changer le pointeur de programme (la ligne affichée).

- Utilisez les touches flèches ▼ et ■ ▲. Le fait d'appuyer sur ▼ à la dernière ligne fait revenir à PRGM TOP, alors que l'appui sur ■ ▲ au niveau de PRGM TOP fait revenir à la dernière ligne du programme.

  Pour déplacer plus d'une ligne à la fois (pour faire « défiler » les lignes, continuez à appuyer sur les touches ▼ ou ▲).
- Appuyez sur GTO · · pour placer le pointeur au PRGM TOP.
- Appuyez sur GTO · *label nn* pour passer à une ligne (dont le numéro est < 100).

Si la saisie de programme n'est pas active (aucune ligne n'est affichée), vous pouvez aussi déplacer le pointeur en appuyant sur ■ GTO *label*.

L'annulation de la saisie de programme ne change *pas* la position du pointeur de programme.

Utilisation de la mémoire

Chaque ligne de programme utilise de 1,5 à 9,5 octets.

- Les nombres occupent 9,5 octets, *sauf* les nombres entiers de 1 à 99 qui n'utilisent que 1,5 octet.
- Toutes les autres instructions occupent 1,5 octet.

Si pendant la saisie de programme, vous rencontrez le message MEMORY FULL, il n'y a plus assez d'espace en mémoire-programme pour la ligne que vous essayez de saisir. Voyez « Effacer un ou plusieurs programmes », ci-dessous, ou « Gestion de la mémoire du calculateur », à l'annexe B.

Le catalogue des programmes (MEM)

C'est une liste de tous les labels de programmes, avec le nombre d'octets qu'utilise chaque label et les lignes qui y sont associées. Appuyez sur ■ [MEM] {PGM} pour afficher le catalogue, et appuyez sur [▼] ou sur ■ [▲] pour vous déplacer à l'intérieur de la liste. Vous pouvez utiliser ce catalogue pour :

- Revoir les libellés présents en mémoire et le « poids » en octets de chaque programme muni d'un label, ou de chaque sous-programme.
- Exécuter un programme muni d'un label (appuyez sur [XEQ] ou sur [R/S] lorsque le label est affiché).
- Afficher un programme affecté d'un label (apuyez sur ■ [PRGM] lorsque le label est affiché).
- Effacer certains programmes (appuyez sur ■ [CLEAR] lorsque le label est affiché).
- Voir la vérification par total de contrôle correspondant à un segment de programme donné (appuyez sur ■ [SHOW]).

Le catalogue vous montre l'utilisation que fait de la mémoire chaque sgement de programme portant un label :

```
LBL C 031,5
```

Nombre d'octets utilisés par le programme C.

Effacer un ou plusieurs programmes

Pour effacer (ou supprimer de la mémoire) un programme précis :

1. Appuyez sur ■ [MEM] {PGM} et affichez (grâce à [▼] et à ■ [▲]) le label de ce programme.

2. Appuyez sur ■ CLEAR.
3. Appuyez sur C pour annuler le catalogue ou sur ◄ pour revenir en arrière.

Pour effacer tous les programmes de la mémoire :

1. Appuyez sur ■ PRGM pour afficher les lignes de programme (**PRGM** apparaît).
2. Appuyez sur ■ CLEAR {PGM} pour effacer la mémoire de programme.
3. Le message CL PGMS? Y N vous demande confirmation. Appuyez sur {Y}.
4. Appuyez sur ■ PRGM pour annuler la saisie de programme.

Le total de contrôle (Angl. Checksum)

C'est une valeur hexadécimale unique donnée à chaque label de programme et aux lignes qui y sont associées. Ce nombre permet une comparaison avec un total de contrôle connu introduit en mémoire de programme. Si le total connu et celui qu'affiche votre calculateur sont identiques, vous avez correctement saisi toutes les lignes du programme. Pour voir le total de contrôle :

1. Appuyez sur ■ MEM {PGM} pour le catalogue des labels.
2. Affichez le label en utilisant les touches flèches si nécessaire.
3. Appuyez et maintenez la pression sur ■ SHOW pour afficher CHKSUM=*valeur*.

Par exemple, pour voir le total de contrôle du programme en cours :

Touches :	**Affichage :**	**Description :**
■ MEM {PGM}	LBL C 031,5	Affcihe le label C, qui occupe 31,5 octets.
■ SHOW *(maintenez)*	CHKSUM=4602	Si *votre* total de contrôle ne correspond pas à ce nombre, vous n'avez pas saisi le programme correctement.

Vous verrez que tous les programmes de contrôle fournis dans la quatrième partie comportent des totaux de contrôle CHKSUM correspondant à chaque programme portant un label, pour vous permettre de vérifier la saisie.

Fonctions non programmables

Les instructions du HP-32S suivantes ne sont pas programmables :

- [■] [CLEAR] {PGM}
- [■] [CLEAR] {ALL}
- [◀]
- [▼], [■] [▲]
- [■] [PRGM]
- [■] [GTO] [·] [·]
- [■] [GTO] [·] *label nn*
- [■] [MEM]
- [■] [SHOW]

Expressions polynomiales et méthode de Horner

Certaines expressions, comme les polynômes, utilisent les mêmes variables à plusieurs reprises pour arriver à leurs solutions. L'expression

$$f(x) = Ax^4 + Bx^3 + Cx^2 + Dx + E$$

par exemple, utilise la variable x à quatre reprises. Un programme qui résoudrait cette équation rappellerait sans cesse une copie stockée de x d'une variable. Une méthode plus économe, cependant, serait d'utiliser une pile remplie d'une constante (Cf. « Remplissage de la pile par une constante » en page 39 du chapitre 2).

La méthode de Horner est un moyen utile de réarranger les expressions polynomiales pour abréger les étapes et le temps d'exécution du calcul. Elle est particulièrement rapide avec SOLVE et ∫FN, deux opérations complexes qui utilisent des sous-programmes.

Cette méthode implique de ré-écrire l'expression de façon emboîtée pour éliminer les exposants plus grands que 1 :

$$Ax^4 + Bx^3 + Cx^2 + Dx + E$$

$$(Ax^3 + Bx^2 + Cx + D)x + E$$

$$((Ax^2 + Bx + C)x + D)x + E$$

$$(((Ax + B)x + C)x + D)x + E$$

Exemple. Ecrivez un programme correspondant à $5x^4 + 2x^3$ as $(((5x + 2)x)x)x$, puis évaluez-le pour $x = 7$.

Touches :	**Affichage :**	**Description :**
■ PRGM		Vous pouvez sauter le
■ GTO · ·	PRGM TOP	GTO si l'affichage affiche déjà PRGM TOP.
■ LBL/RTN {LBL} P	P01 LBL P	
■ INPUT X	P02 INPUT X	Remplit la pile de x,
ENTER	P03 ENTER	puis calcule $5x$.
ENTER	P04 ENTER	
ENTER	P05 ENTER	
5	P06 5	
×	P07 ×	
2	P08 2	
+	P09 +	
×	P10 ×	$(5x + 2)x$.
×	P11 ×	
×	P12 ×	$(5x + 2)x^3$.
■ LBL/RTN {RTN}	P13 RTN	
C		Annule la saisie de programme.

Evaluez maintenant ce polynôme pour $x = 7$.

XEQ P	X?*valeur*	Sollicite pour x.
7 R/S	12.691,0000	Résultat.

Une forme plus générale de ce programme pour n'importe quelle équation $(((Ax + B)x + C)x + D)x + E$ serait celle-ci :

```
P01 LBL P
P02 INPUT A
P03 INPUT B
P04 INPUT C
P05 INPUT D
P06 INPUT E
P07 INPUT X
P08 ENTER
P09 ENTER
P10 ENTER
P11 RCL× A
P12 RCL+ B
P13 ×
P14 RCL+ C
P15 ×
P16 RCL+ D
P17 ×
P18 RCL+ E
P19 RTN
```

6

Techniques de programmation

Le chapitre 5 traitait de la programmation de base. Ce chapitre se consacre à des techniques plus sophistiquées mais plus utiles :

- L'utilisation d'un sous-programme pour simplifier les programmes en séparant et en affectant des labels à des parties du programme traitant de tâches particulières. L'utilisation de sous-programmes raccourcit également tout programme qui doit répéter l'exécution d'une ou plusieurs parties.
- L'utilisation d'instructions conditionnelles (comparaisons et indicateurs) pour déterminer quelle instruction ou sous-programme devrait être utilisé dans un cas particulier.
- L'utilisation de boucles avec compteurs pour exécuter un certain nombre d'instructions un certain nombre de fois.
- Enfin, l'utilisation de l'adressage indirect pour accéder à différentes variables utilisant la même instruction de programme.

Sous-programmes à l'intérieur des programmes

Un programme se compose d'un ou de plusieurs *sous-programmes*. Un sous-programme est une unité fonctionnelle qui a un rôle précis. Les programmes complexes nécessitent des sous-programmes qui regroupent et séparent les tâches. Ils rendent un programme plus aisé à écrire, à lire, à comprendre—et à modifier.

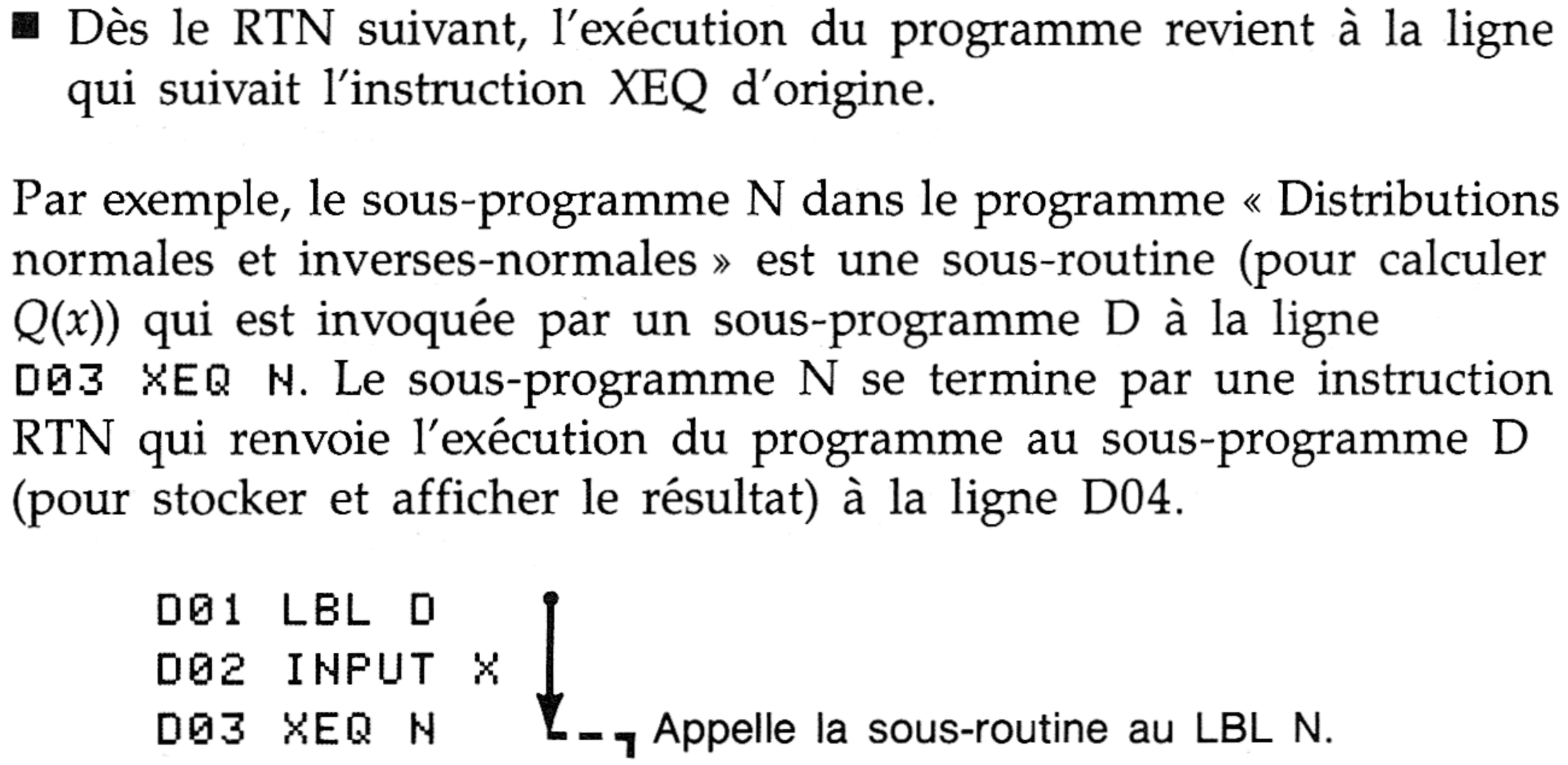
Voyez par exemple le programme écrit pour les « Distributions normales et inverses-normales » en page 215 de la quatrième partie. Ce programme compte quatre sous-programmes, dont les labels sont S, D, N et F. Le sous-programme S « initialise » le programme en rassemblant les données pour la moyenne et l'écart-type. Le sous-programme D détermine les limites de l'intégration, exécute le sous-programme N et affiche le résultat. Le sous-programme N intègre la fonction définie dans le sous-programme F et termine le calcul de probabilité de $Q(x)$.

Appel de sous-routines (XEQ, RTN)

Nous appellerons ici *sous-routine* un sous-programme qui a été *invoqué, appelé* (exécuté) par un autre sous-programme et y renvoie lorsque la sous-routine est terminée. La sous-routine *doit* commencer par un label et se terminer par une instruction RTN. Une sous-routine est essentiellement un sous-programme et peut appeler d'autres sous-routines.

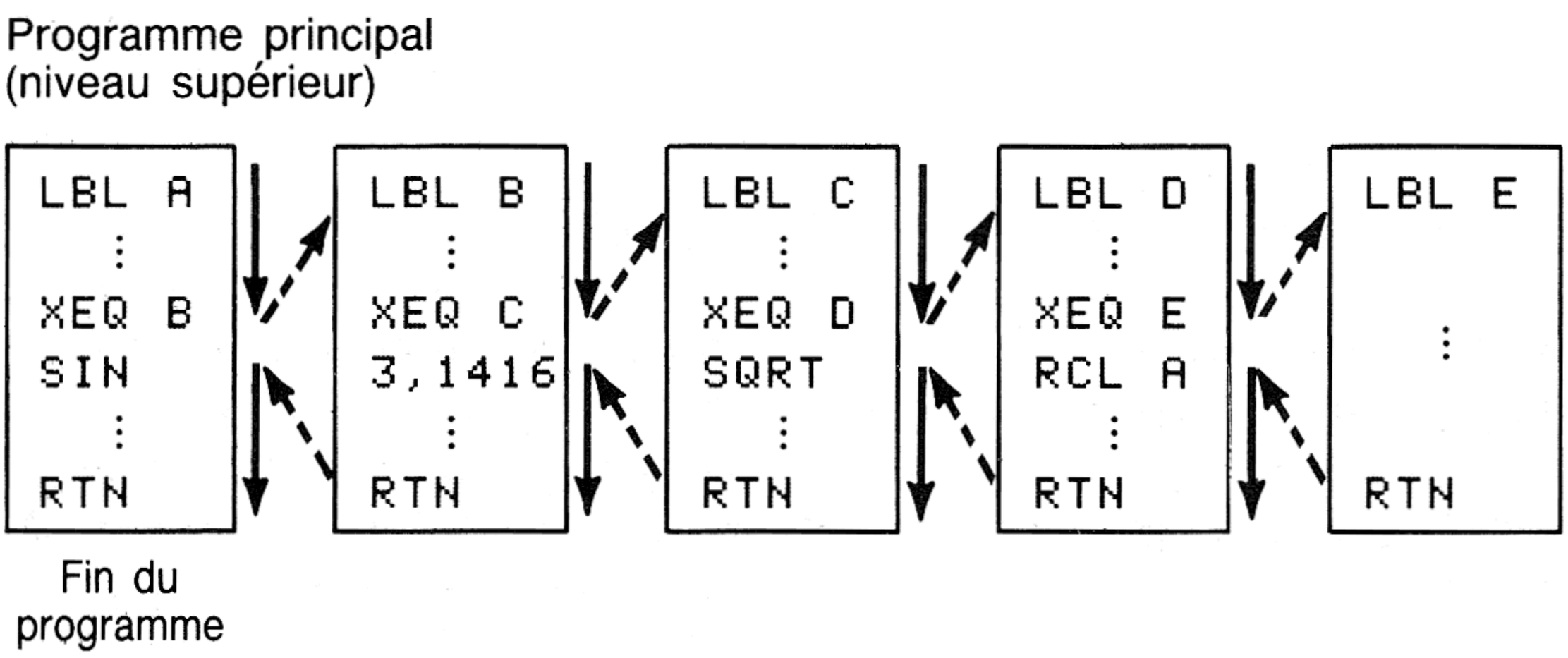
- XEQ doit se brancher à un label (LBL) correspondant à la sous-routine (elle ne peut se brancher à un numéro de ligne).
- Dès le RTN suivant, l'exécution du programme revient à la ligne qui suivait l'instruction XEQ d'origine.

Par exemple, le sous-programme N dans le programme « Distributions normales et inverses-normales » est une sous-routine (pour calculer $Q(x)$) qui est invoquée par un sous-programme D à la ligne `D03 XEQ N`. Le sous-programme N se termine par une instruction RTN qui renvoie l'exécution du programme au sous-programme D (pour stocker et afficher le résultat) à la ligne D04.



Sous-routines emboîtées

Une sous-routine peut en invoquer une autre, et celle-ci à sont tour peut en appeler une autre. Cet « emboîtage » est limité à sept sous-routines (sans compter le programme principal). Voici comment fonctionnent plusieurs sous-routines emboîtées :

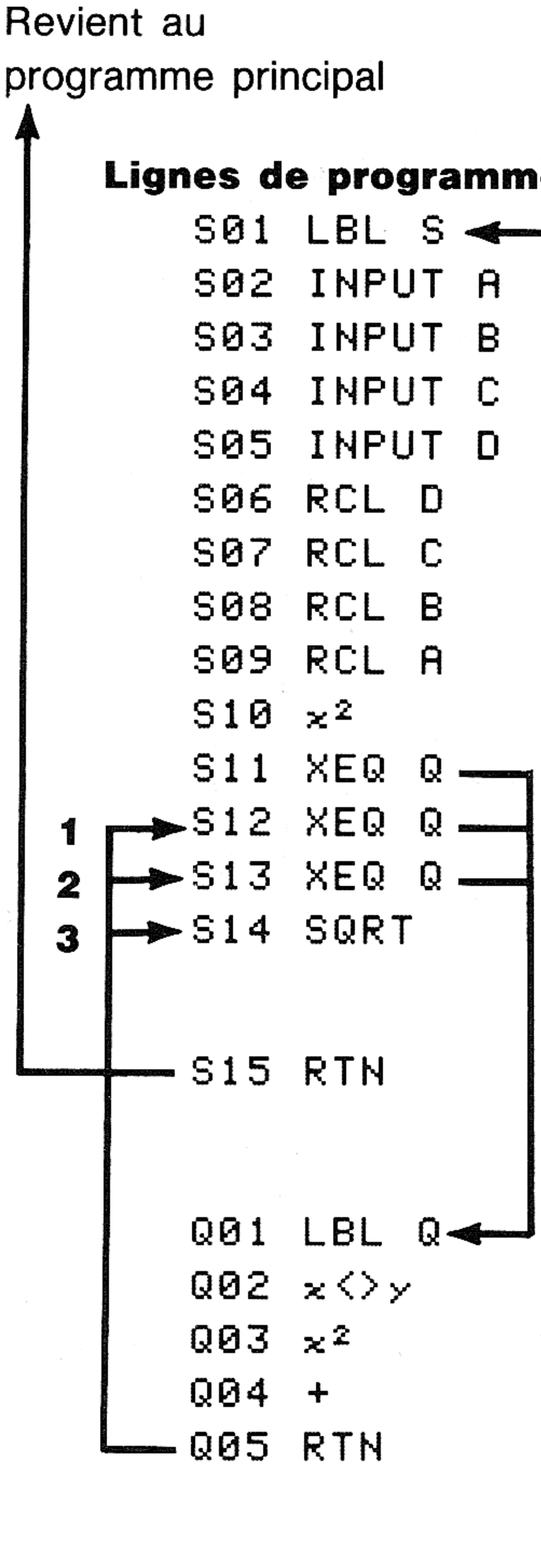


Toute tentative d'exécuter une sous-routine placée plus « profondément » qu'à sept niveaux de profondeur cause une erreur `XEQ OVERFLOW`.

Exemple de sous-routine emboîtée. Cette sous-routine, de label S, calcule la valeur de l'expression

$$\sqrt{a^2 + b^2 + c^2 + d^2}$$

qui est une partie d'une expression plus complexe, dans un programme plus vaste. La sous-routine en appelle une *autre* (qui est donc une *sous-routine emboîtée*), de label Q, qui effectue mises au carré et additions, toutes opérations répétitives. Ceci permet de conserver une programme relativement plus court que si ce système n'était pas utilisé :



Description :

Lance la sous-routine principale.
Introduit A.
Introduit B.
Introduit C.
Introduit D.
Rappelle les données pour le calcul suivant.

Calcule A^2.
Calcule B^2, puis $A^2 + B^2$.
Calcule $A^2 + B^2 + C^2$.
Calcule $A^2 + B^2 + C^2 + D^2$.
Calcule

$$\sqrt{A^2 + B^2 + C^2 + D^2}$$

Termine la sous-routine principale ; reprend l'exécution au programme principal.
Lance la sous-routine emboîtée.
Met le nombre au carré et l'ajoute à l'actuelle somme des carrés.
Termine la sous-routine, Q ; renvoie vers la première sous-routine, S.

Branchements (GTO)

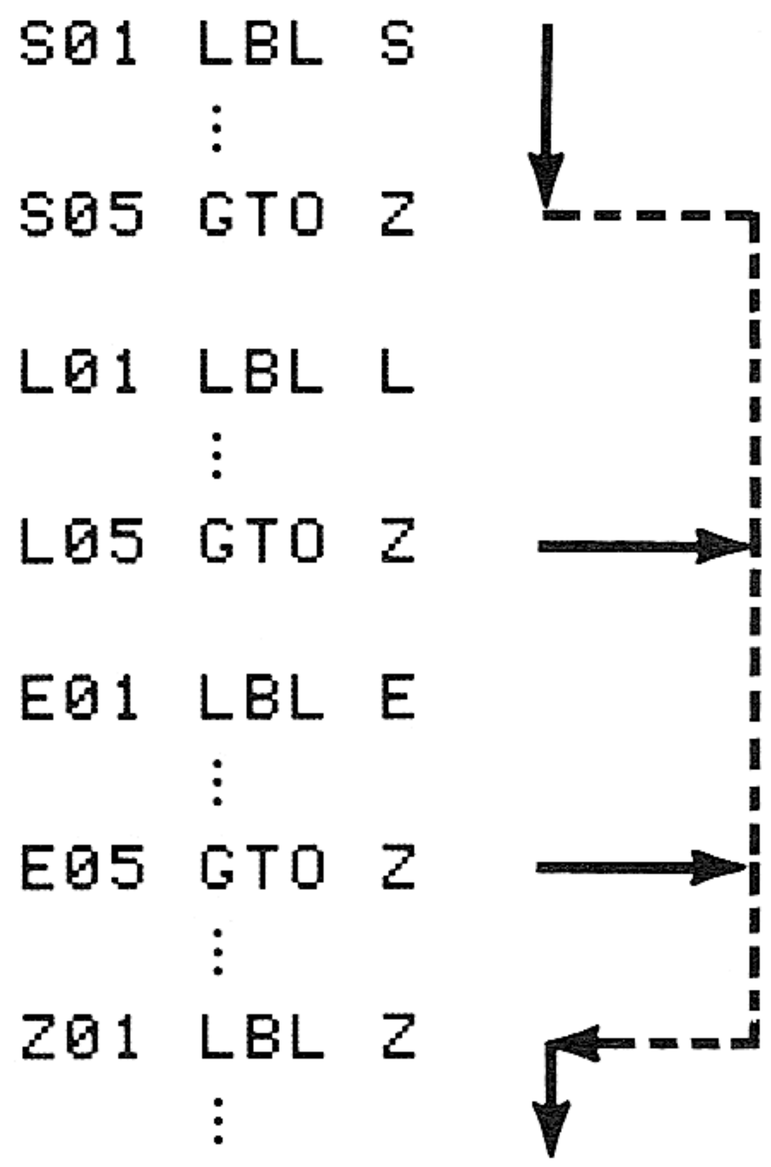
Comme nous venons de le voir avec les sous-routines, il est souvent souhaitable de transférer l'exécution à une partie du programme autre que la ligne suivante. Cette opération est appelée un *branchement*.

Le branchement inconditionnel utilise l'instruction GTO (*go to, aller vers*) pour aller se « brancher » à un *label* de programme. Il n'est pas possible de se brancher à un numéro de ligne spécifique pendant l'exécution d'un programme.

■ GTO *label*

Une instruction GTO programmée. L'instruction GTO *label* transfère l'exécution d'un programme en train de tourner vers la ligne du programme qui contient ce label, où qu'elle se trouve.

Voyez le programme « d'ajustement de courbes », page 204.
L'instruction `GTO Z` branche l'exécution de l'un de trois sous-programmes d'initialisation indépendants vers LBL Z, le sous-programme qui est le point d'entrée commun au cœur du programme :



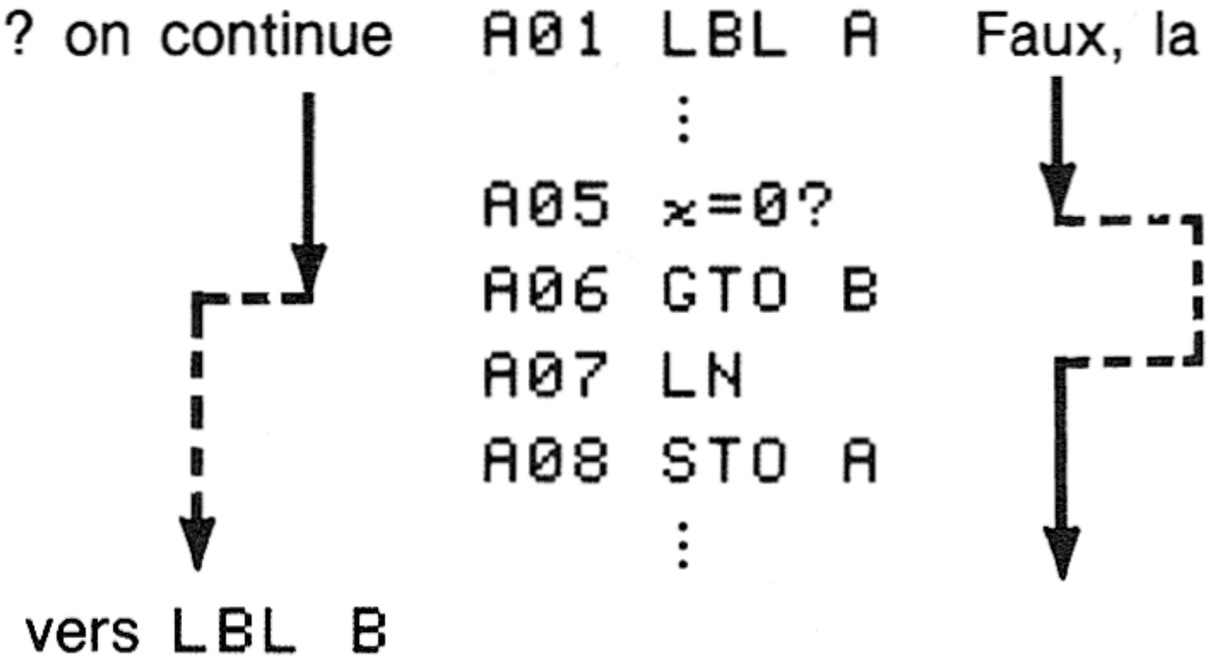
Utilisation de ■ GTO au clavier. Si la saisie de programme n'est *pas* active (aucune ligne de programme n'est affichée et **PRGM** est éteint), vous pouvez utiliser ■ GTO pour déplacer le pointeur de programme vers un libellé ou un numéro de ligne *sans* lancer pour autant l'exécution du programme :

- Vers un label : ■ GTO *label* (Exemple : ■ GTO A .)
- Vers un numéro de ligne : ■ GTO · *label nn* (*nn* < 100. Exemple : ■ GTO · A05.)
- Vers le début du programme, `PRGM TOP` : ■ GTO · ·

Instructions conditionnelles

Une autre façon de modifier la séquence d'exécution d'un programme est l'utilisation d'un *test conditionnel,* un test de type vrai/faux qui compare deux nombres et saute l'instruction suivante si la proposition est fausse.

Par exemple, si la ligne A05 comporte une instruction conditionnelle `x=0?` (*x est-il égal à zéro ?*), le programme compare le contenu du registre X avec zéro. Si le registre X contient effectivement zéro, le programme passe à la ligne suivante. S'il ne contient *pas* zéro, le programme *saute* la ligne suivante, se branchant ainsi à la ligne A07.



L'exemple ci-dessus montre une technique habituelle pour les tests conditionnels : la ligne qui suit immédiatement le test est un branchement vers un autre label. L'effet est donc de passer à un autre sous-programme dans certaines circonstances.

Il y a trois catégories d'instructions conditionnelles :

- Les tests de comparaison. Ils comparent les registres X et Y ou le registre X et zéro.
- Les test d'indicateurs. Ceux-ci testent l'état des indicateurs, qui est soit armé soit désarmé.
- Les compteurs de boucles. Ils sont utilisés pour effectuer une boucle un certain nombre de fois.

Tests de comparison (TESTS)

Huit comparaisons sont disponibles pour la programmation dans le menu TESTS. Appuyez sur ■ TESTS pour afficher les deux catégories de tests :

x?y

Tests de comparaison de *x* et *y*.

x?0

Tests de comparaison de *x* et 0.

Souvenez-vous que *x* réfère au nombre qui se trouve dans le registre X et *y* à celui du registre Y. Ils ne comparent pas les *variables* X et Y.

Choisissez la catégorie de comparaison puis appuyez sur la touche de menu correspondant à l'instruction que vous désirez :

Les touches du menu TESTS

{x?y}	{x?0}
{≠y} pour x≠y?	{≠0} pour x≠0?
{<y} pour x<y?	{<0} pour x<0?
{>y} pour x>y?	{>0} pour x>0?
{=y} pour x=y?	{=0} pour x=0?

Bien qu'il soit possible d'afficher ces menus en dehors de la saisie de programme, ces fonctions n'ont pas d'utilité en dehors des programmes.

Exemple : Le programme « Equation du second degré » en page 191, utilise les tests x=0? et x<0? dans le sous-programme Q.

```
Q01 LBL Q
Q02 INPUT A
Q03 x=0?          Vérifie A, qui ne peut être égal à zéro.
Q04 GTO Q         Si A = 0, le programme recommence.
Q05 INPUT B       Si A ≠ 0, le programme continue.
      ⋮
```

Les lignes Q14 à Q19 calculent $B^2 - 4AC$. Les lignes suivantes testent pour une valeur négative (qui produirait une racine imaginaire).

```
      :
Q20 x<0?          Le résultat est-il négatif ?
Q21 GTO I         Si oui, branchement vers un autre sous-
                  programme.
Q22 SQRT          Si le résultat est positif, extraire la racine
                  carrée.
      :
```

Indicateurs

Ils indiquent un *état,* une situation. Ils sont *armés* (vrais) ou *désarmés* (faux). Le test d'un indicateur suit la même règle que l'autre test conditionnel : si l'indicateur testé est armé, l'exécution se poursuit directement, s'il est désarmé, elle saute une ligne.

Signification des indicateurs. Le HP-32S possède sept indicateurs, numérotés de 0 à 6. Tous peuvent être armés, désarmés et testés par instruction de programme, comme ils peuvent l'être depuis le clavier.*

- Les indicateurs 0, 1, 2, 3 et 4 n'ont pas de signification prédéterminée. Leur statut sera ce que vous voudrez dans un programme donné (voyez l'exemple en page 99).
- L'indicateur 5, lorsqu'il est armé, interrompt le programme lorsqu'un dépassement de capacité se produit, avec affichage de `OVERFLOW` et de ⚠.† S'il est désarmé, un programme avec dépassement de capacité ne s'interrompt pas, bien que `OVERFLOW` s'affiche brièvement lorsque le programme finit par s'arrêter.
- L'indicateur 6 est *automatiquement* armé chaque fois qu'un dépassement de capacité se produit (mais il peut aussi est armé par vous-même). Il n'a pas d'effet, mais peut être testé.

* La seule autre opération qui désarme les indicateurs est l'opération d'effacement total de la mémoire par la séquence de trois touches décrite à l'annexe B.

† Un *dépassement de capacité* se produit lorsqu'un résultat dépasse le nombre le plus grand que le calculateur puisse traiter, et c'est ce nombre qui est substitué au véritable résultat de l'opération.

Les indicateurs 5 et 6 permettent le contrôle des situations de dépassement de capacité lors de l'exécution d'un programme. Armé, indicateur 5 arrête un programme à la ligne qui suit immédiatement celle qui a causé le dépassement. Indicateur 6, testé dans un programme, donne la possibilité de modifier le cours du programme ou de modifier un résultat chaque fois qu'un dépassement se produit.

Les témoins des indicateurs armés. Les indicateurs 0, 1, 2 et 3 font apparaître des témoins sur l'affichage lorsqu'ils sont armés. La présence ou l'absence de **0**, **1**, **2** ou **3** vous renseigne immédiatement sur leur état. Quant à celui de 4, 5 et 6, il ne peut être mis en évidence que par une instruction FS? (voir ci-dessous).

Fonctions pour indicateurs. Le menu ■ FLAGS fait apparaître le menu des indicateurs :

```
SF   CF   FS?
```

Après le choix de la fonction, le HP-32S vous demande le numéro de l'indicateur ; pour armer 0, appuyez sur ■ FLAGS {SF} 0.

Le menu FLAGS

Touche de menu	Description
{SF} *n*	*Set flag (armer indicateur).* Arme *n*.
{CF} *n*	*Clear flag (désarmer ind.).* Désarme *n*.
{FS?} *n*	*Indicateur armé ?* Teste l'état de l'indicateur *n*.

Test des indicateurs (FS?). Un test d'indicateur est, comme un test conditionnel, un test de comparaison qui agit sur l'exécution du programme. L'instruction FS? *n* teste l'état d'un indicateur. S'il est armé, la ligne suivante du programme est exécutée. S'il ne l'est pas, la ligne suivante est sautée.

Il est possible d'exécuter {FS?} en dehors de la saisie de programme, mais le test d'indicateurs n'a pas grande utilité en dehors des programmes.

Il est important, au début d'un programme, de connaître la situation des indicateurs que vous testerez pendant son déroulement ; l'état des indicateurs avant l'exécution dépend des autres programmes qui ont été exécutés précédemment. Il serait dangereux de *supposer* qu'un indicateur est désarmé, et qu'il sera armé par votre programme. *Assurez-vous* de l'état des indicateurs et désarmez-les d'office, avant que ne se produise une situation qui les arme. Voici un exemple :

Exemple : utilisation des indicateurs. Le programme d'équation du second degré, en page 191, utilise l'indicateur 0 avec la comparaison `x<0?` pour enregistrer le signe de *B*. Notez que la ligne Q11 désarme l'indicateur 0 pour s'assurer qu'il ne sera armé *que* par la condition désirée.

```
Q11  CF  0          S'assure que l'indicateur 0 est désarmé.
Q12  x<0?           B (dans le registre X) est-il négatif ?
Q13  SF  0          Arme l'indicateur 0 si B est négatif.
       :
Q23  FS?  0         L'indicateur 0 est-il armé ? (B est-il
                    négatif ?)
Q24  +/-            Si oui, change le signe.
Q25  +              Si non, additionne.
       :
```

Les autres programmes de la quatrième partie du manuel qui utilisent les indicateurs sont « Ajustement de courbes » et « Conversion d'unités ». Ils utilisent les indicateurs pour enregistrer la condition que l'utilisateur désire résoudre (le type de courbe, le type de conversion), ce qui a une influence sur le choix des options ou des calculs.

Boucles (GTO, LOOP)

Le branchement récursif—c'est-à-dire le fait de brancher à un label placé dans une ligne de programme précédente—permet d'untiliser une partie de programme plusieurs fois ; on parle de *boucle* de programme.

```
D01 LBL D
D02 INPUT M
D03 INPUT N
D04 INPUT T
D05 GTO D
```

Ce sous-programme, extrait de « Transformations de coordonnées », en page 198, est un exemple de *boucle sans fin*. Il est utilisé pour récolter les données initiales, avant les transformations de coordonnées. Après la saisie des trois valeurs, c'est à l'utilisateur du programme d'interrompre la boucle en choisissant les transformations qui doivent être exécutées (en appuyant sur XEQ N pour « *old-to-new* », d'ancien en nouveau, ou XEQ O pour « *new-to-old* », de nouveau en ancien).

Boucles conditionnelles (GTO)

Dans les cas où vous désirez exécuter une opération si une certaine condition se réalise, mais où vous ne savez pas combien de fois il est nécessaire de répéter la boucle, vous pouvez créer une boucle qui comporte un test conditionnel et une instruction GTO.

Voici par exemple un sous-programme qui utilise une boucle pour diminuer une valeur *A* d'une quantité constante *B* jusqu'à ce que le résultat *A* soit inférieur ou égal à *B*.

Lignes de programme :	**Description :**
`A01 LBL A`	
`A02 INPUT A`	
`A03 INPUT B`	
`S01 LBL S`	
`S02 RCL A`	Il est plus facile de rappeler *A* que de se souvenir de ce qu'il y avait dans la pile...
`S03 RCL- B`	Calcule $A - B$.
`S04 STO A`	Remplace l'*ancien A* par le *nouveau* résultat.
`S05 RCL B`	Rappelle la constante pour comparaison.
`S06 x<y?`	$B <$ nouvel A ?
`S07 GTO S`	Oui : la boucle répète la soustraction.
`S08 VIEW A`	Non : le nouveau *A* est affiché.
`S09 RTN`	

Boucles avec compteurs (DSE, ISG)

Si vous désirez exécuter une boucle un certain nombre de fois, utilisez les fonctions conditionnelles DSE (*decrement; skip if less than or equal to*, décroissement : sautez si le nombre est inférieur ou égal à) ou ISG (*increment; skip if greater than*, augmentation : sautez si le nombre est supérieur à) du menu LOOP (■ LOOP). A chaque exécution, une valeur-compteur stockée dans une variable est diminuée ou augmentée ; elle compare la valeur actuelle du compteur à une valeur finale que le compteur doit atteindre, puis continue ou sort de la boucle.

Pour une boucle à décompteur, utilisez :

■ LOOP {DSE} *variable*

Pour une boucle à compteur, utilisez :

■ LOOP {ISG} *variable*

Ces fonctions sont très similaires à la boucle FOR-NEXT en BASIC :

```
FOR variable = valeur initiale TO valeur finale STEP incrément
        ⋮
NEXT variable
```

Une instruction DSE (*decrement, skip if less or equal to)* ressemble à une instruction FOR-NEXT dont le compteur est un *décompteur*.

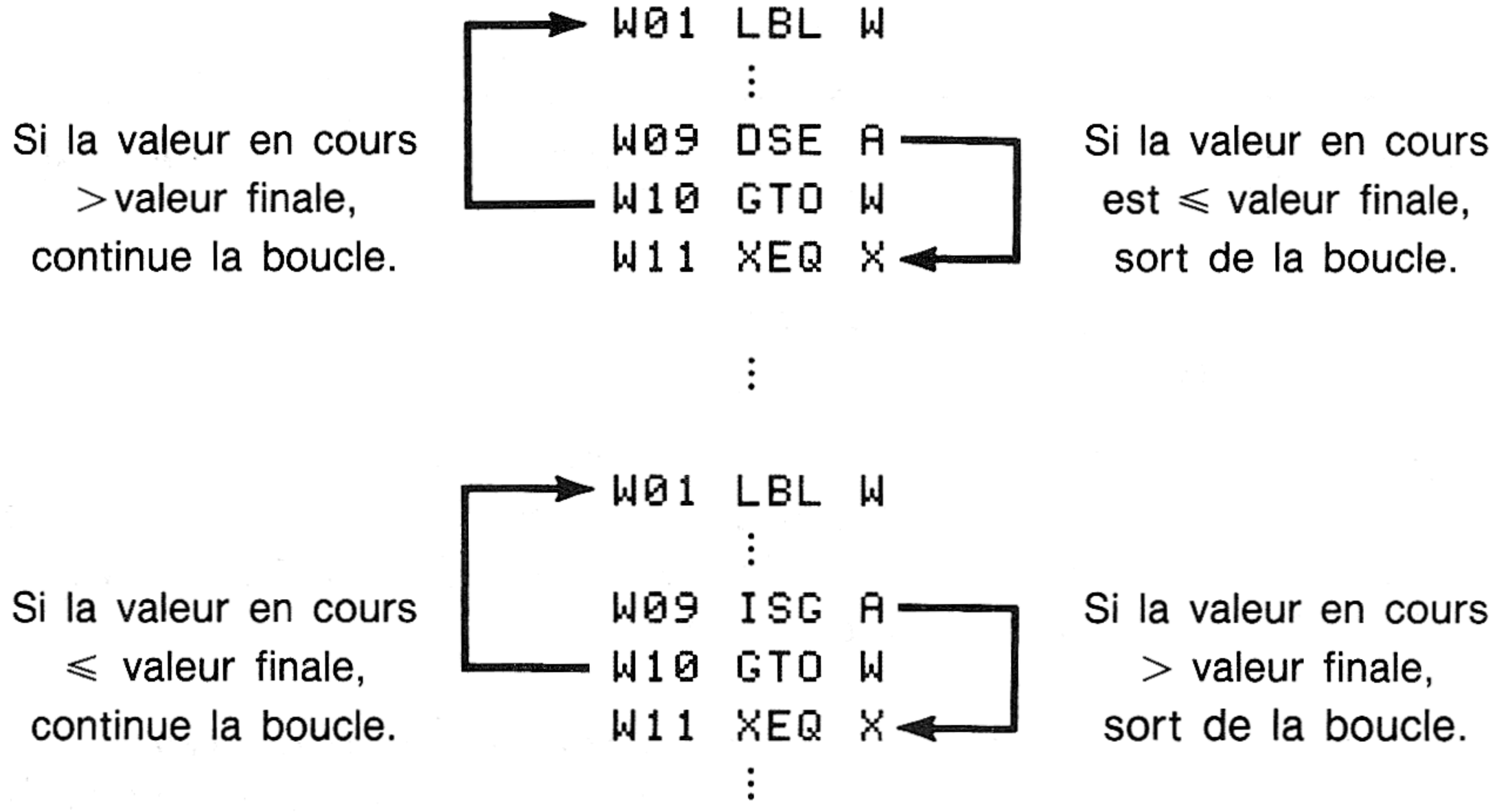
Après avoir appuyé sur la touche de menu {DSE} ou {ISG}, le HP-32S sollicite la lettre de la variable qui contient le *nombre de contrôle de boucle*...

Le nombre de contrôle de boucle. La variable spécifiée contient un nombre de contrôle, ±*ccccccc.fffii*, dans lequel :

- ±*ccccccc* est la valeur en cours du compteur (1 à 12 chiffres). Cette valeur *change* lors de l'exécution de la boucle.
- *fff* est la valeur finale du compteur (trois chiffres) ; cette valeur *ne change pas* lors de l'exécution de la boucle.
- *ii* est l'intervalle d'augmentation et de diminution (deux chiffres). Cette valeur ne change *pas*. Si vous ne précisez rien, la valeur sera 01 (augmentation/diminution de 1).

Etant donné le nombre de compteur de boucle *ccccccc.fffii,* DSE diminue *ccccccc* pour l'amener à *ccccccc* — *ii,* compare le nouveau *ccccccc* avec *fff,* et fait passer l'exécution du programme à la ligne suivante si ce *ccccccc* ⩽ *fff.*

ISG augmente *ccccccc* pour l'amener à *ccccccc* + *ii,* compare le nouveau *ccccccc* avec *fff,* et fait sauter la ligne suivante si *ccccccc* > *fff.*



Par exemple, le nombre de contrôle de boucle 0,050 pour l'instruction ISG signifie : commencer le compte à zéro, compter jusqu'à 50 et augmenter le nombre de 1 à chaque révolution.

Le programme suivant utilise ISG pour effectuer 10 boucles. Le compteur de boucles (0000001,01000) est stocké dans la variable Z. Les zéros à droite et en tête peuvent être ignorés.

```
L01 LBL L
L02 1,01
L03 STO Z
M01 LBL M
M02 ISG Z
M03 GTO M
M04 RTN
```

Appuyez sur ■ VIEW Z pour voir le compteur : 11,0100.

Adressage indirect de variables et de labels

L'*adressage indirect* est une technique de programmation de haut niveau pour spécifier une variable ou un label *sans les spécifier nommément à l'avance*. Ceci dépend des résultats intermédiaires obtenus lors de l'exécution, ou des saisies faites en cours de déroulement du programme.

L'adressage indirect utilise deux touches : [i] (avec [·]) et [(i)] (par la touche [R/S]).* Ces touches sont actives pour plusieurs fonctions qui utilisent des lettres comme variables ou labels.

- *i* est une variable dont le contenu fait référence à une *autre* variable ou libellé. Elle contient un nombre comme toute autre variable.
- [(i)] est une fonction de programmation qui indique d'utiliser le nombre contenu par *i* pour déterminer quelle variable ou label sont désignés ; ceci est un *adressage indirect* (*A* à *Z* sont des *adresses directes*).

[i] et [(i)] sont utilisées ensemble pour créer une adresse indirecte (voyez les exemples en page 105). Au demeurant, *i* est une variable comme les autres. Et [(i)] est soit indéfinie (pas de nombre dans *i*) ou incontrôlée (utilisant n'importe quel nombre laissé dans *i*).

La variable « *i* »

Le contenu de *i* peut être manipulé comme celui de n'importe quelle autre variable. Vous pouvez résoudre pour *i* ou intégrer à l'aide de *i*.

Fonctions utilisant *i* directement

STO *i*	INPUT *i*	DSE *i*
RCL *i*	VIEW *i*	ISG *i*
STO +, −, ×, ÷ *i*	∫FN *i*	
RCL +, −, ×, ÷ *i*	SOLVE *i*	

* La variable *I* n'a rien à voir avec [(i)] ou avec la variable *i*.

L'adresse indirecte (*i*)

Bien des fonctions qui utilisent les lettres A à Z (comme variables ou labels) peuvent utiliser [(i)] pour référer à ces lettres (comme variables ou comme labels) de façon *indirecte*. La fonction [(i)] utilise la valeur de la variable *i* pour déterminer quelle variable ou quel label est désigné. Voici comment :

Adressage indirect

Si *i* contient :	...(*i*) désignera :
± 1 ⋮ ±26	variable *A* ou label A ⋮ variable *Z* ou label Z
⩾27 ou ⩽−27 or 0	erreur : INVALID (i)

Seule la valeur absolue de la partie entière du nombre contenu dans *i* est utilisée pour l'adressage.

Voici les fonctions qui peuvent utiliser (*i*) comme adresse. Pour GTO, XEQ et FN=, (*i*) réfère à un label ; pour toutes les autres fonctions, il fait référence à une variable.

Fonctions utilisant (*i*) pour l'adressage indirect

STO(i)	INPUT(i)
RCL(i)	VIEW(i)
STO+, −, ×, ÷ (i)	DSE(i)
RCL+, −, ×, ÷ (i)	ISG(i)
XEQ(i)	SOLVE(i)
GTO(i)	∫FN(i)
	FN=(i)

Contrôle de programme avec (*i*)

Puisque le contenu de *i* peut changer chaque fois qu'un programme s'exécute—ou même dans différentes parties d'un même programme—une instruction telle que `GTO(i)` peut brancher en direction de labels différents à des moments différents. Ceci conserve une certaine souplesse en laissant sans réponse (jusqu'à l'exécution du programme) la question de savoir quelle variable ou quel programme seront nécessaires (voir le premier exemple ci-dessous).

L'adressage indirect est très utile pour le comptage et le contrôle des boucles. La variable *i* sert d'*index,* conservant l'adresse de la variable qui contient le nombre de contrôle du compteur pour les fonctions DSE et ISG (voir le deuxième exemple ci-dessous).

Exemple : choix de sous-routines avec (*i*). Le programme d'« Ajustement de courbes », page 204, utilise l'adressage indirect pour déterminer le modèle à utiliser pour calculer les valeurs estimées de x et de y (des sous-routines différentes calculent x et y pour les différents modèles). Notez que *i* est stockée et qu'il y est ensuite fait un adressage indirect dans différentes parties, largement distinctes, du programme).

Les quatre premiers sous-programmes (D, L, E, P) du programme spécifient le *modèle* de courbe qui sera utilisé et assignent un numéro (1, 2, 3, 4) à chacun de ces modèles. Le numéro est alors stocké dans le sous-programme Z, le point de saisie commun à chacun des modèles :

```
Z03 STO i
```

Le sous-programme Y utilise *i* pour appeler la sous-routine appropriée au modèle pour calculer les estimations pour x et y. La ligne Y03 appelle la sous-routine de calcul de $\hat{y}$:

```
Y03 XEQ(i)
```

et la ligne Y08 appelle une sous-routine différente pour calculer $\hat{x}$ après que *i* ait été augmentée de 6 :

```
Y06 6
Y07 STO+ i
Y08 XEQ(i)
```

Si *i* contient :	...XEQ(i) appelle :	Pour :
1	LBL A	Calcule *y* pour le modèle en ligne droite.
2	LBL B	Calcule *y* pour le modèle logarithmique.
3	LBL C	Calcule *y* pour le modèle exponentiel.
4	LBL D	Calcule *y* pour le modèle de puissance.
7	LBL G	Calcule *x* pour le modèle en ligne droite.
8	LBL H	Calcule *x* pour le modèle logarithmique.
9	LBL I	Calcule *x* pour le modèle exponentiel.
10	LBL J	Calcule *x* pour le modèle de puissance.

Exemple : contrôle de boucle avec (*i*). Une valeur-index placée dans *i* est utilisée par le programme « Solutions d'équations simultanées—méthode du déterminant », cité en page 175. Ce programme utilise les instructions de boucle `ISG i` et `DSE i` conjointement avec les instructions indirectes `RCL(i)` et `STO(i)` pour remplir et manipuler la matrice.

La première partie de ce programme est la sous-routine A, qui place le nombre de contrôle de boucle initial dans *i*.

Lignes de programme :	Description :
`A01 LBL A`	Le point de départ de l'introduction de données.
`A02 1,012`	Le nombre de contrôle de boucle : bouclage de 1 à 12 par pas de 1.
`A03 STO i`	Stocke le nombre de contrôle de boucle dans *i*.

Le sous-programme suivant est L, une boucle destinée à récolter les 12 valeurs connues pour une matrice à coefficient 3×3 (variables *A*–*I*) et les trois constantes (*J*–*L*) pour les équations.

Lignes de programme :	**Description :**
`L01 LBL L`	Ce sous-programme reçoit toutes les valeurs connues dans trois équations.
`L02 INPUT(i)`	Sollicite et stocke un nombre dans la variable adressée par *i*.
`L03 ISG i`	Ajoute 1 à *i* et répète la boucle jusqu'à ce que *i* atteigne la valeur de 13,012.
`L04 GTO L`	
`L05 GTO A`	Lorsque *i* dépasse la valeur finale du compteur, l'exécution se branche à nouveau sur A.

Troisième partie

Fonctionnement de haut niveau

Page 110 7 : Recherche d'une inconnue

126 8 : Intégration numérique

137 9 : Opérations sur les nombres complexes

144 10 : Conversions de bases et arithmétique

153 11 : Statistiques

7

Recherche d'une inconnue

L'opération SOLVE peut résoudre une équation pour n'importe quelle variable ; par exemple :

$$x^2 - 3y.$$

Cette fonction peut être rendue égale à zéro pour créer l'équation :

$$x^2 - 3y = 0.*$$

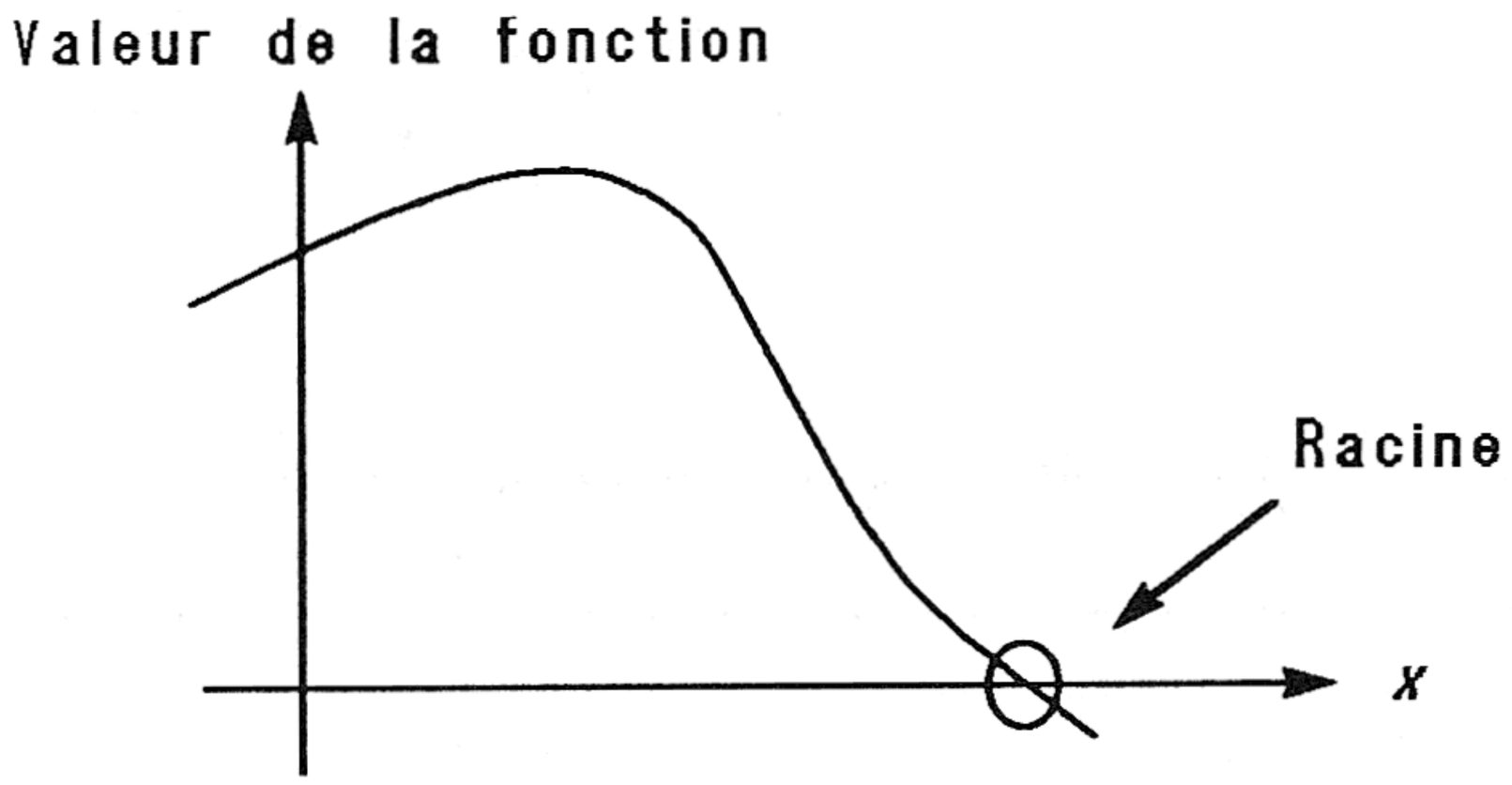
Si vous connaissez la valeur de y dans cette équation, SOLVE peut résoudre l'équation pour x. Si vous connaissez x, SOLVE peut la résoudre pour y. Ceci est vrai également pour les équations sous forme « verbale » :

$$Marge \times Coût = Prix$$
$$Marge \times Coût - Prix = 0.$$

Si vous connaissez l'une de ces variables, SOLVE peut calculer la troisième.

Lorsque l'équation ne compte qu'une variable, ou lorsque des variables connues sont fournies pour toutes les variables sauf une, résoudre pour x revient à trouver la (les) racine(s) de l'équation. La *racine* d'une équation est atteinte lorsque la courbe de la fonction atteint l'axe des x, parce que c'est le point où la valeur de la fonction est égale à zéro.

* En fait, vous pouvez déterminer la fonction pour qu'elle soit égale à n'importe quelle valeur réelle, telle $x^2 - y = 10$. Ceci peut alors être mis sous la forme $x^2 - y - 10 = 0$ pour faire usage de SOLVE.



Utilisation de SOLVE

Pour résoudre une équation à une inconnue :

1. Saisissez un programme qui définisse la fonction (voyez « Rédaction de programmes pour SOLVE », ci-dessous).
2. Choisissez le programme qui définit la fonction à résoudre : appuyez sur
   [■] [SOLVE/∫] {FN} *label*.
3. Solution pour la variable inconnue : appuyez sur
   [■] [SOLVE/∫] {SOLVE} *variable*.

Il est possible d'arrêter un calcul en appuyant sur [C] ou sur [R/S].

Estimations initiales. Pour certaines fonctions, une ou deux estimations de l'inconnue sont utiles (dans le registre de variable et le registre X). Ceci peut accélérer le calcul, diriger la réponse vers une solution proche de la réalité et, si c'est nécessaire, trouver plus d'une solution. Voir « Choix d'estimations de départ pour SOLVE » en page 120.

Résultats. Le registre X et la variable elle-même contiennent l'estimation finale de la racine, le registre Y contient l'estimation précédente et le registre Z contient la valeur de la fonction selon l'estimation la plus basse de la racine.

Dans certaines situations mathématiques complexes, une solution définitive ne peut être trouvée. Voir « Interprétation des résultats » et « Lorsque SOLVE ne peut trouver une Racine », à l'annexe C.

Pour résoudre l'équation avec une autre inconnue : Spécifiez la variable inconnue : ■ SOLVE/∫ {SOLVE} *variable*. Le programme qui a été défini en dernier lieu sera utilisé (FN= *label*).

Création de programmes pour SOLVE

Avant de pouvoir résoudre une équation à une inconnue, il faut écrire un programme ou une sous-routine qui évalue la fonction.*

Création d'une fonction à partir d'une équation. Commencez par simplifier l'équation en combinant toutes les variables semblables et toutes les constantes. Puis placez tous les termes d'un côté de l'équation, ne laissant que zéro de l'autre côté.

Par exemple, l'équation pour le volume d'une boîte est donnée par :

$$\text{Longueur} \times \text{profondeur} \times \text{hauteur} = \text{Volume}.$$

Déplacez les termes de l'équation pour qu'un des côtés soit égal à zéro :

$$\text{Longueur} \times \text{profondeur} \times \text{hauteur} - \text{Volume} = 0 \text{ , ou}$$

$$L \times P \times H - V = 0.$$

Pour rédiger un programme qui évalue une fonction :

1. Commencez par le label, pour que SOLVE puisse invoquer le programme.
2. Prévoyez une instruction INPUT pour chaque variable, y compris l'inconnue (s'il n'y a qu'une variable dans la fonction, omettez l'instruction INPUT, puisqu'elle est ignorée pour l'inconnue).†

* SOLVE ne fonctione qu'avec les nombres reéls. Si cependant vous êtes en présence d'une fonction à valeurs complexes qui permet d'isoler les parties réelle et imaginaire, SOLVE peut alors résoudre ces deux parties de manière séparée.

† Les instructions INPUT sont utiles pour les fonctions à plusieurs variables. Puisque l'instruction INPUT est ignorée pour l'inconnue, il n'est besoin de créer qu'*un seul* programme, qui contient les instructions INPUT pour *toutes* les variables. Vous utiliserez le même programme, quelle que soit l'inconnue.

3. Introduisez les instructions pour l'évaluation de la fonction. Utilisez une instruction RCL partout où une valeur de variable est nécessaire à un calcul.
4. Terminez le programme par un RTN. Le programme devrait se terminer avec la valeur de la fonction dans le registre X.

Chaque fois que SOLVE exécute un programme (plusieurs fois, peut-être), la valeur de la variable inconnue change, tout comme la valeur produite par le programme. Lorsque le programme revient à zéro, la valeur de l'inconnue a été trouvée.

Exemples d'utilisation de SOLVE

Exemple : les dimensions d'une boîte. Le programme suivant utilise l'*arithmétique de rappel,* qui utilise moins de mémoire qu'en rappelant chaque variable et en exécutant une série d'opérations séparées.

```
B01 LBL B
B02 INPUT L
B03 INPUT P
B04 INPUT H
B05 INPUT V
B06 RCL L
B07 RCL× P
B08 RCL× H
B09 RCL- V
B10 RTN
```

Saisissons d'abord le programme B :

Touches :	Affichage :	Description :
■ PRGM		Commence la saisie de programme. Va se placer en tête de mémoire de programme (si nécessaire).
■ GTO · ·	PRGM TOP	

■ LBL/RTN {LBL} B	B01 LBL B	Saisit les lignes de programme.
■ INPUT L	B02 INPUT L	
■ INPUT P	B03 INPUT P	
■ INPUT H	B04 INPUT H	
■ INPUT V	B05 INPUT V	
RCL L	B06 RCL L	
RCL × P	B07 RCL× W	
RCL × H	B08 RCL× H	
RCL − V	B09 RCL- V	
■ LBL/RTN {RTN}	B10 RTN	A ce stade, le registre X contient la valeur de la fonction $L \times P \times H - V$.
C		Termine la saisie de programme. Affiche le contenu du registre X.

Donne le volume d'une boîte de 8,5 cm × 10 cm × 25 cm. Plus loin, nous utiliserons la même fonction pour une variable différente.

■ SOLVE/∫ {FN}	FN= _	Sollicite le label du programme définissant la fonction.
B		Spécifie le programme B.
■ SOLVE/∫ {SOLVE}	SOLVE _	Sollicite la variable inconnue.
V	L?*valeur*	Lance le programme B ; sollicite la saisie des données (sauf *V*, l'inconnue).
25 R/S	P?*valeur*	
10 R/S	H?*valeur*	
8,5		
R/S	SOLVING	Le volume est de 2 125 cm^3.
	V=2.125,0000	

Cherchons maintenant la longueur de la boîte si elle doit avoir un volume de 3 000 cm^3, mais conservons la même hauteur et la même profondeur. Nul besoin de spécifier le label de programme puisque nous utilisons le même programme que précédemment.

■ SOLVE/∫ {SOLVE}		Commence le programme B pour résoudre l'équation pour *L*. Sollicite les valeurs des variables.
L	W?10,0000	
R/S	H?8,5000	
R/S	V?2.125,0000	Pour conserver la même valeur, appuyez simplement sur R/S.
3000 R/S	L=35,2941	

Exemple : l'équation du mouvement linéaire. Voici l'équation du mouvement linéaire pour un objet en chute libre :

$$d = v_0 t + \tfrac{1}{2} g t^2$$

dans laquelle d est la distance, v_0 est la vélocité initiale, t est le temps et g est l'accélération due à la pesanteur. Après simplification :

$$0 = t(v_0 + gt/2) - d.$$

Voici le programme qui évalue cette fonction :

```
M01 LBL M
M02 INPUT V
M03 INPUT T
M04 INPUT G
M05 INPUT D
M06 RCL G
M07 2
M08 ÷
M09 RCL× T
N10 RCL+ V
M11 RCL× T
M12 RCL- D
M13 RTN
```

L'accélération due à la pesanteur, g, est incluse comme variable pour permettre de travailler avec des unités différentes :

$$g = 9{,}8 \text{ m/s}^2 = 32{,}2 \text{ ft/s}^2.$$

Saisit le programme ci-dessus (LBL M). Calcule la distance parcourue par un objet en chute libre en cinq secondes, à partir d'une position de repos.

Touches :	Affichage :	Description :
■ SOLVE/∫ {FN} M	FN= M *(brièvement)*	Spécifie LBL M pour la fonction.
■ SOLVE/∫		*D* est l'inconnue.
{SOLVE}D	V?3,000,0000	Sollicite la valeur de *V* et montre la valeur courante de *V* (utilisée dans le dernier exemple).
0 R/S	T? *valeur*	
5 R/S	G? *valeur*	
9,8 R/S	D=122,5000	La distance, en mètres.

Essayez un autre calcul, en utilisant la même équation : combien de temps faut-il à un objet pour chuter de 500 mètres ? Les valeurs v_0 et g sont déjà stockées.

■ SOLVE/∫		Spécifie une autre inconnue ; sollicite une valeur pour *V*.
{SOLVE}T	V?0,0000	
R/S	G?9,8000	
R/S	D?122,5000	
500 R/S	T=10,1015	Le résultat, en secondes.

Exemple : trouver les racines d'une équation. Soit une équation à une inconnue

$$x^3 - 5x^2 - 10x = -20.$$

Déplacez les termes de l'équation :

$$x^3 - 5x^2 - 10x + 20 = 0.$$

La méthode de Horner (voir chapitre 5) simplifie l'équation pour utiliser moins de mémoire :

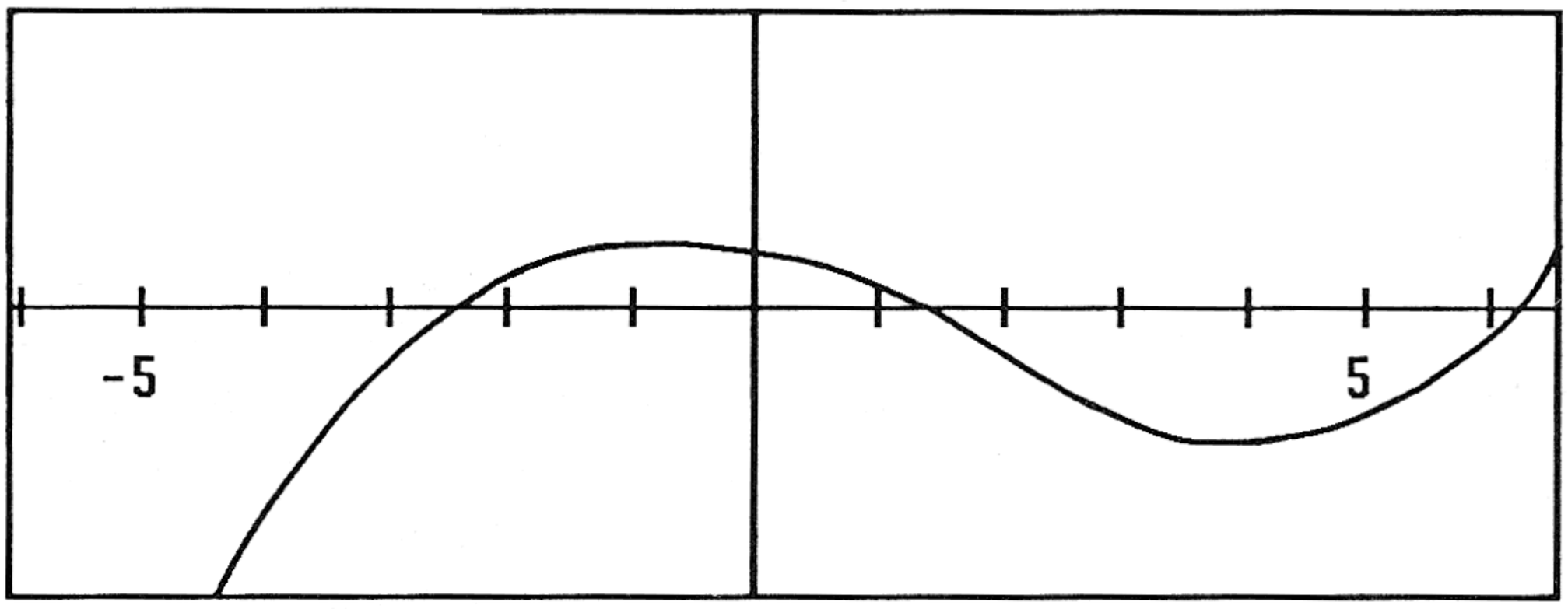
$$x(x(x - 5) - 10) + 20 = 0.$$

Voici le programme qui évalue cette fonction :

```
R01 LBL R
R02 RCL X
R03 5
R04 -
R05 RCL× X
R06 10
R07 -
R08 RCL× X
R09 20
R10 +
R11 RTN
```

Voici le tracé correspondant

$x^3-5x^2-10x+20$



Le tracé indique qu'il y a trois racines—la courbe coupe l'axe des x trois fois. Le calculateur trouve les trois racines si vous exécutez SOLVE trois fois *et si vous fournissez trois estimations chaque fois*. Pour plus de détails, voyez « Choix d'estimations de départ pour SOLVE », en page 120.

Saisissez le programme ci-dessus (LBL R). La courbe indique que la première racine se trouve quelque part entre $x = -3$ et $x = -2$, la deuxième entre 1 et 2 et la troisième entre 6 et 7. Placez chacune de ces estimations dans la variable X et dans le registre X, puis cherchez X.

Touches :	Affichage :	Description :
■ SOLVE/∫ {FN} R		Choisit le programme LBL R.
3 +/− STO X	-3,0000	Premières estimations.
2 +/−	-2_	
■ SOLVE/∫ {SOLVE} X	X=-2,4433	Spécifie l'inconnue et donne la première racine.
1 STO X	1,0000	Deuxièmes estimations.
2	2_	
■ SOLVE/∫ {SOLVE} X	X=1,3416	Deuxième racine.
6 STO X	6,0000	Troisièmes estimations.
7	7_	
■ SOLVE/∫ {SOLVE} X	X=6,1017	La troisième racine.

Si vous ne fournissez *pas* d'estimations, vous n'obtiendrez qu'*une seule* de ces racines. C'est le contenu des variables X et du registre X qui influencera le choix de la racine que vous obtiendrez, puisque le calculateur utilise les valeurs qui y sont placées comme estimations.

Comprendre et contrôler la fonction SOLVE

SOLVE utilise une procédure intérative (répétitive) pour résoudre une équation à une inconnue. La procédure commence par la substitution de deux estimations à l'inconnue définie dans le programme. Selon le résultat obtenu avec ces deux estimations, SOLVE génère une autre estimation, plus précise. Par un processus d'itération, SOLVE trouve la valeur qui rend la fonction égale à zéro.

Dans certains cas, il sera nécessaire d'entrer des estimations vous-même pour trouver la solution (lisez « Choix d'estimations initiales pour SOLVE » en page 120).

Vous trouverez également à l'annexe C des indications plus approfondies sur le fonctionnement de SOLVE.

Vérification du résultat

Après la fin du calcul par SOLVE, vous pouvez vérifier que le résultat est bien une racine de $f(x)$ en passant en revue les valeurs restant dans la pile :

- L'affichage (le registre X) et la variable elle-même contiennent la solution (la racine) de l'inconnue, c'est-à-dire la valeur qui rend $f(x) = 0$.
- Le registre Y (vous pouvez le vérifier en appuyant sur [R↓]) contient l'extimation précédente de la racine. Ce nombre devrait être indentique à celui du registre X. Si ce n'est *pas* le cas, c'est que la racine n'était qu'une *approximation* de la racine et les valeurs des registres X et Y « entourent » la racine. Ces nombres sont normalement proches l'un de l'autre.
- Le registre Z (appuyez une nouvelle fois sur [R↓]) contient la valeur de $f(x)$ selon la valeur donnée par le registre X. Pour une racine exacte, ce devrait être zéro. Si ce n'est pas le cas, c'est que la racine n'était qu'une *approximation,* et ce nombre devrait être proche de zéro.

Si un calcul se termine par `NO ROOT FND`, cela signifie que le calcul n'a pu aboutir à une racine, et les valeurs de X et de Y ne sont probablement pas proches l'une de l'autre. Effacez le message avec [C] ou [←] et lisez la valeur du registre X—la dernière estimation de la racine. Les deux valeurs, X et Y, entourent l'intervalle recherché en dernier lieu pour la racine. Le registre Z contient la valeur de $f(x)$ selon la dernière estimation de la racine. Cette valeur ne devrait pas être proche de zéro.

Interruption du calcul effectué par SOLVE

Pour arrêter le calcul, utilisez [C] ou [R/S]. La meilleure estimation en cours se trouve dans la variable inconnue ; vous pouvez la voir grâce à [■] [VIEW], sans déranger la pile. Pour reprendre le calcul, appuyez sur [R/S].*

* Le fait d'appuyer sur [XEQ], [GTO] ou sur {RTN} *annule* l'opération SOLVE. Dans ce cas, relancez le programme plutôt que d'essayer de le continuer.

Choix d'estimations de départ pour SOLVE

Les deux estimations de départ viennent de :

- Le nombre qui se trouve stocké à ce moment dans la variable inconnue.
- Le nombre stocké dans le registre X (l'affichage).

Ces deux sources sont utilisées *de toute manière*, que vous saisissiez des estimations ou non. Si vous ne saisissez qu'une seule estimation, la deuxième aura la même valeur puisque l'affichage conserve le nombre stocké dans la variable. Vous avez avantage à introduire vos propres estimations :

- La plage de recherche est rétrécie, il faut moins de temps pour arriver à la solution.
- S'il y a plus d'une solution, les estimations dirigent SOLVE vers la réponse correcte ou tout au moins vers la plage de réponses. Par exemple, l'équation du mouvement :

  $$d = d_0 + v_0 t + \tfrac{1}{2} g t^2$$

  a deux solutions possibles pour t. Vous pouvez orienter vers la réponse la plus plausible ($t > 0$) par la saisie d'estimations.*
- Si une équation n'admet pas certaines valeurs pour l'inconnue, les estimations peuvent éviter l'apparition de ces valeurs. Par exemple

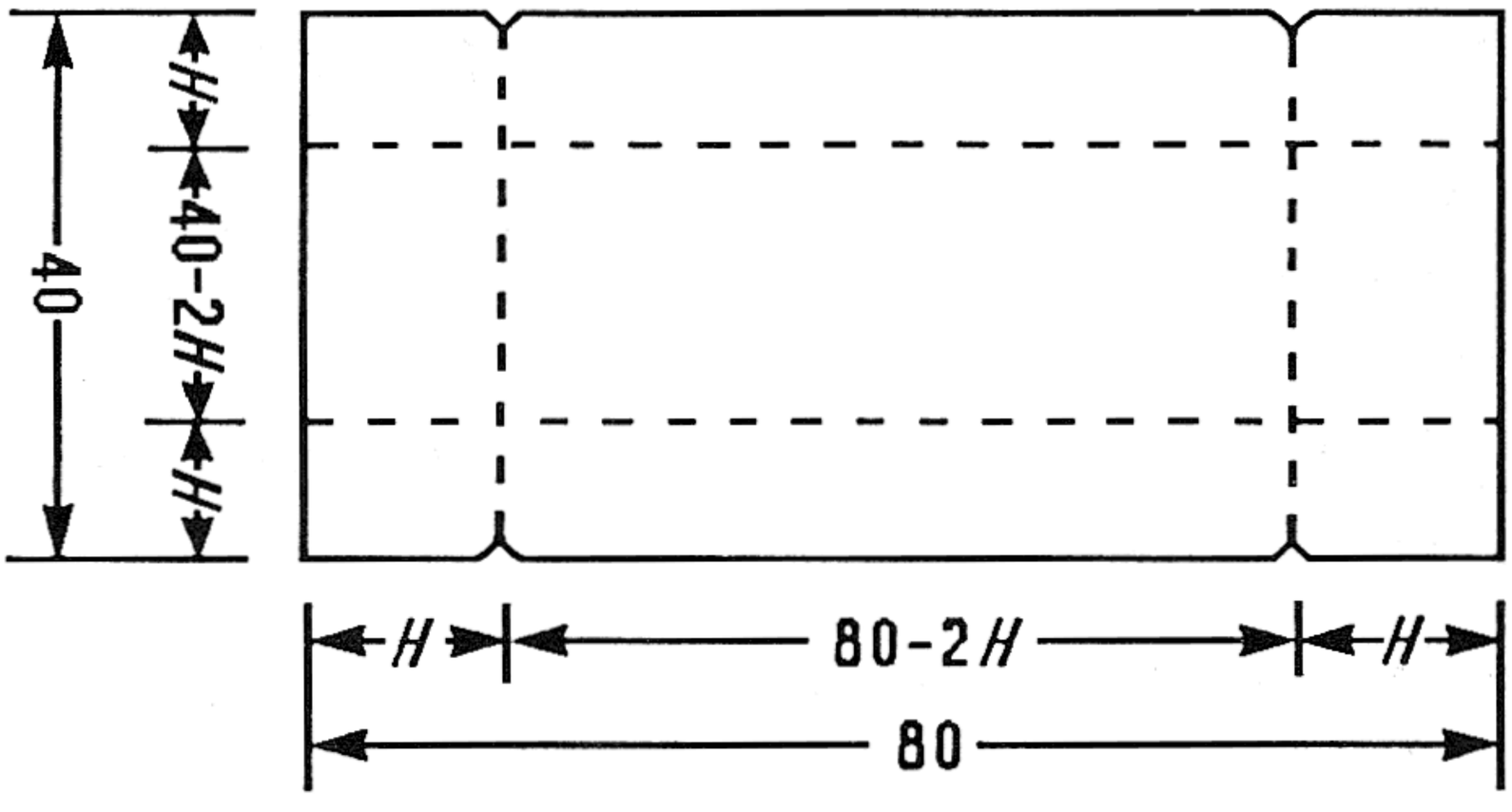
  $$y = t + \log x$$

  aboutit à une erreur si $x \leqslant 0$ (`LOG(0)`, `LOG(NEG)`).

L'exemple de la section précédente (« Trouver les racines d'une équation ») utilisait des estimations pour trouver trois solutions à une équation. Voici un autre exemple qui traite des dimensions d'une boîte, mais avec quelques restrictions.

* L'exemple qui utilisait cette équation en page 116 n'a pas besoin d'estimations pour trouver t parce que dans la première partie de l'exemple nous avons stocké une valeur pour T et trouvé D. La valeur laissée dans T était valable et a été utilisée comme estimation pour trouver T.

Exemple : plier la boîte. Avec une pièce de métal rectangulaire de 40 cm sur 80, formez une boîte sans couvercle d'un volume de 7 500 centimètres-cubes. Quel est la hauteur de la boîte ? *Une boîte de bonne hauteur serait préférable.*



Si H est la hauteur, la longueur de la boîte est $(80 - 2H)$ et la largeur est $(40 - 2H)$. Le volume, V, est :

$$V = (80 - 2H) \times (40 - 2H) \times H$$

et la fonction égale à zéro est

$$\begin{aligned} 0 &= (80 - 2H) \times (40 - 2H) \times H - V \\ &= 4H\,[(40 - H)\,(20 - H)] - V \end{aligned}$$

Voici le programme qui définit cette fonction :

```
V01 LBL V
V02 INPUT H
V03 INPUT V
V04 40
V05 RCL- H
V06 20
V07 RCL- H
V08 ×
V09 4
V10 ×
V11 RCL× H
V12 RCL- V
V13 RTN
```

Il semble raisonnable d'accepter que les deux extrêmes sont soit une boîte haute et étroite, soit une boîte courte et plate. Puisque c'est la première solution qui est préférée, les estimations pour la hauteur sont plus importantes. Mais une hauteur supérieure à 20 centimètres n'est pas possible parce que la feuille de métal n'est large que de 40 centimètres. Des estimations préalables de l'ordre de 10 et 20 centimètres sont donc appropriées.

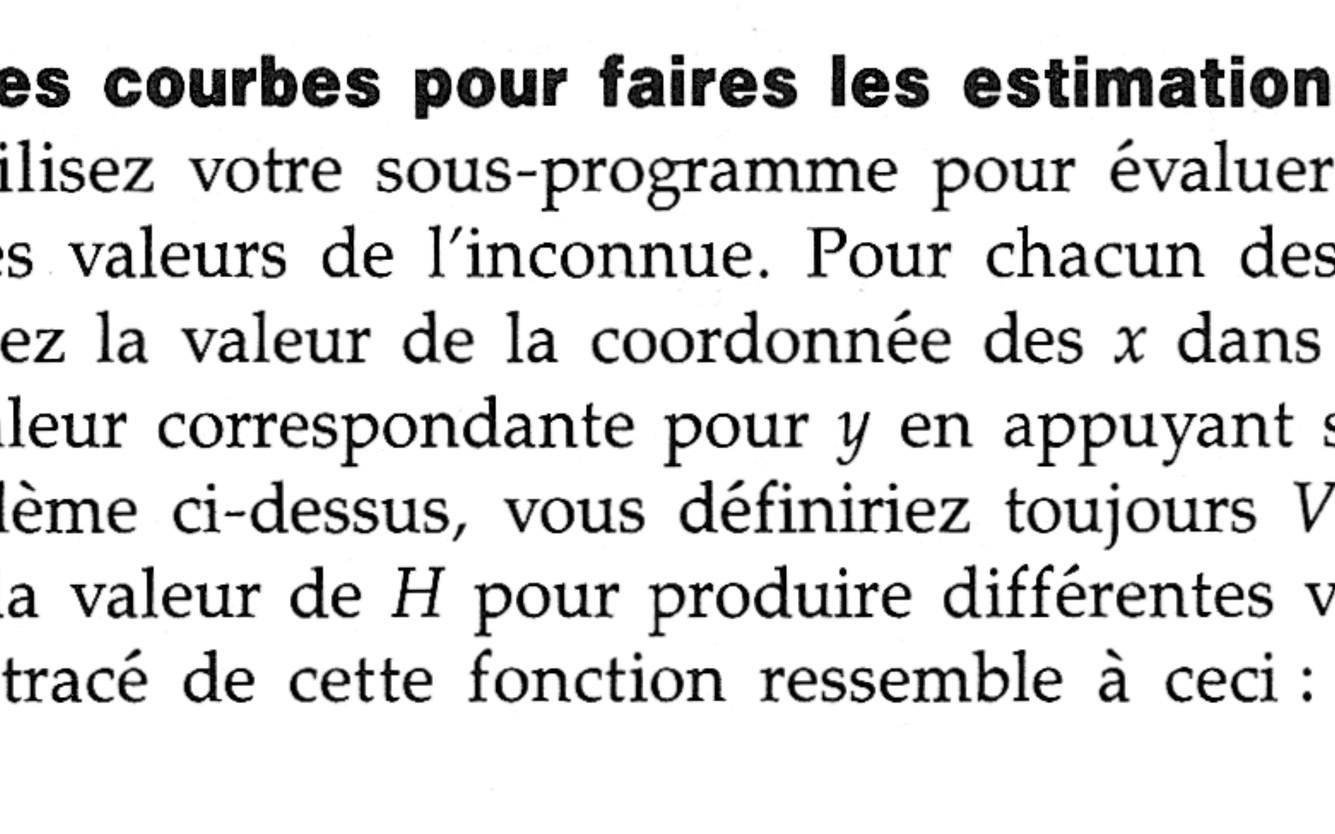
Touches :	**Affichage :**	**Description :**
■ SOLVE/∫ {FN} V		Sélectionne le programme V.
10 STO H 20	10,0000 20_	Stocke les limites supérieure et inférieure.
■ SOLVE/∫ {SOLVE} H	V?*valeur*	Sollicite le volume.
7500 R/S	SOLVING H=15,0000	Voici la hauteur idéale.

Vérifions maintenant la qualité de la solution—le fait de savoir si elle a fourni une racine exacte—en examinant les valeurs de l'équation précédente (dans le registre Y) et de $f(x)$ à la racine (dans le registre Z).

R↓	15,0000	Cette valeur du registre Y est l'estimation faite juste avant l'obtention du résultat final. Puisque c'est la même que la solution, celle-ci représente la racine exacte...
R↓	0,0000	...et, comme le montre cette valeur présente dans le registre Z, $f(x) = 0$ à la racine.

Les dimensions de la boîte sont 50 × 10 × 15 cm. Si vous avez ignoré la limite supérieure apportée à la hauteur (20 cm) et si vous avez utilisé 30 et 40 cm comme estimations de départ, vous obtiendrez une hauteur de 42,0256 cm—une racine qui n'a pas de signification pratique. Si vous avez utilisé des estimations réduites, comme 0 et 10 cm, vous obtiendrez une hauteur de 2,9744 cm—et une boîte plate.

Utlisation des courbes pour faires les estimations initiales. Utilisez votre sous-programme pour évaluer la fonction pour certaines valeurs de l'inconnue. Pour chacun des points de la courbe, stockez la valeur de la coordonnée des x dans la variable, et obtenez la valeur correspondante pour y en appuyant sur XEQ *label*. Pour le problème ci-dessus, vous définiriez toujours $V = 7\,500$ et feriez varier la valeur de H pour produire différentes valeurs pour la fonction. Le tracé de cette fonction ressemble à ceci :



Utilisation de SOLVE dans un programme

Vous pouvez utiliser l'opération SOLVE comme partie d'un programme. Si c'est nécessaire, vous pouvez ajouter des sollicitations pour estimations initiales (saisies dans la variable inconnue et le registre X) avant d'exécuter `SOLVE` *variable*. Les deux instructions nécessaires pour résoudre une équation à une variable inconnue se trouvent dans le programme sous la forme :

```
FN= label
SOLVE variable
```

Documentation d'un résultat. Lorsque SOLVE est programmée, elle ne produit pas un affichage qui documente le résultat (*variable=valeur*) puisque ce résultat n'est peut-être pas significatif—il se peut que vous ayez à faire d'autres calculs avec ce nombre avant de l'afficher. Pour afficher ce résultat, ajoutez une instruction VIEW *variable* après l'instruction SOLVE.

Exécution conditionnelle en cas de non-solution. Si aucune solution n'est trouvée pour la variable inconnue, la ligne suivante du programme est sautée. Le programme devrait alors prévoir le cas où une racine n'est pas trouvée, et par exemple choisir de nouvelles estimations ou changer les valeurs saisies.

Exemple : calculs sur des flux constants. Ce programme du chapitre 14 résout des problèmes de prêts et d'épargne en résolvant une équation à une inconnue. Cette équation est décrite comme une fonction dans le sous-programme T qui traite les valeurs pour le solde actuel, futur, les paiements, le taux d'intérêt et le nombre de paiements.

Si l'une quelconque de ces variables est donnée, SOLVE (à la ligne L04) trouve la solution pour la cinquième :

```
L01 LBL L
L02 STO i      Stocke une valeur d'index qui indique quelle
               variable a été choisie comme inconnue.
L03 FN= T      Sélectionne la fonction définie dans le pro-
               gramme T.
L04 SOLVE(i)   Trouve l'inconnue du programme T.
L05 VIEW(i)    Affiche la solution.
L06 GTO(i)     Renvoie à la sous-routine d'initialisation
               pour préparer un autre calcul.
```

Cette opération fonctionne parfaitement sans estimations initiales fournies par l'utilisateur.

Limitations. L'instruction SOLVE ne peut appeler un sous-programme contenant une autre instruction SOLVE, c'est-à-dire qu'elle ne peut être utilisée de manière récursive ; elle causerait l'erreur `SOLVE(SOLVE)`. SOLVE ne peut pas non plus appeler de sous-programme qui contienne l'instruction `FN=` *label* (erreur `SOLVE ACTIVE`). SOLVE ne peut appeler un sous-programme qui contient une instruction ∫FN (erreur `SOLVE(∫FN)`) ; de même, ∫FN ne peut invoquer un sous-programme contenant une instruction SOLVE (erreur `∫(SOLVE)`).

L'instruction `SOLVE` *variable* dans un programme utilise l'un des sept retours de sous-routines (Cf. « Sous-routines emboîtées », page 120).

Pour plus de renseignements...

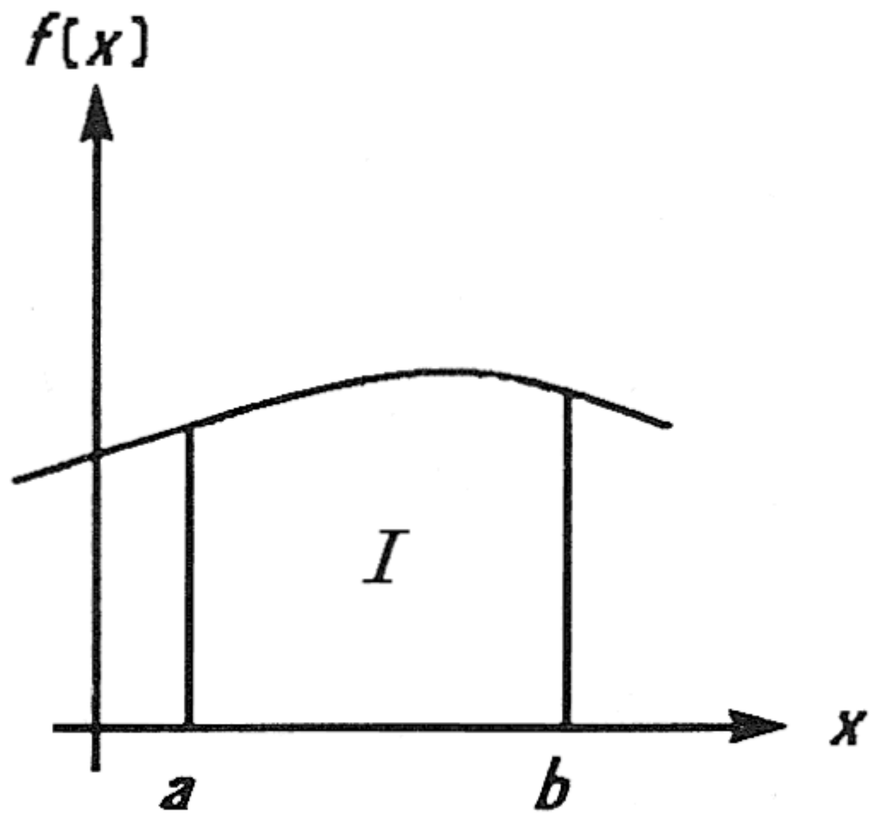
Ce chapitre traite d'équations couvrant un large domaine d'applications. L'annexe C décrit des renseignements complémentaires sur le fonctionnement de l'algorithme de SOLVE, comment interpréter les résultats, ce qui se passe lorsqu'aucun résultat n'est trouvé et les situations qui aboutissent à des résultats incorrects.

8

Intégration numérique

Beaucoup de problèmes en mathématiques, sciences et ingénierie sont résolus par le calcul de l'intégrale définie d'une fonction. Si la fonction est caractérisée par *f(x)* et si l'intervalle d'intégration va de *a* à *b*, l'intégrale peut être exprimée comme ceci :

$$I = \int_a^b f(x)\ dx.$$



I peut être interprétée géométriquement comme une région limitée par la courbe de la fonction *f(x)*, l'axe des *x* et les limites $x = a$ et $x = b$ (pour autant que *f(x)* ne soit pas négative dans l'intervalle d'intégration).

L'opération ∫FN intègre une fonction par rapport à une variable spécifiée.* La fonction doit avoir été définie auparavant dans un programme portant un label, et elle peut comporter plus d'une variable.

* ∫FN ne fonctionne qu'avec des nombres réels.

Comment utiliser l'intégration (∫FN)

Pour intégrer une fonction :

1. Saisissez un programme qui définisse la fonction de l'intégrande (voir « Création de programmes pour ∫FN », en page suivante).
2. Choisissez le programme qui définit la fonction à intégrer : appuyez sur
   [■] [SOLVE/∫] {FN} *label*.
3. Saissez les limites d'intégration ; introduisez la *limite inférieure* et appuyez sur [ENTER], puis la *limite supérieure*.
4. Choisissez la variable d'intégration : appuyez sur
   [■] [SOLVE/∫] {∫FN} *variable*.

   Ceci lance le calcul.

Cette opération utilise plus de mémoire qu'aucune autre opération du calculateur. Si l'exécution de {∫FN} provoque l'apparition d'un message MEMORY FULL, voyez l'annexe B.

Vous pouvez arrêter un calcul d'intégration en appuyant [C] ou sur [R/S]. Aucune information sur l'intégration n'est disponible jusqu'à ce que le calcul se termine normalement. Pour reprendre le calcul, appuyez sur [R/S].*

Précision. Le choix du format d'affichage influence le niveau de précision de la fonction et celui du résultat. L'intégration est plus précise, mais nécessite beaucoup plus de temps avec les choix de {ALL}, de {FX} avec une valeur élevée,{SC} et {EN}.
L'*incertitude* du résultat se termine dans le registre Y, repoussant les limites de l'intégration vers le haut, dans les registres T et Z. Pour plus d'informations, lisez « Précision de l'intégration », en page 131.

Résultats. Le registre X contient l'intégrale, le registre Y contient l'incertitude, le registre Z la limite supérieure et le registre T, la limite inférieure ; la variable d'intégration contient une valeur sans importance.

* Le fait d'appuyer sur [XEQ], [GTO] ou sur {RTN} annule l'opération ∫FN. Dans ce cas, recommencez l'opération plutôt que de la reprendre où vous l'aviez laissée.

Pour intégrer la même fonction avec d'autres données : Sautez les deux premières étapes ci-dessus. Si vous utilisez les mêmes limites, appuyez deux fois sur [R↓] pour les ramener dans les registres X et Y. Si vous *n'utilisez pas* les mêmes limites, répétez le point 3. Exécutez ■ [SOLVE/∫] {∫FN} *variable*. Pour traiter un autre problème en utilisant une fonction *différente*, recommencez avec un autre programme pour la fonction.

Création de programmes pour ∫FN

Comment rédiger un programme définissant la fonction de l'intégrande :

1. Commencez par un label pour que le programme puisse être invoqué par∫FN.
2. Introduisez l'instruction INPUT pour chaque variable, y compris pour la variable d'intégration. S'il n'y a qu'une variable dans la fonction, vous pouvez omettre l'instruction INPUT.*
3. Saissez les instructions pour définir la fonction. Utilisez l'instruction RCL chaque fois qu'une valeur de variable est nécessaire pour un calcul.
4. Terminez le programme par RTN. Le programme devrait se terminer avec la valeur de la fonction affichée dans le registre X.

Exemples d'utilisation de ∫FN

Exemple : fonction de Bessel. La fonction de Bessel de la première catégorie d'ordre zéro est exprimée sous la forme

$$J_0(x) = {}^1/\pi \int_0^\pi \cos\,(x \sin t)\, dt.$$

Trouvez la fonction de Bessel pour des valeurs de x de 2 et 3.

* Les instructions INPUT sont utiles dans les fonctions à plusieurs variables. Puisque l'INPUT pour la variable d'intégration (l'intégrande) est ignorée, vous pouvez écrire un programme qui contienne des instructions INPUT pour *toutes* les variables, que vous pouvez utiliser ensuite sans égard à la variable que vous spécifiez comme étant la variable d'intégration.

Ce programme évalue la fonction $f(t) = \cos(x \sin t)$:

```
J01 LBL J
J02 RAD
J03 INPUT X
J04 INPUT T
J05 RCL T
J06 SIN
J07 RCL× X
J08 COS
J09 RTN
```

Le programme est saisi de la façon suivante :

Touches :	**Affichage :**	**Description :**
■ PRGM		Commence la saisie de
■ GTO · ·	PRGM TOP	programme et place le pointeur au début.
■ LBL/RTN {LBL} J	J01 LBL J	Commence le
■ MODES {RD}	J02 RAD	programme.
■ INPUT X	J03 INPUT X	
■ INPUT T	J04 INPUT T	
RCL T SIN	J06 SIN	
RCL × X COS	J08 COS	
■ LBL/RTN {RTN}	J09 RTN	A ce stade, le registre X contiendra la valeur de la fonction.
C		Termine la saisie de programme.

Intégrez cette fonction par rapport à t de zéro à π ; $x = 2$.

■ SOLVE/∫ {FN}	FN= _	Choisit le sous-programme J pour la
J		fonction.
0 ENTER ■ π	3,1416	Introduit les limites de l'intégration.
■ SOLVE/∫ {∫FN}	∫FN d _	Spécifie T comme variable d'intégration.
T	X?*valeur*	Sollicite la valeur de x.

2 R/S	∫=0,7034	$x = 2$. Commence l'intégration et produit le résultat de $\int_0^{\pi} f(t)$.

Pour terminer le calcul, souvenez-vous de multiplier la valeur de l'intégrale par la constante $1/\pi$ en dehors de l'intégrale. Vous pouvez également inclure cette multiplication dans le programme.

■ π ÷	0,2239	Résultat final de $J_0(2)$.

Calculez maintenant $J_0(3)$ avec les mêmes limites d'intégration. Vous n'avez pas à re-spécifier la fonction (le sous-programme J), mais vous devrez spécifier les limites de l'intégration (0, π) puisqu'elles ont été repoussées par la division suivante par π.

0 ENTER ■ π	3,1416	Affiche la limite supérieure.
■ SOLVE/∫ {∫FN} T	X?2,0000	Commence l'intégration ; sollicite x.
3 R/S	∫=-0,8170	Intégrale de $f(t)$.
■ π ÷	-0,2601	Résultat pour $J_0(3)$.

Exemple : intégrale de sinus. Certains problèmes de la théorie de la communication (par exemple la communication dans des réseaux idéalisés) nécessitent le calcul d'une intégrale de la forme

$$Si(t) = \int_0^t \left(\frac{\sin\ x}{x}\right) dx.$$

Trouvez $Si(2)$.

Entrez le programme suivant pour évaluer la fonction $f(x) = (\sin x) \div x$.*

* Si le calculateur essaie d'évaluer cette fonction à la valeur $x = 0$, la limite inférieure, l'erreur DIVIDE BY 0 se produira. Normalement, l'algorithme d'intégration n'évalue pas de fonction à la limite d'intégration, à moins que les points extrêmes de l'intervalle d'intégration soient très proches et que le nombre de points d'échantillonnage soit extrêmement grand.

```
S01 LBL S
S02 RAD
S03 RCL X
S04 SIN
S05 RCL÷ X
S06 RTN
```

Intégrez cette fonction par rapport à x (c'est-à-dire, X) de zéro à 2 ($t = 2$).

Touches :	**Affichage :**	**Description :**
■ SOLVE/∫ {FN} S		Choisit le sous-programme S pour la fonction.
0 ENTER 2	2_	Introduit les limites de l'intégration (la limite inférieure d'abord).
■ SOLVE/∫ {∫FN}X	∫=1,6054	Résultat pour $Si(2)$.

Précision de l'intégration

Puisque le calculateur est incapable de calculer *exactement* la valeur d'une intégrale, il effectue une *approximation*. Sa précision dépend de la précision de l'intégrande de la fonction elle-même, telle qu'elle a été calculée par votre programme.* L'erreur d'arrondi dans le calculateur et la précision des constantes empiriques sont également un facteur.

* Les intégrales de fonctions avec certaines caractéristiques, comme des pics ou des oscillations très rapides *peuvent* être calculées de manière erronnée, mais cette probabilité est très réduite. Les caractéristiques générales des fonctions qui peuvent susciter des problèmes, et les techniques pour résoudre ces problèmes, sont traitées à l'annexe D.

Choix de la précision

Le réglage du format d'affichage détermine la *précision* du calcul d'intégration : plus grand le nombre de chiffres affichés, plus grande sera la précision de l'intégrale calculée (et plus il faudra de temps pour la résoudre). Moins il y a de chiffres affichés, plus rapide est le calcul, mais le calculateur estimera la précision de la fonction selon le nombre de chiffres spécifiés dans le format d'affichage.

Pour spécifier la *précision* de l'intégration, définissez le format d'affichage de telle façon que l'affichage ne montre pas plus que le nombre de chiffres que vous estimez corrects pour les valeurs de l'intégrande. Le même niveau de précison se reflètera dans le résultat de l'intégration.

Interprétation de la précision

Après le calcul de l'intégrale, le calculateur place l'*incertitude* de ce résultat d'intégrale dans le registre Y. Appuyez sur [x≷y] pour voir la valeur d'incertitude.

Par exemple, si l'intégrale *Si*(2) est 1,6054 ± 0,0001, alors, 0,0001 est son incertitude.

Exemple : choix de la précision. Si le format d'affichage est réglé sur SCI 2, calculez l'intégrale de l'expression de *Si*(2) (reprise de l'exemple précédent).

Touches :	Affichage :	Description :
■ [DISP] {SC} 2	1,61E0	Choisit la notation scientifique avec 2 décimales (fonction précise jusqu'à 2 décimales).
0 [ENTER] 2	2_	Limites.
■ [SOLVE/∫] {FN} S	2,00E0	Choisit le sous-programme S.
■ [SOLVE/∫] {∫FN} X	∫=1,61E0	Intégrale, approximation à 2 décimales.

x≷y	1,00E-3	Incertitude de l'approximation de l'intégrale.

L'intégrale est 1,61 ± 0,00100. Puisque l'incertitude n'influence l'approximation qu'à partir de sa troisième décimale, vous pouvez considérer que tous les chiffres affichés dans cette approximation sont exacts.

Si l'incertitude d'une approximation est plus grande que ce que vous avez choisi d'admettre, vous pouvez élever le nombre de chiffres du format d'affichage et répéter l'intégration (pour autant que $f(x)$ soit encore calculée précisément en fonction du nombre de chiffres affichés). En général, l'incertitude de l'intégration diminue par un facteur de 10 pour chaque chiffre supplémentaire spécifié dans le format d'affichage.

Exemple : changement de la précision. Pour l'intégrale de $Si(2)$ que vous venez de calculer, spécifiez la précision à la quatrième décimale au lieu de la deuxième :

Touches :	Affichage :	Description :
■ DISP {SC} 4	1,0000E-3	Spécifie la précision à quatre décimales. L'incertitude de l'exemple précédent est toujours affichée.
R↓ R↓	2,0000E0	Redescend les limites de l'intégration des registres Z et T dans les registres X et Y.
■ SOLVE/∫ {∫FN} X	∫=1,6054E0	Le résultat.
x≷y	1,0000E-5	Notez que l'incertitude est à peu près 1/100 par rapport au résultat calculé avec SCI 2 précédemment.
■ DISP {FX} 4		Rétablit le format FIX 4.

Cette incertitude indique que le résultat *pourrait* n'être correct que pour quatre décimales. En fait, ce résultat est précis à la *septième* décimale lorsqu'il est comparé à la valeur réelle de cette intégrale. Parce que l'incertitude d'un résultat est calculée avec une marge d'erreur, *l'approximation du calculateur est en général plus précise que son incertitude ne l'indique.*

Pour plus de détails à ce sujet, lisez l'annexe D.

Utilisation de l'intégration dans un programme

Dans un programme, souvenez-vous de solliciter ou d'inclure les limites de l'intégration avant de l'exécuter et souvenez-vous que la précision et le temps d'exécution sont déterminés par le format d'affichage et le temps nécessaire au programme. Les deux instructions d'intégration apparaissent dans le programme sous la forme :

FN= *label*
∫FN d *variable*

Documentation des sorties. L'instruction *programmée* ∫FN ne documente pas l'affichage du résultat (∫=*valeur*), puisque qu'il pourrait ne pas produire une sortie significative de votre programme (vous pourriez désirer inclure ce nombre dans d'autres calculs). Si vous *désirez* que ce résultat soit affiché, ajoutez une instruction PSE (■ LBL/RTN {PSE}) ou STOP (R/S) pour afficher le résultat dans le registre X après l'instruction ∫FN.

Exemple : distribution normale. Le programme de la page 215, « Distributions normales et inverses-normales », comporte une intégration de l'équation pour la fonction de la densité normale,

$$\frac{1}{S\sqrt{2\pi}} \int_M^D e^{-\left(\frac{D-M}{S}\right)^2 \div 2} dD.$$

Cette fonction est définie dans le sous-programme F :

```
F01 LBL F
F02 RCL D
F03 RCL- M
F04 RCL÷ S
F05 x²
F06 2
F07 ÷
F08 +/-
F09 eˣ
F10 RTN
```

D'autres sous-routines sollicitent le valeurs connues et effectuent les autres calculs pour trouver $Q(D)$, la surface de l'extrémité supérieure d'une courbe ordinaire. L'intégration est établie et exécutée à partir du sous-programme Q :

```
Q01 LBL Q
Q02 RCL M          Rappelle la limite inférieure d'intégration.
Q03 RCL X          Rappelle la limite supérieure d'intégration
                   (X = D.)
Q04 FN= F          Spécifie la fonction définie par LBL F pour
                   intégration.
Q05 ∫FN d D        Intègre la fonction normale pour la variable
 ⋮                 D.
```

Limitations. L'instruction ∫FN *variable* ne peut appeler un sous-programme qui contienne une autre instruction ∫FN ; c'est-à-dire qu'elle ne peut être utilisée récursivement. Il est donc impossible de calculer de multiples intégrales (erreur ∫(∫FN)). Impossible également pour ∫FN d'invoquer un sous-programme qui contienne l'instruction FN= *label* (erreur ∫FNACTIVE). ∫FN ne peut invoquer de sous-programme contenant une instruction SOLVE (erreur ∫(SOLVE)), tout comme SOLVE ne peut invoquer de sous-programme contenant une instruction d'intégration (erreur SOLVE(∫FN)).

L'instruction ∫FN d *variable* placée dans un programme utilise l'un des sept retours de sous-routines disponibles dans le calculateur (Cf. « Sous-routines emboîtées », au chapitre 6).

Pour plus d'informations

Ce chapitre vous donne les instructions nécessaires pour utiliser l'intégration sur le HP–32S dans un grand nombre d'applications. Cependant, l'annexe D contient des informations plus détaillées sur le fonctionnement de l'algorithme, sur les conditions de calcul qui pourraient causer des résultats incorrects, les situations qui peuvent prolonger la durée des calculs et l'obtention de l'approximation d'une intégrale.

9

Opérations sur les nombres complexes

Le HP-32S utilise les nombres complexes sous la forme

$$x + iy.$$

Il contient des opérations pour l'arithmétique complexe (+, −, ×, ÷), la trigonométrie complexe (sin, cos, tan) et des fonctions mathématiques $-z$, $1/z$, $z_1{}^{z_2}$, ln z et e^z (dans laquelle z_1 et z_2 sont des nombres complexes).

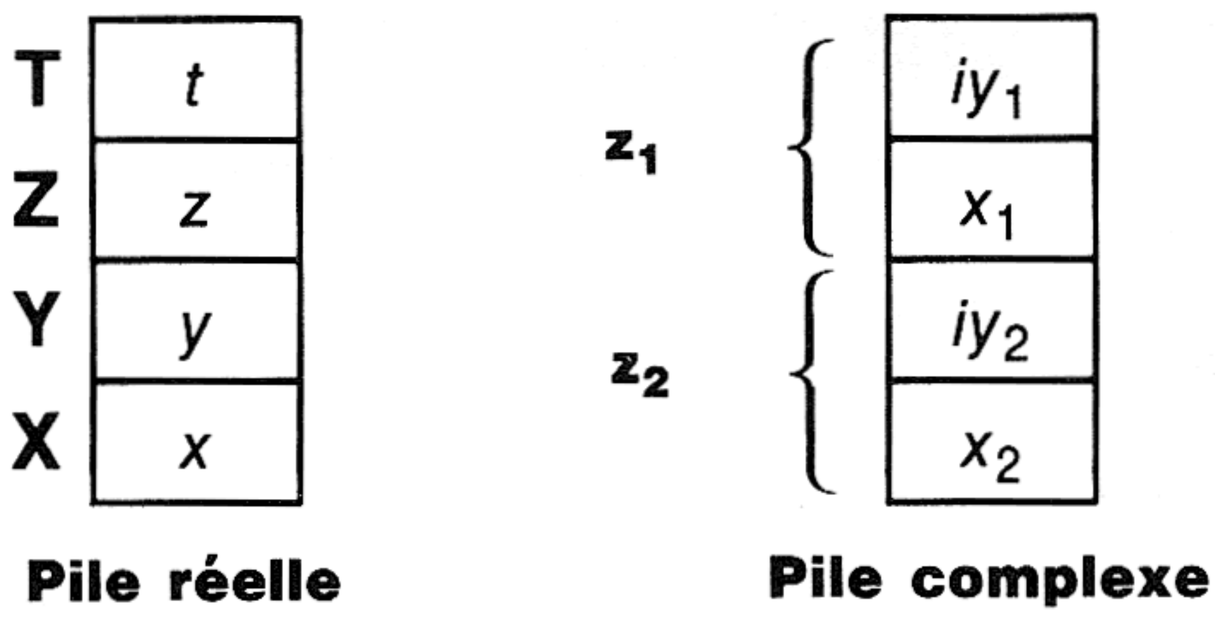
Les nombres complexes dans le HP–32S sont traités en saisissant leurs deux parties (imaginaire et réelle) de manière séparée. Pour saisir *deux* nombres complexes, vous saisissez *quatre* nombres distincts. Pour effectuer une opération sur ces nombres, appuyez sur ■ CMPLX avant l'opérateur. Par exemple, pour faire

$$(2 + i4) + (3 + i5),$$

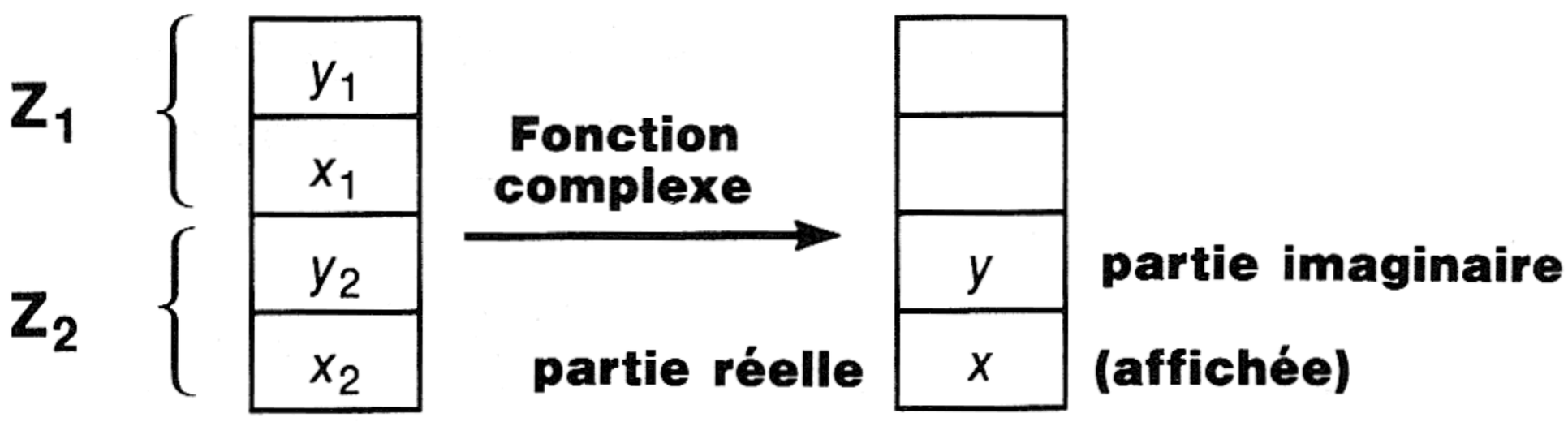
appuyez sur 4 ENTER 2 ENTER 5 ENTER 3 ■ CMPLX +. Le résultat est $5 + i9$ (appuyez sur x≷y pour voir la partie imaginaire).

La pile complexe

C'est en fait la pile de mémoire classique, partagée en deux doubles registres pour pouvoir traiter deux nombres complexes, $z_{1x} + iz_{1y}$ et $z_{2x} + iz_{2y}$:



Puisque les parties imaginaire et complexe d'un nombre sont saisies séparément, il est aisé de travailler avec l'une ou l'autre partie séparément.



Saisissez toujours la partie imaginaire (la partie y) *d'abord*. La partie réelle du résultat (z_x) est affichée ; appuyez sur [x≷y] pour visualiser la partie imaginaire (z_y).

Opérations complexes

Utilisez les opérations complexes de la même façon que les opérations réelles, mais faites précéder les opérateurs de la séquence ■ CMPLX.

Pour effectuer une opération sur un seul nombre complexe :

1. Saisissez le nombre complexe z, composé de $x + iy$, en frappant y ENTER x.
2. Choisissez le fonction complexe :

Fonctions pour un nombre complexe, z

To Calculate:	Appuyer sur :
Changer de signe, $-z$	■ CMPLX +/–
Inverse, $1/z$	■ CMPLX 1/x
Log naturel, ln z	■ CMPLX LN
Antilog. naturel, e^z	■ CMPLX e^x
Sin z	■ CMPLX SIN
Cos z	■ CMPLX COS
Tan z	■ CMPLX TAN

Pour effectuer une opération arithmétique avec deux nombres complexes :

1. Saisissez le premier nombre complexe, z_1 (composé de $x_1 + iy_1$) en frappant y_1 ENTER x_1 ENTER (pour $z_1{}^{z_2}$, saisissez la base, z_1, en premier lieu).
2. Saisissez le second nombre complexe, z_2, en frappant y_2 ENTER x_2 (pour $z_1{}^{z_2}$, frappez l'exposant, z_2, en second lieu).
3. Choisissez l'opération arithmétique :

Arithmétique avec deux nombres complexes, z_1 et z_2

Pour calculer :	Appuyer sur :
Addition, $z_1 + z_2$	■ CMPLX +
Soustraction, $z_1 - z_2$	■ CMPLX −
Multiplication, $z_1 \times z_2$	■ CMPLX ×
Division, $z_1 \div z_2$	■ CMPLX ÷
Fonction de puissance, $z_1^{z_2}$	■ CMPLX y^x

Exemples. Voici quelques exemples de trigonométrie et d'arithmétique avec les nombres complexes :

Evaluez sin $(2 + i3)$.

Touches :	Affichage :	Description :
3 ENTER 2		
■ CMPLX SIN	9,1545	Partie réelle du résultat.
x≷y	−4,1689	Le résultat est $9{,}1545 - i4{,}1689$.

Evaluez l'expression

$$z_1 \div (z_2 + z_3),$$

dans laquelle $z_1 = 23 + i13, \quad z_2 = -2 + i, \quad z_3 = 4 - i3.$

Puisque la pile ne peut conserver que deux nombres complexes en même temps, exécutez le calcul sous la forme :

$$z_1 \times [1 \div (z_2 + z_3)].$$

Touches :	Affichage :	Description :
1 [ENTER] 2 [+/–]		Ajoute $z_2 + z_3$; affiche
[ENTER] 3 [+/–] [ENTER] 4		la partie réelle.
■ [CMPLX] [+]	2,0000	
■ [CMPLX] [1/x]	0,2500	$1 \div (z_2 + z_3)$.
13 [ENTER] 23		$z_1 \div (z_2 + z_3)$.
■ [CMPLX] [×]	2,5000	
[x≷y]	9,0000	Le résultat est $2{,}5 + i9$.

Evaluez $(4 - i2/5)\,(3 - i2/3)$. N'utilisez pas d'opérations complexes en calculant une partie seulement du nombre complexe.

Touches :	Affichage :	Description :
2 [ENTER] 5 [÷] [+/–]	−0,4000	Calcule la partie imaginaire en utilisant des opérations réelles.
4 [ENTER]	4,0000	Saisit la part réelle du premier nombre complexe.
2 [ENTER] 3 [÷] [+/–]	−0,6667	Calcule la partie imaginaire du second.
3 ■ [CMPLX] [×]	11,7333	Termine la saisie du second nombre, puis multiplie les deux nombres complexes.
[x≷y]	−3,8667	Le résultat est 11,7333 $- i3{,}8667$.

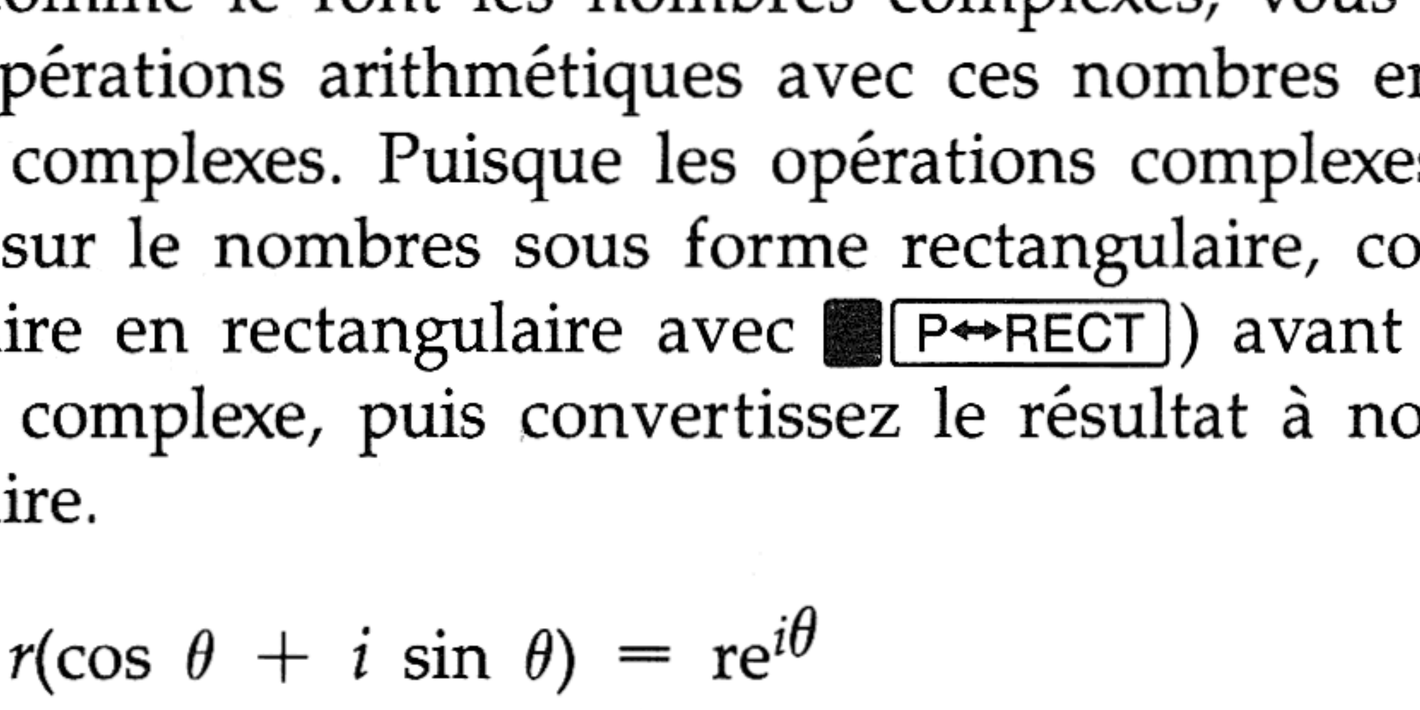
Evaluez $e^{z^{-2}}$, où $z = (1 + i)$. Utilisez ■ [CMPLX] [y^x] pour évaluer z^{-2} ; saisissez -2 sous la forme $-2 + i0$.

Touches :	Affichage :	Description :
1 ENTER 1 ENTER 0 ENTER 2 +/–		Résultat intermédiaire de $(1 + i)^{-2}$.
■ CMPLX e^x	0,8776	Partie réelle du résultat final.
y^x	-0,4794	Le résultat final est $0{,}8776 - i0{,}4794$.

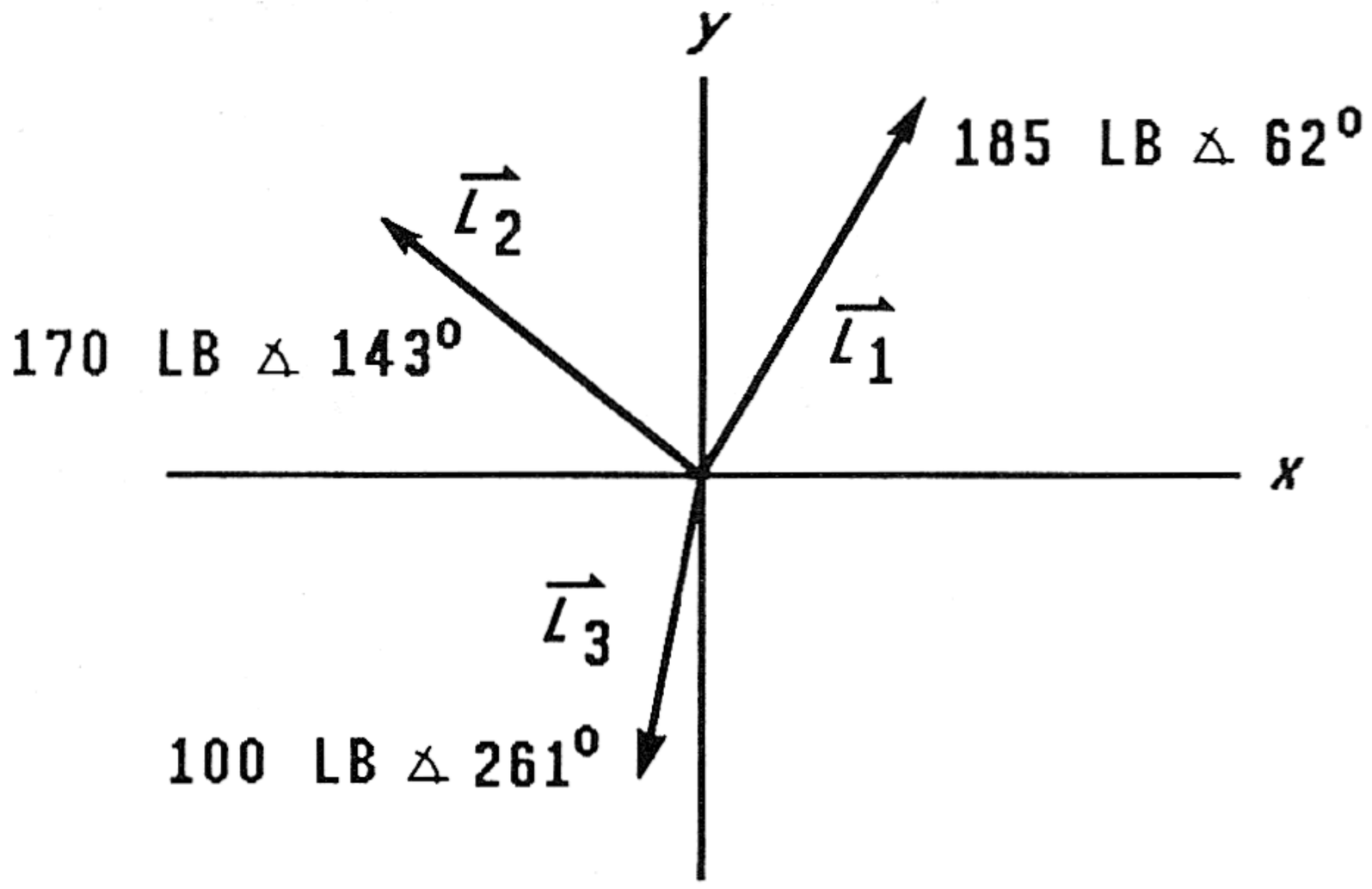
Utilisation de nombres en notation polaire

Beaucoup d'applications utilisent des nombres reéls sous forme *polaire,* ou en en notation *phasor*. Ces formes utilisent des paires de nombres, comme le font les nombres complexes, vous pouvez donc faire des opérations arithmétiques avec ces nombres en utilisant les opérations complexes. Puisque les opérations complexes du HP–32S travaillent sur le nombres sous forme rectangulaire, convertissez la forme polaire en rectangulaire avec ■ P↔RECT) avant d'exécuter l'opération complexe, puis convertissez le résultat à nouveau en forme polaire.

$$a + ib = r(\cos\theta + i \sin\theta) = re^{i\theta}$$
$$= r \angle \theta \quad \text{Forme polaire ou phasor}$$



Exemple : addition de vecteur. Ajoutez les trois charges suivantes. Commencez par convertir les coordonnées polaires en rectangulaires.



Touches :	Affichage :	Description :
■ MODES {DG}		Définit le mode degrés.
62 ENTER 185		
■ P↔RECT {θ,r→y,x}	86,8522	Saisit $\mathbf{L}_1$ et le convertit sous forme rectangulaire.
143 ENTER 170		Saisit et convertit $\mathbf{L}_2$.
■ P↔RECT {θ,r→y,x}	-135,7680	
■ CMPLX +	-48,9158	Ajoute les vecteurs.
261 ENTER 100		Saisit et convertit $\mathbf{L}_3$.
■ P↔RECT {θ,r→y,x}	-15,6434	
■ CMPLX +	-64,5592	Additionne $\mathbf{L}_1 + \mathbf{L}_2 + \mathbf{L}_3$.
■ P↔RECT {y,x→θ,r}	178,9372	Convertit les vecteurs sous forme polaire ; affiche *r*.
x≷y	111,1489	Affiche θ.

10

Conversions de bases et arithmétique

Le menu BASE (■ BASE) permet de changer la base utilisée pour saisir les nombres et les autres opérations (y compris la programmation). Les changements de base changent également l'*affichage* des nombres.

Le menu BASE

Label de menu	Description
{DEC}	*Mode décimal*. Pas de témoin. Convertit les nombres en base 10. Les nombres ont une partie entière et une partie fractionnaire.
{HX}	*Mode hexadécimal*. Témoin **HEX** affiché. Convertit les nombres en base 16 ; utilise uniquement des entiers. Les touches du premier rang deviennent les chiffres de A à F.
{OC}	*Mode octal*. Témoin **OCT** affiché. Convertit les nombres en base 8 ; n'utilise que des entiers. Les touches 8, 9 et du premier rang non-préfixées sont inactives.
{BN}	*Mode binaire*. Témoin **BIN** affiché. Convertit les nombres en base 2 ; n'utilise que des entiers. Les touches de chiffres autres que 0 et 1 et les fonctions non préfixées du premier rang de touches sont inactives. Si un nombre est plus long que 12 chiffres, les deux touches extérieures du premier rang ($\sqrt{x}$ et $\Sigma+$) sont actives pour visualiser les fenêtres (voir « Fenêtres pour les nombres binaires longs »

Examples: conversion de la base d'un nombre. Les frappes suivantes effectuent diverses conversions de bases.

Convertissez $125{,}99_{10}$ en nombres hexadécimaux, octaux et binaires.

Touches :	Affichage :	Description :
125,99 ■ BASE {HX}	7D	Convertit la partie entière seulement (125) du nombre décimal en base 16 et affiche cette valeur.
■ BASE {OC}	175	Base 8.
■ BASE {BN}	1111101	Base 2.
■ BASE {DEC}	125,9900	Rétablit la base 10 ; la valeur décimale d'origine a été conservée, y compris sa partie fractionnaire.

Convertissez $24FF_{16}$ en base binaire. Le nombre binaire sera plus long que 12 chiffres (la longueur maximale).

■ BASE {HX} 24FF	24FF_	Utilise la touche Σ+ pour frapper « F ».
■ BASE {BN}	010011111111	Le nombre binaire est trop grand. Le témoin ← indique que le nombre continue vers la gauche ; le témoin ▼ pointe vers √x.
√x	10	Affiche le reste du nombre. Le nombre entier est 100100111111_2.
Σ+	010011111111	Affiche à nouveau les 12 premeirs chiffres.
■ BASE {DEC}	9.471,0000	Retour à la base 10.

Arithmétique en bases 2, 8 et 16

Vous pouvez exécuter des opérations arithmétiques en utilisant [+], [−], [×] et [÷] dans n'importe quelle base.* L'arithmétique dans les bases 2, 8 et 16 est en forme complément à 2 et n'utilise que des entiers :

- Si un nombre possède une partie fractionnaire, seule la partie entière est utilisée pour le calcul.
- Le résultat d'une opération est toujours un entier (toute partie fractionnaire est éliminée).

Alors que les conversions ne changent que l'affichage du nombre et non le nombre présent dans le registre X, l'arithmétique *modifie* le nombre du registre X.

Si le résultat d'une opération ne peut être représenté sur 36 bits, l'affichage indique `OVERFLOW` et le plus grand nombre positif ou négatif possible.

Exemples. Un peu d'arithmétique en modes hexadécimal, octal et binaire :

$$12F_{16} + E9A_{16} = ?$$

Touches :	**Affichage :**	**Description :**
■ [BASE] {HX}		Définit la base 16 ; le témoin **HEX** est affiché.
12F [ENTER] E9A [+]	FC9	Résultat.

* Les seules touches de fonction qui sont inactivées en dehors du mode décimal sont [$\sqrt{x}$], [e^x], [LN], [y^x], [1/x], [Σ+]. Cependant, la plupart des opérations autres qu'arithmétiques ne produiront pas de résultats significatifs puisque les parties fractionnaires des nombres sont supprimées.

$$7760_8 - 4326_8 = ?$$

■ BASE {OC}	7711	Définit la base 8 ; le témoin **OCT** apparaît. Convertit le nombre affiché en sa forme octale.
7760 ENTER 4326 −	3432	Résultat.

$$100_8 \div 5_8 = ?$$

100 ENTER 5 ÷	14	Partie entière du résultat.

$$5A0_{16} + 1001100_2 = ?$$

■ BASE {HX} 5A0	5A0_	Définit la base 16 ; témoin **HEX** affiché.
■ BASE {BN} 1001100	1001100_	Définit la base 2 ; témoin **BIN** affiché. Ceci met fin à la saisie de chiffres, ENTER n'est pas nécessaire entre les nombres.
+	10111101100	Résultat en base binaire.
■ BASE {HX}	5EC	Résultat en base hexadécimale.
■ BASE {DC}	1.516,0000	Rétablit la base décimale.

La représentation de nombres

Bien que l'*affichage* d'un nombre soit converti lorsque la base est changée, sa forme stockée ne l'est pas, les nombres décimaux ne sont donc pas raccourcis–jusqu'à ce qu'ils soient utilisés en arithmétique.

Lorsqu'un nombre apparaît sous forme hexadécimale, octale ou binaire, il est affiché en tant que nombre entier, justifié à droite, avec un maximum de 36 bits (12 chiffres octaux ou 9 chiffres hexadécimaux). Les zéros de tête ne sont pas affichés, mais ils sont importants parce qu'ils indiquent un nombre positif. La représentation binaire de 125_{10} est :

1111101

ce qui équivaut à ces 36 chiffres :

000000000000000000000000000001111101

Nombres négatifs

Le nombre *bit* placé le plus à gauche (le plus significatif) est le bit de signe ; il est armé (1) pour les nombres négatifs. S'il y a des zéros de tête, qui ne sont pas affichés, le bit de signe est 0 (positif). Un nombre négatif est le complément à deux de son nombre binaire positif.

Touches :	**Affichage :**	**Description :**
546 ■ BASE {HX}	222	Saisit un nombre positif, décimal, puis le convertit en base hexadécimale.
+/−	FFFFFFDDE	Complément à deux (changement de signe).
■ BASE {BN}	11011101111 0	Version binaire ; ← indique la présence d'autres chiffres.
√x √x	11111111111 1	Affiche la partie gauche ; le nombre est négatif, le bit le plus élevé est 1.
■ BASE {DEC}	-546,0000	Nombre décimal négatif.

Plage de nombres

La taille de mots à 36 bits détermine la plage de nombres qui peuvent être représentés sous forme hexadécimale (9 chiffres), octale (12 chiffres) et binaire (36 chiffres), ainsi que la plage de nombres décimaux (11 chiffres) qui peuvent être convertis en ces autres bases.

Plage de nombres pour conversion de bases

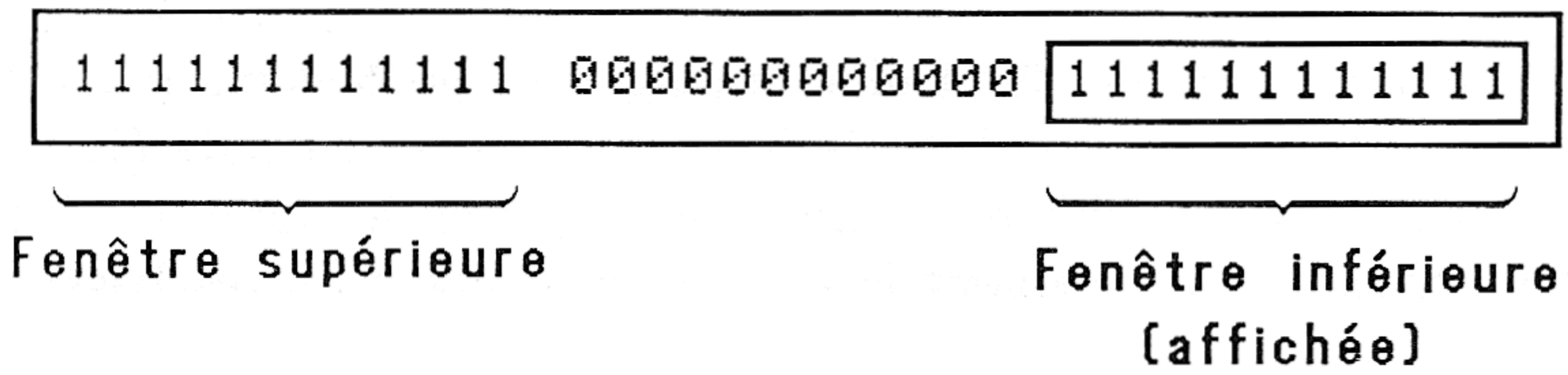
Base	Nombre entier positif le plus grand	Nombre entier négatif le plus grand
Hexadécimale	7FFFFFFFF	800000000
Octale	377777777777	400000000000
Binaire	011111111111111111 111111111111111111	100000000000000000 000000000000000000
Décimale	34.359.738.367	−34.359.738.368

Lorsque vous saisissez des nombres, le calculateur n'accepte pas plus que le nombre maximal admis pour chaque base. Si vous essayez de frapper un nombre hexadécimal de 10 chiffres, la saisie est interrompue et le témoin ⚠ apparaît.

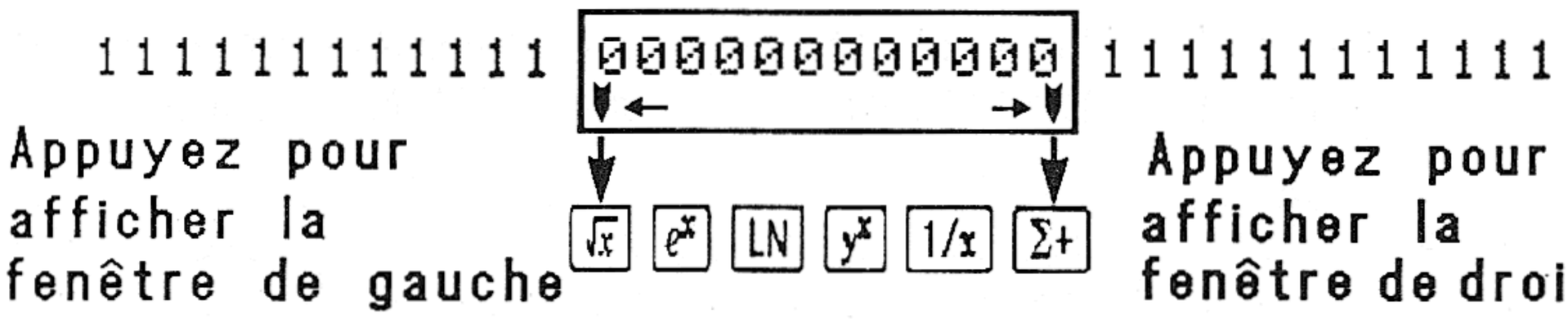
Si un nombre saisi en base décimale est en dehors de la plage définie ci-dessus, il causera l'affichage d'un message `TOO BIG` dans les autres modes de base. Toute opération utilisant `TOO BIG` provoquera un dépassement de capacité, et le HP-32S substituera le plus grand nombre positif ou négatif possible pour ce nombre.

Fenêtres pour les nombres binaires longs

Le nombre binaire le plus long compte 36 chiffres—trois fois plus que ce que contient l'affichage. Chaque affichage de 12 chiffres d'un long nombre est nommé une *fenêtre*.



Lorsque la longueur d'un nombre binaire excède 12 chiffres, le témoin ← ou → apparaît, ou tous deux apparaissent, indiquant de quel(s) côté(s) se trouve(nt) le reste des chiffres. Appuyez sur la touche [√x] ou sur la touche [Σ+] pour voir la fenêtre cachée.



Visualisation de nombres partiallement cachés

Les fonctions ■ [VIEW] et ■ [INPUT] agissent sur les nombres non-décimaux comme sur les nombres décimaux. Cependant, si le nombre octal ou binaire complet est trop grand pour être affiché, les chiffres de *tête* sont remplacés par des points de suspension (...). Appuyez sur ■ [SHOW] pour visualiser les chiffres cachés par A=... ou par A?... .

Touches :	**Affichage :**	**Description :**
■ [BASE] {OC}		Saisit un long nombre octal.
123456712345	23456712345_	
[STO] A	123456712345	

■ VIEW A	A=...456712345	
		Ignore les 3 chiffres placés le plus à gauche.
■ SHOW *(hold)*	123456712345	Montre 12 chiffres.

Programmation avec BASE

Les instructions de changement de base sont programmables avec ■ BASE. Leur fonctionnement est identique ; les programmes acceptent des nombres dans chacune de ces bases, ils permettent d'effectuer des calculs arithmétiques dans toutes ces bases, et aussi d'afficher des résultats sous n'importe quelle forme.

Lors de la création de programmes utilisant une autre base que 10, définissez le mode de base à la fois comme *mode en cours* pour le calculateur et *dans le programme* (en tant qu'instruction).

Choix d'un mode de base dans un programme

Insérez une instruction BIN, OCT ou HEX en début de programme. Il est habituellement conseillé d'ajouter une instruction DEC en fin de programme pour remettre le calculateur en mode décimal.

Une instruction de changement de base incluse dans un programme détermine l'aspect que prendront les sorties *pendant et après l'exécution du programme,* mais elle n'affecte *pas* les lignes de programmes lors de la saisie.

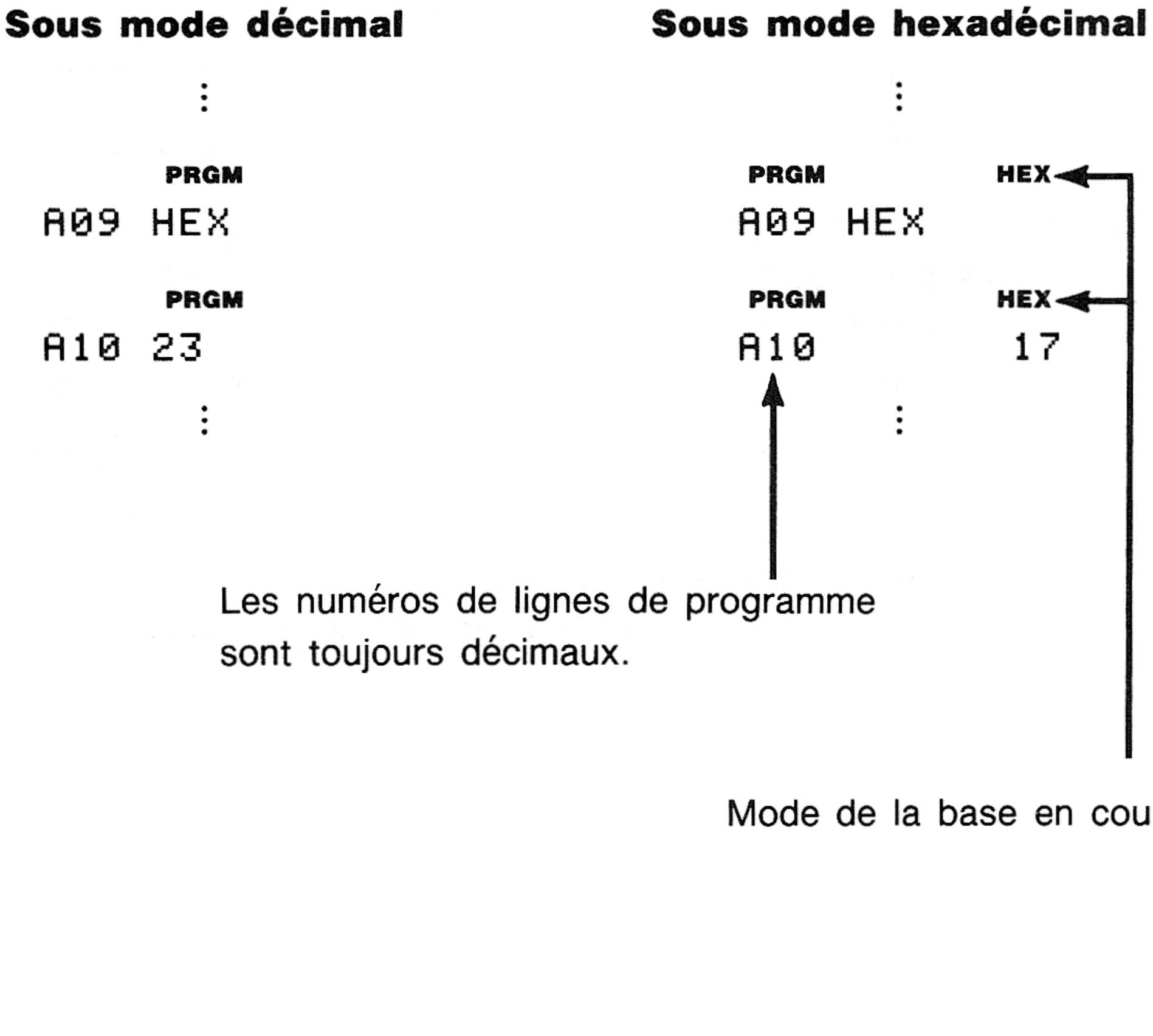
Les opérations SOLVE et ∫FN déterminent automatiquement le mode DEC.

Nombres saisis en lignes de programme

Avant de commencer la saisie d'un programme, définissez la base. Le réglage en cours détermine la base des nombres qui sont saisis dans les lignes de programme. Leur affichage change avec le changement de mode de base.

Les numéros de ligne apparaissent toujours en base 10.

Un témoin vous indique le réglage en cours. Comparez par exemple les lignes de programme ci-dessus, dans les colonnes de droite et de gauche. Remarquez que le nombre hexadécimal, comme tous les nombres non-décimaux, est justifié à droite.

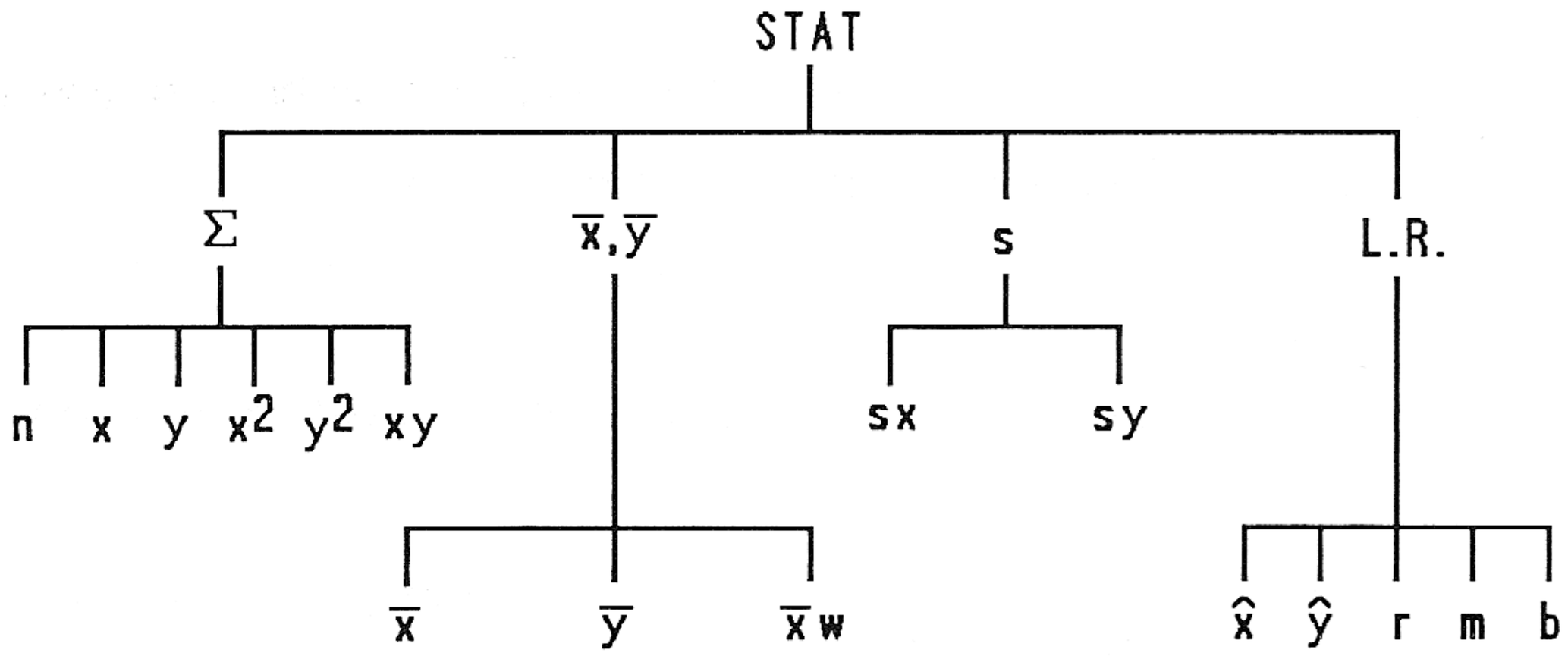


11

Statistiques

Le menu STAT permet d'analyser statistiquement des ensembles de données à une ou deux variables.

- Données à une seule variable : moyenne et écart-standard.
- Données à deux variables (x,y) : régression linéaire et estimation linéaire ($\hat{x}$ and $\hat{y}$).
- Moyenne pondérée (x pondéré par y).
- Statistiques de sommation : n, Σx, Σy, Σx^2, Σy^2 et Σxy.



Saisie de données statistiques (Σ+ , ■ Σ−)

Les données à une et à deux variables sont saisies de même façon. Les valeurs des données sont accumulées comme des statistiques de sommation dans six *registres statistiques* ; leurs valeurs sont affichées par la séquence ■ STAT {Σ} .

Saisie de données à une seule variable

1. Appuyez sur ■ CLEAR {Σ} pour effacer les données statistiques précédentes.
2. Frappez chaque valeur de x et appuyez sur Σ+.
3. L'affichage montre n, le nombre de valeurs statistiques accumulées.*

Pour rappeler une valeur à l'affichage immédiatement après la saisie, appuyez sur ■ LASTx.

Saisie de données à deux variables

Lorsque les données se composent de deux variables, x est la *variable indépendante* et y la *variable dépendante*. Souvenez-vous d'inscrire une paire (x, y) en ordre inverse pour que y aille se placer dans le registre Y et x dans le registre X.

1. Appuyez sur ■ CLEAR {Σ} pour effacer les données statistiques précédentes.
2. Frappez la valeur y en *premier* lieu et appuyez sur ENTER.
3. Frappez la valeur x correspondante et appuyez sur Σ+.
4. L'affichage montre n, le nombre de paires statistiques actuellement accumulées.
5. Continue la saisie de paires x-y. La valeur n est mise à jour à chaque saisie.

Pour rappeler une valeur de x à l'affichage immédiatement après la saisie, appuyez sur ■ LASTx.

* Cette procédure saisit en fait deux variables dans les registres statistiques parce que la valeur déjà présente dans le registre Y est sauvegardée en tant que valeur y. Pour cette raison, la calculateur effectuera une régression linéaire et montrera les valeurs basées sur y même lorsque vous n'avez saisi que des valeurs x—ou même lorsque vous avez saisi un nombre inégal de valeurs x et de valeurs y. Aucune erreur ne se manifeste, mais les résultats sont évidemment vides de sens.

Correction des erreurs de saisie

Si vous faites une erreur de saisie, supprimez la donnée incorrecte et ajoutez la donnée juste. Même s'il n'y a qu'une seule erreur dans une paire *x*-*y*, il faut supprimer et ressaisir les deux valeurs.

Pour corriger des données statistiques :

1. Resaisissez la version corrigée de la valeur, mais au lieu d'appuyer sur [Σ+], appuyez sur ■[Σ−]. Ceci supprime les valeurs et diminue *n*.
2. Saisissez la valeur correcte en utilisant [Σ+].

Si les valeurs incorrectes sont celles que vous *venez* de saisir, appuyez sur ■[LASTx] pour les récupérer, puis sur ■[Σ−] pour les détruire (la valeur *y* incorrecte était toujours dans le registre Y et sa valeur *x* a été sauvegardée dans le registre LAST*x*).

Exemple.

x et *y* initiales	*x*, *y* corrigées
20, 4	20, 5
400, 6	40, 6

Touches :	Affichage :	Description :
■[CLEAR] {Σ}		Efface les données précédentes, puis saisit 2 paires de données. L'affichage montre *n*, le nombre de paires saisies.
4 [ENTER] 20 [Σ+]	1,0000	
6 [ENTER] 400 [Σ+]	2,0000	
■[LASTx]	400,0000	Rappelle la dernière valeur de *x*. Le dernier *y* est toujours dans le registre Y (appuyez deux fois sur [x≷y]).
■[Σ−]	1,0000	Remplace la dernière paire de données par 40 et 6.
6 [ENTER] 40 [Σ+]	2,0000	

4 ENTER 20 ■ Σ−	1,0000	Supprime et remplace la première paire (20, 4 à changer en 20, 5). Deux paires au total.
5 ENTER 20 Σ+	2,0000	

Calculs statistiques

Une fois que vous avez saisi les données statistiques, vous pouvez utiliser les données du menu STAT. Appuyez sur ■ STAT pour l'afficher.

Le menu STAT

Label de menu	Description
{Σ}	Le menu de sommation : n, Σx, Σy, Σx^2, Σy^2, Σxy. Voyez « Statistiques de sommation ».
{x̄,ȳ}	Le menu de moyenne : $\bar{x}$, $\bar{y}$ et $\bar{x}$ pondéré ($\bar{x}w$). Voyez « Ecart-type et moyen ».
{s}	Le menu écart type : s_x et s_y. Voyez « Moyenne et écart-type »."
{L.R.}	Le menu de régression linéaire (*linear-regression*) : ajustement de courbes (r, m, b) et estimation linéaire ($\hat{x}$, $\hat{y}$). Voyez « Régression linéaire ».

Moyenne et écart-type

Le menu moyenne ($\bar{x}$, $\bar{y}$).

- Appuyez sur ■ STAT {x̄,ȳ} {x̄} pour la moyenne arithmétique des valeurs x.
- Appuyez sur ■ STAT {x̄,ȳ} {ȳ} pour la moyenne arithmétique des valeurs y.

- Appuyez sur ■ STAT {x̄,ȳ} {x̄w} pour obtenir la moyenne pondérée des valeurs x en utilisant les valeurs y comme pondération ou comme féquences. Les nombres peuvent être entiers ou non-entiers.

Le menu « écart-type » (s). C'est la mesure de la dispersion des valeurs de données par rapport à la moyenne.

- Appuyez sur ■ STAT {s} {sx} pour l'écart-type des valeurs de x.*
- Appuyez sur ■ STAT {s} {sy} pour l'écart-type des valeurs y.*

Exemple : moyenne et écart-type avec une seule variable. Un chef de production désire connaître la durée d'un certain processus. Il choisit dix personnes au hasard et observe la façon dont chacun d'eux exécute ce processus et enregistre le temps nécessaire :

15,5	9,25
12,5	12,0

10,0
8,5

Calcule la moyenne et l'écart-type des durées (traitez-les comme valeurs x).

Touches :	Affichage :	Description :
■ CLEAR {Σ}		Efface les registres statistiques.
15,5 Σ+	1,0000	Saisit la première durée.
9,25 Σ+ 10 Σ+ 12,5 Σ+ 12 Σ+ 8,5 Σ+	3,0000 5,0000 6,0000	Saisit le reste des données.
■ STAT {x̄,ȳ} {x̄}	11,2917	Calcule la moyenne.

* Ceci calcule l'écart-type échantillonné (en utilisant n-1 comme diviseur), ce qui suppose que les données constituent un échantillon d'un ensemble de données plus grand et plus complet. Si vos données constituent l'ensemble des données, l'*écart-type de la population* peut être trouvé en calculant la moyenne des données initiales, et en ajoutant la moyenne aux données statistiques en utilisant Σ+ et en calculant ensuite l'écart-type.

■ STAT {s} {sx} 2,5808 Calcule l'écart-type.

Exemple : moyenne pondérée. Une fabrique achète une pièce quatre fois par an. L'an dernier, les achats s'échelonnaient de cette façon :

Prix de la pièce (x)	$4,25	$4,60	$4,70	$4,10
Nombre de pièces (y)	250	800	900	1000

Trouvez le prix moyen payé pour cette pièce. Introduisez *y*, le poids (la fréquence), avant *x*, le prix.

Touches :	**Affichage :**	**Description :**
■ CLEAR {Σ}		Efface les registres statistiques.
250 ENTER 4,25 Σ+	1,0000	Saisissez les données
800 ENTER 4,6 Σ+	2,0000	et leur poids.
900 ENTER 4,7 Σ+	3,0000	
1000 ENTER 4,1 Σ+	4,0000	
■ STAT {x̄,ȳ} {x̄w}	4,4314	Calcule le prix moyen pondéré par la quantité achetée.

Régression linéaire

La régression linéaire (nommée aussi *estimation linéaire*) est une méthode statistique pour établir une droite qui corresponde au plus près d'une série de données, *x* et *y*. Saisissez vos valeurs avant d'utiliser ces fonctions.

- Pour trouver une valeur estimée pour *x* (or *y*), introduisez *d'abord* une valeur hypothétique pour *y* (ou *x*), puis appuyez sur ■ STAT {L.R.} {x̂} (ou sur {ŷ}).
- Pour trouver les valeurs qui définissent la ligne qui accommode le mieux vos données, appuyez sur ■ STAT {L.R.} suivie de {r}, {m} ou {b}.

Le menu régression linéaire (Linear Regression, L.R.)

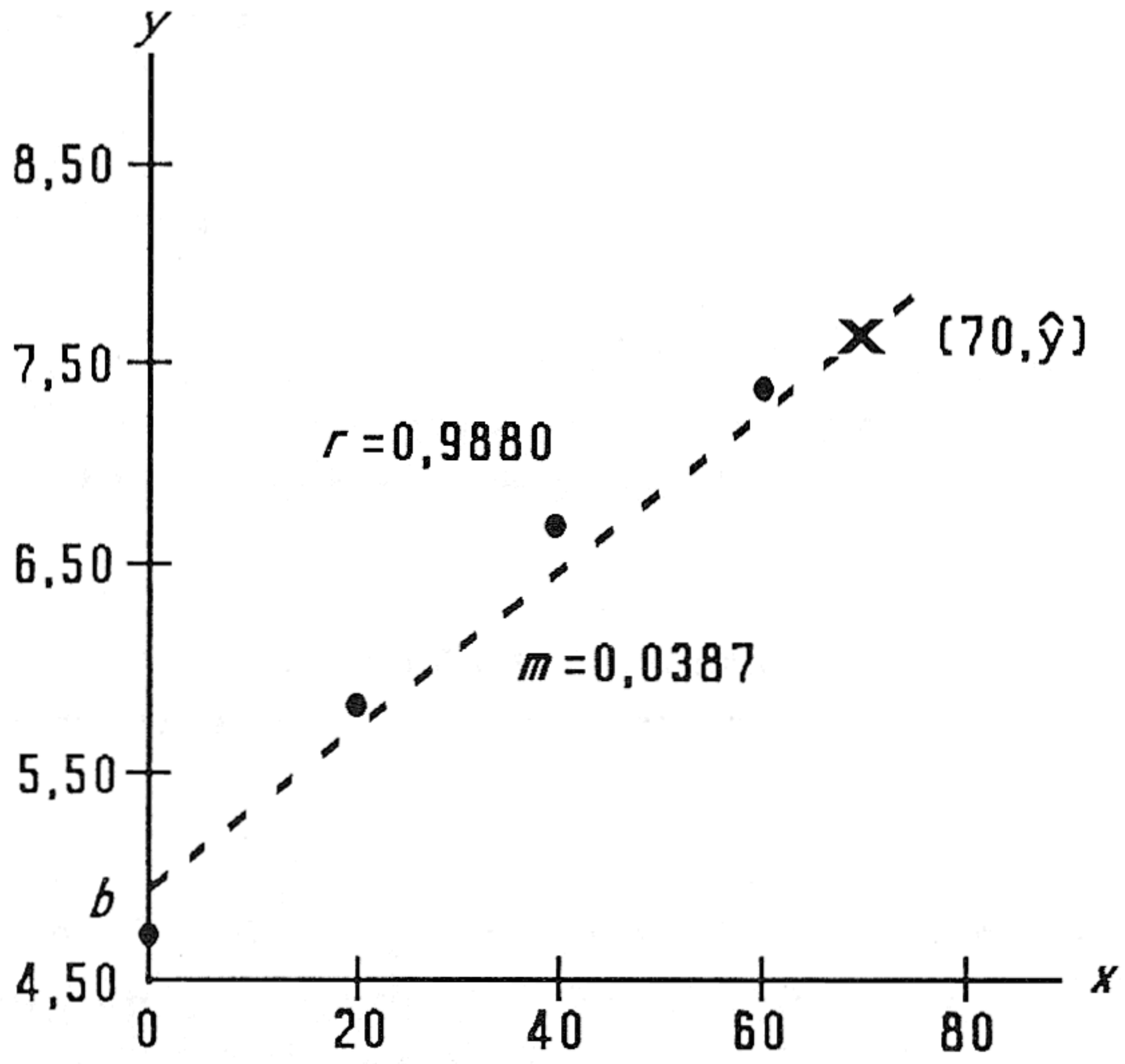
Label de menu	Description
{x̂}	Estime *x* pour une valeur hypothétique de *y*, basée sur une droite calculée pour qu'eile corresponde aux données.
{ŷ}	Estime *y* pour une valeur hypothétique de *x*, basée sur une droite calculée pour qu'elle corresponde aux données.
{r}	Coefficient de corrélation pour les données *x* et *y*. C'est un nombre compris entre −1 et +1 qui mesure la façon dont la droite correspond aux données.
{m}	Pente de la droite.
{b}	Ordonnée à l'origine de la droite.

Exemple : ajustement de courbe. Le rendement d'une nouvelle espèce de riz dépend de son rythme de fertilisation par l'azote. Pour les données suivantes, déterminez la relation inéaire : déterminez le coefficient de corrélation, la pente et la coordonnée à l'origine.

X, azote répandue (kg par hectare)	0,00	20,00	40,00	60,00	80,00
Y, rendement du grain (tonnes par hectare)	4,63	5,78	6,61	7,21	7,78

Touches :	**Affichage :**	**Description :**
■ CLEAR {Σ}		Efface le registre.
4,63 ENTER 0 Σ+		Introduit les données ; affiche *n* : 5 paires de données ont été saisies.
5,78 ENTER 20 Σ+		
6,61 ENTER 40 Σ+		
7,21 ENTER 60 Σ+		
7,78 ENTER 80 Σ+	5,0000	
■ STAT {L.R.}	x̂ ŷ r m b	Affiche le menu (L.R.).
{r}	0,9880	Coefficient de corrélation ; les données sont très proches d'une ligne droite.

■ STAT {L.R.} {m}	0,0387	Pente de la droite.
■ STAT {L.R.} {b}	4,8560	Ordonnée à l'origine.



Que se passerait-il si 70 kg d'azote étaient répandus sur le champ de riz ? Estimez la production sur base des statistiques précédentes.

Touches :	**Affichage :**	**Description :**
70	70_	Saisit la valeur de x hypothétique.
■ STAT {L.R.} {ŷ}	7,5615	La production estimée, en tonnes par hectare.

Limitations de la précision des données

Puisque le calculateur est d'une précision limitée à 12 ou 15 chiffres, il s'ensuit une erreur d'arrondi. En voici deux exemples :

Normalisation de nombres importants et proches. Il est possible que le calculateur soit incapable de calculer correctement l'écart-type et la régression linéaire pour une variable dont les valeurs de données varient d'une façon relativement peu importante. Pour tourner cette difficulté, *normalisez* les données en introduisant chaque valeur en tant que différence par rapport à une valeur centrale (telle que la moyenne). Pour les valeurs normalisées de x, cette différence doit être ajoutée aux calculs de $\bar{x}$ et de $\hat{x}$ et $\hat{y}$ et b doivent également être ajustées. Par exemple, si vos valeurs de x étaient 7 776 999, 7 777 000 et 7 777 001, vous devez introduire ces données sous la forme -1, et 1 ; puis ajouter 7 777 000 à $\bar{x}$ et à $\hat{x}$. Pour b, ajoutez 7 777 000 × m. Pour calculer $\hat{y}$, assurez-vous de fournir une valeur de x qui soit inférieure à 7 777 000.

De telles inexactitudes peuvent également se produire si les valeurs de x et de y diffèrent de façon importante. Ici aussi, la solution est de mettre ces données à l'échelle.

Effets de données retardées. Le fait d'exécuter ■ Σ− n'élimine aucune erreur d'arrondi qui a pu se produire dans les registres statistiques à cause des valeurs de données originales. Cette différence est négligeable jusqu'à ce que les données incorrectes prennent une importance énorme par rapport aux données correctes ; dans ce cas il est à conseiller d'effacer les registres et de recommencer la saisie des données.

Les valeurs de sommation et les registres statistiques

Les registres statistiques ont six emplacements uniques dans la mémoire qui stockent l'accumulation des six valeurs de sommation.

Statistiques de sommation

L'appui sur ■ STAT {Σ} donne accès au contenu des registres :

- Appuyez sur {n} pour voir le nombre d'ensembles de données accumulés.
- Appuyez sur {x} pour voir la somme des valeurs x.
- Appuyez sur {y} pour voir la somme des valeurs y.

- Appuyez sur {x²}, {y²} et sur {xy} pour lire la somme des carrés et la somme des produits, toutes valeurs pertinentes pour le calcul d'autres calculs statistiques que ceux que fournit le calculateur.

Registres statistiques dans la mémoire du calculateur

L'espace mémoire pour les registres statistiques (48 octets) est attribué automatiquement (s'il n'existe pas encore) au moment où vous appuyez sur [Σ+] ou sur [Σ−]. Ils sont supprimés lorsque vous exécutez ■[CLEAR] {Σ}, et l'espace mémoire est à nouveau disponible.

Si la mémoire du calculateur n'est pas assez grande pour contenir les registres statistiques lorsque vous appuyez sur [Σ+] (ou sur [Σ−]) pour la première fois, le calculateur affiche MEMORY FULL. Il vous faudra effacer des variables ou des programmes (ou bien des variables *et* des programmes) pour faire place aux registres statistiques. Consultez « Gestion de la mémoire du calculateur », à l'annexe B, en page 253.

Quatrième partie

Programmes d'application

Page 164 12 : Programmes mathématiques

204 13 : Programmes statistiques

222 14 : Programmes divers

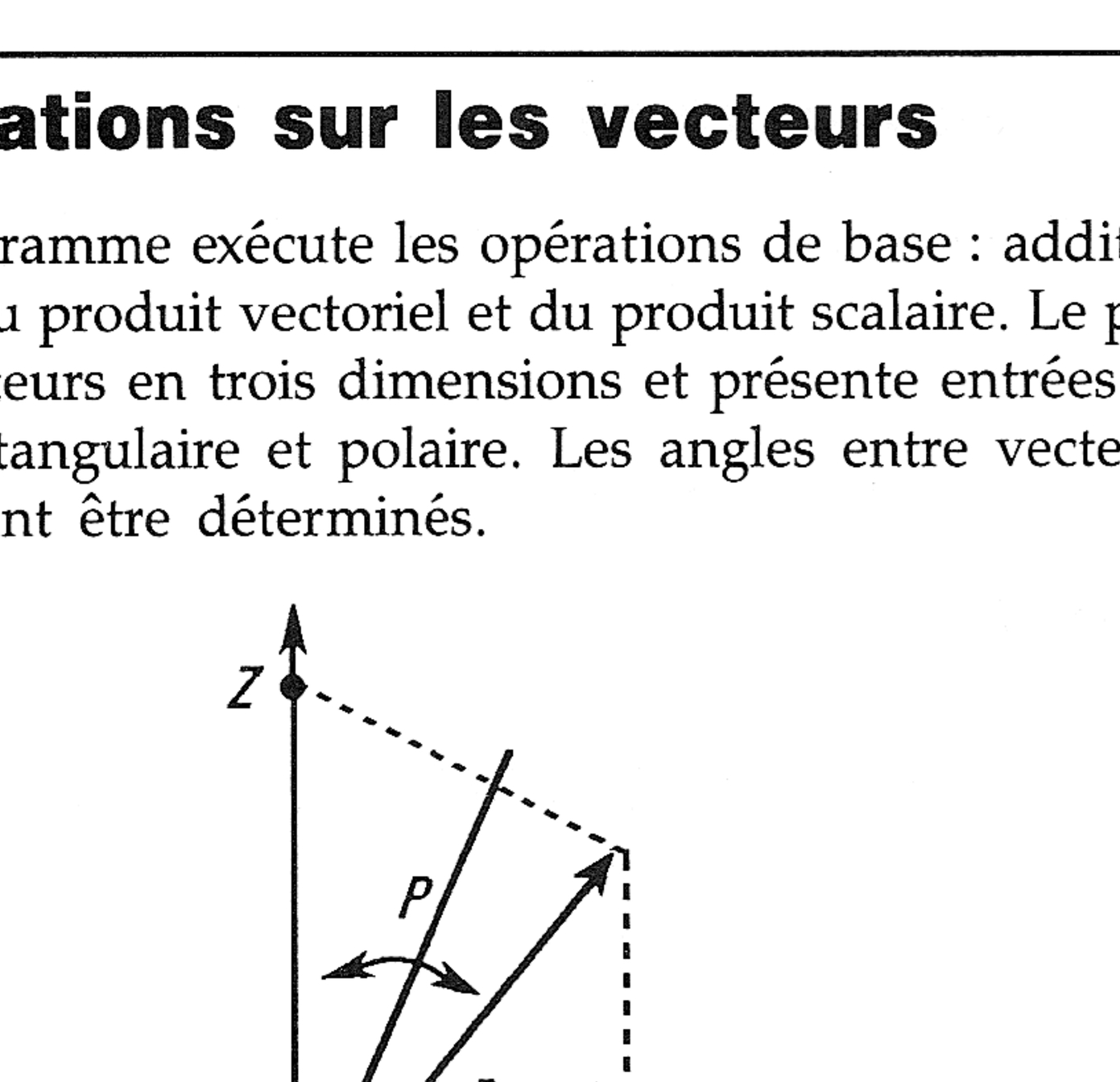
12

Programmes mathématiques

La quantité de mémoire utilisée et le contrôle par totalisation de chaque programme peuvent être vérifiés en utilisant le catalogue de programmes (Cf. page 85).

Opérations sur les vecteurs

Ce programme exécute les opérations de base : addition, soustraction, calcul du produit vectoriel et du produit scalaire. Le programme utilise des vecteurs en trois dimensions et présente entrées et sorties en format rectangulaire et polaire. Les angles entre vecteurs peuvent également être déterminés.



Systèmes de coordonnées de vecteurs

Ce programme utilise les équations suivantes.

Conversion de coordonnées :

$$X = R \sin(P) \cos(T) \qquad R = \sqrt{X^2 + Y^2 + Z^2}$$

$$Y = R \sin(P) \sin(T) \qquad T = \arctan \frac{Y}{X}$$

$$Z = R \cos(P) \qquad P = \arctan \frac{Z}{\sqrt{X^2 + Y^2}}$$

Addition et soustraction de vecteurs :

$$\mathbf{v}_1 + \mathbf{v}_2 = (X + U)\mathbf{i} + (Y + V)\mathbf{j} + (Z + W)\mathbf{k}$$

$$\mathbf{v}_2 - \mathbf{v}_1 = (U - X)\mathbf{i} + (V - Y)\mathbf{j} + (W - Z)\mathbf{k}$$

Produit vectoriel :

$$\mathbf{v}_1 \times \mathbf{v}_2 = (YW - ZV)\mathbf{i} + (ZU - XW)\mathbf{j} + (XV - YU)\mathbf{k}$$

Produit scalaire :

$$D = XU + YV + ZW$$

Angle entre vecteurs (γ) :

$$G = \arccos \frac{D}{R_1 \times R_2},$$

où

$$\mathbf{v}_1 = X\mathbf{i} + Y\mathbf{j} + Z\mathbf{k}$$

et

$$\mathbf{v}_2 = U\mathbf{i} + V\mathbf{j} + W\mathbf{k}$$

Le vecteur affiché par les sous-programmes de saisie (LBL P et LBL R) est V_1.

Liste de programme :

Lignes de programme :	Description :
R01 LBL R	Définit le début du sous-programme de saisie/d'affichage rectangulaire.
R02 INPUT X	Affiche ou accepte la saisie de X.
R03 INPUT Y	Affiche ou accepte la saisie de Y.
R04 INPUT Z	Affiche ou accepte la saisie de Z.
Octets et totalisation de contrôle : 006,0, 80FB	
Q01 LBL Q	Définit le début du processus de conversion « rectangulaire/polaire ».
Q02 RCL Y	
Q03 RCL X	
Q04 y,x→θ,r	Calcule $\sqrt{(X^2 + Y^2)}$ et arctan(Y/X).
Q05 x<>y	
Q06 STO T	Sauvegarde T = arctan(Y/X).
Q07 R↓	Rappelle $\sqrt{(X^2 + Y^2)}$.
Q08 RCL Z	
Q09 y,x→θ,r	Calcule $\sqrt{(X^2 + Y^2 + Z^2)}$ et P.
Q10 STO R	Sauvegarde R.
Q11 x<>y	
Q12 STO P	Sauvegarde P.
Octets et totalisation de contrôle : 018,0, D6D5	
P01 LBL P	Définit le début du sous-programme de saisie/d'affichage.
P02 INPUT R	Affiche ou accepte la saisie de R.
P03 INPUT T	Affiche ou accepte la saisie de T.
P04 INPUT P	Affiche ou accepte la saisie de P.
P05 RCL T	
P06 RCL P	
P07 RCL R	
P08 θ,r→y,x	Calcule R cos(P) et Rsin(P).
P09 STO Z	Stocke $Z = R$ cos(P).
P10 R↓	
P11 θ,r→y,x	Calcule R sin(P) cos(T) et R sin(P) sin(T).
P12 STO X	Sauvegarde $X = R$ sin(P) cos(T).
P13 x<>y	

```
P14 STO Y
P15 GTO P
```

Sauvegarde $Y = R \sin(P) \sin(T)$.
Boucle récursivement pour un autre affichage de forme polaire.

Octets et totalisation de contrôle : 022,5, AA98

```
E01 LBL E
E02 RCL X
E03 STO U
E04 RCL Y
E05 STO V
E06 RCL Z
E07 STO W
E08 GTO Q
```

Définit le début du sous-programme de saisie de vecteur.
Copie les valeurs de X, Y et Z respectivement dans U, V et W.
Boucle récursivement pour la conversion polaire et l'affichage/saisie.

Octets et totalisation de contrôle : 012,0, 7137

```
X01 LBL X
X02 RCL X
X03 RCL U
X04 STO X
X05 x<>y
X06 STO U
X07 RCL Y
X08 RCL V
X09 STO Y
X10 x<>y
X11 STO V
X12 RCL Z
X13 RCL W
X14 STO Z
X15 x<>y
X16 STO W
X17 GTO Q
```

Définit le début du sous-programme d'échange de vecteur.
Remplace X, Y et Z respectivement par U, V et W.
Boucle récursivement pour la conversion polaire et l'affichage/saisie.

Octets et totalisation de contrôle : 025,5, EAD8

```
A01 LBL A
```
Définit le début du programme d'addition de vecteurs.

```
A02 RCL X
A03 RCL+ U
A04 STO X
```
Sauvegarde $X + U$ dans X.

```
A05 RCL V
A06 RCL+ Y
A07 STO Y
```
Sauvegarde $V + Y$ dans Y.

```
A08 RCL Z
A09 RCL+ W
A10 STO Z
```
Sauvegarde $Z + W$ dans Z.

```
A11 GTO Q
```
Boucle récursivement pour la conversion polaire et l'affichage/saisie.

Octets et totalisation de contrôle : 016,5, F888

```
S01 LBL S
```
Définit le début du sous-programme de soustraction de vecteurs.

```
S02 -1
```
Multiplie X, Y et Z par (-1) pour changer de signe.

```
S03 STO× X
S04 STO× Y
S05 STO× Z
```

```
S06 GTO A
```
Passe au sous-programme d'addition de vecteurs.

Octets et totalisation de contrôle : 017,0, 250B

```
C01 LBL C
```
Définit le début du sous-programme de produit vectoriel.

```
C02 RCL Y
C03 RCL× W
C04 RCL Z
C05 RCL× V
C06 -
```
Calcule $(YW - ZV)$, qui est la composante X.

```
C07 RCL Z
C08 RCL× U
C09 RCL X
C10 RCL× W
C11 -
```
Calcule $(ZU - WX)$, qui est la composante Y.

```
C12 RCL X
C13 RCL× V
C14 RCL Y
C15 RCL× U
C16 -
C17 STO Z
C18 R↓
C19 STO Y
C20 R↓
C21 STO X
C22 GTO Q
```

Stocke ($XV - YU$), qui est la composante Z.

Stocke la composante *Y*.

Stocke la composante *X*.

Boucle récursivement pour la conversion polaire et l'affichage/saisie.

Octets et totalisation de contrôle : 033,0, D74B

```
D01 LBL D
```

Définit le début du sous-programme du produit scalaire et d'angle-vecteur.

```
D02 RCL X
D03 RCL× U
D04 RCL Y
D05 RCL× V
D06 +
D07 RCL Z
D08 RCL× W
D09 +
D10 STO D
```

Stocke le produit scalaire de $XU + YV + ZW$.

```
D11 VIEW D
```

Affiche le produit scalaire.

```
D12 RCL D
D13 RCL R
D14 ÷
```

Divise le produit scalaire par la taille du vecteur *X*. *Y*. Z.

```
D15 RCL W
D16 RCL V
D17 RCL U
D18 y,x→θ,r
D19 x<>y
D20 R↓
D21 y,x→θ,r
D22 x<>y
D23 R↓
```

Calcule la taille du vecteur *U*, *V*, *W*.

```
D24 ÷
D25 ACOS
D26 STO G
D27 VIEW G
D28 GTO P
```

Divise le résultat précédent par la taille.
Calcule l'angle.
Affiche l'angle.
Boucle récursivement pour les saisies polaires et d'affichage.

Octets et totalisation de contrôle : 042,0, 739F

Indicateurs utilisés : aucun.

Mémoire nécessaire : 280,5 octets : 192,5 pour le programme, 88 pour les variables.

Remarques : La longueur du sous-programme S peut être raccourcie de 6,5 octets. La valeur −1 utilise 9,5 octets. Si elle apparaît sous la forme 1 suivie de +/− , elle ne nécessitera que 3 octets. Pour arriver à ce résultat, il suffit de saisir une étape factice entre le 1 et le +/−, puis de supprimer l'étape factice.

Les termes « polaire » et « rectangulaire », qui réfèrent à des systèmes bi-dimensionnels, sont utilisés au lieu des termes plus appropriés de « sphérique » et « cartésien ». Ceci permet aux labels d'être associés avec leur fonction sans conflits induisant la confusion. Par exemple si LBL C avait été associé avec la saisie de coordonnées cartésiennes elle n'aurait pas été disponible pour le produit vectoriel.

Instructions du programme :

1. Saisissez les sous-programmes ; appuyez sur [C] lorsque vous avez terminé.
2. Si votre vecteur est sous forme rectangulaire, appuyez sur [XEQ] R et passez au point 4. S'il est sous forme polaire, appuyez sur [XEQ] P et continuez au point 3.
3. Saisissez *R* et appuyez sur [R/S], saisissez *T* et appuyez sur [R/S], saissez *P* et appuyez sur [R/S]. Passez au point 5.
4. Saisissez *X* et appuyez sur [R/S], saisissez *Y* et appuyez sur [R/S], saisissez *Z* et appuyez sur [R/S].
5. Pour saisir un deuxième vecteur, appuyez sur [XEQ] E (pour *Enter*) et passez au point 2.

6. Exécutez l'opération vectorielle désirée :

a. Ajoutez les vecteurs en appuyant sur [XEQ] A ;

b. Soustrayez vecteur 1 de vecteur 2 en appuyant sur [XEQ] S ;

c. Calculez le produit vectoriel en appuyant sur [XEQ] C ;

d. Calculez le produit scalaire en appuyant sur [XEQ] D et l'angle entre vecteurs en appuyant sur [R/S].

7. Optionnel : pour examiner $\mathbf{v}_1$ sous forme polaire, appuyez sur [XEQ] P, puis appuyez plusieurs fois sur [R/S] pour examiner les divers éléments.

8. Optionnel : pour voir $\mathbf{v}_1$ sous forme rectangulaire, appuyez sur [XEQ] R, puis appuyez plusieurs fois sur [R/S] pour voir les éléments un à un.

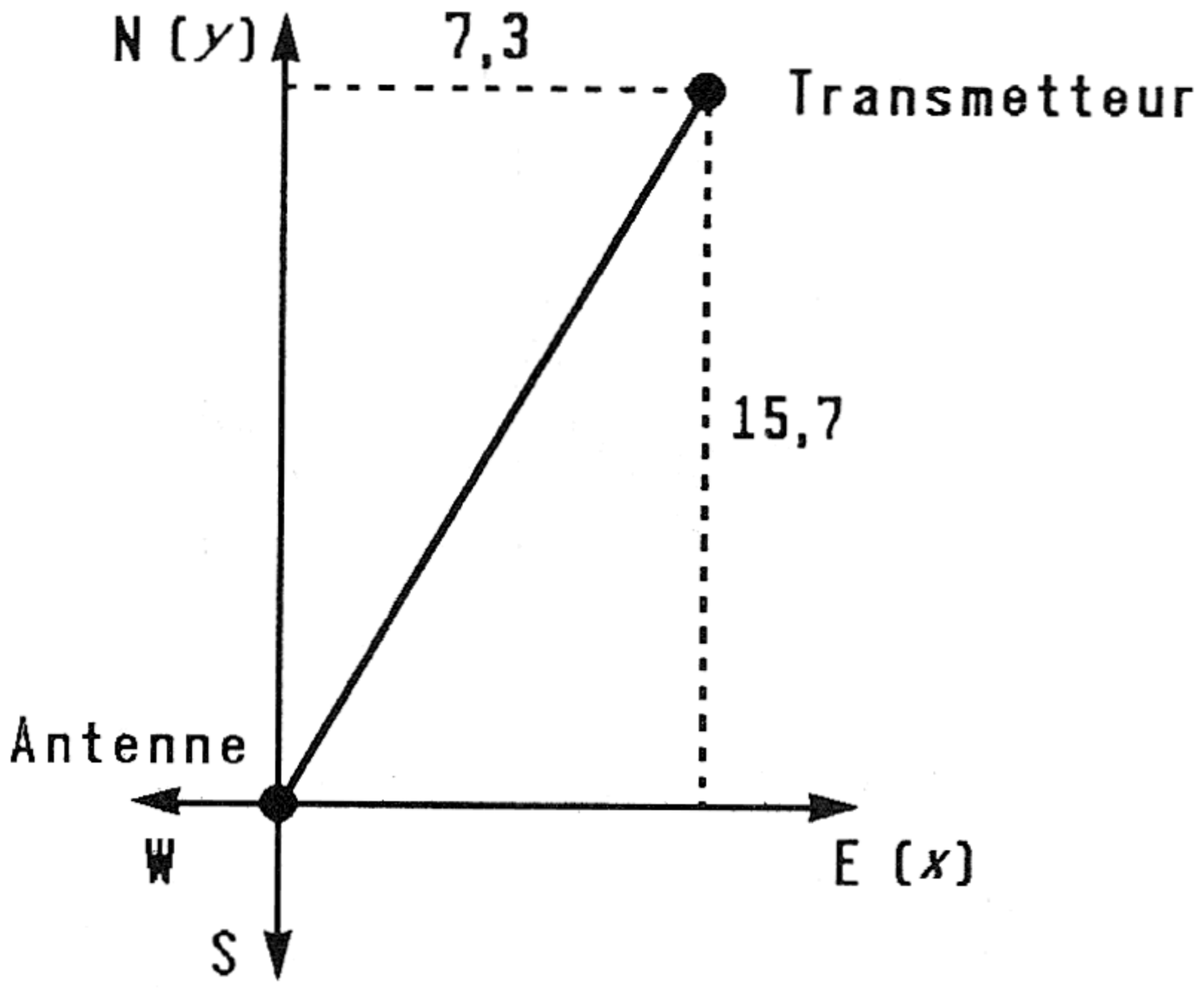
9. Si vous ajoutez, soustrayez ou calculez le produit vectoriel, $\mathbf{v}_1$ a été remplacé par le résultat. $\mathbf{v}_2$ n'est pas modifié. Pour continuer les calculs basés sur le résultat, souvenez-vous d'appuyer sur [XEQ] E avant de saisir un nouveau vecteur.

10. Passez au point 2 pour continuer les calculs sur les vecteurs.

Variables utilisées :

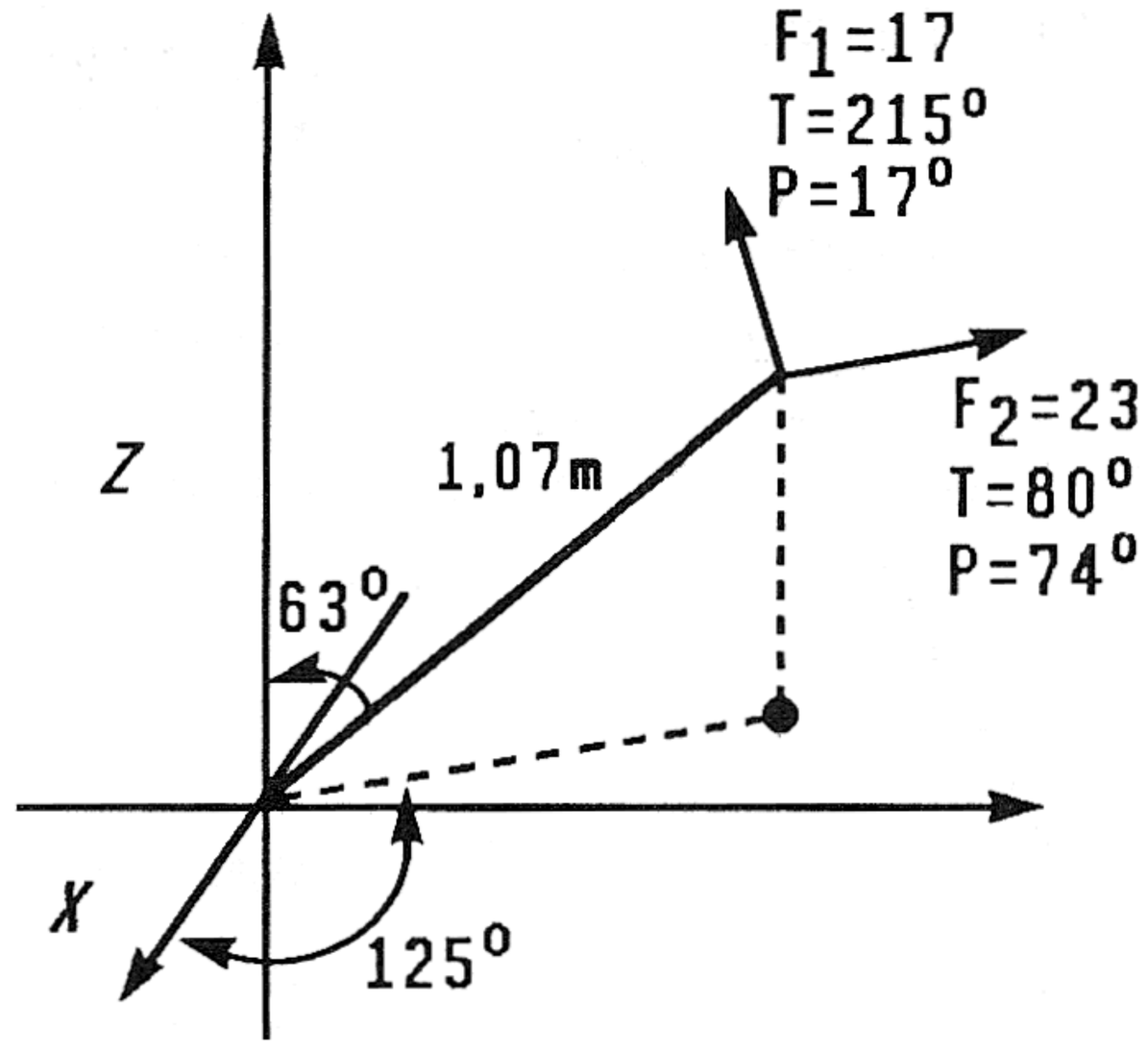
X, Y, Z	Les composants rectangulaires de $\mathbf{v}_1$.
U, V, W	Les composants rectangulaires de $\mathbf{v}_2$.
R, T, P	Le rayon, l'angle dans le plan x y (θ) et l'angle de l'axe Z de $\mathbf{v}_1$ (Φ).
D	Le produit scalaire.
G	L'angle entre les vecteurs (γ).

Exemple 1. Une antenne à micro-ondes est orientée vers un émetteur se trouvant à 15,7 kilomètres au nord, 7,3 kilomètres à l'est et 0,76 km plus bas. Utilisez la conversion rectangulaire en polaire pour trouver la distance totale et la direction de l'émetteur.



Touches :	Affichage :	Description :
XEQ R	X?*valeur*	Lance le sous-programme de saise/affichage rectangulaire.
7,3 R/S	Y?*valeur*	Définit X égal à 7,3.
15,7 R/S	Z?*valeur*	Définit Y égal à 15,7.
,76 +/− R/S	R=17,3308	Définit Z égal à −0,76 et calcule R, le rayon.
R/S	T=65,0631	Calcule T, l'angle dans le plan x/y.
R/S	P=92,5134	Calcule P, l'angle depuis l'axe z.

Exemple 2. Quel est le moment à l'origine du levier indiqué ci-dessous ? Quel est le composant de force le long de ce levier ? Quel est l'angle entre la résultante des vecteurs de force et le levier ?



Commencez par additionner les vecteurs de force.

Touches :	**Affichage :**	**Description :**
XEQ P	R?*valeur*	Lance le sous-programme de saisie polaire.
17 R/S	T?*valeur*	Définit le rayon : 17.
215 R/S	P?*valeur*	Définit T égal à 215.
17 R/S	R?17,0000	Définit P égal à 17.
XEQ E	R?17,0000	Saisit le vecteur en le copiant dans $\mathbf{v}_2$.
23 R/S	T?-145,0000	Définit le rayon de $\mathbf{v}_1$: 23.
80 R/S	P?17,0000	Définit T égal à 80.
74 R/S	R?23,0000	Définit P égal à 74.
XEQ A	R?29,4741	Ajoute les vecteurs et affiche la résultante R.
R/S	T?90,7032	Affiche T du vecteur résultant.

R/S	P?39,9445	Affiche *P* du vecteur résultant.
XEQ E	R?29,4741	Saisit le vecteur résultant.

Puisque le moment est égal au produit vectoriel du vecteur de rayon et du vecteur de force ($\mathbf{r} \times \mathbf{F}$), introduisez le vecteur représentant le levier et trouvez le produit vectoriel.

Touches :	**Affichage :**	**Description :**
1,07 R/S	T?90,7032	Définit *R* comme égal à 1,07.
125 R/S	P?39,9445	Définit *T* comme égal à 125.
63 R/S	R?1,0700	Définit *P* comme égal à 63.
XEQ C	R?18,0209	Calcule le produit vectoriel et affiche *R* du résultat.
R/S	T?55,3719	Affiche *T* du produit vectoriel.
R/S	P?124,3412	Affiche *P* du produit vectoriel.
XEQ R	X?8,4554	Affiche la forme rectangulaire du produit vectoriel.
R/S	Y?12,2439	
R/S	Z?-10,1660	

Le produit scalaire peut être utilisé pour trouver la force exercée (toujours dans $\mathbf{v}_2$) le long de l'axe du levier.

Touches :	Affichage :	Description :
XEQ P	R?18,0209	Lance le sous-programme de saisie polaire.
1 R/S	T?55,3719	Définit le rayon : un vecteur d'une unité.
125 R/S	P?124,3412	Définit *T* comme égal à 125.
63 R/S	R?1,0000	Définit *P* comme égal à 63.
XEQ D	D=24,1882	Calcule le produit scalaire.
R/S	G=34,8490	Calcule l'angle entre le vecteur de force résultante et le levier.
R/S	R?1,0000	Ramène au sous-programme de saisie.

Solutions d'équations simultanées— Méthode du déterminant

Ce programme résout des équations linéaires simultanées à deux ou trois inconnues. Le programme utilise la méthode de Cramer, également connue sous le nom de *méthode des déterminants.*

Soit un système de trois équations linéaires

$$AX + DY + GZ = J$$

$$BX + EY + HZ = K$$

$$CX + FY + IZ = L$$

Les trois inconnues *X*, *Y* et *Z* peuvent être calculées à partir des déterminants.

$$X = \frac{Dét_x}{Dét} \qquad Y = \frac{Dét_y}{Dét} \qquad Z = \frac{Dét_z}{Dét}$$

$$Dét = \begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix} \qquad Dét_x = \begin{bmatrix} J & D & G \\ K & E & H \\ L & F & I \end{bmatrix}$$

$$Dét_y = \begin{bmatrix} A & J & G \\ B & K & H \\ C & L & I \end{bmatrix} \qquad Dét_z = \begin{bmatrix} A & D & J \\ B & E & K \\ C & F & L \end{bmatrix}$$

Liste de programme :

Lignes de programme :	Description :
A01 LBL A	Point de départ de la saisie des valeurs connues.
A02 1,012	Valeur de contrôle de boucle : de 1 à 12, 1 à la fois.
A03 STO i	Stocke la valeur de contrôle dans la variable de l'index.

Octets et totalisation de contrôle : 012,5, 7878

L01 LBL L	Lance la boucle de saisie.
L02 INPUT(i)	Sollicite et stocke la variable indiquée par *i*.
L03 ISG i	Ajoute un à *i*.
L04 GTO L	Si *i* est inférieur à 13, retourne à LBL L et appelle la valeur suivante.
L05 GTO A	Revient à LBL A pour revoir les valeurs.

Octets et totalisation de contrôle : 007,5, C1DE

S01 LBL S	Point de départ des solutions des équations simultanées.
S02 9	Valeur de l'index de *I* pour adressage indirect.
S03 STO i	Stocke la valeur d'index.

```
S04 XEQ E
```
Permute la colonne de solution et celle des coefficients contenant *I*.

```
S05 XEQ D
```
Calcule le déterminant.

```
S06 STO Z
```
Sauvegarde le déterminant dans Z.

```
S07 XEQ E
```
Rétablit le dét. ss forme d'origine.

```
S08 6
```
Valeur d'index de *F* pour adressage indirect.

```
S09 STO i
```
Stocke la valeur d'index.

```
S10 XEQ E
```
Permute col. de solution et col. cont. *F*.

```
S11 XEQ D
```
Calcule le déterminant.

```
S12 STO Y
```
Sauvegarde le déterminant dans *Y*.

```
S13 XEQ E
```
Rétablit le dét. ss forme originelle.

```
S14 3
```
Valeur d'index de *C* pour adressage indirect.

```
S15 STO i
```
Stocke la val. d'index ds une var. d'index.

```
S16 XEQ E
```
Permute col. de solution et col. conten. *F*.

```
S17 XEQ D
```
Calcule le déterminant.

```
S18 STO X
```
Sauvegarde le déterminant dans *X*.

```
S19 XEQ E
```
Rétablit le déterminant ss forme originale.

```
S20 XEQ D
```
Calcule le déterminant des coefficients d'origine.

```
S21 STO÷ X
```
Divise par le déterminant d'origine.

```
S22 STO÷ Y
S23 STO÷ Z
```

```
S24 RCL X
```
Rappelle et affiche le résultat de *X*, *Y* et Z.

```
S25 VIEW X
S26 RCL Y
S27 VIEW Y
S28 RCL Z
S29 VIEW Z
```

```
S30 RTN
```
Revient au programme d'appel ou à `PRGM TOP`.

Octets et totalisation de contrôle : 045,0, 3971

```
E01 LBL E
```
Ce sous-programme échange les colonnes et la règle de Cramer.

```
E02 RCL(i)
```
Appelle le dernier élément de la colonne du déterminant de coefficient.

```
E03 RCL L
```
Appelle le dernier élément du vecteur de solution.

```
E04 STO(i)
```
Sauvegarde le vecteur d'élément dans le déterminant.

```
E05 x<>y
E06 STO L
E07 DSE i
```
Rappelle l'élément de coefficient.
Sauvegarde l'élément dans le vecteur.
Définit la valeur d'index pour indiquer l'élément du milieu dans la colonne du déterminant.

```
E08 RCL(i)
```
Appelle l'élément du milieu de la colonne du déterminant.

```
E09 RCL K
```
Obtient l'élément du milieu du vecteur.

```
E10 STO(i)
```
Sauvegarde l'élément vectoriel dans le déterminant.

```
E11 x<>y
```
Rappelle l'élément coefficient.

```
E12 STO K
```
Sauvegarde l'élém. coeff. ds le vecteur.

```
E13 DSE i
```
Définit la valeur d'index pour indiquer l'élém. sup. de la col. du dét.

```
E14 RCL(i)
```
Appelle l'élém. sup. de la colonne du dét.

```
E15 RCL J
```
Appelle l'élément supérieur du vecteur.

```
E16 STO(i)
```
Sauvegarde l'élém. du vecteur ds le dét.

```
E17 x<>y
```
Rappelle l'élément du déterminant.

```
E18 STO J
```
Sauvegarde l'élém. dét. ds le vecteur.

```
E19 2
E20 STO+ i
```
Remet *i* à sa valeur d'origine, au début du sous-programme.

```
E21 RTN
```
Revient au programme d'appel ou à `PRGM TOP`.

Octets et totalisation de contrôle : 031,5, 8420

```
D01 LBL D
```
Ce sous-programme calcule le déterminant.

```
D02 RCL A
D03 RCL× E
D04 RCL× I
```
Calcule $A \times E \times I$.

```
D05 RCL D
D06 RCL× H
D07 RCL× C
D08 +
```
Calcule $(A \times E \times I) + (D \times H \times C)$.

```
D09 RCL G
D10 RCL× F
D11 RCL× B
D12 +
```
Calcule
$(A \times E \times I) + (D \times H \times C) + (G \times F \times B)$.

```
D13 RCL G
D14 RCL× E
```

D15 RCL× C	
D16 −	$(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C)$.
D17 RCL A	
D18 RCL× F	
D19 RCL× H	
D20 −	$(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C) - (A \times F \times H)$.
D21 RCL D	
D22 RCL× B	
D23 RCL× I	
D24 −	$(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C) - (A \times F \times H) - (D \times B \times I)$.
D25 RTN	Revient au programme d'appel ou à PRGM TOP.

Octets et totalisation de contrôle : 037,5, 152E

Indicateurs utilisés : aucun.

Besoins en mémoire : 262 octets : 134 pr le prog., 128 pr les var.

Instructions du programme :

1. Frappez les sous-programmes ; appuyez sur [C] lorsque vous aurez terminé.
2. Appuyez sur [XEQ] A pour saisir les coefficients (de *A* à *L*) des équations linéaires.
3. Frappez le coefficient (*A* à *L*) et appuyez sur [R/S].
4. Optionnel : pour calculer le déterminant d'un système de 3 × 3, [XEQ] D.
5. Calculez la solution du système d'équations en appuyant sur [XEQ] S.
6. Lisez la valeur de *X* et appuyez sur [R/S] pour voir la valeur de *Y*.
7. Appuyez sur [R/S] pour voir la valeur de *Z*.
8. Pour un nouveau cas, revenez à l'étape 2.

Variables utilisées :

A à *I*	Coefficients des équations.
J à *L*	Côtés droits des équations.
X à *Z*	Inconnues.
i	Valeur de contrôle de boucle (index).

Remarques : Ce programme est prévu pour un système de deux ou trois équations (c'est-à-dire une matrice de $n \leqslant 3$).

Pour 2 × 2 solutions, utilisez zéro pour les coefficients *C*, *F*, *G*, *H* et *L*. Utilisez 1 comme *I*. Pour les matrices non carrées, utilisez zéro pour les coefficients « manquants ».

Tous les systèmes d'équations n'ont pas une solution. S'ils n'en ont pas, ils provoquent une erreur `DIVIDE BY 0` à la ligne S21.

Exemple 1. Pour le système ci-dessous, calculez le déterminant et la solution du système. Ensuite, remplacez les valeurs dans la première équation pour vérifier que le côté gauche de l'équation est bien égal à son côté droit (1).

$$23X + 15Y + 17Z = 1$$

$$8X + 11Y - 6Z = 1$$

$$4X + 15Y + 12Z = 1$$

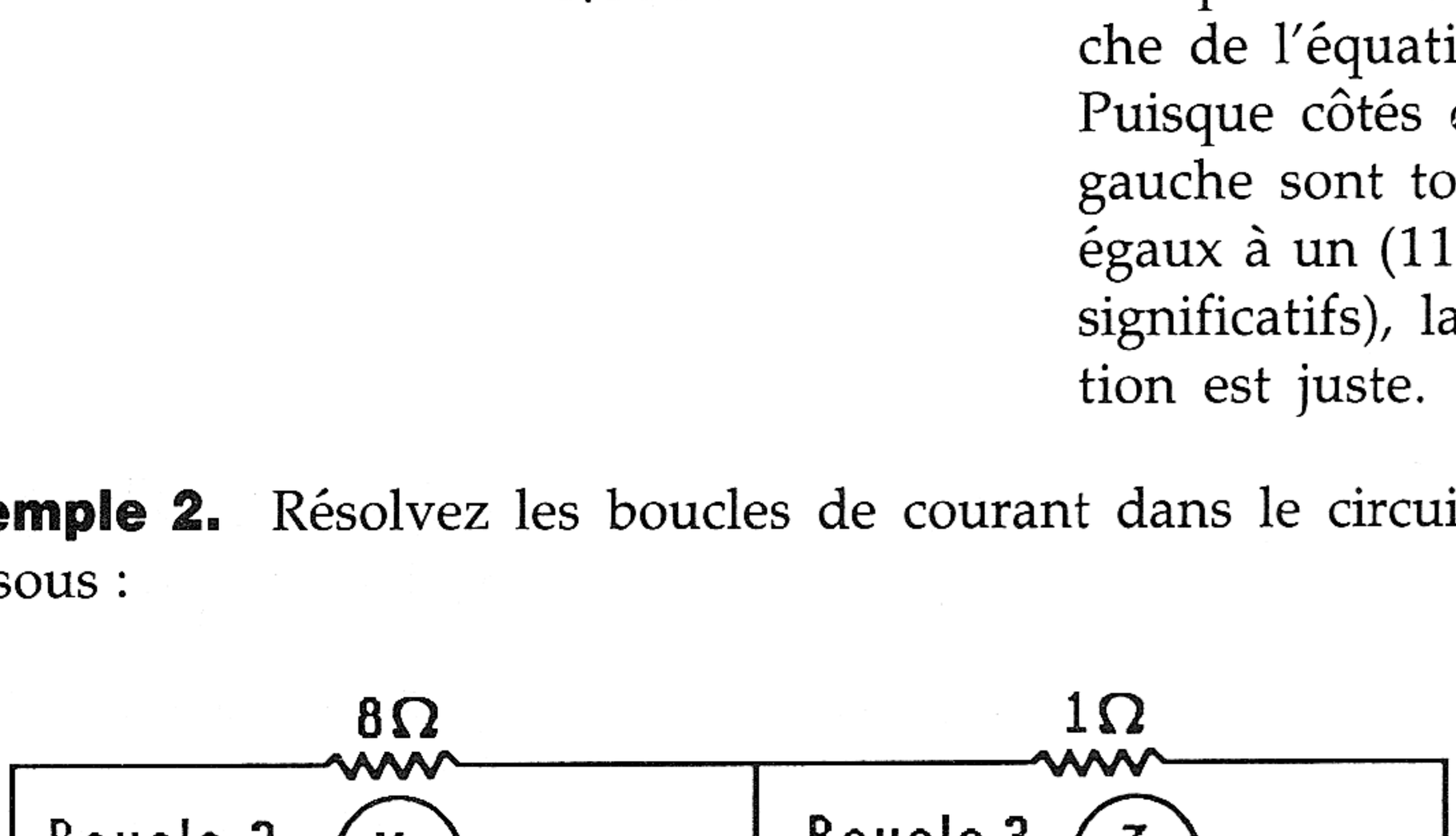
Touches :	**Affichage :**	**Description :**
XEQ A	`A?`*valeur*	Lance le sous-programme de saisie.
23 R/S	`B?`*valeur*	Définit le premier coefficient, *A*, égal à 23.
8 R/S	`C?`*valeur*	Définit *B* égal à 8.
4 R/S	`D?`*valeur*	Définit *C* égal à 4.
15 R/S	`E?`*valeur*	Définit *D* égal à 15.
⋮	⋮	Continue la saisie pour toutes les val. (*E* à *L*).
1 R/S	`A?23,0000`	Revient au premier coefficient qui ait été saisi.
XEQ D	`4.598,0000`	Calcule le déterminant.

XEQ S	X=0,0043	Résout le système d'équations et affiche X.
R/S	Y=0,0787	Affiche Y.
R/S	Z=-0,0165	Affiche Z.

Maintenant, vérifions le résultat :

Touches :	**Affichage :**	**Description :**
23 RCL × X	0,1000	Multiplie X par 23.
15 RCL × Y	1,1809	Multiplie Y par 15.
+	1,2810	Ajoute les deux derniers résultats.
17 RCL × Z	-0,2810	Multiplie Z par 17.
+	1,0000	Complète le côté gauche de l'équation. Puisque côtés droit et gauche sont tous deux égaux à un (11 chiffres significatifs), la solution est juste.

Exemple 2. Résolvez les boucles de courant dans le circuit ci-dessous :



Rédigez tout d'abord les équations pour calculer la perte de courant à chaque passage de boucle.

Pour la boucle n^0 1 : $4X - 4Y + 15X - 15Z - 40 = 0$

Pour la boucle n^0 2 : $4Y - 4X + 8Y + 10Y - 10Z = 0$

Pour la boucle n^0 3 : $10Z - 10Y + Z + 15Z - 15X = 0$

Après la combinaison de termes semblables, les équations deviennent

$$19X - 4Y - 15Z = 40$$

$$-4X + 22Y - 10Z = 0$$

$$-15X - 10Y + 26Z = 0$$

Touches :	**Affichage :**	**Description :**
XEQ A	A?*valeur*	Lance le sous-programme de saisie.
19 R/S	B?*valeur*	Définit le premier coefficient, *A*, comme étant égal à 19.
4 +/– R/S	C?*valeur*	Définit *B* comme étant égal à −4.
15 +/– R/S	D?*valeur*	Définit *C* comme étant égal à −15.
⋮		Continue la saisie de *D* à *L*.
0 R/S	A?19,0000	Saisit *L* et renvoie le premier coefficient saisi.
XEQ S	X=7,8601	Résout le système d'équations et affiche X.
R/S	Y=4,2298	Affiche *Y*.
R/S	Z=6,1615	Affiche Z.

Solutions d'équations simultanées— Méthode de l'inversion de matrice

Ce programme résout les équations linéaires simultanées à deux ou trois inconnues. Il le fait par inversion et multipication de matrice.

Un système de trois équations linéaires

$$AX + DY + GZ = J$$

$$BX + EY + HZ = K$$

$$CX + FY + IZ = L$$

peut être représenté par l'équation matricielle ci-dessous.

$$\begin{bmatrix} A & D & G \\ B & E & H \\ C & F & I \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} J \\ K \\ L \end{bmatrix}$$

L'équation matricielle peut être résolue pour X, Y et Z en multipliant la matrice-résultat par l'inverse de la matrice coefficient.

$$\begin{bmatrix} A' & D' & G' \\ B' & E' & H' \\ C' & F' & I' \end{bmatrix} \begin{bmatrix} J \\ K \\ L \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix}$$

Les détails du processus d'inversion sont donnés dans les commentaires du sous-programme d'inversion, I.

Liste de programme :

Lignes de programme :	Description :
A01 LBL A	Point de départ de la saisie des coefficients.
A02 1,012	Valeur de contrôle de boucle : boucle de 1 à 12, 1 à la fois.
A03 STO i	Stocke la valeur de contrôle dans une variable d'index.

Octets et totalisation de contrôle : 012,5, 7878

L01 LBL L	Lance la boucle de saisie.
L02 INPUT(i)	Sollicite et stocke la variable adressée par *i*.
L03 ISG i	Ajoute *un* à *i*.
L04 GTO L	Si $i < 13$, revient à LBL L et appelle la valeur suivante.
L05 GTO A	Revient à LBL A pour revoir les valeurs.

Octets et totalisation de contrôle : 007,5, C1DE

I01 LBL I	Ce sous-programme inverse une matrice de 3×3.
I02 XEQ D	Calcule le déterminant et sauvegarde la valeur correpondant à la boucle de division, J.
I03 STO W	
I04 RCL A	
I05 RCL× I	
I06 RCL C	
I07 RCL× G	
I08 -	
I09 STO X	Calcule $E' \times$ déterminant $= AI - CG$.
I10 RCL C	
I11 RCL× D	
I12 RCL A	
I13 RCL× F	
I14 -	
I15 STO Y	Calcule $F' \times$ déterminant $= CD - AF$.
I16 RCL B	
I17 RCL× G	
I18 RCL A	
I19 RCL× H	
I20 -	
I21 STO Z	Calcule $H' \times$ déterminant $= BG - AH$.
I22 RCL A	

```
I23 RCL× E
I24 RCL B
I25 RCL× D
I26 -
I27 STO i
I28 RCL E
I29 RCL× I
I30 RCL F
I31 RCL× H
I32 -
I33 STO A
I34 RCL C
I35 RCL× H
I36 RCL B
I37 RCL× I
I38 -
I39 RCL B
I40 RCL× F
I41 RCL C
I42 RCL× E
I43 -
I44 STO C
I45 R↓
I46 STO B
I47 RCL F
I48 RCL× G
I49 RCL D
I50 RCL× I
I51 -
I52 RCL D
I53 RCL× H
I54 RCL E
I55 RCL× G
I56 -
I57 STO G
I58 R↓
I59 STO D
I60 RCL i
I61 STO I
I62 RCL X
I63 STO E
```

Calcule $I' \times$ déterminant $= AE - BD$.

Calcule $A' \times$ déterminant $= EI - FH$.

Calcule $B' \times$ déterminant $= CH - BI$.

Calcule $C' \times$ déterminant $= BF - CE$.

Stocke B'.

Calcule $D' \times$ déterminant $= FG - DI$.

Calcule $G' \times$ déterminant $= DE - EG$.

Stocke D'.

Stocke I'.

Stocke E'.

```
I64 RCL Y
I65 STO F          Stocke F'.
I66 RCL Z
I67 STO H          Stocke H'.
I68 9
I69 STO i          Définit la valeur d'index pour qu'elle
                   indique le dernier élément de la matrice.
I70 RCL W          Rappelle la valeur du déterminant.
```

Octets et totalisation de contrôle : 105,0, E5C1

```
J01 LBL J          Ce sous-programme termine l'inversion en
                   divisant par le déterminant.
J02 STO÷(i)        Divise l'élément.
J03 DSE i          Diminue la valeur de l'index pour qu'elle
                   soit plus proche de A.
J04 GTO J          La boucle passe à la valeur suivante.
J05 RTN            Revient au progr. d'appel/PRGM TOP.
```

Octets et totalisation de contrôle : 007,5, A354

```
M01 LBL M          Ce sous-programme multipie une matrice
                   colonne et une matrice 3 × 3.
M02 7              Définit la valeur de l'index pour qu'elle
                   indique le dernier élémt de la 1e ligne.
M03 XEQ N
M04 8              Définit la valeur d'index pour qu'elle
                   indique le dernier élément de la deuxième
                   ligne.
M05 XEQ N
M06 9              Définit la valeur de l'index por qu'elle
                   indique le dernier élémt de la 3e ligne.
```

Octets et totalisation de contrôle : 009,0, 0A85

```
N01 LBL N          Ce sous-programme calcule le produit d'un
                   vecteur colonne et de la ligne indiquée par
                   l'index.
N02 STO i          Sauvegarde la valeur d'index dans i.
N03 RCL J          Rappelle J de la matrice colonne.
N04 RCL K          Rappelle K de la matrice colonne.
N05 RCL L          Rappelle L du vecteur colonne.
N06 RCL×(i)        Multiplie par le dernier élément de la ligne.
N07 XEQ P          Multiplie par le deuxième élément de la li-
                   gne et additionne.
```

N08 XEQ P	Multiplie par le troisième élément de la ligne et additionne.
N09 23	Définit la valeur d'index pour afficher *X*, *Y* ou Z en fonction de la ligne de saisie.
N10 STO+ i	
N11 R↓	Obtient le résultat.
N12 STO(i)	Stocke le résultat.
N13 VIEW(i)	Affiche le résultat.
N14 RTN	Revient au programme d'appel ou à PRGM TOP.

Octets et totalisation de contrôle : 021,0, BBBF

P01 LBL P	Ce sous-programme multiplie et additionne les valeurs dans une même ligne.
P02 x<>y	Appelle la valeur de la colonne suivante.
P03 DSE i	Définit la valeur d'index pour qu'elle indique la valeur de la ligne suivante.
P04 DSE i	
P05 DSE i	
P06 RCL×(i)	Multiplie la valeur de la colonne par celle de la ligne.
P07 +	Ajoute le produit à la somme précédente.
P08 RTN	Revient au programme d'appel.

Octets et totalisation de contrôle : 012,0, 520E

D01 LBL D	Ce sous-programme calcule le déterminant.
D02 RCL A	
D03 RCL× E	
D04 RCL× I	Calcule $A \times E \times I$.
D05 RCL D	
D06 RCL× H	
D07 RCL× C	
D08 +	Calcule $(A \times E \times I) + (D \times H \times C)$.
D09 RCL G	
D10 RCL× F	
D11 RCL× B	
D12 +	Calcule $(A \times E \times I) + (D \times H \times C) + (G \times F \times B)$.
D13 RCL G	
D14 RCL× E	
D15 RCL× C	
D16 -	$(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C)$.

D17 RCL A	
D18 RCL× F	
D19 RCL× H	
D20 −	$(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C) - (A \times F \times H)$.
D21 RCL D	
D22 RCL× B	
D23 RCL× I	
D24 −	$(A \times E \times I) + (D \times H \times C) + (G \times F \times B) - (G \times E \times C) - (A \times F \times H) - (D \times B \times I)$.
D25 RTN	Revient au programme d'appel ou à PRGM TOP.

Octets et totalisation de contrôle : 037,5, 152E

Indicateurs utilisés : Aucun.

Besoins en mémoire : 348 octets: 212 pour le programme, 136 pour les variables.

Instructions du programme :

1. Frappez les sous-programmes ; appuyez sur [C] lorsque vous aurez terminé.
2. Appuyez sur [XEQ] A pour saisir les coefficients de la matrice et du vecteur colonne.
3. Frappez le coefficient ou la valeur du vecteur (*A* à *L*) à chaque sollicitation et appuyez sur [R/S].
4. Optionnel : appuyez sur [XEQ] D pour calculer le déterminant du système 3 × 3.
5. Appuyez sur [XEQ] I pour calculer l'inverse de la matrice 3 × 3.
6. Optionnel : appuyez sur [XEQ] A et appuyez plusieurs fois sur [R/S] pour revoir les valeurs de la matrice inverse.
7. Appuyez sur [XEQ] M pour multiplier la matrice inverse par le vecteur colonne et pour voir les valeurs de *X*. Appuyez sur [R/S] pour voir la valeur de *Y*, puis appuyez encore sur [R/S] pour voir la valeur de Z.
8. Pour un nouveau cas, revenez au point 2.

Variables utilisées :

A à *I*	Coefficients de la matrice.
J à *L*	Valeurs du vecteur colonne.
W	Variable « brouillon » utilisée pour stocker le déterminant.
X à Z	Valeurs de sortie du vecteur ; utilisées aussi comme brouillon.
i	Valeur de contrôle de la boucle (variable index) ; utilisée également comme brouillon.

Remarques : Pour les solutions 2 × 2 utilisez zéro pour les coefficients *C, F, H, G* et pour *L*. Utilisez 1 pour le coefficient *I*.

Notez que les sous-programmes A, L et D sont communs à ce programme et au programme « Solutions d'équations simultanées—Méthode du déterminant ».

Exemple. Pour le système ci-dessous, calculez l'inverse et la solution du système. Visualisez la matrice inverse. Inversez la matrice à nouveau et vérifiez le résultat pour vous assurer que la matrice originale est renvoyée.

$$23X + 15Y + 17Z = 31$$

$$8X + 11Y - 6Z = 17$$

$$4X + 15Y + 12Z = 14$$

Touches :	**Affichage :**	**Description :**
XEQ A	A?*valeur*	Commence la saisie.
23 R/S	B?*valeur*	Définit le premier coefficient, *A*, égal à 23.
8 R/S	C?*valeur*	Définit *B* égal à 8.
4 R/S	D?*valeur*	Définit *C* égal à 4.
15 R/S	E?*valeur*	Définit *D* égal à 15.
⋮	⋮	Continue la saisie de *D* à *L*.

14 [R/S]	A?23,0000	Revient au premier coefficient qui a été saisi.
[XEQ] I	4.598,0000	Calcule l'inverse et affiche le déterminant.
[XEQ] M	X=0,9306	Multiplie par le vecteur colonne pour calculer X.
[R/S]	Y=0,7943	Calcule et affiche *Y*.
[R/S]	Z=-0,1364	Calcule et affiche Z.
[XEQ] A	A?0,0483	Commence la révision de la matrice inverse.
[R/S]	B?-0,0261	Affiche la valeur suivante.
[R/S]	C?0,0165	Valeur suivante.
[R/S]	D?0,0163	Valeur suivante.
[R/S]	E?0,0452	Valeur suivante.
[R/S]	F?-0,0620	Valeur suivante.
[R/S]	G?-0,0602	Valeur suivante.
[R/S]	H?0,0596	Valeur suivante.
[R/S]	I?0,0289	Valeur suivante.
[XEQ] I	0,0002	Inverse l'inverse pour retrouver la matrice d'origine.
[XEQ] A	A?23,0000	Commence la révision de l'inverse de la matrice inverse.
[R/S]	B?8,0000	Affiche la valeur suivante,...
⋮	⋮	...etc.

Equation du second degré

Ce programme utilise la formule du second degré pour une équation qui recherche les racines réelles et complexes d'un polynôme du second degré.

Dans un polynôme du second degré

$$ax^2 + bx + c = 0$$

x peut être trouvé en utilisant la formule du second degré

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

dans laquelle $b^2 - 4ac$ est le discriminant. Dans le cas des racines complexes (dans lesquelles le discriminant est négatif), la partie réelle est

$$R = \frac{-b}{2a},$$

et la partie imaginaire

$$I = \pm \frac{i\sqrt{|b^2 - 4ac|}}{2a}.$$

Pour le racines réelles, le programme calcule toujours la racine de la valeur absolue la plus grande d'abord, ceci pour minimiser les inexactitudes qui peuvent être introduites si la racine carrée du discriminant est presque égale à b. Une fois la première racine trouvée, x_1, la seconde, x_2, est calculée en utilisant la relation

$$x_2 = \frac{c}{ax_1}.$$

Tout ordinateur utilisant un nombre fini dans ses calculs amènera une erreur si aucune précaution n'est prise. Les résultats incorrects induits par l'ordinateur peuvent bien souvent être évités lors de la conception du logiciel. L'exemple 4 illustre le problème numérique évité dans ce programme d'équation du second degré.

Liste de programme :

Lignes de programme :	Description :
Q01 LBL Q	Définit le début du sous-programme d'équation du second degré.
Q02 INPUT A	Sollicite et stocke la valeur de *A*.
Q03 x=0?	Si *A* est égal à zéro, revient et re-sollicite la valeur de *A*.
Q04 GTO Q	
Q05 INPUT B	Sollicite et stocke la valeur de *B*.
Q06 INPUT C	Sollicite et stocke la valeur de *C*.
Q07 x=0?	Si *C* est égal à zéro, revient et re-sollicite toutes les saisies.
Q08 GTO Q	
Q09 RCL B	Rappelle *B*.
Q10 +/-	$-B$.
Q11 CF 0	Efface l'indicateur 0 (suppose que $(-B)$ est positif).
Q12 x<0?	$(-B)$ est-il négatif ?
Q13 SF 0	S'il l'est, arme l'indicateur 0.
Q14 RCL B	
Q15 x^2	Calcule B^2.
Q16 4	
Q17 RCL× A	
Q18 RCL× C	
Q19 -	Calcule $B^2 - 4AC$.
Q20 x<0?	Teste pour voir si les racines sont imaginaires.
Q21 GTO I	Branche vers le sous-programme si elles le sont.
Q22 SQRT	$\sqrt{(B^2 - 4AC)}$.
Q23 FS? 0	Teste pour voir si $(-B)$ est négative.
Q24 +/-	Choisit la racine de la valeur absolue la plus grande possible.
Q25 +	$-B - \sqrt{(B^2 - 4AC)}$ or $-B + \sqrt{(B^2 - 4AC)}$.
Q26 2	
Q27 ÷	
Q28 RCL÷ A	Calcule X de la valeur absolue la plus grande.
Q29 STO X	Stocke la valeur de X de valeur absolue la plus grande.

```
Q30 VIEW X
Q31 RCL C
Q32 RCL÷ A
Q33 RCL÷ X
Q34 STO X
Q35 VIEW X
Q36 GTO Q
```

Affiche X.
Calcule la seconde valeur de X.
Calcule $X = C \div AX$.
Stocke la seconde valeur de X.
Affiche X.
Revient pour un deuxième cas.

Octets et totalisation de contrôle : 054,0, A04D

```
I01 LBL I
I02 ABS
I03 SQRT
I04 2
I05 RCL× A
I06 ÷
I07 STO I
I08 RCL B
I09 +/-
I10 LAST×
I11 ÷
I12 STO R
I13 RCL I
I14 RCL R
I15 VIEW R
I16 VIEW I
I17 GTO Q
```

Définit le début du sous-programme de calcul imaginaire.

Calcule la valeur absolue de $\sqrt{(B^2 - 4AC)} \div 2A$.
Stocke la partie imaginaire.

Récupère $2A$.

Stocke la partie réelle dans R.
Récupère la partie imaginaire de X.
Récupère la partie réelle de X.
Affiche la partie réelle.
Affiche la partie imaginaire.
Revient pour traiter un autre cas.

Octets et totalisation de contrôle : 025,5, DA3B

Indicateurs utilisés : L'indicateur 0 est utilisé pour retenir le signe de $(-B)$. Si $(-B)$ est négatif, l'indicateur 0 est armé. L'indicateur 0 est testé plus tard pour s'assurer que la première racine réelle calculée est bien celle de la plus grande valeur. Si $(-B)$ est négative (indicateur 0 armé), le sous-programme soustrait la racine carrée du discriminant de $(-B)$. Si $(-B)$ est positive (indicateur 0 désarmé), le sous-programme ajoute la racine carrée.

Besoins en mémoire : 127,5 octets: 79,5 pour le programme, 48 pour les variables.

Remarques : L'extension de ce programme pour lui donner la possibilité de traiter les équations cubiques devrait être simple. Les équations cubiques possèdent toujours au moins une racine réelle, et la fonction SOLVE pourrait être utilisée pour la trouver. La division synthétique devrait ensuite réduire l'équation cubique à une équation du second degré qui serait résolue par ce programme.

Instructions du programme :

1. Frappez les sous-programmes ; appuyez sur [C] lorsque vous avez terminé.
2. Appuyez sur [XEQ] Q pour commencer le sous-programme d'équation du second degré.
3. Frappez *A* et appuyez sur [R/S].
4. Frappez *B* et appuyez sur [R/S].
5. Frappez *C* et appuyez sur [R/S].
6. Visualisez la première valeur de *X*, si les racines sont réelles, ou la partie réelle, *R*, si les racines sont imaginaires.
7. Appuyez sur [R/S] pour voir la seconde valeur de *X*, ou pour voir la partie imaginaire, *I*, si les racines sont imaginaires.
8. Pour un nouveau cas, appuyez sur [R/S] et revenez à l'étape 3.

Variables utilisées :

A Coefficient de x^2.

B Coefficient de x.

C Constante.

X La première ou seconde valeur réelle de x.

R La partie réelle de la racine complexe.

I La partie positive, imaginaire de la racine complexe.

Exemple 1. Trouvez les racines de $3x^2 + 5x - 3 = 0$.

Touches :	Affichage :	Description :
XEQ Q	A?*valeur*	Lance le programme d'équation du second degré.
3 R/S	B?*valeur*	Stocke 3 dans *A*.
5 R/S	C?*valeur*	Stocke 5 dans *B*.
3 +/− R/S	X=-2,1350	Stocke −3 dans *C* et calcule la première valeur de *X*.
R/S	X=0,4684	Calcule la seconde valeur de *X*.

Exemple 2. Trouvez les racines de $3x^2 + 5x + 3 = 0$. Notez que la seule différence entre ce problème et l'exemple 1 est le signe de *C*. Si vous avez déjà exécuté l'exemple 1, tout ce que vous aurez à faire est de changer le signe de *C* :

Touches :	Affichage :	Description :
R/S	A?3,0000	Recommence le programme.
R/S	B?5,0000	Conserve *A*.
R/S	C?-3,0000	Conserve *B*.
+/− R/S	R=-0,8333	Change le signe de *C* et calcule la partie réelle de la racine complexe.
R/S	I=0,5528	Calcule la valeur positive de la racine imaginaire.

Exemple 3. Une balle est lancée verticalement à une vitesse de vingt mètres à la seconde d'une hauteur de deux mètres. En ne tenant pas compte de la résistance de l'air, à quel moment atteindra-t-elle le sol ? L'accélération due à la pesanteur est de 9,81 mètres par seconde^2.

Selon la mécanique newtonienne, ce problème peut être exprimé sous forme d'un polynôme du second degré, dans lequel T est le temps en secondes.

$$f(T) = (-9{,}81 \div 2)T^2 + 20T + 2$$

Touches :	**Affichage :**	**Description :**
XEQ Q	A?*valeur*	Lance le programme d'équation du second degré.
9,81 +/− ENTER 2 ÷		Stocke (−9,81/2) ds *A*.
R/S	B?*valeur*	
20 R/S	C?*valeur*	Stocke 20 dans *B*.
2 R/S	X=4,1751	Stocke 2 ds *C* et calcule *X* (aussi connu sous le nom de *T*).
R/S	X=-0,0977	Calcule l'autre racine.

Notez que puisque le temps négatif n'a aucune signification dans ce problème, c'est le premier résultat, 4,1751 s, la réponse.

Exemple 4. Trouvez les racines du polynôme du seond degré suivant, utilisant le programme tel qu'il est listé. Puis changez le signification de la comparaison de la ligne Q12 pour que la seconde racine soit trouvée en premier lieu et que les résultats soient comparés ensuite. Souvenez-vous de rétablir la ligne d'origine ou d'effacer le programme lorsque vous aurez terminé cet exemple.

$$x^2 + (3 \times 10^6)x + 1 = 0$$

Touches :	**Affichage :**	**Description :**
XEQ Q	A?*valeur*	Commence le programme.
1 R/S	B?*valeur*	Stocke 1 dans *A*.
3 E 6 R/S	C?*valeur*	Stocke 3×10^6 dans *B*.
1 R/S	X=-3.000.000,00	Stocke 1 dans *C* et calcule la première racine.

R/S	X=-3,3333E-7	Calcule la deuxième racine.
■ PRGM	Q36 GTO Q	Passe en mode de saisie de programme.
■ GTO · Q12	Q12 x<0?	Déplace le pointeur de programme vers la ligne Q12.
←	Q11 CF 0	Supprime la ligne Q12.
■ TESTS {x?0} {>0}	Q12 x>0?	Ajoute le test conditionnel $x>0$?.
C	-3,3333E-7	Annule la saisie de programme.
XEQ Q	A?1,0000	Lance le programme.
R/S R/S		Saute la saisie de données puisque les valeurs sont déjà stockées.
R/S	X=0,0000	Calcule la première racine en utilisant les saisies précédentes.
R/S	DIVIDE BY 0	Essaie de calculer la deuxième racine.

Comme vous pouvez le voir, le résultat d'un simple changement dans l'ordre des calculs peut être très significatif.

Si vous replacez les premières valeurs calculées dans l'équation, vous verrez que le côté gauche de l'équation est zéro pour la racine de la valeur absolue la plus petite (comme le suggère la théorie) et 1 pour la racine de la valeur absolue la plus grande. Cela signifie-t-il que le résultat de $-3\,000\,000{,}0000$ est incorrect ? Non. Si vous augmentez ou diminuez cette valeur d'un dans le chiffre le moins significatif et si vous placez le résultat dans l'équation d'origine, le côté gauche de l'équation sera 31 ou -29. Donc $-3\,000\,000{,}0000$, bien que n'étant pas exactement correct, est le meilleur résultat possible compte tenu de la résolution à 12 chiffres de la machine.

Transformations de coordonnées

Ce programme permet la translation de coordonnées bi-dimensionnelles et la rotation.

Les formules suivantes sont utilisées pour convertit un point P venant de la paire de coordonnées cartésiennes (x, y) dans l'ancien système en la paire (u, v) dans le nouveau, après translation et rotation.

$$u = (x - m)\ \cos\theta + (y - n)\ \sin\theta$$

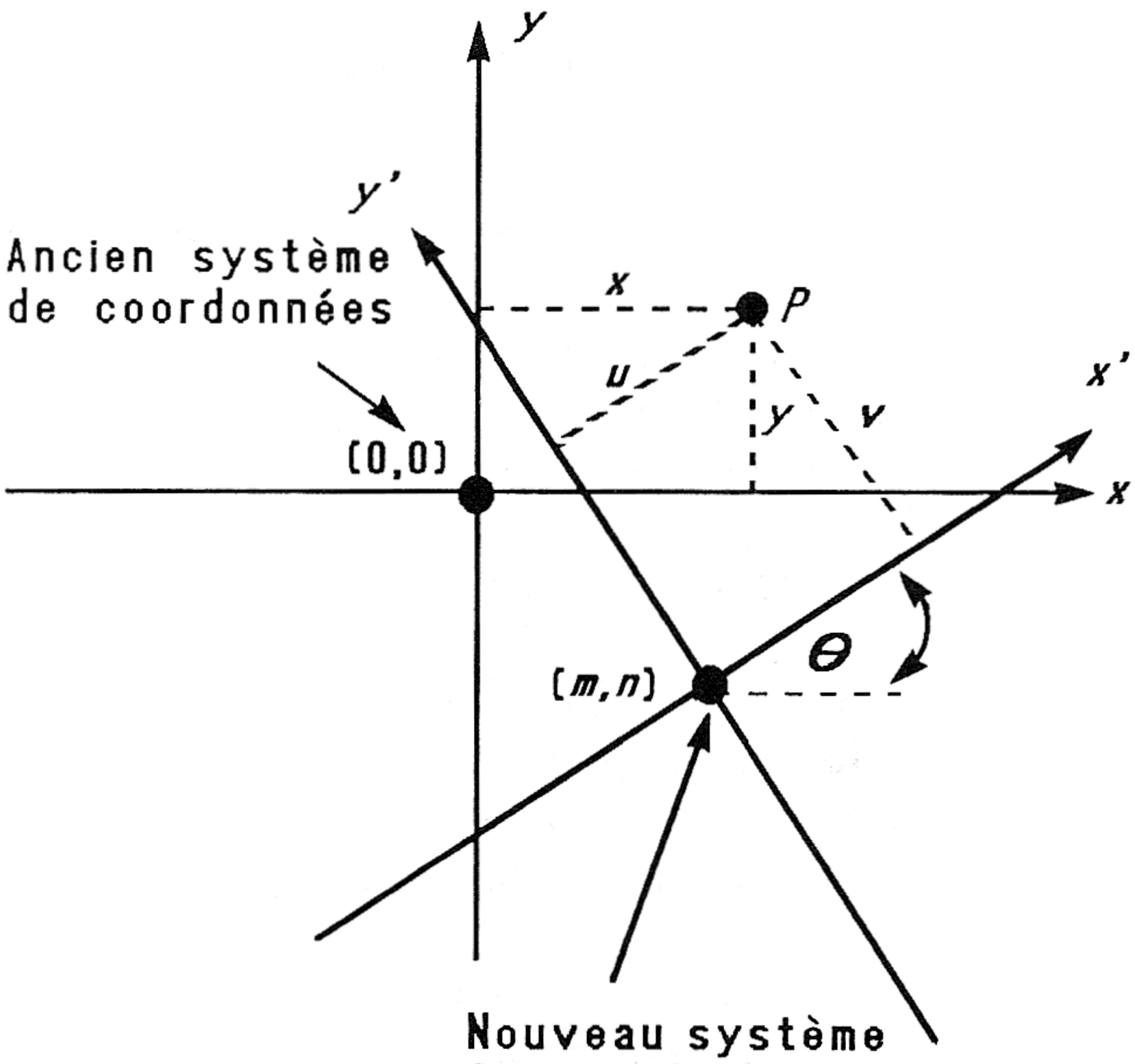
$$v = (y - n)\ \cos\theta - (y - n)\ \sin\theta$$

La transformation inverse est accomplie grâce aux formules :

$$x = u\ \cos\theta - v\ \sin\theta + m$$

$$y = u\ \sin\theta + v\ \cos\theta + n$$

Les fonctions complexes et polaires-rectangulaires du HP-32S simplifient ces calculs.



Une rotation en deux dimensions autour de l'axe

Liste de programme :

Lignes de programme :	Description :
D01 LBL D	Ce sous-programme définit le nouveau système de coordonnées.
D02 INPUT M	Sollicite et stocke *M*, la nouvelle coordonnée *x* de l'origine.
D03 INPUT N	Sollicite et stocke *N*, la nouvelle coordonnée *y* de l'origine.
D04 INPUT T	Sollicite et stocke *T*, l'angle θ.
D05 GTO D	Boucle pour permettre la vérification de saisies.
Octets et totalisation de	contrôle : 007,5, 1CD9
N01 LBL N	Ce sous-programme convertit l'ancien système en nouveau système.
N02 INPUT X	Sollicite et stocke X, l'ancienne coord. *x*.
N03 INPUT Y	Sollicite et stocke Y, l'ancienne coord. *y*.
N04 RCL X	Remonte Y et rappelle X dans le registre X.
N05 RCL N	Remonte X et Y et rappelle *N* dans le registre X.
N06 RCL M	Remonte *N*, X et Y et rappelle *M*.
N07 CMPLX-	Calcule $(X-M)$ et $(Y-N)$.
N08 RCL T	Remonte $(X-M)$ et $(Y-N)$ et rappelle *T*.
N09 +/-	Change le signe de *T* parce que $\sin(-T)$ est égal à $-\sin(T)$.
N10 1	Définit un rayon 1 pour les calculs de $\cos(T)$ et de $-\sin(T)$.
N11 θ,r→y,x	Calcule $\cos(T)$ et $-\sin(T)$ dans les registres X et Y.
N12 CMPLX×	Calcule $(X-M)\cos(T) + (Y-N)\sin(T)$ et $(Y-N)\cos(T) - (X-M)\sin(T)$.
N13 STO U	Stocke la coordonnée *x* dans la variable *U*.
N14 x<>y	Echange les positions des coordonnées.
N15 STO V	Stocke la coordonnée *y* dans la variable *V*.
N16 x<>y	Echange à nouveau les positions des coordonnées.
N17 VIEW U	Arrête le programme pour afficher *U*.
N18 VIEW V	Arrête le programme pour afficher *V*.
N19 GTO N	Revient au point de départ pour un autre calcul.
Octets et totalisation de	contrôle : 028,5, 6078

001 LBL 0	Ce sous-programme convertit l'ancien système en nouveau système.
002 INPUT U	Sollicite et stocke U.
003 INPUT V	Sollicite et stocke V.
004 RCL U	Remonte V et rappelle U.
005 RCL T	Remonte U et V et rappelle T.
006 1	Définit un rayon de 1 pour le calcul de $\sin(T)$ et de $\cos(T)$.
007 θ,r→y,x	Calcule $\cos(T)$ et $\sin(T)$.
008 CMPLX×	Calcule $U\cos(T) - V\sin(T)$ et $U\sin(T) + V\cos(T)$.
009 RCL N	Remonte les résultats précédents et rappelle N.
010 RCL M	Remonte les résultats et rappelle M.
011 CMPLX+	Termine le calcul en ajoutant M et N aux résultats précédents.
012 STO X	Stocke la coordonnée x dans la variable X.
013 x<>y	Echange les positions des coordonnées.
014 STO Y	Stocke la coordonnée y dans la variable Y.
015 x<>y	Echange à nouveau les positions des coordonnées.
016 VIEW X	Arrête le programme pour afficher X.
017 VIEW Y	Arrête le programme pour afficher Y.
018 GTO 0	Revient au point de départ pour un nouveau calcul.

Octets et totalisation de contrôle : 027,0, 9AE6

Indicateurs utilisés : aucun.

Besoins en mémoire : 119 octets: 63 pour le programme, 56 pour les variables.

Instructions du programme :

1. Frappez les sous-programmes du programme ; appuyez sur [C] lorsque vous avez terminé.
2. Appuyez sur [XEQ] D pour lancer la séquence de sollicitation qui définit la transformation de coordonnées.
3. Frappez la coordonnée x de l'origine du nouveau système M et appuyez sur [R/S].
4. Frappez la coordonnée y de l'origine du nouveau système N et appuyez sur [R/S].

5. Frappez l'angle de rotation T et appuyez sur [R/S].

6. Pour passer de l'ancien système (a.s.) au nouveau, continuez avec l'étape 7. Pour passer du nouveau système (n.s.) à l'ancien, passez directement à l'étape 12.

7. Appuyez sur [XEQ] N pour lancer le sous-programme de transformation ancien-nouveau.

8. Frappez X et appuyez sur [R/S].

9. Frappez Y, appuyez sur [R/S] et voyez la coordonnée x, U dans le nouveau système (n.s.).

10. Appuyez sur [R/S] et voyez la coordonnée, y, V dans le n. s.

11. Pour une autre transformation ancien-nouveau, appuyez sur [R/S] et passez au point 8. Pour une transformation nouveau-ancien, continuez avec l'étape 12.

12. Appuyez sur [XEQ] O pour lancer le ss-prog. nouveau-ancien.

13. Frappez U (la coord. x dans le n. s.) et appuyez sur [R/S].

14. Frappez V (la coord. y dans le n. s.) et [R/S] pour voir X.

15. Appuyez sur [R/S] pour voir Y.

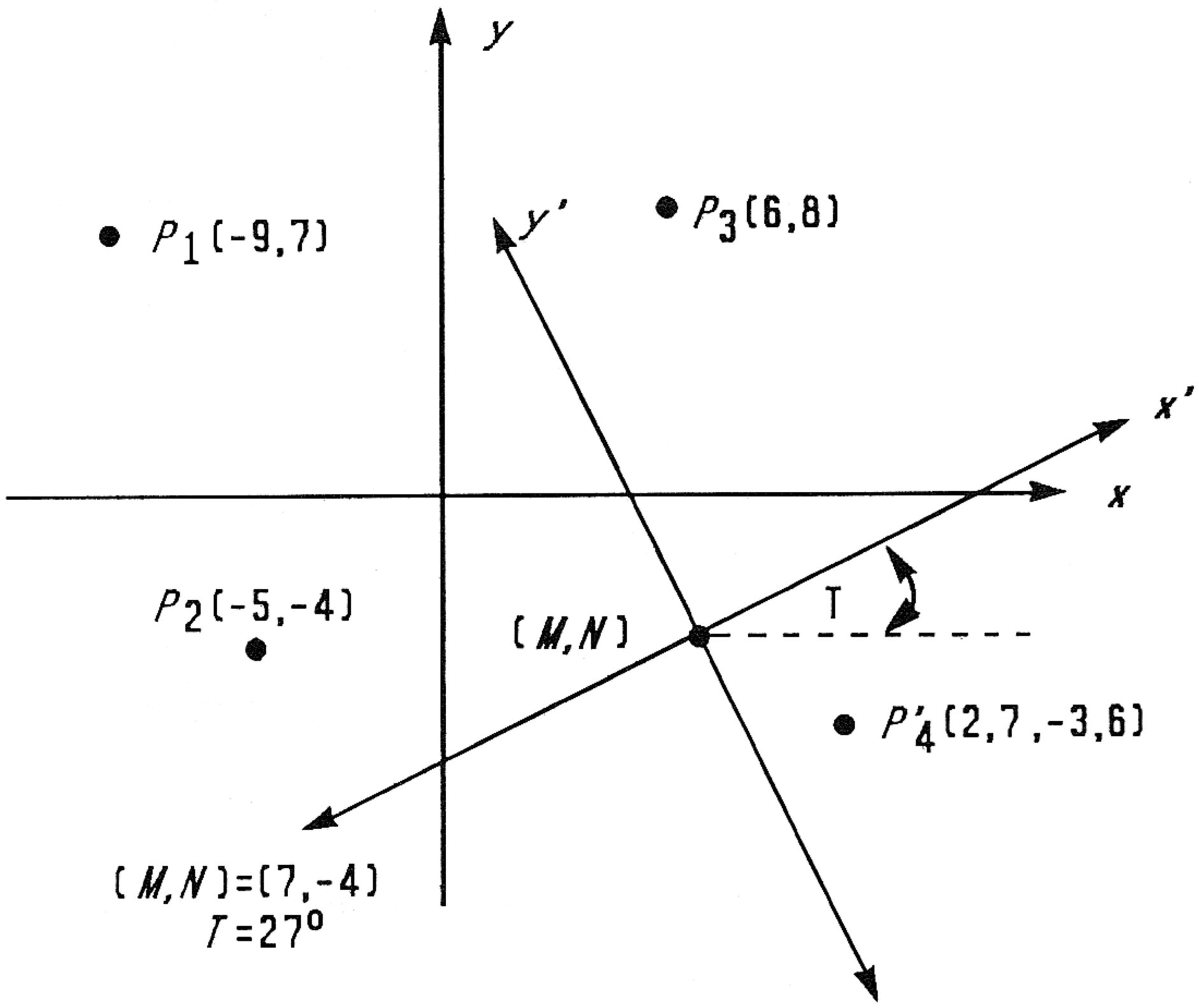
16. Pour une autre transformation nouveau-ancien, appuyez sur [R/S] et passez au point 13. Pour une transformation ancien-nouveau, revenez au point 7.

Variables utilisées :

M	La coordonnée x à l'origine du nouveau système.
N	La coordonnée y à l'origine du nouveau système.
T	L'angle de rotation, θ, entre les deux systèmes.
X	La coordonnée x d'un point dans l'ancien système.
Y	La coordonnée y d'un point dans l'ancien système.
U	La coordonnée x d'un point dans le nouveau système.
V	La coordonnée y d'un point dans le nouveau système.

Remarques : Pour la translation seulement, frappez une valeur zéro pour T. Pour la rotation seulement, frappez zéro pour M et N.

Exemple : Pour le système de coordonnées ci-dessous, convertissez les points P_1, P_2 et P_3, qui se trouvent actuellement dans le système (X,Y), en des points du système (X',Y'). Convertissez le point P'_4, du système (X',Y'), au système (X,Y).



Touches :	Affichage :	Description :
■ MODES {DG}		Définit le mode degrés puisque T est donné en degrés.
XEQ D	M?*valeur*	Lance le sous-programme qui définit la transformation.
7 R/S	N?*valeur*	Stocke 7 dans M.
4 +/− R/S	T?*valeur*	Stocke −4 dans N.
27 R/S	M?7,0000	Stocke 27 dans T.

XEQ N	X?*valeur*	Lance le sous-programme ancien-nouveau.
9 +/- R/S	Y?*valeur*	Stocke −9 dans *X*.
7 R/S	U=-9,2622	Stocke 7 dans *Y* et calcule *U*.
R/S	V=17,0649	Calcule *V*.
R/S	X?-9,0000	Reprend le sous-programme ancien-nouveau pour le problème suivant.
5 +/- R/S	Y?7,0000	Stocke −5 dans *X*.
4 +/- R/S	U=-10,6921	Stocke −4 dans *Y*.
R/S	V=5,4479	Calcule *V*.
R/S	X?-5,0000	Reprend le sous-programme ancien-nouveau pour le problème suivant.
6 R/S	Y?-4,0000	Stocke 6 dans *X*.
8 R/S	U=4,5569	Stocke 8 dans *Y* et calcule *U*.
R/S	V=11,1461	Calcule *V*.
XEQ O	U?4,5569	Lance le sous-programme nouveau-ancien.
2,7 R/S	V?11,1461	Stocke 2,7 dans *U*.
3,6 +/- R/S	X=11,0401	Stocke −3,6 dans *V* et calcule *X*.
R/S	Y=-5,9818	Calcule *Y*.

13

Programmes statistiques

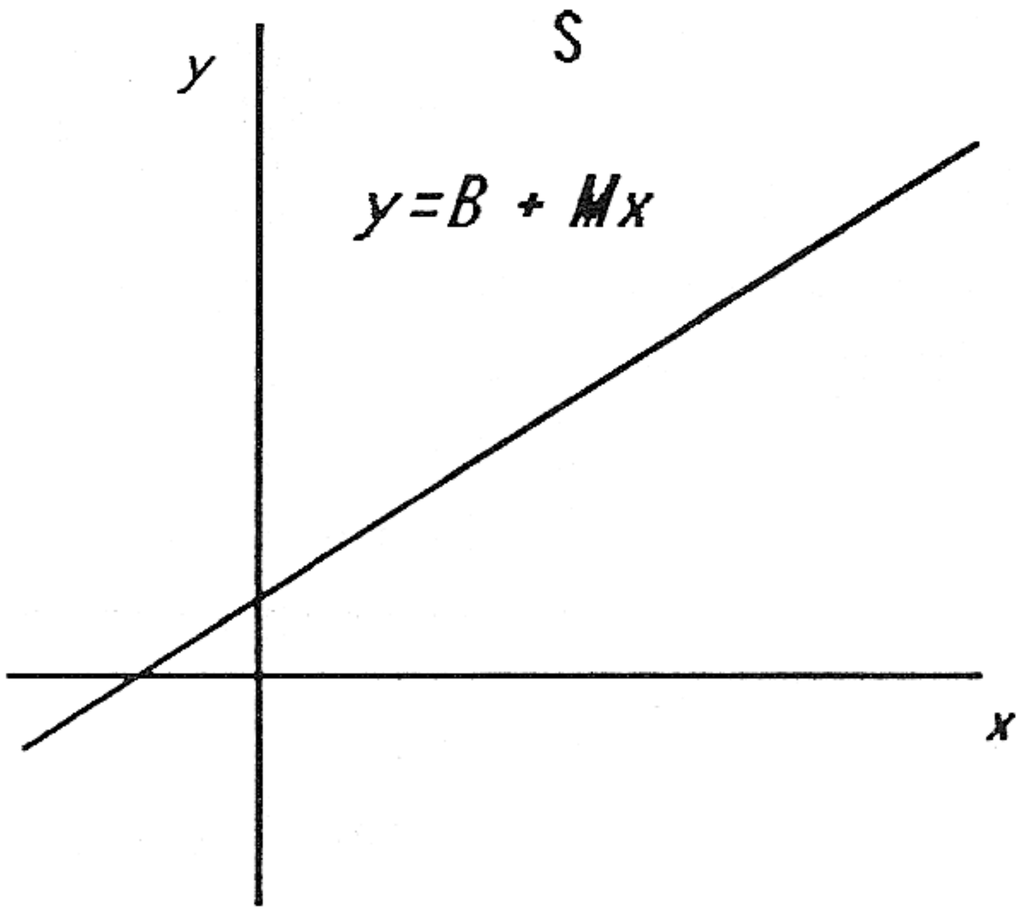
L'utilisation que fait chaque programme de la mémoire et la totalisation de contrôle peuvent être vérifiés en utilisant le catalogue des programmes (Cf. page 85).

Ajustement de courbes

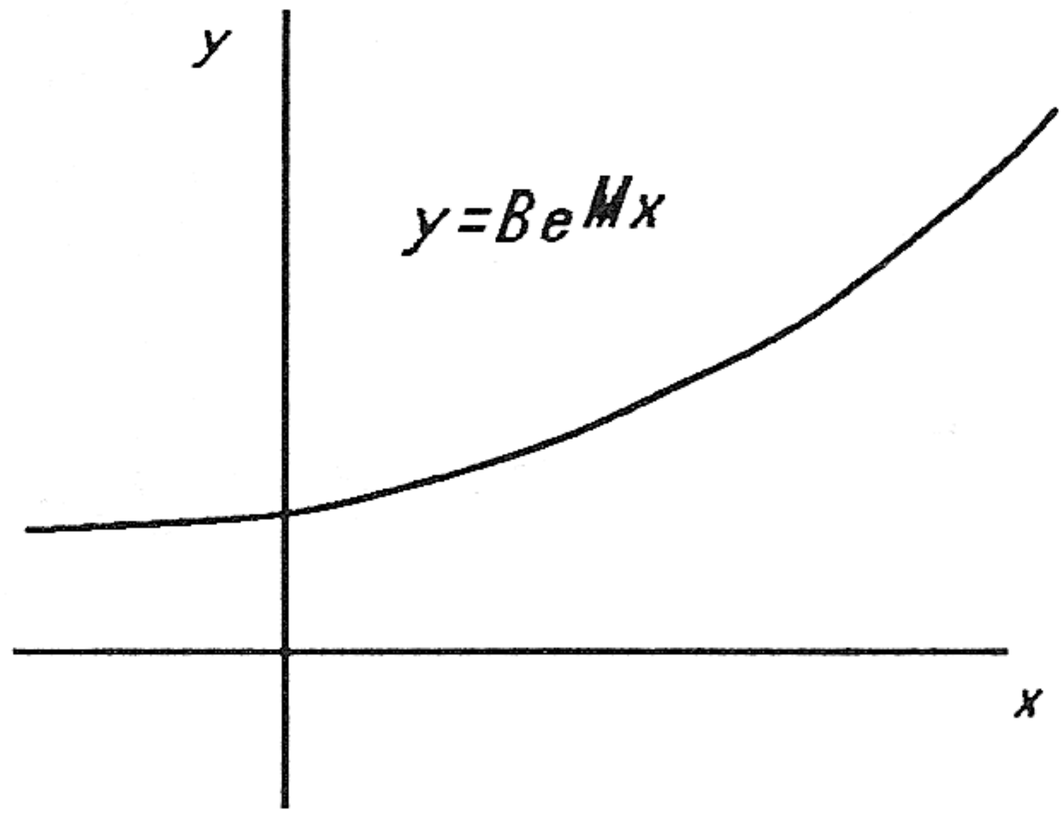
Ce programme peut être utilisé pour faire correspondre un modèle d'équation à vos données ; il y a quatre modèles : la ligne droite, la courbe logarithmique, la courbe exponentielle et la courbe de puissance. Le programme accepte deux ou plusieurs paires de données (x, y) et calcule ensuite le coefficent de corrélation, r, et les deux coefficients de régression, m et b. Le programme comprend un sous-programme pour calculer les estimations $\hat{x}$ et $\hat{y}$ (pour la définition de ces valeurs, lisez « Régression linéaire », au chapitre 11, page 158).

Des exemples des courbes et les équations qui y correspondent sont indiqués ci-après. Les fonctions de régression interne du HP-32S sont utilisées pour calculer les coefficients de régression.

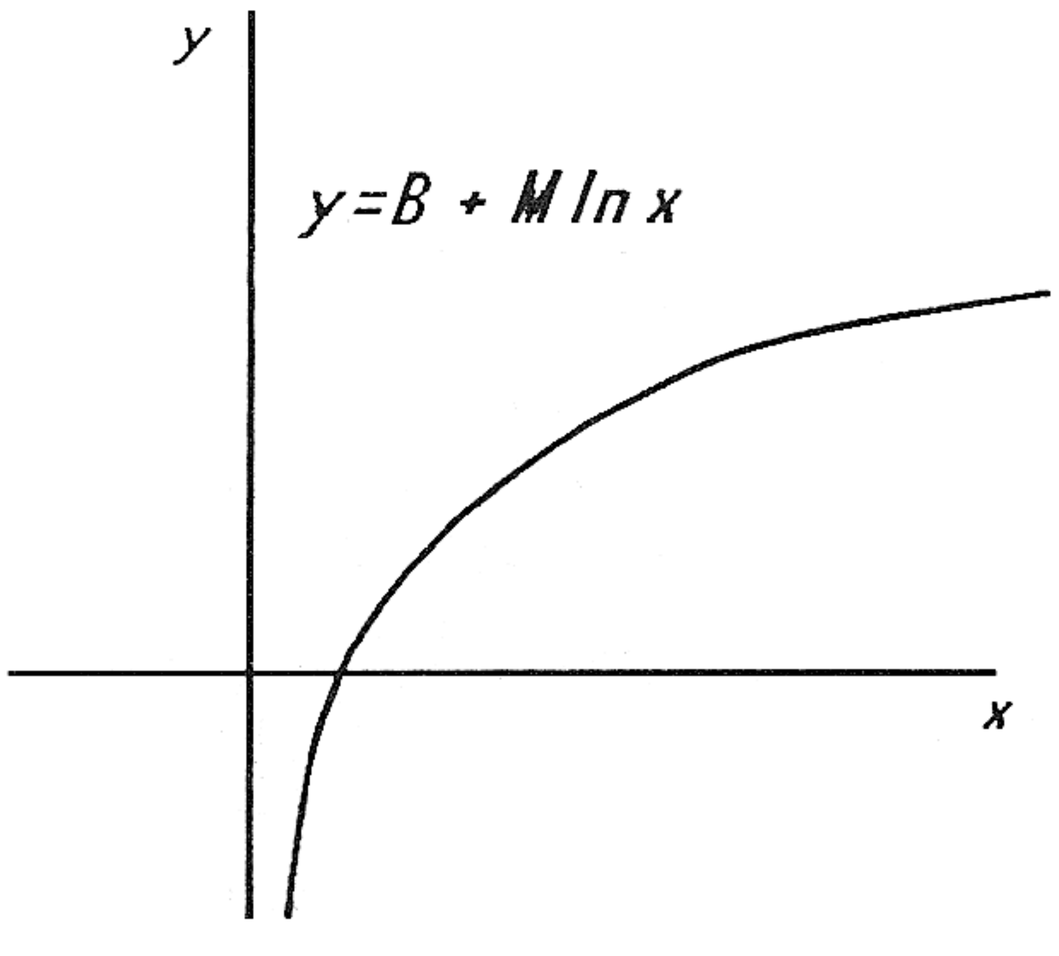
Ligne droite
(Straight Line)
S



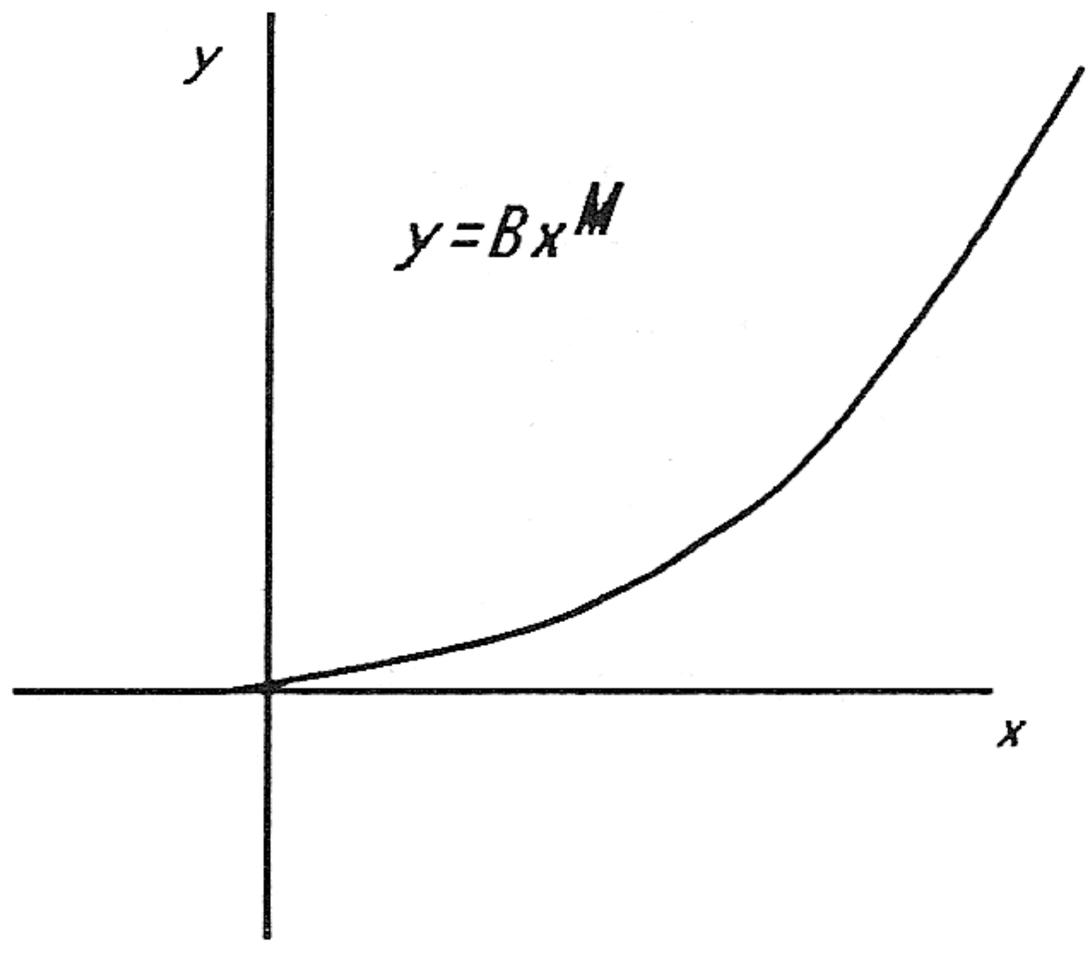
Courbe exponentielle
E



Courbe logarithmique
L



Courbe de puissance
P



Pour ajuster des courbes logarithmiques, les valeurs de x doivent être positives. Pour ajuster des courbes exponentielles, les valeurs de y doivent être positives. Une erreur `LOG(NEG)` se produira si un nombre négatif est introduit dans ces cas.

Les valeurs de données de très grande taille mais présentant de petites différences peuvent provoquer des problèmes de précision, tout comme les valeurs de données dont les tailles diffèrent forte-ment. Voyez « Limitations de la précision des données » au chapitre 11, page 160.

Liste de programme :

Lignes de programme :	Description :
S01 LBL S	Ce sous-programme définit l'état du modèle de ligne droite.
S02 1	Saisit une valeur d'index pour stockage ultérieur dans *i* (pour adressage indirect).
S03 CF 0	Désarme l'indicateur 0, l'indicateur de lnX.
S04 CF 1	Désarme l'indicateur 1, l'indicateur de lnY.
S05 GTO Z	Se branche sur le point d'entrée commun, Z.
Octets et totalisation de	contrôle : 007,5, 17CA
L01 LBL L	Ce sous-programme définit l'état du modèle logarithmique.
L02 2	Saisit la valeur d'index pour stockage ultérieur dans *i* (pour l'adressage indirect).
L03 SF 0	Définit l'indicateur 0, l'indicateur de lnX.
L04 CF 1	Désarme l'indicateur 1, l'indicateur de lnY.
L05 GTO Z	Se branche sur le point d'entrée commun, Z.
Octets et totalisation de	contrôle : 007,5, 6047
E01 LBL E	Ce sous-programme définit l'état du modèle exponentiel.
E02 3	Saisit la valeur d'index pour stockage ultérieur dans *i* (pour l'adressage indirect).
E03 CF 0	Désarme l'indicateur 0, l'indicateur de lnX.
E04 SF 1	Définit l'indicateur 1, l'indicateur de lnY.
E05 GTO Z	Se branche sur le point d'entrée commun Z.
Octets et totalisation de	contrôle : 007,5, C0F1
P01 LBL P	Ce sous-programme définit l'état du modèle de puissance.
P02 4	Saisit la valeur d'index pour stockage ultérieur dans *i* (pour l'adressage indirect).
P03 SF 0	Définit l'indicateur 0, l'indicateur de lnX.
P04 SF 1	Définit indicateur 1, l'indicateur de lnY.
Octets et totalisation de	contrôle : 006,0, A26B

Z01 LBL Z	Définit un point d'entrée commun pour tous les modèles.
Z02 CLΣ	Efface les registres statistiques.
Z03 STO i	Stocke la valeur d'index dans *i* pour adressage indirect.
Z04 0	Met le compteur de boucle à zéro pour la première saisie.

Octets et totalisation de contrôle : 006,0, CC1B

W01 LBL W	Définit le début de la boucle de saisie.
W02 1	Ajuste le compteur de boucle d'une unité pour solliciter la saisie.
W03 +	
W04 STO X	Stocke le compteur de boucle dans *X* pour qu'il apparaisse avec la solicitation de *X*.
W05 INPUT X	Affiche le compteur avec la sollicitation et stocke la saisie pour *X*.
W06 FS? 0	Si l'indicateur 0 est armé...
W07 LN	...prend le log naturel de la saisie.
W08 STO B	Stocke cette valeur pour le sous-programme de correction.
W09 INPUT Y	Sollicite, et stocke *Y*.
W10 FS? 1	Si l'indicateur 1 est armé...
W11 LN	...prend le log naturel de la saisie.
W12 STO R	
W13 RCL B	
W14 Σ+	Accumule *B* et *R* comme paires de données *x,y* dans les registres statistiques.
W15 GTO W	Boucle pour une autre paire *X*, *Y*.

Octets et totalisation de contrôle : 022,5, 1A43

U01 LBL U	Définit le début du sous-programme « Undo », défaire.
U02 RCL R	Rappelle la paire de données la plus récente.
U03 RCL B	
U04 Σ-	Supprime cette paire de l'accumulation statistique.
U05 GTO W	Boucle pour une autre paire *X,Y*.

Octets et totalisation de contrôle : 007,5, 5D02

`R01 LBL R`	Définit le début du sous-programme de sorties.
`R02 r`	Calcule le coefficient de corrélation.
`R03 STO R`	Stocke ce coefficient dans *R*.
`R04 VIEW R`	Affiche le coefficient de corrélation.
`R05 b`	Calcule le coefficient *b*.
`R06 FS? 1`	Si l'indicateur 1 est armé, prend le log naturel de *b*.
`R07 e^x`	
`R08 STO B`	Stocke *b* dans *B*.
`R09 VIEW B`	Affiche la valeur.
`R10 m`	Calcule le coefficient *m*.
`R11 STO M`	Stocke *m* dans *M*.
`R12 VIEW M`	Affiche la valeur.

Octets et totalisation de contrôle : 018,0, 7492

`Y01 LBL Y`	Définit le début de la boucle d'estimation (de projection).
`Y02 INPUT X`	Affiche, sollicite, et, s'il y a eu modification, stocke la valeur x dans X.
`Y03 XEQ(i)`	Appelle la sous-routine de calcul de $\hat{y}$.
`Y04 STO Y`	Stocke la valeur $\hat{y}$ dans Y.
`Y05 INPUT Y`	Affiche, sollicite et, s'il y a eu modification, stocke la valeur y dans Y.
`Y06 6`	
`Y07 STO+ i`	Ajuste la valeur de l'index pour qu'il s'adresse à la sous-routine appropriée.
`Y08 XEQ(i)`	Appelle la sous-routine pour calculer $\hat{x}$.
`Y09 STO X`	Stocke $\hat{x}$ dans X pour la boucle suivante.
`Y10 GTO Y`	Boucle pour une autre estimation.

Octets et totalisation de contrôle : 015,0, 9AEA

`A01 LBL A`	Cette sous-routine calcule $\hat{y}$ pour le modèle de ligne droite.
`A02 RCL M`	
`A03 RCL× X`	
`A04 RCL+ B`	Calcule $\hat{y} = MX + B$.
`A05 RTN`	Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 007,5, 0E85

```
G01 LBL G
```
Cette sous-routine calcule $\hat{x}$ pour le modèle de droite.

```
G02 STO- i
```
Remet l'index à sa valeur d'origine.

```
G03 RCL Y
G04 RCL- B
G05 RCL÷ M
```
Calcule $\hat{x} = (Y - B) \div M$.

```
G06 RTN
```
Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 009,0, FDF1

```
B01 LBL B
```
Cette sous-routine calcule $\hat{y}$ pour le modèle logarithmique.

```
B02 RCL X
B03 LN
B04 RCL× M
B05 RCL+ B
```
Calcule $\hat{y} = M \ \ln X + B$.

```
B06 RTN
```
Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 009,0, 1B06

```
H01 LBL H
```
Cette sous-routine calcule $\hat{x}$ pour le modèle logarithmique.

```
H02 STO- i
```
Rétablit la valeur de l'index à sa valeur originale.

```
H03 RCL Y
H04 RCL- B
H05 RCL÷ M
H06 e^x
```
Calcule $\hat{x} = e^{(Y - B) \div M}$.

```
H07 RTN
```
Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 010,5, C783

```
C01 LBL C
```
Cette sous-routine calcule $\hat{y}$ pour le modèle exponentiel.

```
C02 RCL M
C03 RCL× X
C04 e^x
C05 RCL× B
```
Calcule $\hat{y} = Be^{MX}$.

```
C06 RTN
```
Revient au programme d'appel.

Octets et totalisation de contrôle : 009,0, B411

```
I01 LBL I
```
Cette sous-routine calcule $\hat{x}$ pour le modèle exponentiel.

```
I02 STO- i
```
Remet le nombre de l'index à sa valeur initiale.

```
I03 RCL Y
I04 RCL÷ B
I05 LN
I06 RCL÷ M
```
Calcule $\hat{x} = (\ln(Y \div B)) \div M$.

```
I07 RTN
```
Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 010,5, 01D6

```
D01 LBL D
```
Cette sous-routine calcule $\hat{y}$ pour le modèle de puissance.

```
D02 RCL X
D03 RCL M
D04 y^x
D05 RCL× B
```
Calcule $Y = B(X^M)$.

```
D06 RTN
```
Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 009,0, B4D4

```
J01 LBL J
```
Cette sous-routine calcule $\hat{x}$ pour le modèle de puissance.

```
J02 STO- i
```
Remet l'index à sa valeur d'origine.

```
J03 RCL Y
J04 RCL÷ B
J05 RCL M
J06 1/x
J07 y^x
```
Calcule $\hat{x} = (Y/B)^{1/M}$.

```
J08 RTN
```
Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 012,0, FAA4

Indicateurs utilisés : L'ndicateur 0 est armé si un logarithme naturel est requis pour la saisie de X. L'indicateur 1 est armé si le logarithme naturel est requis pour la saisie de *Y*.

Besoins en mémoire : 270 octets : 174 pour le programme, 96 pour les données (48 registres statistiques).

Instructions du programme :

1. Frappez les sous-programmes du programme ; appuyez sur [C] lorsque vous avez terminé.
2. Appuyez sur [XEQ] et choisissez le type de courbe vous désirez ajuster en appuyant sur :
   - S pour une droite (*straight line*) ;
   - L pour une courbe logarithmique ;
   - E pour une courbe exponentielle ; ou sur
   - P pour une courbe de puissance.
3. Frappez la valeur de x et appuyez sur [R/S].
4. Frappez la valeur de y et appuyez sur [R/S].
5. Répétez les étapes 3 et 4 pour chaque paire de données. Si vous avez fait une erreur après avoir appuyé sur [R/S] au point 3 (lorsque la sollicitation `Y?`*valeur* était encore visible), appuyez à nouveau sur [R/S] (pour afficher la sollicitation `X?`*valeur*) et appuyez sur [XEQ] U pour *défaire* (*Undo*) la dernière paire de données. Si vous avez fait une erreur après l'étape 4, appuyez sur [XEQ] U. Dans tous les cas, reprenez avec l'étape 3.
6. Après la saisie de toutes les données, appuyez sur [XEQ] R pour voir le coefficient de corrélation, R.
7. Appuyez sur [R/S] pour voir le coefficient de régression, B.
8. Appuyez sur [R/S] pour voir le coefficient de régression, M.
9. Appuyez sur [R/S] pour voir la sollicitation `X?`*valeur* pour le sous-programme d'estimation de $\hat{x}$-, $\hat{y}$.
10. Si vous désirez estimer $\hat{y}$ sur base de x, frappez x après la sollicitation `X?`*valeur*, puis appuyez sur [R/S] pour voir $\hat{y}$ (`Y?`).
11. Si vous désirez estimer $\hat{x}$ sur base de y, appuyez sur [R/S] jusqu'à ce que vous voyez la sollicitation `Y?`*valeur*, frappez y, puis appuyez sur [R/S] pour $\hat{x}$ (`X?`).
12. Pour d'autres estimations, passez aux points 10 ou 11.
13. Pour un nouveau cas, passez au point 2.

Variables utilisées :

B	Coefficient de régression (ordonnée à l'origine d'une ligne droite) ; utilisé comme brouillon.
M	Coeff. de régression (pente d'une ligne droite).
R	Coeff. de corrélation ; utilisé comme brouillon.
X	La valeur x d'une paire de données lors de la saisie des données ; x hypothétique lors de la projection de $\hat{y}$; ou bien $\hat{x}$ (estimation pour x) en présence d'un y hypothétique.
Y	Valeur y d'une paire de données lors de la saisie de données ; y hypothétique lors de la projection de $\hat{x}$; ou bien $\hat{y}$ (estimation pour y) en présence d'un x hypothétique.
i	Variable d'index utilisée pour l'adressage indirect de l'équation correcte de projection pour $\hat{x}$-, $\hat{y}$.
Registres statistiques	Accumulation statistique et calcul.

Exemple 1. Ajustez une droite aux données ci-dessous. Faites une « erreur » lors de la saisie de la troisième paire de données, puis corrigez-la à l'aide du sous-programme « *undo* ». Estimez également y pour une valeur de x de 37. Estimez x pour une valeur de y de 101.

X	40,5	38,6	37,9	36,2	35,1	34,6
Y	104,5	102	100	97,5	95,5	94

Touches :	**Affichage :**	**Description :**
XEQ S	X?1,0000	Sous-programme de la droite (*straight line*).
40,5 R/S	Y?*valeur*	Saisit la valeur x de la paire de données.
104,5 R/S	X?2,0000	Saisit la valeur y de la paire de données.

38,6 [R/S]	Y?104,5000	Saisit la valeur x de la paire de données.
102 [R/S]	X?3,0000	Saisit la valeur y de la paire de données.

Saisit intentionnellement 379 au lieu de 37,9 pour tester la correction des saisies.

379 [R/S]	Y?102,0000	Saisit une valeur x erronée.
[R/S]	X?4,0000	Récupère la sollicitaion X?.
[XEQ] U	X?3,0000	Supprime la dernière paire.

C'est le moment de faire la correction.

37,9 [R/S]	Y?102,0000	Saisit la valeur x correcte pour cette paire de données.
100 [R/S]	X?4,0000	Saisit la valeur y de la paire.
36,2 [R/S]	Y?100,0000	Saisit la valeur x de la paire.
97,5 [R/S]	X?5,0000	Saisit la valeur y de la paire.
35,1 [R/S]	Y?97,5000	Saisit la valeur x de la paire.
95,5 [R/S]	X?6,0000	Saisit la valeur y de la paire.
34,6 [R/S]	Y?95,5000	Saisit la valeur x de la paire.
94 [R/S]	X?7,0000	Saisit la valeur y de la paire.
[XEQ] R	R=0,9955	Calcule le coefficient de corrélation.

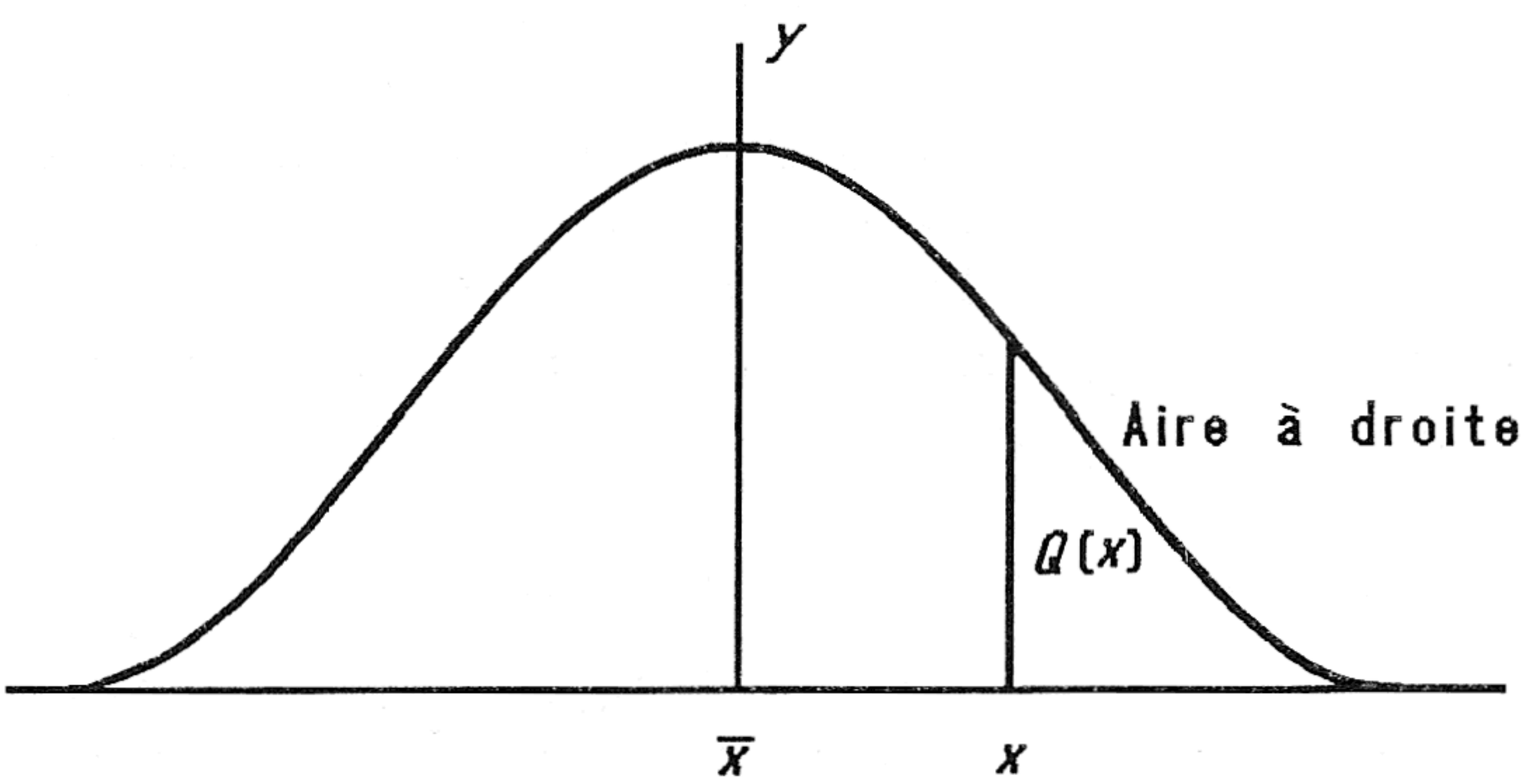
R/S	B=33,5271	Calcule le coefficient de régression B.
R/S	M=1,7601	Calcule le coefficient de régression M.
R/S	X?7,0000	Sollicite une valeur de x hypothétique.
37 R/S	Y?98,6526	Stocke 37 dans X et calcule $\hat{y}$.
101 R/S	X?38,3336	Stocke 101 dans Y et calcule $\hat{x}$.

Exemple 2. Répétez l'exemple 1 (avec les mêmes données) pour des ajustements de courbes logarithmiques, exponentielles et de puissance. Le tableau ci-dessous vous donne le label et les résultats (les coefficients de corrélation, de régression, et les estimations pour x et y) pour chaque type de courbe). Il faudra re-saisir les valeurs des données chaque fois que vous exécutez le programme pour une courbe différente.

	Logarithmic	**Exponential**	**Puissance**
Pour commencer :	XEQ L	XEQ E	XEQ P
R	0,9965	0,9945	0,9959
B	−139,0088	51,1312	8,9730
M	65,8446	0,0177	0,6640
Y ($\hat{y}$ si X=37)	98,7508	98,5870	98,6845
X ($\hat{x}$ si Y=101)	38,2857	38,3628	38,3151

Distributions normales et inverses-normales

La distribution est fréquemment utilisée pour illustrer le comportement des variations aléatoires autour d'une moyenne. Ce modèle prend pour hypothèse de départ que la distribution-échantillon est symétrique par rapport à la moyenne, *M*, avec un écart-type (*Angl. Standard Deviation*), *S* et à une forme semblable à l'illustration ci-dessous. Pour une valeur *x* donnée, ce programme calcule la probabilité qu'une sélection aléatoire des données échantillon aura une valeur supérieure. Ceci est décrit en littérature anglaise sous le nom de *Upper Tail Area,* aire droite *Q*(*x*). Ce programme présente aussi l'inverse : pour une valeur donnée de *Q*(*x*), le programme calcule la valeur correspondante de *x*.



$$Q(x) = 0{,}5 - \frac{1}{\sigma\sqrt{2\pi}} \int_{\bar{x}}^{x} e^{-((x - \bar{x}) \div \sigma)^2 \div 2}\, dx.$$

Ce programme utilise la caractéristique d'intégration du HP–32S pour intégrer l'équation de la courbe de fréquence normale. L'inverse est obtenu par la méthode de Newton pour rechercher de manière itérative une valeur de *x* qui donne la probabilité *Q*(*x*).

Liste de programme :

Lignes de programme :	Description :
S01 LBL S	Ce sous-programme commence le programme d'écart-type.
S02 0	Stocke la valeur par défaut de la moyenne.
S03 STO M	
S04 INPUT M	Sollicite et stocke la moyenne, *M*.

```
S05 1
S06 STO S
S07 INPUT S
S08 RTN
```

Stocke la valeur d'écart-type par défaut.

Sollicite et stocke l'écart-type (*Standard deviaton*), *S*.

Arrête l'affichage de l'écart-type.

Octets et totalisation de contrôle : 012,0, 1F60

```
D01 LBL D
D02 INPUT X
D03 XEQ Q
D04 STO Q
D05 VIEW Q
D06 GTO D
```

Ce sous-programme calcule Q(*X*) connaissant *X*.

Sollicite et stocke *X*.

Calcule l'aire droite de la courbe.

Stocke la valeur de *Q*.

Affiche *Q*(*X*).

Boucle pour calculer un autre *Q*(*X*).

Octets et totalisation de contrôle : 009,0, 002C

```
I01 LBL I
I02 INPUT Q
I03 RCL M
I04 STO X
```

Ce sous-programme calcule *X* connaissant Q(*X*).

Sollicite et stocke *Q*(*X*).

Rappelle la moyenne.

Stocke la moyenne comme estimation de *X*, nommée X_{estim}.

Octets et totalisation de contrôle : 006,0, ED6E

```
T01 LBL T
T02 XEQ Q
T03 RCL- Q
T04 RCL X
T05 STO D
T06 R↓
T07 XEQ F
T08 RCL÷ T
T09 ÷
T10 STO+ X
T11 ABS
T12 0,0001
T13 x<y?
T14 GTO T
T15 RCL X
```

Ce label définit le début de la boucle.

Calcule ($Q(X_{estim}) - Q(X)$).

Calcule la dérivée à X_{estim}.

Calcule la correction pour X_{estim}.

L'ajoute pour un nouveau X_{estim}.

Teste pour voir si la correction est significative.

Revient pour refaire une boucle si la correction est significative. Continue si la correction n'est pas significative.

```
T16 VIEW X
T17 GTO I
```
Affiche la valeur calculée de X.
Boucle pour calculer un autre X.

Octets et totalisation de contrôle : 033,5, 4355

```
Q01 LBL Q
```
Cette sous-routine calcule l'aire droite de la courbe $Q(x)$.

```
Q02 RCL M
```
Rappelle la valeur inférieure.

```
Q03 RCL X
```
Rappelle la valeur supérieure.

```
Q04 FN= F
```
Choisit la fonction définie par LBL F pour intégration.

```
Q05 ∫FN d D
```
Intègre la fonction normale en utilisant la variable factice D.

```
Q06 2
Q07 π
Q08 ×
Q09 SQRT
Q10 RCL S
Q11 ×
Q12 STO T
Q13 ÷
Q14 +/-
Q15 0,5
Q16 +
```
Calcule $S \times \sqrt{2\pi}$.
Stocke temporairement le résultat.
Ajoute la moitié de l'aire droite de la courbe (nous avons utilisé la moyenne comme limite inférieure).

```
Q17 RTN
```
Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 033,5, 4B20

```
F01 LBL F
```
Cette sous-routine calcule l'intégrande pour la fonction normale $e^{-((X-M)\div S)^2\div 2}$.

```
F02 RCL D
F03 RCL- M
F04 RCL÷ S
F05 x²
F06 2
F07 ÷
F08 +/-
F09 e^x
F10 RTN
```
Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 015,0, 034D

Indicateurs utilisés : aucun.

Besoins en mémoire : 157 octets : 109 pour le programme, 48 pour les variables.

Remarques : La précision de ce programme dépend du réglage de l'affichage. Pour les saisies dans la plage de ±3 écarts-types, un affichage de quatre chiffres significatifs ou plus est acceptable pour la plupart des applications. Avec la pleine précision, la limite de saisie devient ±5 écarts-types. Le temps de calcul est réduit significativement si le nombre de chiffres affichés est moins élevé.

Dans le sous-programme N, la constante 0,5 peut être remplacée par 2 et [1/x]. Ceci épargnera 6,5 octets mais diminuera la précision.

Nul besoin de saisir le sous-programme inverse (dans les sous-programmes I et T) si vous n'avez pas besoin de la caractéristique d'inversion.

Instructions pour le programme :

1. Frappez les sous-programmes du programme ; appuyez sur [C] lorsque vous avez terminé.
2. Appuyez sur [XEQ] S.
3. Après la sollicitation de *M*, frappez la moyenne de la population et appuyez sur [R/S] (si la moyenne est zéro, appuyez seulement sur [R/S].)
4. Après la sollicitation de *S*, (écart-type), frappez l'écart-type de la population et appuyez sur [R/S] (si l'écart-type est 1, appuyez seulement sur [R/S].)
5. Pour calculer *X* connaissant *Q*(*X*), passez à l'étape 9.
6. Pour calculer *Q*(*X*) connaissant *X*, [XEQ] D.
7. Après la sollicitation, frappez la valeur de *X* et appuyez sur [R/S]. Le résultat, *Q*(*X*), est affiché.
8. Pour calculer *Q*(*X*) pour un nouvel *X* avec la même moyenne et le même écart-type, appuyez sur [R/S] et passez au point 7.
9. Pour calculer *X* connaissant *Q*(*X*), appuyez sur [XEQ] I.
10. Après la sollicitation, frappez la valeur de *Q*(*X*) et appuyez sur [R/S]. Le résultat, *X*, est affiché.
11. Pour calculer *X* pour une nouvelle valeur de *Q*(*X*) avec une moyenne et un écart-type identiques, appuyez sur [R/S] et passez au point 10.

Variables utilisées :

D *(Angl. Dummy)*, variable factice d'intégration.

M Moyenne de la population, valeur par défaut zéro.

Q Probabilité correspondant à l'aire droite de la courbe.

S (*Angl. Standard Deviation*) Ecart-type de la population, valeur par défaut 1.

T Variable temporaire pour passer la valeur $S\sqrt{2\pi}$ au programme d'inversion.

X Valeur de saisie : côté gche de l'aire droite de la courbe.

Exemple 1. Votre meilleur ami(e) vous informe qu'il va vous présenter une de ses relations. Il précise que son intelligence est de l'ordre de « 3σ ». Vous en déduisez que cette personne est plus intelligente que le reste de la population locale, sauf pour les personnes placées à plus de quatre écarts-types au-dessus de la moyenne. Vous supposez que la population locale compte environ 10 000 personnes susceptibles d'être ainsi rencontrées. Combien parmi elles se trouvent dans cette catégorie des « 3σ » ? Utilisez les valeurs par défaut de zéro pour *M* et 1 pour *S*.

Touches :	**Affichage :**	**Description :**
XEQ S	M?0,0000	Lance le ss-pr. d'initialisation.
R/S	S?1,0000	Accepte la valeur par défaut de zéro pour *M*.
R/S	1,0000	Accepte la valeur par défaut de 1 pour *S*.
XEQ D	X?*valeur*	Lance le progr. de distribution, sollicite *X*.
3 R/S	Q=0,0014	Saisit 3 pour *X* et lance le calcul de *Q*(*X*). Affiche le taux de la population plus intelligente que les personnes présentant trois écarts-types par rapport à la moyenne...

10000 [×]	13,5049	Multiplie par la population. Affiche le nombre approximatif de personnes susceptibles de vous être présentées dans la population du lieu qui satisfont à ce critère.

Mais on dit que l'on prie ses saints comme on les connaît...Votre ami exagère parfois, cela s'est vu, et vous décidez d'évaluer un « 2σ ». Le programme peut être ré-exécuté en appuyant sur [R/S].

Touches :	**Affichage :**	**Description :**
[R/S]	X?3,0000	Reprend le programme.
2 [R/S]	Q=0,0227	Saisit 2 comme valeur X et calcule $Q(X)$.
10000 [×]	227,4937	Multiplie par la population pour trouver la nouvelle estimation.

Exemple 2. La moyenne des résultats d'un test est de 55. L'écart-type est 15,3. En supposant que la courbe standard normale modèle fidèlement la distribution, quelle est la probabilité qu'un étudiant, sélectionné au hasard, obtienne un résultat de 90 ? Et quel est le résultat que seuls 10 % des étudiants peuvent espérer avoir dépassé ? Quel serait le résultat que seulement 20 % n'auraient pu atteindre ?

Touches :	Affichage :	Description :
XEQ S	M?0,0000	Lance le sous-programme d'initialisation.
55 R/S	S?1,0000	Stocke 55 pour la moyenne.
15,3 R/S	15,3000	Stocke 15,3 pour l'écart-type.
XEQ D	X?*valeur*	Lance le programme de distribution et sollicite X.
90 R/S	Q=0,0111	Saisit 90 pour X et calcule Q(X).

Seul 1 pour-cent des étudiants atteindrait un score supérieur à 90...

Touches :	Affichage :	Description :
XEQ I	Q?0,0111	Lance le sous-programme d'inversion.
,1 R/S	X=74,6078	Stocke 0,1 (10 pour-cent) dans *Q*(X) et calcule X.
R/S	Q?0,1000	Reprend le sous-programme d'inversion.
,8 R/S	X=42,1232	Stocke 0,8 (100 pour-cent moins 20 pour-cent) dans *Q*(X) et calcule X.

14

Programmes divers

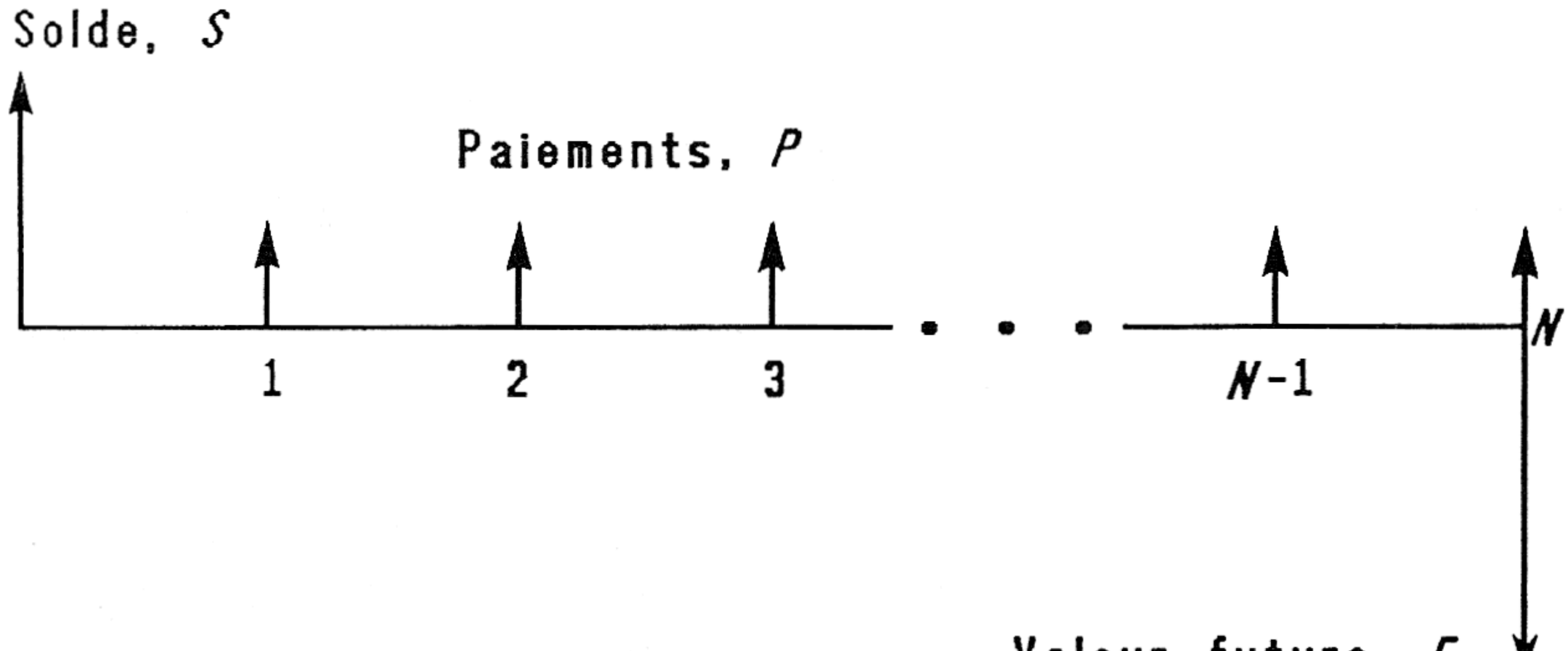
L'utilisation que fait chaque programme de la mémoire et la totalisation de contrôle peuvent être vérifiés dans le catalogue de programmes (page 85).

Calculs financiers sur des flux constants

Si quatre des cinq valeurs de l'équation de calcul sur des flux constants sont connus, ce programme trouve le cinquième. Il est utile en finance comme pour le calcul de prêts faits à de consomateurs ou aux acheteurs d'un immeuble.

Voici l'équation utilisée pour résoudre ces problèmes :

$$P\left[\frac{1-(1+Z)^{-N}}{Z}\right]+F(1+Z)^{-N}+S=0.$$



Un diagramme de calcul sur des flux constants

Les signes des valeurs des sommes d'argent (solde, *S* ; paiement, *P* et solde futur, *F*) correspondent à la *direction* des flux. L'argent reçu est de signe positif tandis que celui qui est versé est de signe négatif. Tout problème peut être vu de deux perspectives différentes ; prêteur et emprunteur voient le problème avec des signes différents.

Liste de programme :

Lignes de programme :	**Description :**
`N01 LBL N`	Ce sous-programme calcule le nombre de paiements, *N*.
`N02 14`	Saisit le nombre correspondant à N pour l'adressage indirect.
`N03 GTO L`	Branche vers le sous-programme de contrôle commun, L.
Octets et totalisation de contrôle : 004,5, 61E5	
`I01 LBL I`	Ce sous-programme calcule le taux d'intérêt, *I*.
`I02 9`	Saisit le nombre correspondant à I pour l'adressage indirect.
`I03 GTO L`	Branche vers le sous-programme de contrôle commun, L.
Octets et totalisation de contrôle : 004,5, DA04	
`B01 LBL S`	Ce sous-programme calcule le solde de départ, *S*.
`B02 2`	Saisit le nombre qui correspond à S pour l'adressage indirect.
`B03 GTO L`	Branche vers le sous-programme de contrôle commun, L.
Octets et totalisation de contrôle : 004,5, 98EB	
`P01 LBL P`	Ce sous-programme calcule le paiement périodique, *P*.
`P02 16`	Saisit le nombre qui correspond à P pour adressage indirect.
`P03 GTO L`	Branche vers le sous-programme de contrôle commun, L.
Octets et totalisation de contrôle : 004,5, A556	
`F01 LBL F`	Ce sous-programme calcule la valeur future.
`F02 6`	Saisit le nombre correspondant à F pour l'adressage indirect.
Octets et totalisation de contrôle : 003,0, 6779	

L01 LBL L	Ce label contrôle le calcul de la variable choisie.
L02 STO i	Stocke la valeur d'index (pour adressage indirect) dans *i*.
L03 FN= T	Choisit the sous-programme T, qui contient l'équation, pour SOLVE.
L04 SOLVE(i)	Trouve la variable indirectement adressée par *i*.
L05 VIEW(i)	Affiche the résultat addressé par *i*.
L06 GTO(i)	Revient pour un autre calcul.

Octets et totalisation de contrôle : 009,0, 7878

T01 LBL T	Ce sous-programme contient l'équation définissant la méthode de calcul sur des flux constants.
T02 INPUT N	Sollicite et stocke *N*.
T03 INPUT I	Sollicite et stocke *I*.
T04 INPUT S	Sollicite et stocke *S*.
T05 INPUT P	Sollicite et stocke *P*.
T06 INPUT F	Sollicite et stocke *F*.
T07 RCL I	Rappelle le taux d'intérêt en pour-cent.
T08 x=0?	Si $I=0\ldots$
T09 GTO K	...alors on utilise l'équation du sous-programme K.
T10 100	
T11 ÷	
T12 STO Z	Convertit *I* en forme décimale et le stocke dans Z.
T13 1	
T14 +	Calcule $(1 + Z)$.
T15 RCL N	
T16 +/-	
T17 y^x	Calcule $(1 + Z)^{-N}$.
T18 ENTER	Reproduit cette quantité pour qu'elle puisse être utilisée plus tard.
T19 +/-	
T20 1	
T21 +	Calcule $1 - (1 + Z)^{-N}$.
T22 RCL÷ Z	Calcule $(1 - (1 + Z)^{-N}) \div Z$.

T23 RCL× P	Calcule $P\times(1-(1+Z)^{-N})\div Z$.
T24 x<>y	Echange la copie de $(1+Z)^{-N}$ (en ligne T18) dans le registre X.
T25 RCL× F	Calcule $F\times(1+Z)^{-N}$.
T26 +	Calcule $F\times(1+Z)^{-N} + P\times(1-(1+Z)^{-N})\div Z$.
T27 RCL+ B	Calcule $F\times(1+Z)^{-N} + P\times(1-(1+Z)^{-N})\div Z+B$.
T28 RTN	Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 050,0, 429C

K01 LBL K	Ce sous-programme est invoqué si $I = 0$.
K02 RCL P	
K03 RCL× N	
K04 RCL+ F	
K05 RCL+ B	Calcule $P \times N + F + B$.
K06 RTN	Revient au sous-programme d'appel.

Octets et totalisation de contrôle : 009,0, F2E0

Indicateurs utilisés : aucun.

Besoins en mémoire : 145 octets: 89 pour le programme, 56 pour les variables.

Remarques : Puisque tous les calculs sont faits dans les sous-programmes T et K, il est possible de raccourcir le programme en éliminant les autres sous-programmes d'interface avec l'utilisateur. Pour exécuter le programme en sa forme raccourcie, choisissez la fonction définie par LBL T (■ SOLVE/∫ {FN=} T) puis recherchez la valeur de la variable dont vous avez besoin (■ SOLVE/∫ {SOLVE} *variable*).

Instructions de programmation :

1. Frappez les sous-programmes du programme ; appuyez sur C lorsque vous avez fini.
2. Choisissez le sous-programme approprié :
   - XEQ N pour calculer le nombre de périodes de composition ;
   - XEQ I pour calculer l'intérêt périodique ;
   - XEQ B pour calculer le solde initial pour un prêt ou un compte d'épargne ;

- [XEQ] P pour calculer le paiement périodique ;
- [XEQ] F pour calculer la valeur future, ou solde, d'un prêt.

3. Frappez la valeur des quatre autre variables au fur et à mesure des sollicitations et appuyez sur [R/S] après chaque valeur.

4. Après le dernier [R/S], le résultat est affiché.

5. Pour recalculer la même variable avec des données différentes, appuyez sur [R/S] et passez au point 3.

6. Pour un cas entièrement nouveau, passez au point 2.

Variables utilisées :

N Le nombre de périodes de composition.

I L'intérêt *périodique* comme taux de pourcentage (par exemple si le taux *annuel* est 15 % mais qu'il y a 12 paiements par an, le taux d'intérêt *périodique* est $15 \div 12 = 1{,}25$ %.)

Z Le taux d'intérêt *périodique* sous forme décimale.

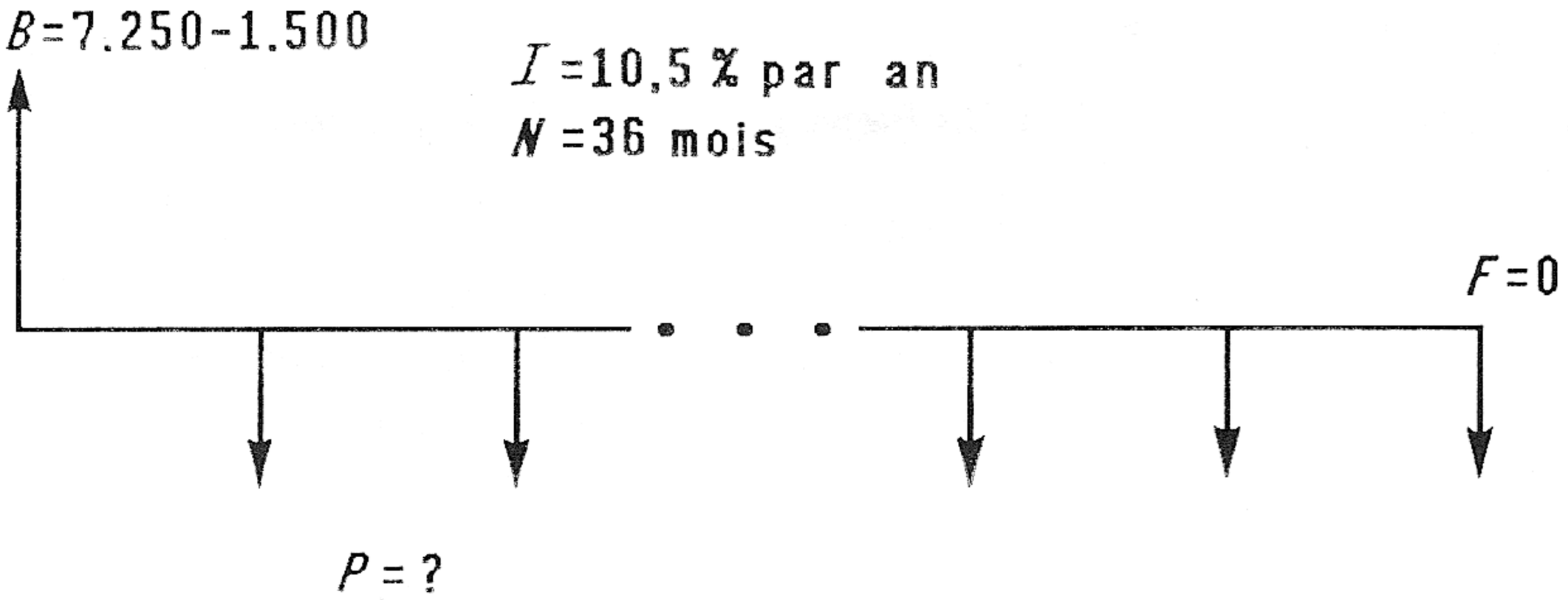
B Le solde initial du prêt ou du compte d'épargne.

P Le paiement périodique.

F La valeur future d'un plan d'épargne ou le solde d'un prêt.

i La variable d'index, utilisée ici pour l'adressage indirect.

Exemple : 1e partie. Vous financez l'achat d'une voiture pour votre fils, étudiant aux Etats-Unis, avec un prêt sur 3 ans à 10,5 % d'intérêt annuel, composé mensuellement. Le prix d'achat est 7 250 dollars U.S. Votre acompte est de 1 500 dollars. Quels sont vos paiements mensuels ?



Touches :	Affichage :	Description :
■ DISP {FX}2		Définit le format d'affichage FIX 2.
XEQ P	N?*valeur*	Choisit le sous-programme P, qui calcule le paiement périodique.
36 R/S	I?*valeur*	Stocke 36 dans *N*.
10,5 ENTER 12 ÷	I?0,88	Convertit le taux d'intérêt annuel en un taux mensuel.
R/S	B?*valeur*	Stocke le taux d'intérêt mensuel dans *I*.
7250 ENTER 1500 −	B?5.750,00	Calcule le solde de départ du prêt.
R/S	F?*valeur*	Définit *B* comme étant égal au solde de départ.
0 R/S	P=-186,89	Stocke zéro dans *F*, le solde futur ou définitf, et calcule le paiement du prêt.

La réponse est négative—le prêt a été envisagé du point de vue de l'emprunteur. L'argent reçu par l'emprunteur (le solde de départ) est positif, l'argent versé est négatif.

2e partie. Quel taux d'intérêt réduirait le paiement mensuel de 10 dollars ?

Touches :	Affichage :	Description :
XEQ I	N?36,00	Choisit le sous-programme I, qui calcule le taux d'intérêt périodique.
R/S	B?5.750,00	Accepte 36 comme nombre de paiements.
R/S	P?-186,89	Accepte 5 750 dollars comme solde de départ.
ENTER	P?-186,89	Copie le paiement sur la pile pour que vous puissiez l'inclure dans le calcul (le registre X sera écrasé lorsque le nombre suivant qui sera saisi).
10 +	P?-176,89	Reduit le paiement mensuel de 10 dollars.
R/S	F?0,00	Stocke la nouvelle valeur du paiement.
R/S	I=0,56	Accepte zéro en tant que solde futur et calcule *I*, le taux d'intérêt mensuel.
12 ×	6,75	Calcule le taux d'intérêt annuel.

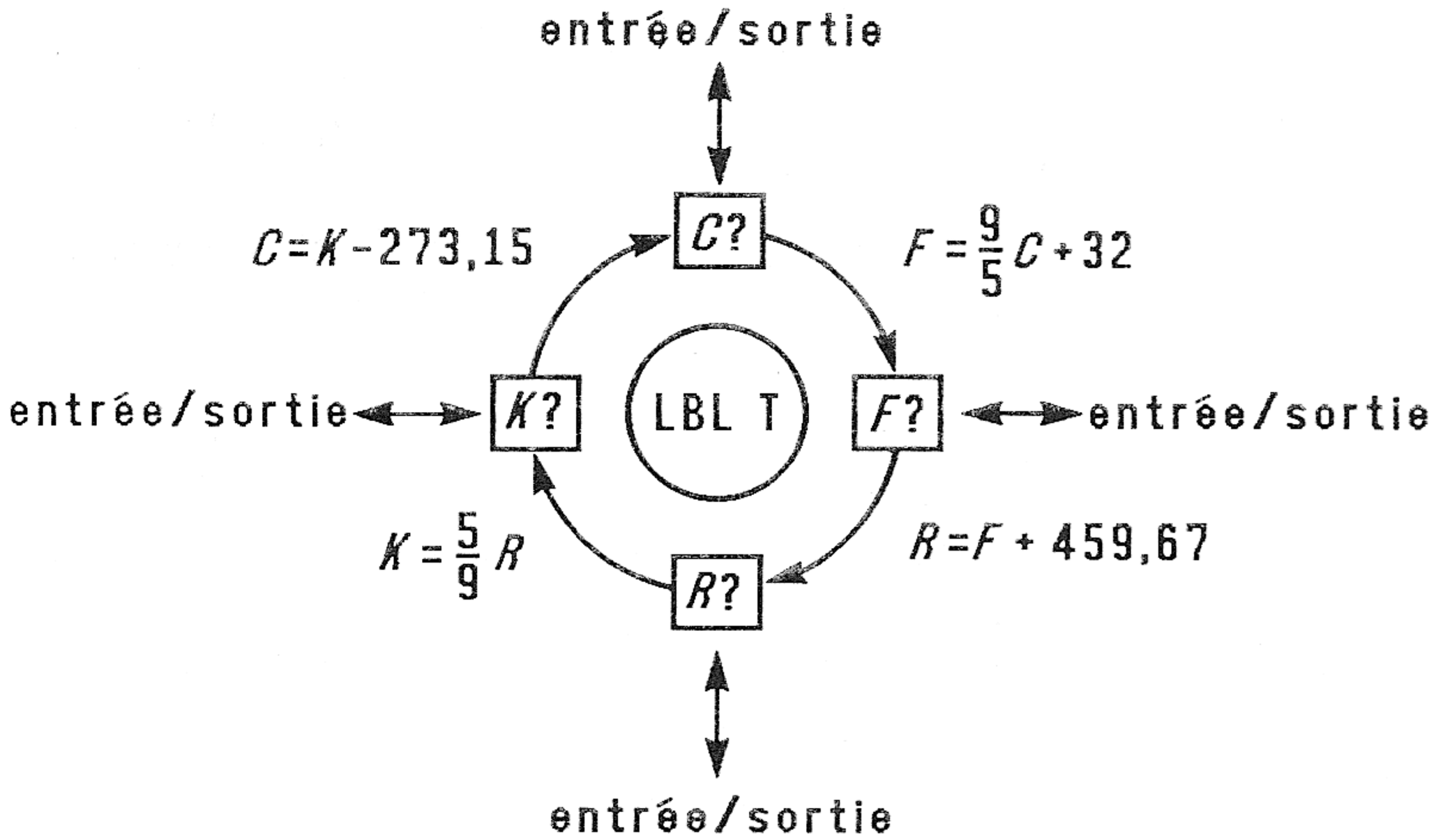
3e partie. En utilisant le taux d'intérêt de 6,75 %, supposez que vous vendiez cette voiture après deux ans—votre fils rentre en France pour se marier. De quelle somme êtes-vous redevable ? Autrement dit, quel sera le solde du prêt dans deux ans ?

Touches :	Affichage :	Description :
XEQ F	N?36,00	Choisit sous-programme F, qui calcule la valeur future.
24 R/S	I?0,56	Change le nombre de paiements en 24 mois.
R/S	B?5.750,00	Accepte le taux d'intérêt mensuel.
R/S	P?-176,89	Accepte 5 750 dollars comme solde de départ.
R/S	F=-2047,04	Accepte la valeur de paiement et calcule le solde futur. Ici aussi le signe est négatif, indiquant que vous devez payer cet argent.

Conversions d'unités

Ce programme consiste en deux sous-programmes qui convertissent un type d'unité en un autre. L'un d'eux établit des conversions entre unités Celsius, Fahrenheit, Rankine et Kelvin. L'autre sous-programme convertit inches, feet et mètres ainsi que pouces carrés (*square inches*), pieds carrés (*square feet*) et mètres carrés. Ces programmes peuvent être modifiés aisément pour convertir d'autres types d'unités de votre choix.

Le programme principal a une structure circulaire. Lorsqu'il commence, il passe par un cycle de sollicitation de saisies. Il suffit d'ignorer les sollicitations dont vous n'avez pas besoin (en appuyant sur [R/S]) jusqu'à ce que les sollicitations correspondant aux unités qui vous intéressent apparaissent. Si vous désirez saisir des degrés Kelvin, vous lancez le programme de température en appuyant sur [XEQ] T puis sur [R/S] jusqu'à ce que la sollicitation K? apparaisse. Vous frappez alors la température en kelvins et appuyez sur [R/S] jusqu'à ce que vous rencontriez la sollicitation corespondant à l'unité désirée (C? pour Celsius, par exemple). La valeur affichée avec la sollicitation serait alors la température en degrés Celsius. Pour quitter le programme, appuyez sur [C].



Structure circulaire pour conversions de températures

Ce programme a été conçu pour minimiser l'utilisation de la pile. Lorsque le programme est terminé, les valeurs présentes dans les registres X et Y sont laissées, respectivement, dans les registres Y et Z, et la valeur convertie est affichée. Si la valeur que vous désirez convertir est présente dans le registre X (l'affichage) lortsque vous lancez le programme, appuyez sur [R↓] pour la récupérer lorsque vous obtiendrez le message de sollicitation correct.

Les programmes de conversion de longueurs et de surface sdont un bon exemple de l'utilisation des indicateurs. Notez que l'indicateur 2 a été choisi, le témoin **2** indique que l'unité a été mise au *carré*.

Liste de programme :

Lignes de programme :	Description :
T01 LBL T	Lance le sous-programme de conversion de températures.
T02 INPUT C	Affiche le température en ° C, ou sollicite et stocke la saisie en Celsius.
T03 9	
T04 ×	
T05 5	
T06 ÷	
T07 32	
T08 +	Convertit de Celsius en Fahrenheit.
T09 STO F	Stocke la température en Fahrenheit dans *F*.
T10 R↓	Descend la pile de sorte qu'un seul niveau est utilisé.
T11 INPUT F	Affiche la température en ° F, ou sollicite et stocke la saisie en Fahrenheit.
T12 459,67	
T13 +	Convertit de Fahrenheit en Rankine.
T14 STO R	Stocke la température en Rankine dans *R*.
T15 R↓	Descend la pile de sorte qu'un seul niveau est utilisé.
T16 INPUT R	Affiche la température en ° R, ou sollicite et stocke la saisie Rankine.
T17 9	
T18 ÷	
T19 5	
T20 ×	Convertit Rankine en Kelvin.
T21 STO K	Stocke la température en Kelvin dans *K*.
T22 R↓	Descend la pile de sorte qu'un seul niveau soit utilisé.
T23 INPUT K	Affiche la température en kelvins, ou sollicite et stocke la saisie Kelvin.
T24 273,15	
T25 −	Convertit Kelvin en Celsius.
T26 STO C	Stocke la température en Celsius dans C.
T27 R↓	Desecend la pile de sorte qu'un seul niveau est utilisé.
T28 GTO T	Revient en arrière pour d'autres conversions.

Octets et totalisation de contrôle : 058,0, 9EF4

A01 LBL S	Lance le sous-programme de conversion de surfaces.
A02 SF 2	Définit l'indicateur 2 pour indiquer la conversion d'unités de surface.
A03 GTO Q	Branche vers le sous-programme de conversion.

Octets et totalisation de contrôle : 004,5, 2A52

L01 LBL L	Lance le programme de conversion de longueurs.
L02 CF 2	Efface l'indicateur de conversion de surfaces.

Octets et totalisation de contrôle : 003,0, 2C3C

Q01 LBL Q	Lance le sous-programme combinant conversions de longueurs et de surfaces.
Q02 INPUT I	Affiche des pouces ou accepte les saisies.
Q03 12	Saisit le facteur de conversion pouces/pieds.
Q04 FS? 2	Teste si ceci est une conversion de surfaces.
Q05 x^2	Si oui, met le facteur au carré.
Q06 ÷	Calcule le résultat.
Q07 STO F	Stocke les pieds ou pieds carrés.
Q08 R↓	Descend la pile.
Q09 INPUT F	Affiche les pieds ou pieds carrés, ou accepte des saisies.
Q10 0,3048	Saisit le facteur de conversion des pieds en mètres.
Q11 FS? 2	Teste pour savoir si ceci est une conversion de surfaces.
Q12 x^2	Si oui, met le facteur au carré.
Q13 ×	Calcule le résultat.
Q14 STO M	Stocke les mètres ou les mètres carrés.
Q15 R↓	Descend la pile.
Q16 INPUT M	Affiche les mètres ou les mètres carrés, ou accepte des saisies.
Q17 0,0254	Saisit le facteur de conversion de mètres en pouces.
Q18 FS? 2	Teste pour savoir si ceci est une conversion de surfaces.
Q19 x^2	Si oui, met le facteur au carré.
Q20 ÷	Calcule le résultat.
Q21 STO I	Stocke les pouces ou les pouces carrés.
Q22 R↓	Descend la pile.
Q23 GTO Q	Boucle en arrière pour d'autres conversions.

Octets et totalisation de contrôle : 050,5, 2A15

Indicateurs utilisés : Indicateur 2.

Besoins en mémoire : 164 octets : 116 p. le pr. 48 pr les var.

Instructions de programme :

1. Frappez les sous-programmes du programme ; appuyez sur [C] lorsque vous aurez terminé.
2. Appuyez sur [XEQ] et sur le label approprié.
   - *S* pour une conversion de surfaces, ou
   - *L* pour une conversion de longueurs, ou
   - *T* pour une conversion de températures.
3. Appuyez sur [R/S] jusqu'à ce que la sollicitation de saisie appropriée apparaisse.
4. Frappez votre saise (ou appuyez sur [R↓] pour récupérer la saisie si elle était à l'affichage).
5. Appuyez sur [R/S] jusqu'à ce que la sollicitation (avec un résultat) corresponde aux unités que vous cherchez.
6. Passez au point 3 pour une autre conversion.
7. Appuyez sur [C] pour effacer la sollicitation et sortir.

Variables utilisées :

C Température en degrés Celsius.

F Température en degrés Fahrenheit, ou pieds (*feet*).

R Température en degrés Rankine.

K Température en kelvins.

I Inches (pouces).

M Mètres.

Exemple 1. Convertit 212 °F en kelvins.

Touches :	**Affichage :**	**Description :**
[XEQ] T	C?*valeur*	Conversion de température.
[R/S]	F?*valeur*	Sollicite les degrés Fahrenheit.

212 [R/S]	R?671,6700	Saisit la température en Fahrenheit et la convertit en degrés Rankine.
[R/S]	K?373,1500	Convertit en kelvins.

Exemple 2. Un plancher mesure 108 pouces × 127 pouces. Quelle est sa surface en pieds carrés ?

Touches :	**Affichage :**	**Description :**
108 [ENTER]		
127 [×]	K?13.716,0000	Calcule la surface en pouces carrés.
[XEQ] A	I?*valeur*	Choisit le sous-programme de onversion de surfaces.
[R↓]	I?13.716,0000	Redescend du registre Y la valeur qui a été précédemment calculée pour les pouces carrés (in^2).
[R/S]	F?95,2500	Calcule les pieds carrés.
[C]	95,2500	Annule la sollicitation et met fin au programe.

Exemple 3. Supposons que le résultat d'un long calcul soit 3,787, et que ce résultat se trouve à l'affichage de votre calculateur. Supposez aussi que cette valeur doive être divisée par 25,73 pieds. Calculez la réponse finale.

Touches :	Affichage :	Description :
3,787	3,787	Saisit le résultat hypothétique.
XEQ L	I?*valeur*	Choisit le sous-programme de conversion de longueurs.
R/S	F?*valeur*	Amène jusqu'à la sollicitation de pieds.
25,73 R/S	M?7,8425	Saisit le diviseur en pieds et le convertit en mètres.
C	7,8425	Annule la sollicitation et termine le programme.
÷	0,4829	Calcule le résultat final.

Générateur de nombres premiers

Ce programme accepte n'importe quel nombre entier impair plus grand que 3. Si le nombre est un nombre premier (il n'est divisible que par lui-même et par 1), le programme renvoie la valeur saisie. Si la saisie n'est pas un nombre premier, le programme renvoie le nombre premier le plus grand suivant celui qui a été saisi.

Le programe identifie les nombre qui ne sont pas premiers en essayant tous les facteurs possibles. Si un nombre n'est pas premier, le programme y ajoute 2 (en supposant que la valeur soit toujours impaire) et teste pour voir s'il a trouvé un nombre premier. Ce processus continue jusqu'à ce qu'un nombre premier soit trouvé.

Note : x est la valeur dans le registre x

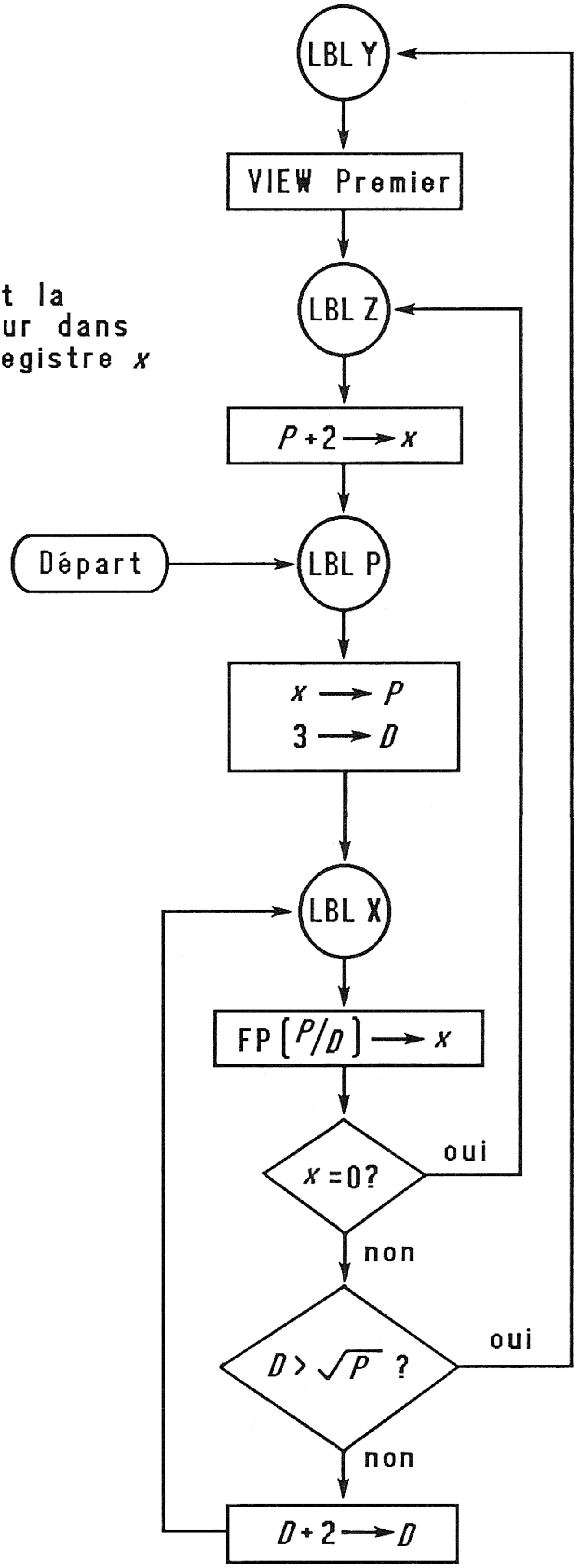


Schéma du programme des nombres premiers

Liste de programme :

Lignes de programme :	Description :
Y01 LBL Y	Ce sous-programme affiche le nombre premier.
Y02 VIEW P	Affiche le nombre premier *P*.
Octets et totalisation de contrôle : 003,0, 9D08	
Z01 LBL Z	Ce sous-programme ajoute 2 à *P* avant de tester pour voir si *P* est premier.
Z02 2	
Z03 RCL+ P	
Octets et totalisation de contrôle : 004,5, E455	
P01 LBL P	Ce sous-programme stocke la valeur de saisie pour *P*.
P02 STO P	
P03 3	Stocke 3 dans le diviseur-test, *D*.
P04 STO D	
Octets et totalisation de contrôle : 006,0, 9E38	
X01 LBL X	Ce sous-programme teste *P* pour voir s'il est premier.
X02 RCL P	
X03 RCL÷ D	
X04 FP	Trouve la partie fractionnaire de $P \div D$.
X05 x=0?	Teste pour savoir si le reste est zéro (nombre non premier).
X06 GTO Z	Si le nombre n'est pas premier, essaie la possibilité suivante.
X07 RCL P	
X08 SQRT	
X09 RCL D	
X10 x>y?	Teste pour voir si tous les facteurs ont été essayés.
X11 GTO Y	S'ils ont tous été essayés, branche vers le sous-programme d'affichage.
X12 2	Calcule le facteur possible suivant, $D + 2$.
X13 STO+ D	

X14 GTO X	Branche pour tester le nombre premier potentiel avec le nouveau facteur.

Octets et totalisation de contrôle : 021,0, 43F2

Indicateurs utilisés : aucun.

Besoins en mémoire : 50,5 octets : 34,5 pour le programme, 16 pour les variables.

Instructions de programmation :

1. Frappez les sous-programmes du programme ; appuyez sur [C] lorsque vous avez terminé.
2. Frappez un entier impair.
3. Appuyez sur [XEQ] *P* pour lancer le programme. Le nombre premier, *P*, sera affiché.
4. Pour voir le nombre premier suivant, appuyez sur [R/S].

Variables utilisées :

P Valeur du nombre premier et valeurs potentielles de nombre premier.

D Diviseur utilisé pour tester la valeur de *P* à ce moment.

Remarques : aucun test n'est fait pour s'assurer que le nombre saisi est un nombre inpair, entier et positif plus grand que 3.

Exemple. Quel est le nombre premier suivant 789 ? Et quel est le nombre premier suivant celui-là ?

Touches :	Affichage :	Description :
789 [XEQ] P	P=797,0000	Saisit 789 et lance le programme ; affiche le nombre premier suivant immédiatement 789.
[R/S]	P=809,0000	Calcule le nombre premier suivant 797.

Cinquième partie

Annexes et matériel de référence

Page 240 A : Assistance, piles et service après-vente

253 B : Mémoire utilisateur et pile des résultats

259 C : Résolution des équations : notions approfondies

273 D : Intégrales : notions approfondies

281 Liste des messages

286 Index des fonctions

299 Index

A

Assistance, piles et service après-vente

Demande d'aide pendant l'utilisation du calculateur

Hewlett-Packard s'est engagé à offrir un support régulier aux utilisateurs des calculateurs HP. Vous pouvez obtenir des réponses à vos questions concernant l'utilisation du calculateur en contactant le département de support technique (adresse à l'intérieur du dos de couverture).

Nous vous recommandons de lire les pages suivantes avant de contacter Hewlett-Packard. Notre expérience nous indique que nos clients ont souvent les mêmes questions.

Réponses aux questions fréquemment posées

Q. Comment puis-je déterminer si le calculateur fonctionne bien ?

A. Lisez la decription du test automatique, en page 246.

Q. Comment puis-je changer le nombre de décimales affichées à l'écran ?

A. Utilisez la fonction ■ DISP (page 30).

Q. Mes nombres utilisent le point comme séparateur décimal au lieu de la virgule. Comment puis-je rétablir la virgule ?

A. Utilisez la fonction ■ MODES (page 29).

Q. Coment puis-je effacer tout ou partie de la mémoire ?

A. ■ [CLEAR] affiche le menu CLEAR qui permet d'effacer toutes les variables, tous les programmes (en mode de saisie de programme uniquement), les registres statistiques ou la totalité de la mémoire disponible à l'utilisateur (sauf durant la saisie de programme).

Q. Que signifie le « E » dans un nombre (2,51E−13) ?

A. *Exposant* de dix ; c'est-à-dire, $2{,}51 \times 10^{-13}$.

Q. Le calculateur affiche le message `MEMORY FULL`. Que faire ?

A. Il est nécessaire d'effacer une partie de la mémoire avant de continuer (voyez l'annexe B).

Q. Pourquoi le calcul du sinus ou de la tangente de π radians donne-t-il pour résultat un petit nombre et non zéro ?

A. π ne peut être représenté *exactement* vu la limitation de la précision du calculateur à 12 chiffres.

Q. Pourquoi obtient-on des résultats erronés lors de l'utilisation des fonctions trigonométriques ?

A. Assurez-vous que le calculateur utilise le mode angulaire correct (■ [MODES]).

Q. Que signifie le petit symbole qui est apparu sur l'écran ?

A. Nous l'appelons un *témoin,* il indique l'état dans lequel se trouve le calculateur. Lisez « Témoins », chapitre 1, page 20.

Alimentation et piles

Le HP-32S est livré avec trois piles alcalines. Un jeu de piles neuves assure entre 7 et 24 mois d'utilisation (les piles au mercure ou à l'oxide d'argent durent environ deux fois plus longtemps). Souvenez-vous néanmoins que la durée de vie des piles dépend de l'utilisation du calculateur. Les calculs longs (tels que [IRR%]) utilisent le plus de puissance.

N'utilisez que des piles bouton neuves. N'utilisez pas de piles rechargeables. Nous recommandons l'utilisation des piles suivantes. Ces piles ne sont pas disponibles dans tous les pays.

Alcaline	**Mercure**	**Oxide d'argent**
Panasonic LR44	Panasonic NP675	Panasonic SR44 ou SP357
Eveready A76	Eveready EP675E	Eveready 357
Varta V13GA	Duracell MP675H	Ray-O-Vac 357
Duracell LR44	Toshiba NR44 ou MR44	Varta V357
	Radio Shack NR44 ou MR44	

Témoin de faible charge

Lorsque le témoin de faible charge (▭) s'allume, le calculateur peut continuer à fonctionner pendant quelques heures. Si vous éteignez le calculateur, le contenu de la mémoire permanente sera conservé pendant plusieurs semaines.

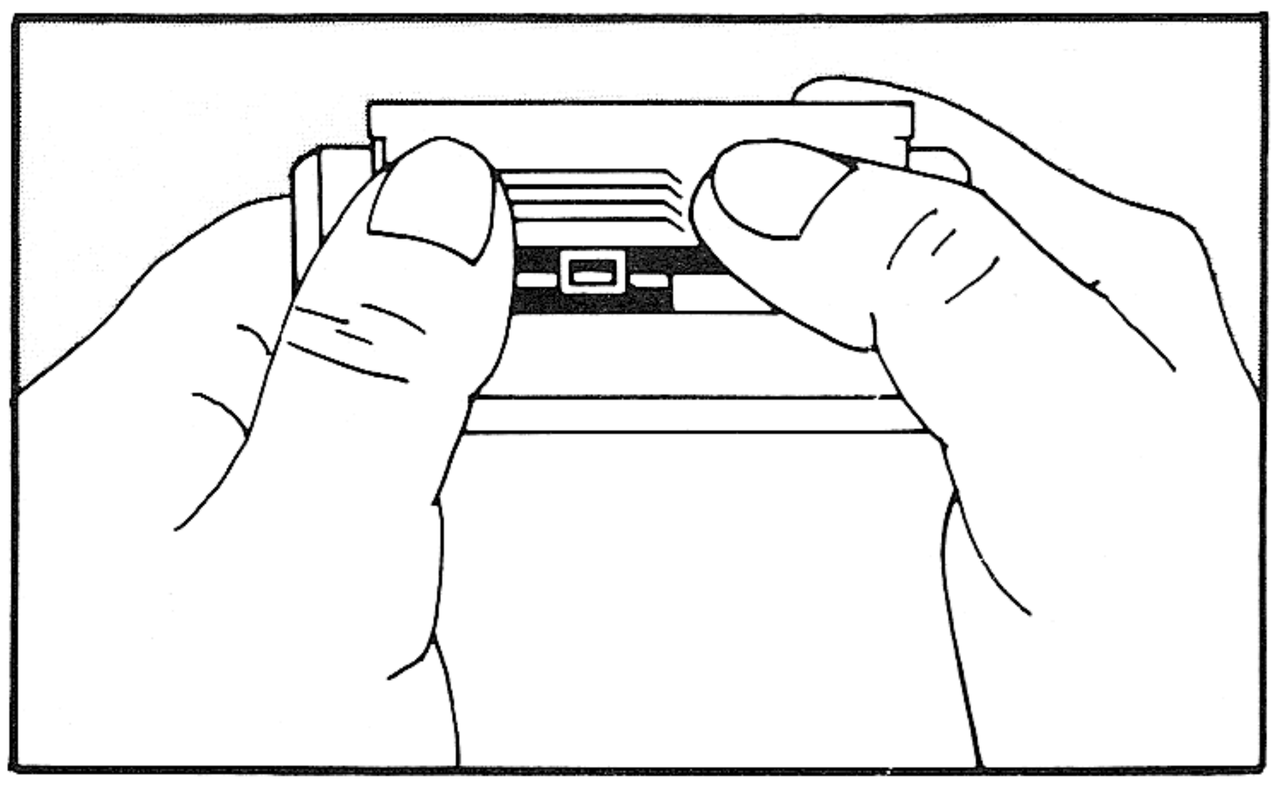
Si vous continuez à utiliser le calculateur après l'apparition du témoin de faible charge, la charge peut baisser au point où l'intensité de l'affichage diminue. Si vous continuez à utiliser le calculateur, les données stockées peuvent être affectées. Dans ce cas, le calculateur nécessite de nouvelles piles avant de fonctionner correctement à nouveau.

Si vous continuez à utiliser le calculateur après l'apparition du témoin de faible charge, la charge peut baisser au point où l'intensité de l'affichage diminue. Si vous continuez à utiliser le calculateur, les données stockées peuvent être affectées. Dans ce cas, le calculateur nécessite de nouvelles piles avant de fonctionner correctement à nouveau. Si les données stockées ont été perdues à cause d'une baisse de charge, le HP-32S affiche `MEMORY CLEAR`.

Installation des piles

Une fois le compartiment des piles ouvert, vous disposez d'une minute pour remplacer les piles sans perdre le contenu de la mémoire permanente. Par conséquent, nous vous recommandons d'avoir un nouveau jeu de piles prêt à mettre en place avant d'ouvrir le compartiment des piles. Veillez en outre à ce que le calculateur soit éteint avant de changer les piles.

1. Placez trois piles bouton à portée de main.
2. Vérifiez que le calculateur est éteint. **N'appuyez pas sur [C] avant d'avoir terminé toute la procédure de remplacement des piles. Le remplacement des piles lorsque le calculateur est allumé peut provoquer l'effacement de la mémoire permanente.**
3. Tenez le calculateur comme indiqué. Pour retirer le volet du compartiment des piles, appuyez sur la partie rainurée jusqu'à ce que le volet glisse.



4. Retournez le calculateur et faites tomber les piles.

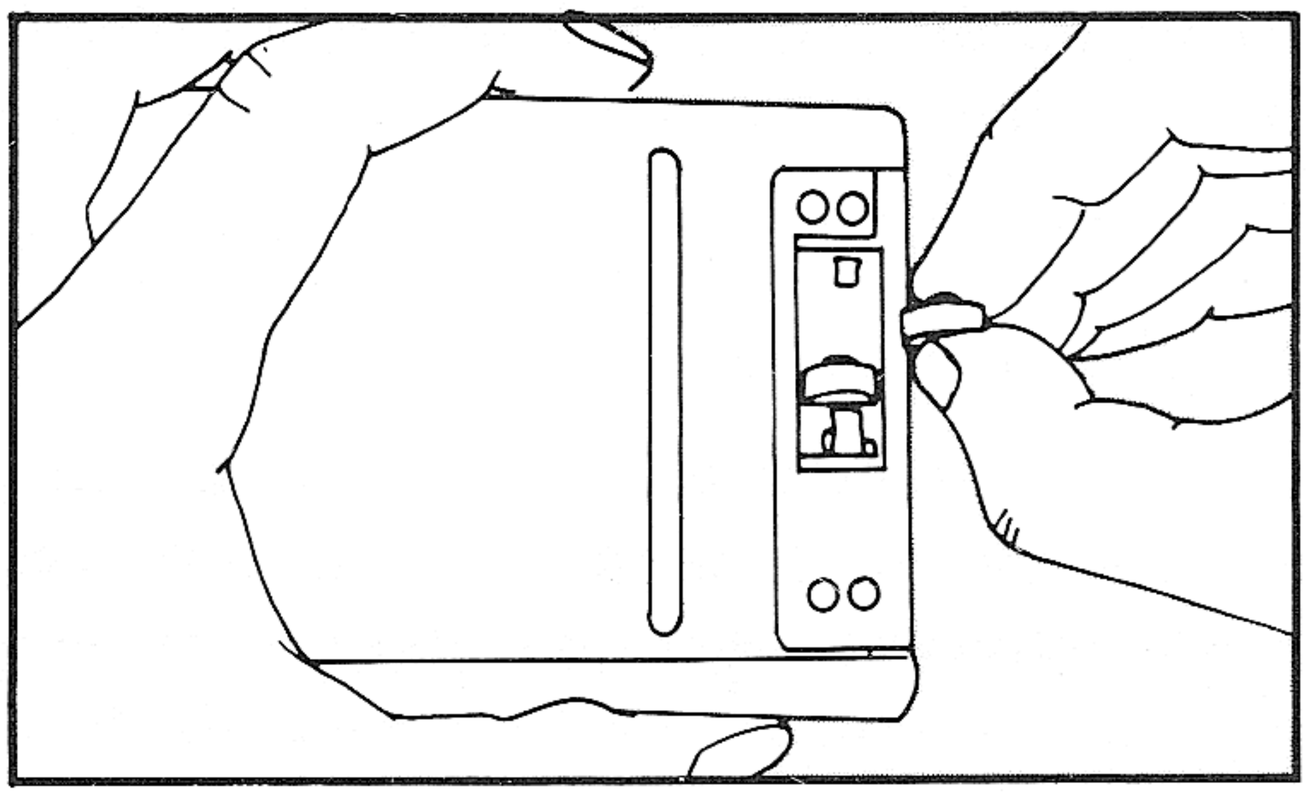


Danger

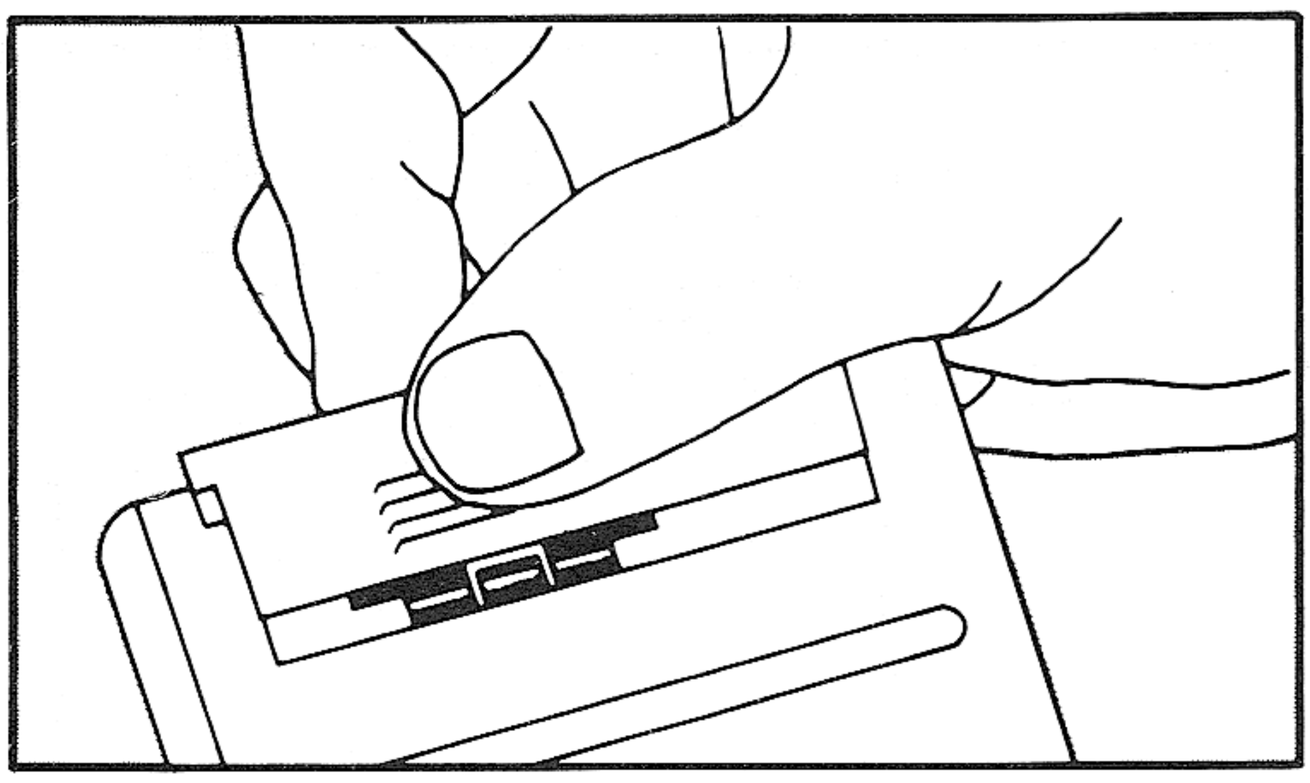
N'endommagez pas et ne brûlez pas les piles. Elles peuvent exploser en émettant des produits chimiques dangereux.

---

5. Tenez le calculateur comme indiqué et insérez trois piles dans le compartiment. Orientez les piles comme indiqué sur le diagramme imprimé dans le compartiment. Veillez à ce que leurs positions correspondent bien à celles du diagramme.



6. Insérez l'onglet du volet du compartiment dans la fente du boîtier, comme illustré ci-dessous :



Remettez le calculateur sous tension. S'il ne fonctionne pas, vous avez peut-être pris trop de temps pour remplacer les piles ou vous avez peut-être allumé le calculateur alors que les piles étaient retirées. *Enlevez les nouvelles piles* et appuyez légèrement *pendant quelques secondes* sur les contacts dans le calculateur avec une pièce de monnaie. Remettez les piles dans le calculateur et rallumez ce dernier ; il devrait afficher `MEMORY CLEAR`.

Environnement

Pour maintenir le bon fonctionnement du calculateur, les limites de température et d'humidité suivantes doivent être respectées :

- Température de fonctionnement : 0 ° à 45 °C.
- Température de stockage : − 20 ° à 65 °C.
- Humidité de fonctionnement et de stockage 90 % d'humidité relative à 40 °C maximum.

Votre calculateur doit-il être envoyé au service après-vente ?

Suivez la procédure ci-dessous pour déterminer si votre calculateur doit être réparé. Si c'est le cas, lisez la section « Informations » en page 161.

- **Si le calculateur ne peut pas être allumé (rien n'est affiché) :**

  1. Essayez de réinitialiser le calculateur (voir page 157).

  2. Si le calculateur ne répond pas à l'étape 1, remplacez les piles.

  Si la situation ne change pas, le calculateur doit être réparé.

■ **Si le calculateur ne répond aux frappes sur le clavier (rien ne se passe lorsque vous appuyez sur des touches) :**

1. Essayez de réinitialiser le calculateur (maintenez la touche [C] et appuyez sur [LN]).
2. Si le calculateur ne répond pas à l'étape 1, essayez d'effacer la mémoire (la procédure est décrite en page 255). Ceci effacera toutes les informations stockées.
3. Si la situation ne change pas, retirez les piles (voir page 243) et appuyez légèrement avec une pièce de monnaie sur les contacts dans le compartiment des piles. Remettez les piles dans le compartiment et allumez le calculateur. Il devrait afficher `MEMORY CLEAR`.

Si la situation ne change pas, le calculateur doit être réparé.

■ **Si le calculateur répond aux séquences de touches mais si vous soupçonnez un mauvais fonctionnement :**

1. Effectuez l'auto-test (voir ci-dessous). Si le calculateur ne passe pas le test, il doit être réparé.
2. Si le calculateur passe le test, vous avez sans doute fait une erreur de manipulation. Relisez la partie du manuel concernant le type ce calcul effectué et consultez la section « Réponses à quelques questions fréquemment posées » en page 240.
3. Si la réponse à votre problème ne se trouve pas dans le manuel, consultez votre distributeur ou le bureau commercial HP le plus proche (adresses à l'intérieur du dos de couverture).

Test automatique

Lorsque l'affichage s'allume mais que le calculateur ne semble pas fonctionner correctement, exécutez le test automatique.

Pour exécuter le test automatique :

1. Pour lancer le test, maintenez la touche [C] enfoncée et appuyez sur la touche [y^x]*.

* Maintenez [C] tout en appuyant sur [1/x] pour lancer un test automatique continu utilisé en usine. Si vous lancez ce test accidentellement, vous pouvez y mettre fin en appuyant sur n'importe quelle touche.

2. Appuyez huit fois sur une touche quelconque et regardez les divers motifs affichés. Après la huitième pression de touche, le calculateur doit afficher le message `COPR. HP 1987` suivi de `KBD 01`.

3. En commençant par l'angle suppérieur gauche du clavier ([√x]), appuyez dans l'ordre sur les touches du rang supérieur. Appuyez ensuite de gauche à droite sur les touches du deuxième rang, puis du troisième, etc. jusqu'à ce que vous ayez appuyé sur toutes les touches.

- Si vous avez appuyé sur les touches dans l'ordre correct et si elles fonctionnent correctement, le calculateur affiche `KBD` suivi d'un nombre à deux chiffres (le calculateur compte les touches en hexadécimal).
- Si vous n'appuyez pas sur les touches dans l'ordre indiqué ou si une touche ne fonctionne pas correctement, la séquence suivante n'affiche pas de message (voir étape 4).

4. L'auto-test donne l'un des deux résultats ci-dessous :

- Le calculateur affiche `32S -OK`, si l'auto-test a réussi. Allez à l'étape 5.
- Le calculateur affiche `32S -FAIL`, suivi d'un chiffre, si l'auto-test n'a pas réussi. Si vous obtenez ce message parce que vous n'avez pas appuyé sur les touches dans l'ordre indiqué, vous devez réinitialiser le calculateur (maintenez [C] enfoncée et appuyez sur [LN]) puis ré-exécutez l'auto-test. Si vous avez appuyé sur les touches dans le bon ordre, et si vous obtenez le message ci-dessus, répétez l'auto-test pour confirmer le résultat. Si vous obtenez le même résultat, le calculateur doit être réparé (voir page 249). Indiquez le message affiché par le calculateur lorsque vous l'envoyez pour réparation.

5. Pour sortir de l'auto-test, réinitialisez le calculateur (maintenez la touche [C] enfoncée et appuyez sur [LN]).

Garantie

Le HP-14B (sauf pour ce qui concerne les piles ou d'éventuels dommages causés par les piles) est garanti par Hewlett-Packard contre tout vice de matière et de fabrication pour une durée d'un an à partir de la date de livraison, la facture d'achat faisant foi. Hewlett-Packard s'engage à réparer ou, éventuellement, à remplacer les pièces qui se révèleraient défectueuses pendant la période de garantie. Cette garantie couvre les pièces et la main d'œuvre*. Elle disparaît en cas d'utilisation en dehors des spécifications ou de modification ou maintenance par un centre non reconnu par Hewlett-Packard.

Seuls les essais effectués à partir des programmes Hewlett-Packard seront considérés comme faisant foi lors des litiges concernant le fonctionnement du matériel. Aucune autre garantie explicite ou implicite n'est accordée. La responsabilité de Hewlett-Packard ne peut être engagée dans le cas d'une application particulière. La société ne peut être tenue responsable des dommages indirects.

Modifications

Les appareils vous sont livrés selon les spécifications en vigueur au moment de la fabrication. Hewlett-Packard n'est pas tenu de modifier les appareils déjà vendus.

Coût de la maintenance

Les réparations sont effectuées pour un prix forfaitaire incluant pièces et main d'œuvre. Ce forfait est sujet à la T.V.A. en France ou à des taxes similaires dans d'autres pays. Ces taxes apparaissent en détail sur les factures.

* Valable pour la France uniquement.
Lorsque l'acheteur est non-professionnel, ou consommateur au sens de la loi 78-23 du 10 janvier 1978, les obligations de HP, définies ci-dessus, ne sont pas exclusives de la garantie légale en matière de vices cachés (article 1641 et suivants du Code Civil).

Les calculateurs endommagés par accident ou utilisation hors des spécifications ne sont pas couverts par le coût forfaitaire. Le prix de la réparation est alors fonction des pièces changées et du temps passé.

Garantie des réparations effectuées après la période de garantie

Tout appareil réparé par Hewlett-Packard est garanti, pièces et main d'œuvre, pendant 90 jours à compter de la date de réparation.

Informations

Si vous désirez des informations plus précises concernant les termes de cette garantie, veuillez contacter :

- **En Europe :** Contactez votre distributeur ou le bureau commercial HP le plus proche ou le Siège Social de HP Europe. *N'envoyez pas votre calculateur pour réparation avant d'avoir contacté un bureau Hewlett-Packard ;*

  Hewlett-Packard S.A.
  150, Route du Nant-d'Avril
  P.O. Box CH 1217 Meyrin 2
  Genève, Suisse
  Téléphone : (022) 82 81 11

- **Aux Etats-Unis :**

  Hewlett-Packard Co.
  1000, N.E. Circle Boulevard
  Corvallis, OR 97330
  (503) 757-2000

- **Dans les autres pays :**

  Hewlett-Packard International
  3495 Deer Creek Road
  Palo Alto, CA 94304
  U.S.A.
  Téléphone : (415) 857-1501

Instructions d'expédition

Si vous devez envoyer votre calculateur pour réparation, conformez-vous aux indications suivantes :

- Joignez au calculateur la carte de maintenance portant la description de la panne.
- Si l'appareil est sous garantie, joignez une copie de la facture ou une preuve de la date d'achat.
- Expédiez le calculateur et les différents documents dans la boîte d'origine, ou, éventuellement, dans un autre emballage de protection pour éviter toute détérioration en cours de transport qui ne serait pas couverte par la garantie. Nous vous conseillons d'assurer le colis.
- Que le calculateur soit sous garantie ou non, les frais d'expédition, et éventuellement de douane, sont à votre charge. Le retour est effectué port payé.

Maintenance

Les appareils sont généralement réparés et ré-expédiés dans un délai de cinq jours ouvrables à dater de leur réception au centre de réparation approprié. Il s'agit d'un délai moyen pouvant varier selon l'époque de l'année et la charge de travail du service après-vente.

Réparations

Si votre calculateur doit être réparé, adressez-vous à un distributeur officiel de Hewlett-Packard, qui le fera parvenir à un centre de maintenance Hewlett-Packard.

Tous les centres de maintenance Hewlett-Packard ne sont pas équipés pour assurer la maintenance des calculateurs. Cependant, s'il y a un distributeur officiel Hewlett-Packard dans le pays où vous vous trouvez, vous pouvez être sûr que HP dispose d'un centre de maintenance dans ce pays.

S'il n'y a pas de distributeur officiel Hewlett-Packard dans le pays où vous vous trouvez, vous pouvez contacter le bureau commercial Hewlett-Packard le plus proche pour plus d'informations.

Sécurité - conformité aux normes

Le HP-14B a été testé selon les normes en vigueur aux Etats-Unis et dans le reste du monde. Ces tests passent en revue la sécurité mécanique et électrique du calculateur, les possibilités d'interférences radio, des études ergonomiques et acoustiques ainsi que l'innocuité des matériaux employés. Là où les règlements nationaux l'exigent, des approbations par les différents organismes concernés ont été obtenues et elles figurent sur l'étiquette du produit.

Interférences radio : France

Le HP-14B a été testé et a été trouvé conforme à toutes les normes françaises concernant les interférences radio pour les limites de classe B.

Si vous utilisez du matériel non fabriqué ou non recommandé par Hewlett-Packard, la configuration doit être en accord avec ces normes.

Utilisation en aviation (U.S.A.)

Le HP-14B satisfait aux spécifications du RTCA (Radio Technical Commission for Aeronautics), Docket 160B, Section 21. La plupart des compagnies aériennes permettent l'utilisation de calculateurs en vol sur la base de cette qualification. Une simple vérification auprès d'un membre de l'équipage vous indiquera la position de la compagnie aérienne sur l'usage de calculateurs en vol.

B

Mémoire utilisateur et pile des résultats

Cette annexe traite

- Du partage de la mémoire et des besoins en mémoire,
- De la réinitialisation du calculateur sans perte de données,
- De l'effacement de toute la mémoire utilisateur et du rétablissement des valeurs par défaut, et
- Des opérations qui modifient l'ordre de la pile.

Gestion de la mémoire du calculateur

Le HP–32S possède 390 octets de mémoire disponible à l'utilisateur, qui vous sont disponibles pour n'importe quelle combinaison de données stockées (variables ou lignes de programme). SOLVE, ∫FN et les calculs statistiques nécessitent également de la mémoire utilisateur (particulièrement l'opération ∫FN).

Toutes vos données sont conservées jusqu'à ce que vous les effaciez. Le message `MEMORY FULL` signifie qu'il n'y a à ce moment plus assez de mémoire pour l'opération que vous avez lancée. Il est nécessaire d'effacer une partie—ou la totalité—de la mémoire disponible à l'utilisateur. Vous pouvez par exemple :

- Effacer le contenu de plusieurs, ou de toutes les variables (page 50).
- Effacer un, ou tous les programmes (page 85).
- Effacer les registres statistiques (appuyez sur ■ CLEAR {Σ}).
- Effacer toute la mémoire utilisateur (appuyez sur ■ CLEAR {ALL}).

Besoins en mémoire

Données ou opération	Quantité de mémoire
Variables	8 octets par valeur non égale à zéro (les valeurs zéro n'en utilisent pas).
Instructions dans les lignes de programme	1,5 octet.
Nombres dans les lignes de programme	Entiers, de 0 à 99 : 1,5 octet. Tous les autres nombres : 9,5 octets.
Données statistiques	48 octets.
Calculs avec SOLVE	33,5 octets.
Calculs avec ∫FN (intégration)	140 octets.

Pour prendre connaissance des besoins en mémoire de programmes spécifiques, appuyez sur ■ [MEM] {PGM} . Appuyez sur [▼] ou sur ■ [▲] pour visualiser les saisies (voir page 86.)

Pour réattribuer de la mémoire allouée à un calcul SOLVE ou ∫FN interrompu, appuyez sur ■ [LBL/RTN] {RTN} . Ce processus se fait automatiquement chaque fois que vous exécutez un programme ou un calcul SOLVE ou ∫FN.

Remise à zéro du calculateur

Si le calculateur ne répond pas au frappes sur les touches, ou si son comportement est inhabituel, faites une réinitialisation. La remise à zéro interrompt le calcul ou le programme en cours et annule le mode de saisie de programme, la saisie de chiffres, un calcul SOLVE ou ∫FN, un affichage VIEW ou INPUT. Les données stockées restent en général intactes.

Pour réinitialiser le calculateur, maintenez la touche [C] et appuyez sur [LN]. Si la réinitialisation ne se fait pas, essayez d'installer de nouvelles piles. Si cela ne permet toujours pas de réinitialiser, essayez d'effacer la mémoire en utilisant la procédure décrite dans la section suivante.

Il se peut que le calculateur effectue de lui-même une réinitialisation, s'il subit un choc violent ou si l'alimentation est interrompue.

Effacement de la mémoire

La façon habituelle d'effacer la mémoire utilisateur est d'appuyer sur ■ [CLEAR] {ALL}. Il existe cependant une procédure plus puissante dans ses effets, à laquelle vous pouvez recourir si le clavier ne fonctionne plus correctement.

Si le calculateur ne répond plus aux frappes, et si vous ne pouvez le rénitialiser normlament ou en changeant les piles, essayez la procédure suivante. Son effet est d'effacer la totalité de la mémoire, d'effectuer une réinitialisation et de rétablir les choix par défaut (les choix faits d'office en usine, en page suivante).

1. Appuyez sur [C] et maintenez la pression.
2. Appuyez sur [$\sqrt{x}$].
3. Appuyez sur [Σ+] (appuyez sur ces trois touches simultanément). Lorsque vous relâchez ces trois touches, l'écran affiche MEMORY CLEAR si l'opération a réussi.

Les réglages par défaut

Catégorie	Etat par défaut
Mode angulaire.	Degrés.
Mode de base.	Décimal.
Réglage du contraste.	Moyen.
Séparateur décimal.	« . »
Format d'affichage.	FIX 4.
Indicateurs.	Mis à zéro.
FN= (fonction en cours).	Nulle.
Pointeur de programme (ligne en cours).	PRGM TOP.
Mémoire de programme.	Effacée.
Valeur de départ (nbres aléatoires).	Zéro.
Mouvements de la pile.	Autorisés.
Registres de la pile.	Remis à zéro.
Variables.	Remises à zéro.

La mémoire peut être effacée par inadvertance, par exemple si le calculateur subit un choc violent ou si l'alimentation est interrompue.

Situation de la montée de la pile

Les quatre registres de la pile sont toujours présents et la pile se trouve toujours dans un état ou dans l'autre : la montée de pile est soit *autorisée,* soit *inactivée* lors de l'arrivée du nombre suivant dans le registre X (voyez le chapitre 2, « La pile automatique »).

Toute fonction non citée dans les deux paragraphes suivants autorisera la montée de la pile de résultats.

Opérations inactivant la montée de la pile

Il y a quatre opérations qui inactivent la montée de la pile. Un nombre, saisi après l'une des opérations suivantes « écrasera » le nombre qui se trouve dans le registre X. Les registres Y, Z et T restent inchangés.

ENTER Σ+ Σ− CLx

De plus, dans les cas où [C] et [←] agissent comme CLx, ils désactivent également la montée de la pile.

La fonction INPUT *inactive* la montée de la pile puisqu'elle arrête un programme pour permettre une sollicitation de saisie (le nombre introduit viendra écraser le contenu du registre X), mais il *autorise* la montée de la pile lorsque le programme reprend son cours.

Opérations sans effet sur la montée de la pile

Les opérations suivantes sont sans influence quant à la montée de la pile :

DEG,RAD, GRAD	FIX,SCI, ENG,ALL	DEC,HEX, OCT,BIN	CLVARS
PSE	SHOW	RADIX.;RADIX,	CLΣ
[OFF]	[R/S] et STOP	[▲], [▼]	[C]*, [←]*
[MEM]{VAR}†	[MEM]{PGM}†	[GTO] [·][·]	[GTO] [·] *label nn*
Echange de fenêtres binaires	Saisie de chiffres	Erreurs	[PRGM] et saisie de programme

* Sauf lorsqu'il est utilisé comme CL*x*.

† Y compris toutes les opérations effectuées lorsque le catalogue est affiché, sauf {VAR} [ENTER] et {PGM} [XEQ], qui autorise la montée de la pile.

Situation de registre LAST X

Les opérations suivantes sauvegardent x dans le registre LAST X :

$+, -, \times, \div$	SQRT, x^2	e^x, 10^x
LN, LOG	y^x	$1/x$
%, %CHG	$\Sigma+$, $\Sigma-$	RCL$+,-,\times,\div$*
$\hat{x}$, $\hat{y}$	SIN, COS, TAN	ASIN, ACOS, ATAN
SINH, COSH, TANH	ASINH, ACOSH, ATANH	IP, FP, RND, ABS
$y,x \rightarrow \theta$,r; θ,r$\rightarrow y,x$	$\rightarrow$HR, $\rightarrow$HMS	$\rightarrow$DEG, $\rightarrow$RAD
Cn,r; Pn,r	$x!$	CMPLX$+/-$
CMPLX$+,-,\times,\div$	CMPLX e^x, LN,y^x, $1/x$	CMPLX SIN,COS,TAN

* Notez que la séquence de frappe de rappel x [RCL] [+] *variable* stocke dans le registre LAST X une valeur différente de celle qu'y stocke la séquence x [RCL] *variable* [+]. La première stocke x dans LAST X ; la deuxième stocke *le nombre rappelé* dans LAST X.

C

Résolution des équations : notions approfondies

Cette annexe supplémente les informations qui sont données sur la fonction SOLVE au chapitre 7.

Comment SOLVE trouve une racine

SOLVE est une opération *interactive* ; c'est-à-dire qu'elle exécute la fonction spécifiée de manière répétitive. Elle commence par une estimation de la variable inconnue, *x*, et améliore cette estimation à chaque exécution de la fonction *f(x)*.*

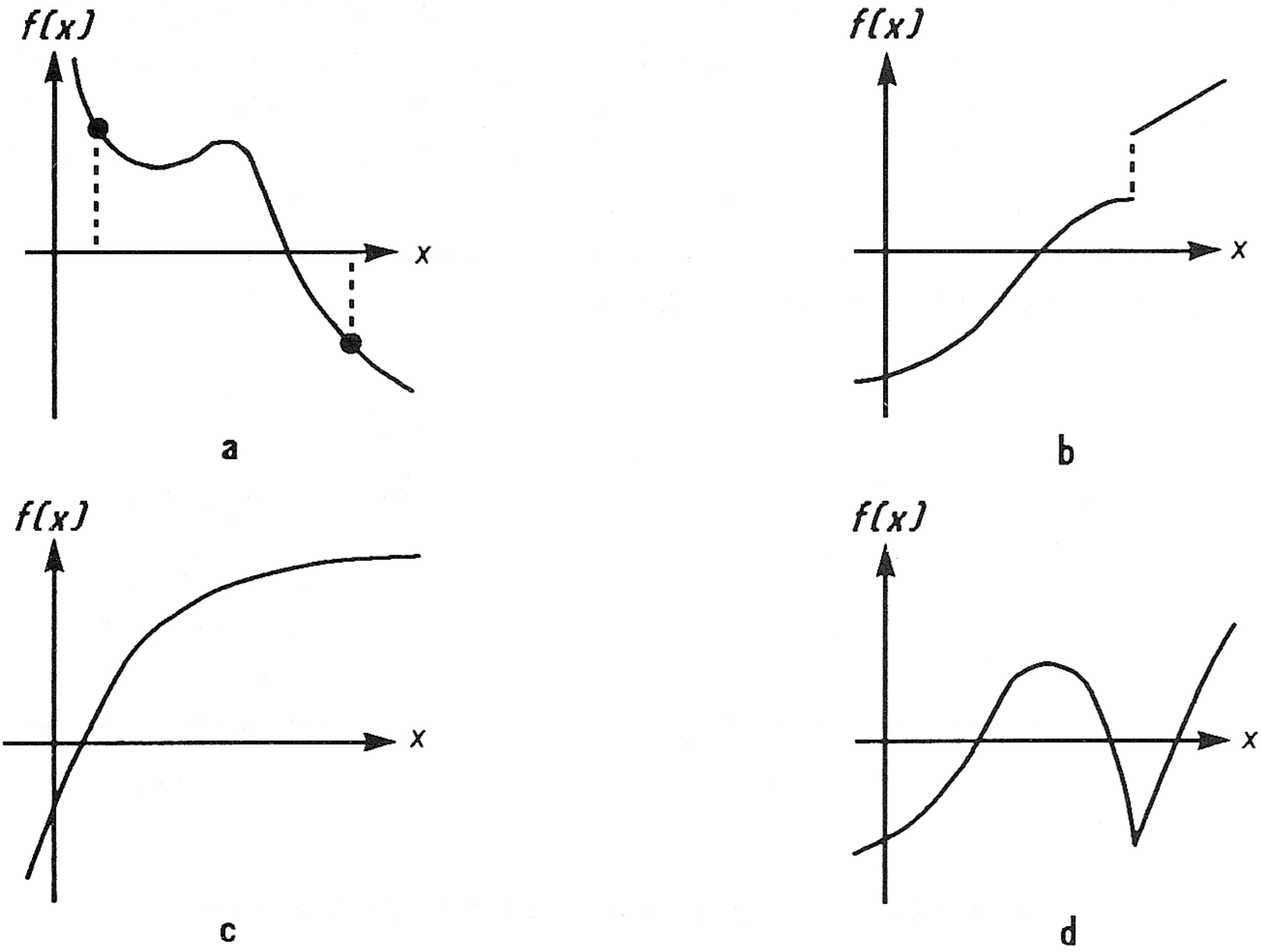
Si deux estimations successives de la fonction *f(x)* ont des signes différents, SOLVE suppose que la fonction *f(x)* rencontre l'axe des *x* à au moins un endroit situé entre les deux estimations. Cet intervalle est systématiquement rétréci jusqu'à ce que la racine soit trouvée.

Pour que SOLVE puisse trouver une racine, celle-ci doit se trouver dans la plage de nombres que peut traiter le calculateur et elle doit être définie mathématiquement là où la recherche itérative s'effectue. SOLVE trouve toujours une racine pour autant qu'une racine existe (dans les limites de la capacité du calculateur), si l'une ou plusieurs de ces conditions sont satisfaites :

- Deux estimations permettent d'obtenir des valeurs *f(x)* de signes opposés, et le tracé de la fonction rencontre l'axe des *x* à au moins un endroit placé entre ces deux estimations (figure a, en page suivante).

* *f(x)* est une notation mathématique d'une fonction définie par l'inconnue *x*.

- *f(x)* augmente ou diminue toujours lorsque *x* augmente (figure b, ci-dessous).
- Le tracé de *f(x)* est soit concave en tous points soit convexe en tous points (figure c, ci-dessous).
- Si *f(x)* possède un ou plusieurs minimums ou maximums locaux, chacun d'eux est placé isolément entre des racines adjacentes de *f(x)* (figure d, ci-dessous).



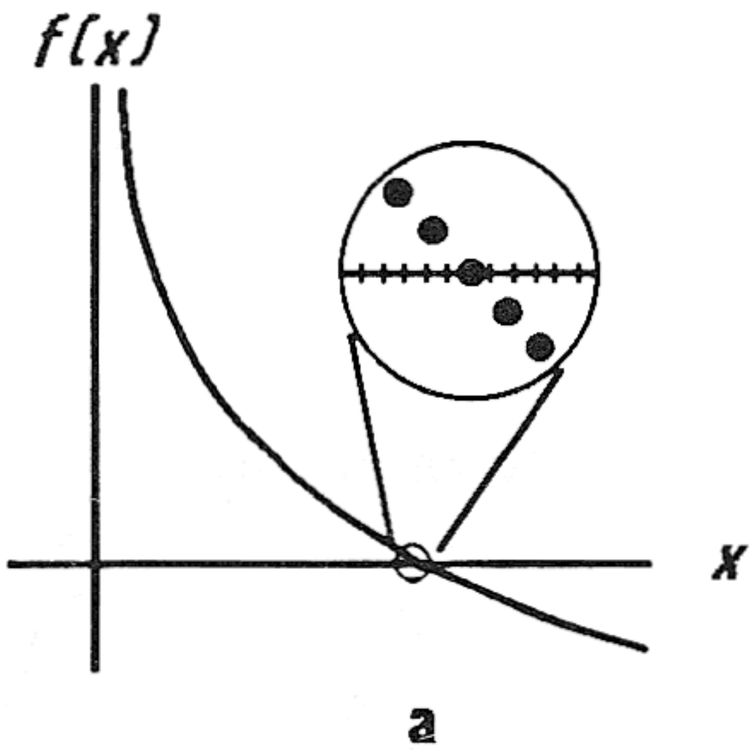
Fonctions dont les racines peuvent être trouvées

Dans la plupart des situations, la racine calculée est une estimation correcte de la racine théorique, infiniment précise de l'équation. Une solution « idéale » serait une situation dans laquelle $f(x)=0$. Cependant une valeur très réduite, non nulle, de *f(x)* est souvent acceptable, parce qu'elle peut résulter de l'approximation de nombres dont la précision (à 12 chiffres) est limitée.

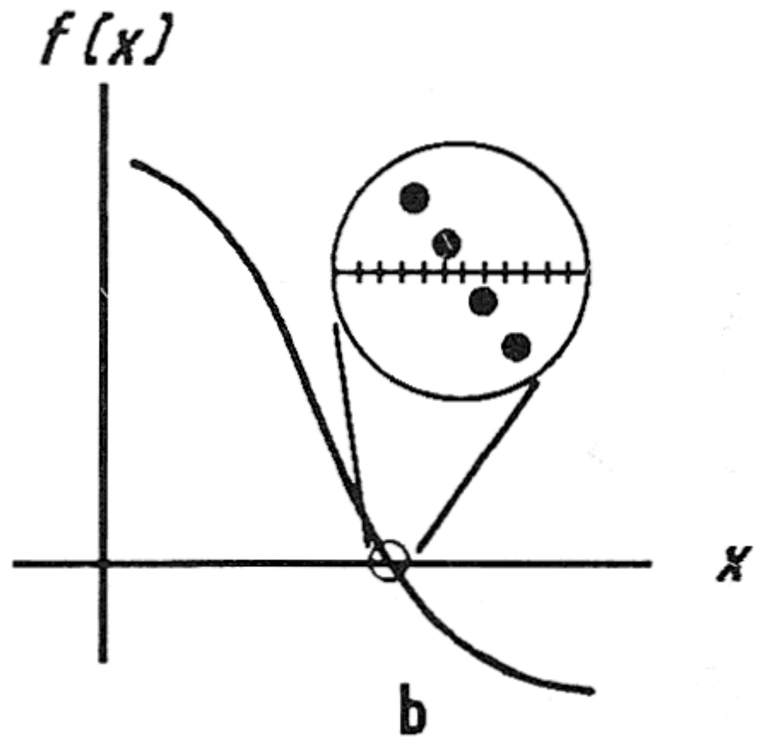
Interprétation des résultats

L'opération SOLVE donnera une solution dans l'une de ces deux conditions :

- Si elle trouve une estimation pour laquelle *f(x)* est égale à zéro (figure a, ci-dessous)
- Si elle trouve une estimation pour laquelle *f(x)* n'est pas égale à zéro, mais que la racine calculée est un chiffre de douze chiffres adajacent à l'endroit où la courbe de la fonction rencontre l'axe des *x* (figure b, ci-dessous). Ceci se produit lorsque les deux estimations finales sont voisines (elles ne diffèrent que par 1, dans le dernier chiffre), et que la valeur de la fonction est positive pour une des estimations et négative pour l'autre.* Dans *la plupart des cas, f(x)* sera relativement proche de zéro.



a



b

Cas dans lesquels une racine est trouvée

Pour obtenir des renseignements complémentaires sur ce résultat, appuyez sur [R↓] pour voir la première estimation de la racine (*x*) laissée dans le registre Y. Appuyez à nouveau sur [R↓] pour voir la valeur de *f(x)*, laissée dans le registre Z. Si *f(x)* égale zéro ou est relativement petite, il est probable qu'une solution a été trouvée. Cependant, si *f(x)* est relativement importante, la prudence est de rigueur lors de l'interprétation des résultats.

* Ou s'ils sont $(0, 10^{-499})$ ou $(0, -10^{-499})$.

Exemple : une équation à une seule racine. Trouvez la racine de l'équation :

$$-2x^3 + 4x^2 - 6x + 8 = 0,$$

qui, en utilisant la méthode de Horner (chapitre 5), se simplifie ainsi :

$$x(x(-2x + 4) - 6) + 8 = 0.$$

Saisissez la fonction dans le programme suivant :

```
A01 LBL A
A02 -2
A03 RCL× X
A04 4
A05 +
A06 RCL× X
A07 6
A08 -
A09 RCL× X
A10 8
A11 +
A12 RTN
```

Touches :	**Affichage :**	**Description :**
■ SOLVE/∫ {FN}A 0 STO X 10 ■ SOLVE/∫ {SOLVE}X	X=1,6506	Calcule x en utilisant les estimations 0 et 10.
R↓	1,6506	Les deux dernières estimations sont les mêmes jusqu'à la quatrième décimale.
R↓	-1,0000E-11	$f(x)$ est très petite, l'estimation est donc juste.

Exemple : une équation à deux racines. Trouvez les deux racines de l'équation parabolique :

$$x^2 + x - 6 = 0.$$

Saisissez l'équation dans le programme suivant :

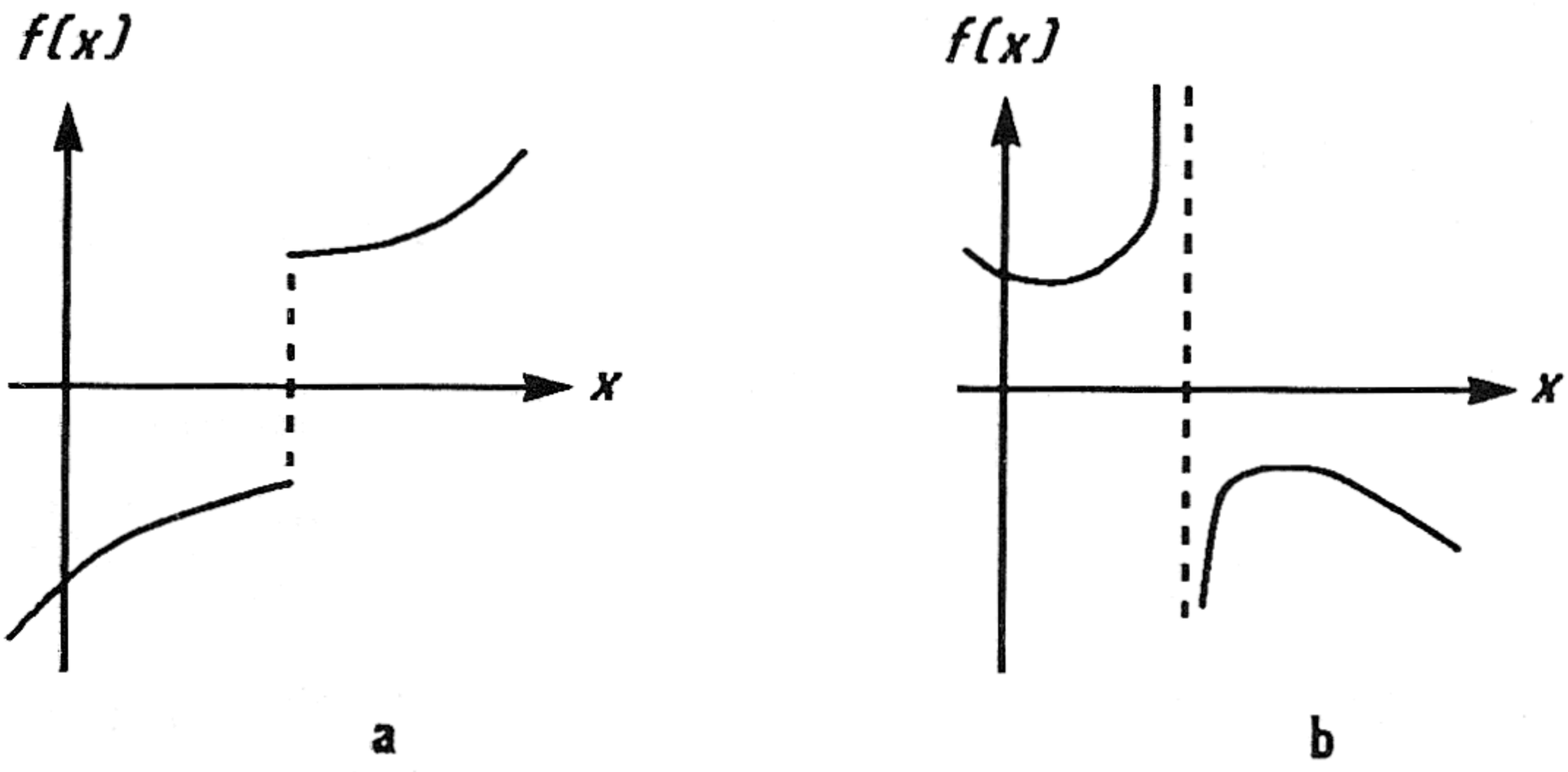
```
D01 LBL D
D02 RCL X
D03 x²
D04 RCL+ X
D05 6
D06 -
D07 RTN
```

Touches :	Affichage :	Description :
■ SOLVE/∫ {FN} D		Calcule la racine positive en utilisant les estimations 0 et 10.
0 STO X 10		
■ SOLVE/∫ {SOLVE}		
X	X=2,0000	
R↓	2,0000	Les deux estimations finales sont les mêmes.
R↓ ■ SHOW	0,00000000000	$f(x) = 0$.
0 STO X 10 +/−		Calcule la racine négative en utilisant les estimations 0 et −10.
■ SOLVE/∫		
{SOLVE}X	X=-3,0000	
R↓ R↓ ■ SHOW	0,00000000000	$f(x) = 0$.

Certains cas nécessitent un traitement particulier :

- Si la courbe de la fonction est discontinue et rencontre l'axe des x par l'une de ses parties, l'opération SOLVE renvoie une valeur adjacente à la discontinuité (figure a, page suivante). Dans ce cas il se peut que $f(x)$ soit relativement importante.

- Les valeurs de *f(x)* peuvent approcher l'infini à l'endroit où la courbe change de signe (voir figure b, ci-dessous). Ceci se nomme un *pôle*. Puisque SOLVE a déterminé qu'il y a changement de signe entre les deux valeurs voisines de *x*, elle renvoie la racine possible. Cependant, la valeur de *f(x)* sera relativement importante. Si le pôle se situe à une valeur de *x* qui est parfaitement représentée par 12 chiffres, cette valeur provoque l'arrêt du calcul avec un message d'erreur.



a b

Cas particuliers : une discontinuité et un pôle

Exemple : une fonction discontinue. Trouvez la racine de l'équation :

$$\mathrm{IP}(x) - 1{,}5 = 0.$$

Saisissez la fonction dans le programme :

```
E01 LBL E
E02 RCL X
E03 IP
E04 1,5
E05 -
R06 RTN
```

Touches :	**Affichage :**	**Description :**
■ SOLVE/∫ {FN}E 0 STO X 5 ■ SOLVE/∫ {SOLVE}X	X=2,0000	Trouve une racine avec les estimations 0 et 5.
■ SHOW	1,99999999999	Montre la racine et 11 décimales.
R↓ ■ SHOW	2,00000000000	L'estimation précédente est légèrement supérieure.
R↓	-0,5000	*f(x)* est relativement important.

Notez la différence entre les dernières deux estimations, et la valeur relativement importante de *f(x)*. Le problème est qu'il n'y a pas de valeur de x pour laquelle *f(x)* égale zéro. Cependant, pour $x = 1{,}99999999999$, il y a une valeur voisine de x qui est de signe opposé pour *f(x)*.

Exemple : un pôle. Trouvez les racines de l'équation

$$\frac{x}{x^2 - 6} - 1 = 0.$$

Si x approche $\sqrt{6}$, f(*x*) devient un nombre positif ou négatif très important.

Saisissez la fonction dans le programme suivant :

```
F01 LBL F
F02 RCL X
F03 x²
F04 6
F05 -
F06 RCL X
F07 x<>y
F08 ÷
F09 1
F10 -
F11 RTN
```

Il est possible de raccourcir ce programme en supprimant les lignes F06–F07 et en ajoutant une seconde instruction `RCL X` après la ligne F02.

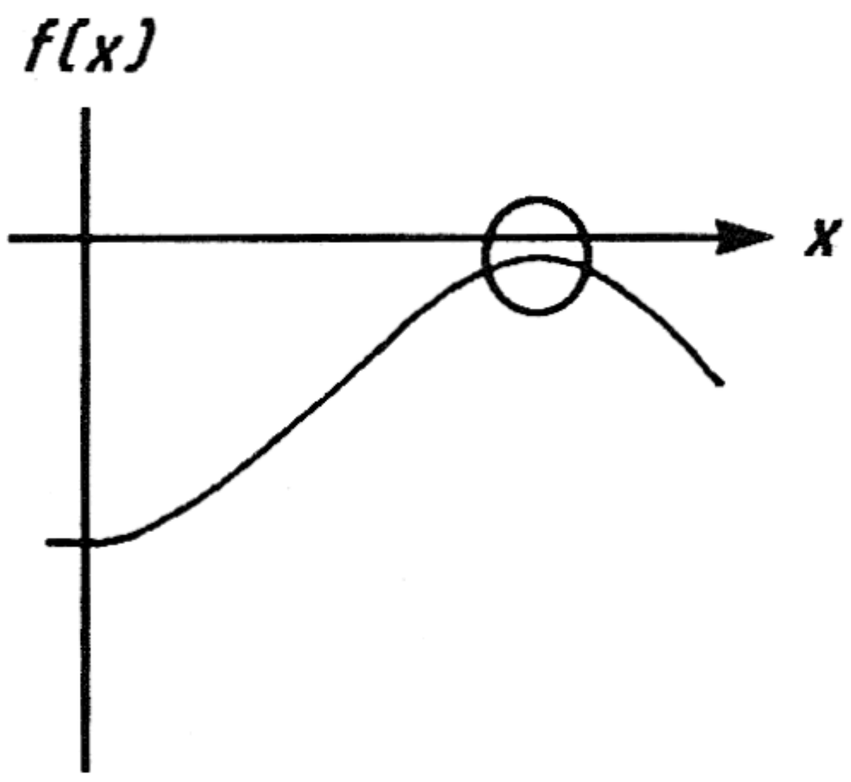
Touches :	**Affichage :**	**Description :**
■ SOLVE/∫		Calcule la racine en utilisant des estimations qui approchent $\sqrt{6}$.
{FN}F		
2,3 STO X 2,7		
■ SOLVE/∫		
{SOLVE}X	X=2,4495	
R↓ R↓	81.649.658.092,0	*f(x)* est relativement important.

Il y a un pôle entre les deux estimations. Les estimations initiales ont obtenu des signes opposés pour *f(x)*, et l'intervalle entre les estimations successives s'est rétréci au point que deux valeurs voisines ont été trouvées. Elles font malheureusement approcher *f(x)* d'un pôle au lieu de l'axe des *x*. La fonction possède des racines aux valeurs −2 et 3, qui peuvent être trouvées en saisissant de meilleures estimations.

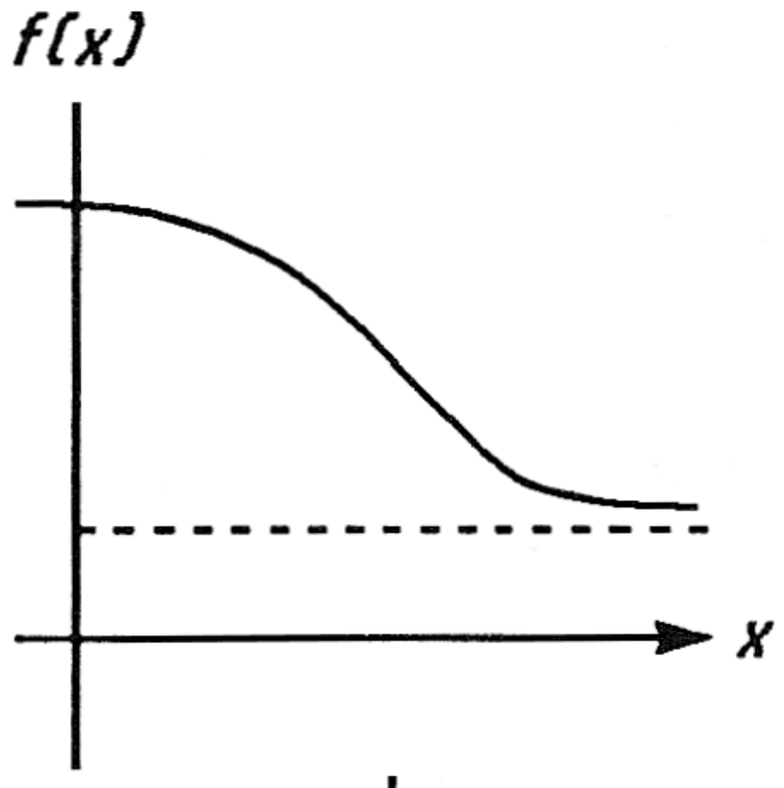
Lorsque SOLVE ne peut trouver une racine

Parfois SOLVE est incapable de trouver une racine. Les situations suivantes provoquent l'affichage du message NO ROOT FND :

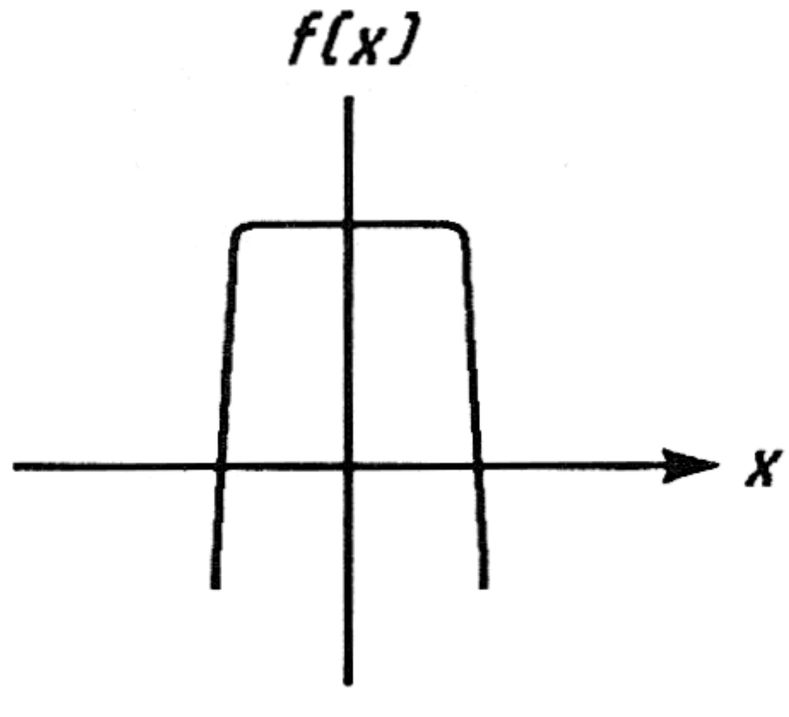
- La recherche s'arrête près d'un minimum ou d'un maximum local (voir figure a, ci-dessous). Si la valeur finale de *f(x)* (stockée dans le registre Z) est proche de zéro, il est possible qu'une racine ait été trouvée ; le nombre stocké dans la variable inconnue peut être un nombre de 12 chiffres très proche de la racine théorique.
- La recherche s'arrête parce que SOLVE travaille sur une asymptote horizontale—une surface dans la quelle *f(x)* est constante pour une grande plage de valeurs de *x* (figure b, ci-dessous). La valeur finale de *f(x)* est la valeur de l'asymptote potentielle.
- La recherche se concentre sur une région « plate » de la fonction (figure c, ci-dessous). La valeur finale de *f(x)* est la valeur de cette fonction dans cette région.



a



b



c

Cas où une racine ne peut être trouvée

L'opération SOLVE renvoie une erreur mathématique si une estimation provoque une opération non correcte—par exemple, une division par zéro, une racine carrée d'un nombre négatif, ou un logarithme de zéro. Souvenez-vous que SOLVE peut générer des estimations sur une plage très large. Les erreurs mathématiques peuvent parfois être évitées en utilisant de bonnes estimations. Si une erreur mathématique se produit, appuyez sur [RCL] *variable inconnue* (ou sur ■[VIEW] *variable*) pour lire la valeur qui a provoqué l'erreur.

Exemple : un minimum relatif. Calculez la racine de cette équation parabolique :

$$x^2 - 6x + 13 = 0.$$

Elle est à son a minimum pour $x = 3$.

Saisissez la fonction dans le programme suivant :

```
G01 LBL G
G02 RCL X
G03 x²
G04 6
G05 RCL× X
G06 -
G07 13
G08 +
G09 RTN
```

Touches :	**Affichage :**	**Description :**
■[SOLVE/∫] {FN}G 0 [STO] X 10 ■[SOLVE/∫] {SOLVE}X	NO ROOT FND	La recherche échoue avec les estimations 0 et 10.
[◄] ■[SHOW]	3,000000010001	Affiche l'estimation finale pour x.
[R↓] ■[SHOW]	3,00000468443	L'estimation précédente n'était pas la même.
[R↓]	4,0000	La valeur finale de $f(x)$ est relativement grande.

Exemple : une asymptote. Trouvez les racines de l'équation

$$10 - \frac{1}{x} = 0.$$

Saisissez la fonction dans le programme :

```
H01 LBL H
H02 10
H03 RCL X
H04 1/x
H05 -
H06 RTN
```

Touches :	**Affichage :**	**Description :**
■ SOLVE/∫ {FN}H ,005 STO X 5 ■ SOLVE/∫		Trouve x avec les estimations 0,005 et 5.
{SOLVE}X	X=0,1000	
R↓	0,1000	L'estimation précédente est la même.
R↓ ■ SHOW	999999999999	$f(x) = 0$.

Voici ce qui se passe lorsque vous utilisez des valeurs d'estimations négatives :

Touches :	**Affichage :**	**Description :**
1 +/- STO X 2 +/- ■ SOLVE/∫		Pas de racine pour $f(x)$.
{SOLVE}X	NO ROOT FND	
◄	-46.666.666.692,1	Affiche la dernière estimation de x.
R↓	-5,7750E15	L'estimation précédente était bien plus grande.
R↓	10,0000	$f(x)$ pour la dernière estimation est plutôt grande.

Il est évident, après examen de l'équation, que si x est un nombre négatif, la valeur la plus petite que puisse prendre $f(x)$ est 10. $f(x)$ *approche* de 10 au fur et à mesure que x devient un nombre négatif de plus grande taille.

Exemple : erreur mathématique. Trouvez la racine de l'équation :

$$\sqrt{[x \div (x + 0{,}3)]} - 0{,}5 = 0.$$

Saissez la fonction dans le programme :

```
I01 LBL I
I02 RCL X
I03 0,3
I04 RCL+ X
I05 ÷
I06 SQRT
I07 0,5
I08 -
I09 RTN
```

Première tentative de trouver une racine positive.

Touches :	**Affichage :**	**Description :**
■ SOLVE/∫ {FN} I		Calcule la racine en utilisant les estimations 0 et 10.
0 STO X 10		
■ SOLVE/∫ {SOLVE}		
X	X=0,1000	

Essayez maintenant de trouver une racine négative en saissant les estimations 0 et -10. Remarquez que la fonction est indéfinie pour les valeurs de x entre 0 et $-0{,}3$ puisque ces valeurs font obtenir un dénominateur positif mais un numérateur négatif, provoquant un résultat négatif.

0 STO X 10 +/−		Erreur mathématique.
■ SOLVE/∫ {SOLVE}		
X	SQRT(NEG)	
■ VIEW X	X=-0,1308	Affiche l'estimation finale pour x.

Exemple : une région « plate ». Trouvez la racine de la fonction

$$f(x) = \begin{cases} x + 2 \text{ if } x < -1 \\ 1 \text{ pour } -1 \leqslant x \leqslant 1 \\ -x + 2 \text{ if } x > 1 \end{cases} \qquad \text{(une région plate)}$$

Saissez la fonction dans le programme :

```
J01 LBL J
J02 1
J03 ENTER *
J04 2
J05 RCL+ X
J06 x<y?
J07 RTN
J08 4
J09 -
J10 +/-
J11 x>y?
J12 R↓
J13 RTN
```

Résolvez pour X en utilisant comme estimations initiales 10^{-8} et -10^{-8}.

Touches :	**Affichage :**	**Description :**
■ SOLVE/∫ {FN} J		Aucune racine n'a pu être trouvée en utilisant des estimations très faibles, proches de zéro (restreignant la recherche à la région plate de la fonction).
E 8 +/- STO X		
1 +/- E 8 +/-		
■ SOLVE/∫ {SOLVE}		
X	NO ROOT FND	
←	1,0000E-8	Les deux dernières estimations sont très éloignées et la valeur finale de *f(x)* est grande.
R↓	0,0025	
R↓	1,0000	

Si vous utilisez de plus grandes estimations, SOLVE peut trouver les racines, qui se trouvent en dehors de la région plate (à $x=2$ et $x=-2$).

* La ligne J03 peut être éliminée pour épargner de l'espace.

Erreur d'arrondi et dépassemement négatif de capacité

Erreur d'arrondi. La précision limitée (à 12 chiffres) du calculateur est due à l'arrondi, qui a un effet défavorable sur les recherches de solutions itératives de SOLVE et d'intégration. Par exemple,

$$[(|x| + 1) + 10^{15}]^2 - 10^{30} = 0$$

n'a pas de racines parce que *f(x)* est toujours plus grande que zéro. Mais avec les estimations 1 et 2, SOLVE donne la réponse 1,0000–provoquée par l'erreur d'arrondi.

L'erreur d'arrondi peut aussi empêcher SOLVE de trouver une racine. L'équation

$$|x^2 - 7| = 0$$

a une racine à $\sqrt{7}$. Cependant aucun nombre de 12 chiffres n'est *exactement* égal à $\sqrt{7}$, ce qui empêche le calculateur de faire atteindre zéro à la fonction. De plus, la fonction ne change jamais de signe. SOLVE renvoie le message `NO ROOT FND`. Toutefois, l'estimation finale de *x* (appuyez sur ◀ pour la voir) est l'approximation de 12 chiffres la meilleure qui puisse être faite de la racine lorsque le sous-programme s'arrête.

Dépassement négatif de capacité. Cette erreur se produit lorsque la taille d'un nombre est plus petite que ce que peut représenter le calculateur, et il lui substitue zéro. Ceci peut influencer les résultats de SOLVE. Voyez par exemple, l'équation

$$\frac{1}{x^2}$$

dont la racine est de valeur infinie. A cause de ce dépassement négatif, SOLVE renvoie une valeur importante comme racine (le calculateur est de toute manière incapable de représenter l'infini).

D

Intégrales : notions approfondies

Cette annexe donne des informations complémentaires aux informations données au chapitre 8.

Comment une intégrale est-elle évaluée ?

L'algorithme utilisé par l'opération d'intégration, ∫FN d *x*, calcule l'intégrale d'une fonction *f*(*x*) en calculant une moyenne pondérée de la valeur de la fonction pour plusieurs valeurs de *x* (que l'on nomme points d'échantillonnage) dans l'intervalle d'intégration. La précision du résultat d'un tel processus d'échantillonnage dépend du nombre de points pris en considération : plus il y a de points, plus grande est la précision. Si *f*(*x*) pouvait être évaluée avec un nombre infini de points d'échantillonnage, l'algorithme serait toujours en mesure de fournir une réponse exacte, pour autant que l'on néglige l'inexactitude inhérente au calcul de la fonction *f*(*x*).

L'évaluation de la fonction avec un nombre infini d'échantillons prendrait un temps très long. Ce n'est cependant pas nécessaire puisque la précision maximum de l'intégrale est par ailleurs limitée par les valeurs de la fonction calculée. Avec un nombre réduit de points d'échantillonnage, l'algorithme peut calculer une intégrale qui est aussi précise qu'il est nécessaire, vu l'incertitude inhérente à *f*(*x*).

L'algorithme d'intégration ne considère au départ que quelques points d'échantillonnage, fournissant des estimations relativement inexactes. Si ces estimations ne sont pas encore aussi précises que celles que permet la fonction *f*(*x*), l'algorithme est répété avec un nombre plus important de points d'échantillonnage. Les itérations continuent, doublant le nombre d'échantillons à chaque passage, jusqu'à ce que l'estimation soit aussi précise qu'il est possible.

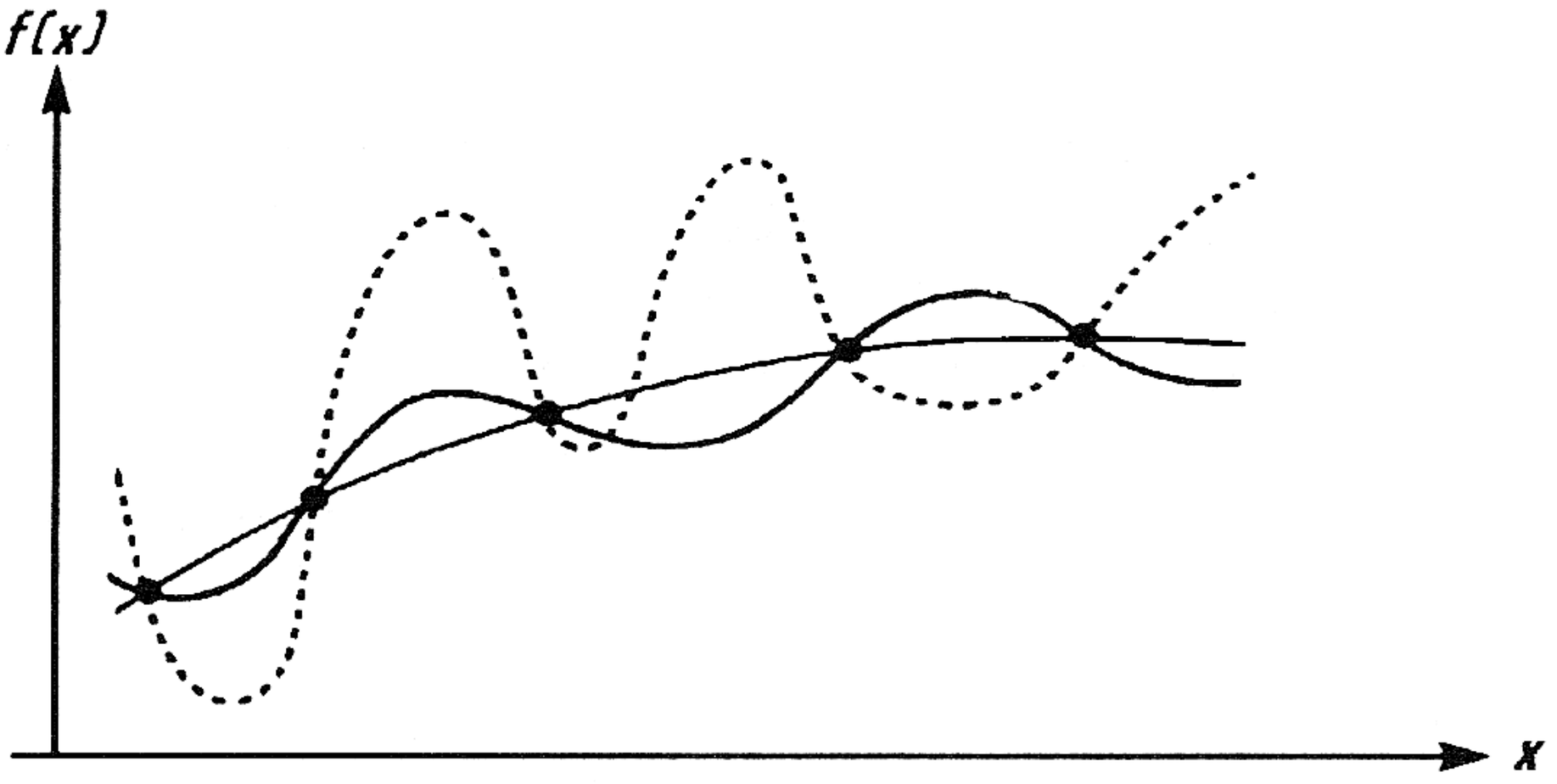
Le chapitre 8 explique que l'incertitude de l'estimation finale est la conséquence du format d'affichage choisi, qui détermine l'incertitude de la fonction. A la fin de chaque itération, l'algorithme compare l'approximation qui vient d'être calculée durant cette itération avec les estimations calculées durant les itérations précédentes. Si la différence entre l'une de ces trois estimations et les deux autres est inférieure à l'incertitude acceptable dans l'estimation finale, la calcul prend fin, laissant l'estimation en cours dans le registre X et son incertitude dans le registre Y.

Il est extrêmement improbable que les erreurs survenues lors de chacune de ces estimations successives—la différence entre l'intégrale réelle et les estimations—soient plus grandes que la différence entre les estimations elles-mêmes. En conséquence, l'erreur contenue dans l'estimation finale sera inférieure à son incertitude (pour autant que $f(x)$ ne varie pas rapidement). Bien qu'il soit impossible de connaître le degré d'erreur dans l'estimation finale, il est très improbable que l'erreur excède l'incertitude de l'estimation. Autrement dit, l'estimation d'incertitude contenue dans le registre Y constitue presqu'à coup sûr une limite supérieure de la différence entre l'estimation et l'intégrale réelle.

Situations qui pourraient provoquer des résultats incorrects

Bien que l'algorithme d'intégration du HP–32S soit l'un des meilleurs, dans certains cas il peut—comme tous les algorithmes d'intégration numérique— vous donner une réponse incorrecte. Cette possibilité est heureusement extrêmement réduite, l'algorithme ayant été conçu pour donner des résultats corrects avec presque toutes les fonctions régulières. Seules les fonctions ayant un comportement extrêmement irrégulier présenteront un risque d'obtenir un résultat inexact. De telles fonctions sont relativement rares dans les problèmes relatifs à des situations réelles ; lorsqu'elles apparaissent, elles peuvent être identifiées et traitées simplement.

Malheureusement, puisque les seuls éléments de $f(x)$ connus de l'algorithme sont ses valeurs aux points d'échantillonnage, il lui est impossible de distinguer entre $f(x)$ et tout autre fonction qui serait en accord avec $f(x)$ à hauteur des points d'échantillonage. Cette situation est illustrée en page suivante :



Dans le cas de ces points d'échantillonnage, l'algorithme calculera la même estimation pour l'intégrale de n'importe laquelle des fonctions illustrées ici. Les intégrales réelles des fonctions illustrées par des lignes courbes sont à peu près semblables, et l'estimation devrait être à peu près exacte si $f(x)$ est l'une de ces fonctions. Cependant, l'intégrale réelle de la fonction illustrée en pointillés est différente de celle des autres, ce qui fait que l'estimation en cours sera plutôt inexacte si $f(x)$ est cette fonction.

L'algorithme estime le comportement général de la fonction en effectuant un échantillonnage de la fonction à un nombre de points de plus en plus grand. Si une fluctuation de la fonction dans une de ses régions est identique à son comportement dans le reste de l'intervalle d'intégration, l'algorithme détectera cette fluctuation après un certain nombre d'itérations. Le nombre de points d'échantillonnage est alors augmenté jusqu'à ce que les itérations successives produisent des estimations qui prennent en compte la présence des fluctuations les plus rapides, mais les plus caractéristiques.

Considérez par exemple l'estimation de

$$\int_0^\infty xe^{-x}dx.$$

Puisque l'évaluation de cette intégrale se fait numériquement, vous pouvez être amené à penser que la limite supérieure de l'intégration est 10^{499}, qui est pratiquement le plus grand nombre qui puisse être saisi dans le calculateur. Essayez cette fonction pour voir les résultats : saissez ce programme qui évalue la fonction $f(x) = xe^{-x}$.

```
F01 LBL F
F02 RCL X
F03 +/-
F04 e^x
F05 RCL× X
F06 RTN
```

Définit le format d'affichage SCI 3, définit les limites inférieure et supérieure de l'intégration comme étant zéro et 10^{499}, puis lance l'intégration.

Touches :	**Affichage :**	**Description :**
■ DISP {SC} 3		Définit le niveau de
0 ENTER E 499	1E499_	précision et les limutes
		d'intégration.
■ SOLVE/∫ {FN} F		Estimation de
■ SOLVE/∫ {∫FN} X	∫=0,000E0	l'intégrale.

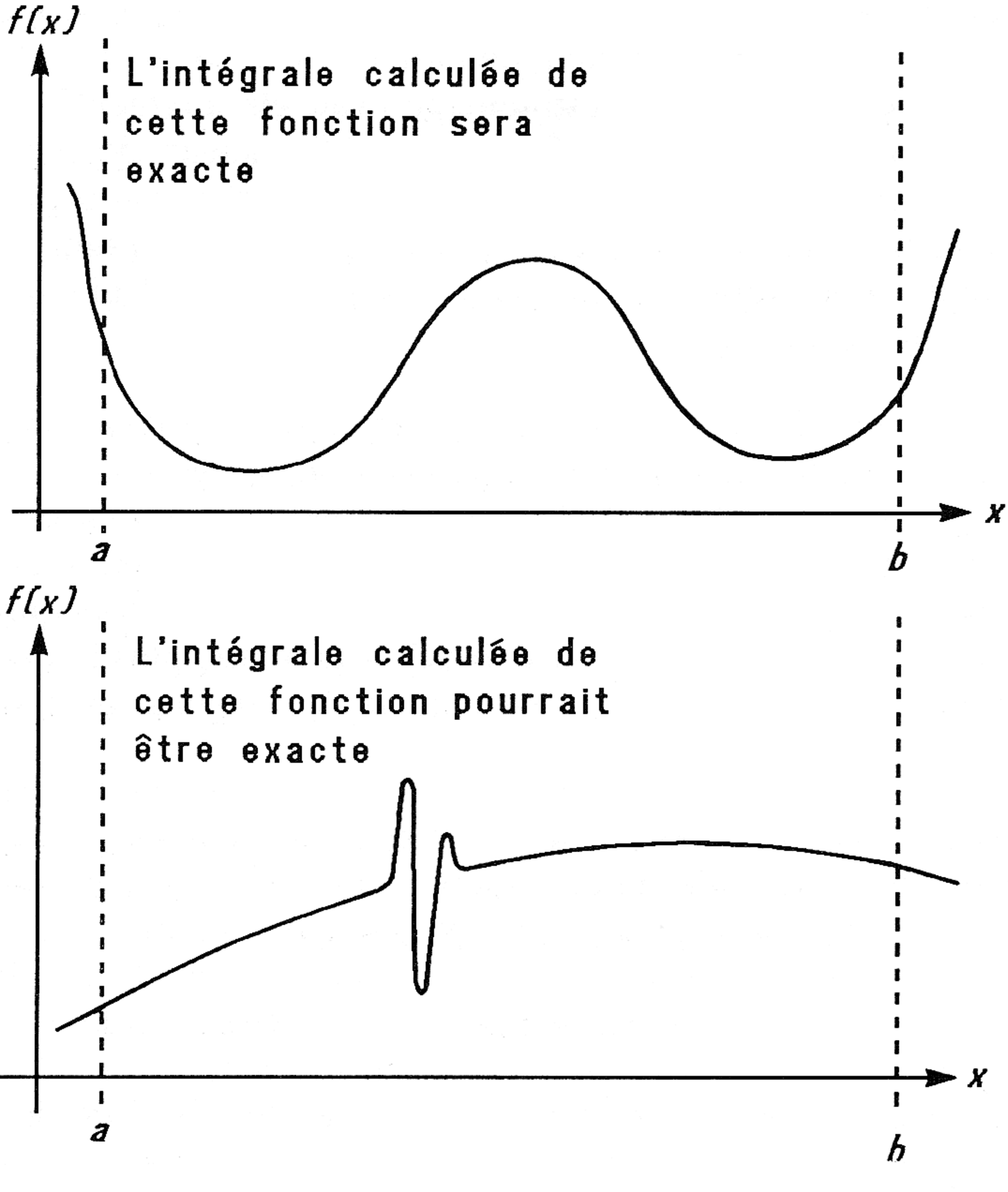
La réponse donnée par le calculateur est visiblement inexacte, puisque l'intégrale réelle de $f(x) = xe^{-x}$ de zéro à ∞ est exactement 1. Mais là n'est pas le problème ; il n'est pas que ∞ a été représenté par 10^{499}, puisque l'intégrale réelle de cette fonction de zéro à 10^{499} est très proche de 1. La raison pour laquelle la réponse est erronnée apparaît clairement sur le graphe de $f(x)$ couvrant l'intervalle d'intégration :



La courbe présente un pic très proche de l'origine. Mais aucun point d'échantillonnage n'a découvert ce pic et l'algorithme a supposé que $f(x)$ est égale à zéro tout au long de l'intervalle d'intégration. Même si le nombre de points d'échantillonnage avait été augmenté en calculant l'intégrale en mode SCI 11 ou ALL, aucun des points d'échantillonnage ainsi ajoutés n'aurait découvert le pic présenté par cette fonction sur cet intervalle particulier. Lisez, en page 279, le paragraphe « Situations qui prolongent le temps de calcul » qui explique ce phénomène.

Heureusement, de telles fonctions sont assez rares et il est peu probable que vous y soyez confronté. Une fonction susceptible de mener à des résultats incorrects peut être identifiée simplement en examinant la vitesse avec laquelle elle-même et ses dérivées varient au long de l'intervalle d'intégration. Plus rapide la variation de la fonction ou de ses dérivées, plus bas sera l'ordre de telles dérivées, plus lentement se déroulera le calcul et moins le résultat de l'approximation sera exact.

Notez que la *rapidité* avec laquelle la fonction varie (elle-même ou ses dérivées d'ordre inférieur) doit être déterminée par rapport à la longueur de l'intervalle d'intégartion. Pour un nombre de points d'échantillonage donné, une fonction $f(x)$ qui présente trois fluctuations peut être bien mieux caractérisée par ses échantillons lorsque ces variations sont présentes sur la plus grande partie de l'intervalle que lorsqu'elles sont confinées à une portion réduite de celui-ci (ces deux situations sont illustrées en page suivante). En considérant les variations ou fluctuations comme un type d'oscillation de la fonction, le critère intéressant est le rapport entre la période des oscillations et la largeur de l'intervalle d'intégration : plus grand est ce rapport, plus rapidement le calcul prendra fin et plus digne de confiance sera l'estimation qui en résultera.



Dans bien des cas vous connaîtrez suffisamment bien la fonction que vous désirez intégrer pour savoir si elle présente des pics dans l'intervalle d'intégration. Si vous ne connaissez pas la fonction, et si vous redoutez des problèmes de cet ordre, vous pouvez tracer rapidement quelques points en évaluant la fonction avec la sous-routine que vous avez créée à cet effet.

Si, pour une raison quelconque, après avoir obtenu une estimation de l'intégrale, vous n'êtes pas sûr de sa validité, il existe un moyen simple de vérifier : subdivisez l'intervalle d'intégration en deux ou plusieurs sous-intervalles adjacents, intégrez la fonction pour chacun d'eux, puis additionnez les estimations qui en résultent. Ceci permettra d'échantillonner la fonction en plusieurs points, réduisant le risque d'éviter l'un ou l'autre pic. Si l'estimation initiale était valable, elle sera égale à la somme des estimations des intervalles.

Situations qui prolongent le temps de calcul

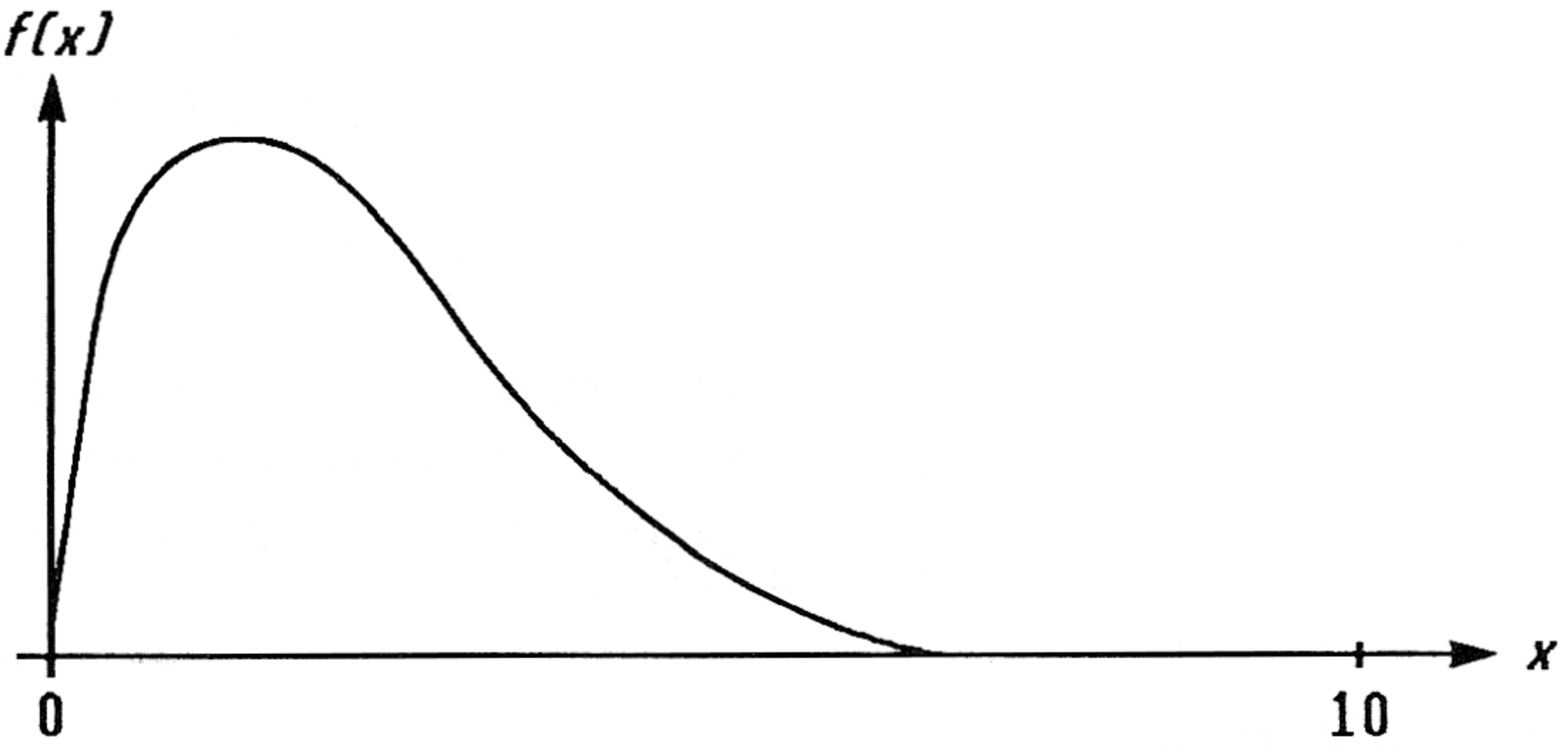
Dans l'exemple précédent, l'algorithme a donné un résultat incorrect parce qu'il n'avait pu déceler le pic de la fonction. Ceci s'est produit parce que la variation de la fonction était trop rapide par rapport à la longueur de l'intervalle d'intégration. Si la longueur de l'intervalle était plus réduite, vous obtiendriez la réponse exacte ; mais il faudrait au calculateur un très long délai pour cela.

Imaginons une intégrale pour laquelle l'intervalle d'iuntégration soit assez long pour nécessiter un long délai de calcul, mais pas assez long cependant pour empêcher un calcul correct. Notez qu'à cause du fait que $f(x) = xe^{-x}$ approche de zéro très rapidement lorsque x approche ∞, la contribution de la fonction pour des valeurs importantes à l'intégrale est négligeable. Vous pouvez donc évaleur l'intégrale en remplaçant ∞, limite supérieure d'intégration, par un nombre inférieur à 10^{499}—par exemple 10^3.

Exécutez à nouveau le problème d'intégration précédent avec cette nouvelle limite. Si vous n'avez pas effectué d'autres intégrations entretemps, vous n'aurez pas à re-définir `FN= F`.

Touches :	**Affichage :**	**Description :**
0 [ENTER] [E] 3	`1E3_`	Nouvelle limite supérieure.
■[SOLVE/∫] {∫FN} X	`∫=1,000E0`	Intégrale (le calcul prend un certain temps).
[x≷y]	`1,824E-4`	Incertitude de l'estimation.

C'est la réponse correcte, mais il a fallu un certain temps pour l'obtenir. Pour comprendre pourquoi, comparez le tracé de la fonction entre $x = 0$ et $x = 10^3$, qui ressemble très fort à celui de la page 276, avec le graphe de la fonction entre $x = 0$ et $x = 10$:



La fonction n'est « intéressante » que pour de faibles valeurs de x. Pour des valeurs plus importantes, elle diminue rapidement et graduellement de façon prévisible.

L'algorithme échantillonne la fonction avec un plus grand nombre de points d'échantillonnage jusqu'à ce que la différence entre les estimations successives devienne suffisamment petit. Pour un intervalle réduit, dans une région où la fonction est intéressante, il faut moins de temps pour atteindre ce nombre critique d'échantillons.

Pour atteindre le même nombre de points d'échantillonnage, le nombre total requis sur l'intervalle plus grand est bien plus grand que le nombre nécessité par l'intervalle court. En conséquence, un plus grand nombre d'itérations sera nécessaire sur l'intervalle le plus grand pour obtenir une estimation présentant la même précision, et le calcul de l'intégrale nécessitera un temps plus considérable.

Le calcul de l'intégration d'une fonction sera prolongé si l'intervalle d'intégration comprend surtout des régions où la fonction n'est pas intéressante, parce que le temps de calcul dépend du délai nécessaire pour atteindre un certain nombre de points d'échantillonnage dans la région où la fonction est intéressante. Heureusement, si vous devez calculer une telle intégrale, vous pouvez modifier le problème de telle façon que le temps de calcul soit très sensiblement réduit. Une de ces techniques est la subdivision de l'intervalle d'intégration, l'autre la transformation des variables. Ces méthodes permettent de changer la fonction ou les limites de l'intégration de telle façon que l'intégrande se comporte mieux dans l'intervalle—ou les intervalles—d'intégration.

Liste des messages

Le calculateur réagit à certaines situations ou à certaines séquences de touches en affichant un message. Le symbole ⚠ attire votre attention sur ce message. Dans les cas importants, le message reste à l'affichage et il vous faut l'effacer. Pour ce faire, appuyez sur [C] ou sur [←] ; le fait d'appuyer sur n'importe quelle autre touche effacera le message *et* exécutera la fonction correspondant à la touche.

∫FN ACTIVE
Un programme en cours d'exécution a essayé de choisir un label de programme (FN=*label*) pendant un calcul d'intégration.

∫(∫FN)
Un programme en cours d'exécution a essayé de calculer une intégrale (∫FN d *variable*) alors qu'un autre calcul d'intégration était en cours.

∫(SOLVE)
Un programme en cours d'exécution a essayé d'exécuter une opération SOLVE alors qu'un calcul d'intégration était en cours.

ALL VARS=0
La catalogue des variables ([■] [MEM] {VAR}) indique qu'aucune value n'est stockée.

CALCULATING
Le calculateur exécute une fonction.

DIVIDE BY 0
Tentative de division par zéro (par exemple [%CHG] si le registre Y contenait zéro).

DUPLICAT, LBL
Tentative d'enregistrer un label de programme existant pour un autre sous-programme.

INTEGRATING
Le calcul d'une intégrale est en cours, attendez quelques instants.

INVALID DATA
Erreur de donnée.

- Tentative de calculer des combinaisons ou des permutations avec $r>n$, avec un non-entier r ou n, ou avec $n\geqslant 10^{12}$.
- Tentative d'utiliser une fonction trigonométrique ou hyperbolique avec un argument illégal : [TAN] avec x, multiple impair de 90 ° ; [ACOS] ou [ASIN] avec $x < -1$ ou $x > 1$; [HYP] [ATAN] avec $x \leqslant -1$ ou $x \geqslant 1$; [HYP] [ACOS] avec $x < 1$.

INVALID x!
Tentative d'exécuter une factorielle ou une opération gamma avec x comme entier négatif.

INVALID y^x
Erreur d'exponentiation :

- Tentative d'élever 0 à la puissance zéro ou à une puissance négative.
- Tentative d'élever un nombre négatif à une puissance non-entière.
- Tentative d'élever un nombre complexe $(0 + i0)$ à un nombre avec une partie réelle négative.

INVALID (i)
Tentative d'exécuter une opération avec adresse indirecte, mais le nombre dans l'indicateur d'index est incorrect ($|i| \geqslant 27$ ou $0 \leqslant |i| < 1$).

LOG(0)
Tentative de prendre le logarithme de zéro ou $(0 + i0)$.

LOG(NEG)
Tentative de prendre un logarithme d'un nombre négatif.

MEMORY CLEAR
Toute la mémoire utilisateur a été effacée (Cf. page 255).

MEMORY FULL
Le calculateur n'a pas assez de mémoire pour effectuer l'opération ; consultez l'annexe B.

NONEXISTENT
Tentative de se référer à un label ou numéro de ligne inexistant avec [GTO], [GTO] [.], [XEQ] ou {FN}. Notez que cette erreur NONEXISTENT peut signifier soit (1) vous avez (à partir du clavier) appelé un *label* de programme inexistant ; ou (2) le programme que vous avez appelé fait référence à un *autre* label qui n'existe pas.

NO LABELS
Le catalogue de progammes ([■] [MEM] {PGM}) indique qu'aucun label de programme n'est stocké.

NO ROOT FND
SOLVE ne peut trouver la racine de l'équatioon avec les estimations initiales fournies (Cf. page 261). Une opération SOLVE exécutée dans un programme ne génère pas cette erreur ; la même situation l'amène à sauter la ligne de programme suivante (celle sui suit l'instruction SOLVE *variable*).

NO STAT DATA
Tentative de d'effectuer un calcul statistique alors qu'aucune donnée statistique n'a été stockée.

OVERFLOW
Ce message est affiché brièvement ; la taille du résultat est trop importante pour que le calculateur puisse la traiter. Il renvoie ±9,999999999E499 dans le format d'affichage en cours. Voyez « Capacité de la machine » en page 24. Cette situation arme l'indicateur 6. Si l'indicateur 5 est armé, le dépassement de capacité a pour effet d'arrêter un programme en cours d'exécution et de laisser le message affiché jusqu'à ce que vous appuyez sur une touche.

PRGM TOP
Indique le haut de la mémoire de programme. La mémoire est structurée de manière circulaire, donc PRGM TOP est aussi la ligne qui suit la dernière ligne présente en mémoire de programme.

RUNNING
Le calculateur exécute un programme (autre qu'un sous-programme SOLVE ou ∫FN).

SELECT FN

Tentative de d'exécuter SOLVE *variable* ou ∫FN d *variable* sans label de programme. Ceci se produit seulement la première fois que vous utilisez SOLVE ou ∫FN après un message MEMORY CLEAR, ou cela peut se produire si le label en cours n'existe plus.

SOLVE ACTIVE

Un programme en cours d'exécution a essayé de choisir un label de programme (FN=*label*) alors qu'une opération SOLVE est en cours.

SOLVE(SOLVE)

Un programme en cours d'exécution a essayé d'exécuter une opération SOLVE alors qu'une autre opération SOLVE est elle-même en cours.

SOLVE(∫FN)

Un programme, lors de son exécution, essaie de calculer une intégrale alors qu'une opération SOLVE est en cours.

SOLVING

Le calculateur est en train de rechercher la racine d'une équation. Cette opération va durer quelques instants.

SQRT(NEG)

Tentative de calcul de la racine carrée d'un nombre négatif.

STAT ERROR

Erreur statistique.

- Tentative de calcul de s_x, s_y, $\hat{x}$, $\hat{y}$, m, r ou de b sans $n = 1$.
- Tentative de calcul de r, $\hat{x}$, ou de $\bar{x}w$ avec uniquement des données x—toutes les valeurs y sont égales à zéro.
- Tentative de calcul de $\hat{x}$, $\hat{y}$, r, m ou b, toutes valeurs de x étant égales.
- Tentative d'effectuer un calcul statistique après que [Σ−] ait réduit n à zéro.

TOO BIG
La taille du nombre est trop grande pour qu'il puisse être converti en base HEX, OCT ou BIN. Le nombre doit être dans la plage $-34\ 359\ 738\ 368 \leqslant n \leqslant 34\ 359\ 738\ 367$.

XEQ OVERFLOW
Un programme en cours d'exécution a essayé de créer un huitième label XEQ *label* emboîté (un maximum de sept sous-routines peuvent être emboîtées). Puisque SOLVE et ∫FN utilisent chacun un niveau, ils peuvent aussi causer cette erreur.

Index des fonctions

Cette section constitue un guide de référence pour toutes les fonctions et les opérations et leur formules, lorsque c'est nécessaire. La liste est classée en ordre alphabétique. Le nom de la fonction est celui qui est utilisé dans les lignes de programme. Par exemple, la fonction FIX *n* est utilisée sous la forme ■ DISP {FX}*n*.

Le nom des fonctions qui ne sont pas programmables sont encadrés de gris, comme ◀.

Les caractères autres que des lettres et les lettres d'alphabet grec sont placés en premier lieu ; les noms de fonctions précédés de flèches (par exemple →DEG) sont classés sans tenir compte de la flèche.

Nom de la fonction	Séquence et description	Page
+/−	+/− Change le signe d'un nombre.	21
+	+ *Addition.* Renvoie *y* + *x*.	25
−	− *Soustraction.* Renvoie *y* − *x*.	25
×	× *Multiplication.* Renvoie *y* × *x*.	25
÷	÷ *Division.* Renvoie *y* ÷ *x*.	25
◀	Supprime le dernier chiffre saisi ; efface *x* ; efface un menu ; supprime une étape de programme.	16, 19, 32, 83
■▲	Affiche la rubrique précédente du catalogue ; déplace le pointeur de programme vers l'étape précédente.	33, 76

Nom de la fonction	Séquence et description	Page
▼	Affiche la rubrique suivante du catalogue ; déplace le pointeur de programme vers l'étape suivante (en saisie de programme) ; exécute la ligne de programme en cours (sf en saisie de programme).	33, 76
1/x	1/x *Réciproque.*	24
10^x	■ 10^x *Exposant commun.* Renvoie 10 élevé à la puissance *x*.	55
%	■ % *Pourcentage.* Renvoie $(y \times x) \div 100$.	59
%CHG	■ %CHG *Pourcentage de changement.* Renvoie $(x - y)(100 \div y)$.	59
π	■ π Renvoie l'approximation 3,14159265359.	56
Σ+	Σ+ Accumule (*y*, *x*) dans les registres statistiques.	154
Σ−	■ Σ− Retire (*y*, *x*) des registres statistiques.	155
Σx	■ STAT {Σ} {x} Renvoie la somme des valeurs *x*.	161
Σx^2	■ STAT {Σ} {x²} Renvoie la somme des carrés des valeurs *x*.	162
Σxy	■ STAT {Σ} {xy} Renvoie la somme des produits des valeurs *x* et *y*.	162
Σy	■ STAT {Σ} {y} Renvoie la somme des valeurs *y*.	161
Σy^2	■ STAT {Σ} {y²} Renvoie la somme des carrés des valeurs *y*.	162

Nom de la fonction	Séquence et description	Page
θ,r→y,x	■ P↔RECT {θ,r→y,x} *Polaire en rectangulaire.*	61
∫FN d *variable*	■ SOLVE/∫ {∫FN} *variable* *Intègre* la fonction en cours par rapport à la *variable*, avec la limite inférieure dans le registre Y et la limite supérieure dans le registre X.	127
ABS	■ PARTS {ABS} *Valeur absolue.* Renvoie |*x*|.	67
ACOS	■ ACOS *Arc cosinus.* Renvoie $\cos^{-1} x$.	57
ACOSH	■ HYP ■ ACOS *Arc cosinus hyperbolique.* Renvoie $\cosh^{-1} x$.	59
ALL	■ DISP {ALL} Affichage de ts les ch. significatifs.	30
ASIN	■ ASIN *Arc sinus.* Renvoie $\sin^{-1} x$.	57
ASINH	■ HYP ■ ASIN *Arc sinus hyperbolique.* Renvoie $\sinh^{-1} x$.	59
ATAN	■ ATAN *Arc tangent.* Renvoie $\tan^{-1} x$.	57
ATANH	■ HYP ■ ATAN *Arc tangent hyperbolique.* Renvoie $\tanh^{-1} x$.	59
b	■ STAT {L.R.} {b} Renvoie *l'ordonnée à l'origine* de la ligne de régression : $\bar{y} - m\bar{x}$.	159
■ BASE	Menu de conversions de bases.	144
BIN	■ BASE {BIN} Choisit le mode binaire (base 2).	144
C	Allume le calculateur ; efface *x* ; efface les messages et sollicitations ; annule les menus ; annule les catalogues ; annule la saisie de programme ; arrête l'exécution d'un programme.	14, 16, 19, 32, 36, 73

Nom de la fonction	Séquence et description	Page
CF *n*	■ FLAGS {CF}*n* *Désarme l'indicateur n* (0 ⩽ *n* ⩽ 6).	98
■ CLEAR	Affiche le menu pour l'effacement de nombres ou de parties de la mémoire ; ou efface la variable ou le programme indiqués du ctalogue MEM.	16, 33
■ CLEAR {ALL}	Efface toutes les données et programmes stockés.	34
■ CLEAR {PGM}	Efface les programmes.	86
CLΣ	■ CLEAR {Σ} Efface les registres statistiques.	154
CLVARS	■ CLEAR {VARS} Met toutes les variables à zéro.	50
CLx	■ CLEAR {x} Met *x* à zéro.	36, 40, 73
■ CMPLX	Affiche le préfixe CMPLX_ pour les fonctions complexes.	139
CMPLX+/−	■ CMPLX +/− *Changement de signe complexe.* Renvoie $-(z_x + iz_y)$.	139
CMPLX+	■ CMPLX + *Addition complexe.* Renvoie $(z_{1x} + iz_{1y}) + (z_{2x} + iz_{2y})$.	140
CMPLX−	■ CMPLX − *Soustraction complexe.* Renvoie $(z_{1x} + iz_{1y}) - (z_{2x} + iz_{2y})$.	140
CMPLX×	■ CMPLX × *Multiplication complexe.* Renvoie $(z_{1x} + iz_{1y}) \times (z_{2x} + iz_{2y})$.	140

Nom de la fonction	Séquence et description	Page
CMPLX÷	■ CMPLX ÷ *Division complexe.* Renvoie $(z_{1x} + iz_{1y}) \div (z_{2x} + iz_{2y})$.	140
CMPLX1/x	■ CMPLX 1/x *Réciproque complexe.* Renvoie $1/(z_x + iz_y)$.	139
CMPLXCOS	■ CMPLX COS *Cosinus complexe.* Renvoie cos $(z_x + iz_y)$.	139
CMPLX e^x	■ CMPLX e^x *Exposant naturel complexe.* Renvoie $e^{z_x + iz_y}$.	139
CMPLXLN	■ CMPLX LN *Logarithme naturel complexe.* Renvoie $\log_e (z_x + iz_y)$.	139
CMPLXSIN	■ CMPLX SIN *Sinus complexe.* Renvoie sin $(z_x + iz_y)$.	139
CMPLXTAN	■ CMPLX TAN *Tangente complexe.* Renvoie tan $(z_x + iz_y)$.	139
CMPLX y^x	■ CMPLX y^x *Puissance complexe.* Renvoie $(z_{1x} + iz_{1y})^{(z_{2x} + iz_{2y})}$.	140
Cn,r	■ PROB {Cn,r} *Combinaisons* de *n* articles pris *r* à la fois. Renvoie $n! \div (r!(n - r)!)$.	65
COS	COS *Cosinus.* Renvoie cos *x*.	57
COSH	■ HYP COS *Cosinus hyperbolique.* Renvoie cosh *x*.	59
DEC	■ BASE {DEC} Choisit le mode décimal.	144
→DEG	■ D↔RAD {→DEG} *Radians en degrés.* Renvoie $(360/2\pi)x$.	64

Nom de la fonction	Séquence et description	Page
DEG	■ MODE {DG} Choisit le mode angulaire « degrés ».	57
■ DISP	Affiche le menu pour ajuster le format d'affichage.	30
■ D↔RAD	Affiche le menu pour convertir entre degrés et radians.	64
DSE *variable*	■ LOOP {DSE} *variable* *Decrement, Skip if Equal* (diminuez, sautez si égal) ou moins. Soustrait *ii* (pour le nombre de contrôle *ccccccc.fffii* stocké dans la variable), de *ccccccc* (valeur-compteur) et, si le résultat est ⩽ *fff* (valeur finale), saute la ligne de programme suivante.	101
E	Commence la saisie d'exposants et ajoute un « E » au nombre saisi. Indique qu'une puissance de 10 suit.	22
e^x	e^x *Exposant naturel.* Renvoie *e* élevé à la puissance *x*.	55
ENG *n*	■ DISP {EN} *n* Choisit l'affichage d'ingénierie avec *n* chiffres suivant le premier. $0 \leqslant n \leqslant 11$.	30
ENTER	ENTER Sépare deux nombres saisis en séquence ; copie *x* dans le registre Y, monte *y* dans le registre Z et monte *z* dans le registre T et perd *t*.	23, 39
FIX *n*	■ DISP {FX} *n* Choisit l'affichage Fix avec *n* décimales. $0 \leqslant n \leqslant 11$.	30
■ FLAGS	Affiche le menu pour arner, désarmer et tester les indicateurs.	98
FN= *label*	■ SOLVE/∫ {FN} *label* Choisit le programme cité dans *label* comme fonction en cours (utilisé par SOLVE et ∫FN).	111, 127
FP	■ PARTS {FP} *Part fractionnaire* de *x*.	67

Nom de la fonction	Séquence et description	Page
FS? *n*	■ FLAGS {FS?} *n* Si l'indicateur *n* ($0 \leqslant n \leqslant 6$) est armé, exécute la ligne suivante ; si *n* est désarmé, saute la ligne suivante.	98
GRAD	■ MODES {GR} Définit le mode angulaire « grades ».	57
GTO *label*	■ GTO *label* Place le pointeur de programme en face du *label*.	93, 100
■ GTO · *label nn*	Place le pointeur s. la ligne *label nn*.	94
■ GTO · ·	Place le pointeur en haut de programme (PRGM TOP).	94
HEX	■ BASE {HX} Choisit le mode de base hexadécimale (base 16).	144
■ HYP	Affiche le préfixe HYP_ pour les fonctions hyperboliques.	59
■ H↔HMS	Affiche le menu pour convertir heures et fractions d'heures en h-min-s.	63
→HMS	■ H↔HMS {→HMS} *Hures en heures, minutes, secondes.* Convertit *x* de l'état de fraction décimale en format min-s.	64
→HR	■ H↔HMS {→HR} *Heures, minutes, secondes en heures.* Convertit *x* de la forme min-s. à la forme de fraction décimale.	64
(i)	Le paramètre indirect. Il adresse (indirectement) la variable ou le label dont la lettre correspond à la valeur numérique contenue dans *i*.	103
INPUT *variable*	■ INPUT *variable* Rappelle la *variable* dans le registre X, affiche le nom de la variable avec le contenu du registre X et arrête l'exécution du programme ; le fait d'appuyer sur R/S (ou sur ▼) stocke le nombre dans la variable.	77
IP	■ PARTS {IP} *Part entière* de *x*.	67

Nom de la fonction	Séquence et description	Page
ISG *variable*	■ LOOP {ISG} *variable* *Increment, Skip if Greater* (augmenter, sauter si >). Ajoute *ii* (pour le nombre de contrôle *ccccccc.fffii* stocké dans la *variable*), to *ccccccc* (valeur compteur) et, si le résultat > *fff* (valeur finale), saute la ligne suivante du programme.	101
LASTx	■ LASTx Renvoie le nombre stocké ds LAST X.	41
LBL *lettre*	■ LBL/RTN {LBL} *label* Donne un *label* d'une seule lettre à un porgramme pour référence par des opérations XEQ, GTO ou FN (prog.).	71
■ LBL/RTN	Affiche le menu pour LBL, RTN et PSE.	71
LN	LN *Logarithme naturel.* Renvoie $\log_e x$.	55
LOG	■ LOG *Logarithme commun.* Renvoie $\log_{10} x$.	55
■ LOOP	Affiche le menu pour DSE et ISG.	99
{L.R.}	■ STAT {L.R.} Affiche le menu de régression linéaire.	158
m	■ STAT {L.R.} {m} Renvoie la pente de régression linéaire : $[\Sigma(x_i - \bar{x})(y_i - \bar{y})] \div \Sigma(x_i - \bar{x})^2$.	159
■ MEM	Affiche le montant de mémoire disponible et le menu de catalogue.	33
■ MEM {PGM}	Ouvre le catalogue de programmes.	85
■ MEM {VAR}	Ouvre le catalogue de variables.	49
■ MODES	Affiche le menu pour définir les modes angulaires et le séparateur décimal.	29
n	■ STAT {Σ} {n} Renvoie le nombre d'ensembles de points de données.	161
OCT	■ BASE {OC} Choisit le mode de base octale (base 8).	144
■ OFF	Eteint le calculateur.	14
■ P↔RECT	Affiche le menu pour conversion de coordonnées polaires en rectangulaires.	60

Nom de la fonction	Séquence et description	Page
■ PARTS	Affiche le menu de sélection de parties de nombres.	67
Pn,r	■ PROB {Pn,r} *Permutations* de *n* articles pris *r* à la fois. Renvoie $n! \div (n - r)!$.	65
■ PRGM	Active ou annule la saisie de programme.	72
■ PROB	Affiche le menu des fonctions de probabilités.	65
PSE	■ LBL/RTN {PSE} *Pause.* Arrête brièvement l'exécution du programme pour afficher *x*, puis reprend le programme (progr. uniqut).	82
r	■ STAT {L,R.} {r} Renvoie le coefficient de corrélation entre les valeurs *x* et *y* : $\Sigma(x_i - \bar{x})(y_i - \bar{y}) \div \sqrt{\Sigma(x_i - \bar{x})^2 \times \Sigma(y_i - \bar{y})^2}$.	159
→RAD	■ D↔RAD {→RAD} *Degrés en radians.* Renvoie $(2\pi/360)x$.	64
{R}	■ PROB {R} Affiche le menu de nombre aléatoire.	65
RAD	■ MODES {RD} Choisit le mode angulaire « radians ».	57
RADIX,	■ MODES { , } Choisit la virgule comme séparateur décimal.	29
RADIX.	■ MODES { . } Choisit le point comme séparateur décimal.	29
RANDOM	■ PROB {R} {RANDOM} Renvoie un nombre aléatoire dans la plage $0 \leqslant x < 1$.	65
RCL *variable*	RCL *variable* *Recall* (rappel). Copie la *variable* dans le registre X.	48
RCL+ *variable*	RCL + *variable* Renvoie *x* + *variable*.	51
RCL− *variable*.	RCL − *variable*. Renvoie *x* − *variable*.	51

Nom de la fonction	Séquence et description	Page
RCL× *variable*.	RCL × *variable*. Renvoie *x* × *variable*.	51
RCL÷ *variable*.	RCL ÷ *variable*. Renvoie *x* ÷ *variable*.	51
RND	■ PARTS {RN} *Arrondit x* à *n* décimales (en option FIX *n*) ou à *n* + 1 chiffres significatifs (en mode SCI *n* ou ENG *n*).	67
RTN	■ LBL/RTN {RTN} *Return.* Marque la fin d'un programme ; le pointeur revient en haut du sous-programme d'appel.	71, 91
R/S	*Run/stop* (marche/arrêt). Commence l'exécution de programme à la ligne de programme en cours *ou* arrête un programme en cours.	82
R↓	R↓ *Roll down* (défilement vers le bas). Déplace *t* vers le registre Z, *z* vers le registre Y, *y* vers le registre X et *x* vers le registre T.	36
SCI *n*	■ DISP {SC}*n* Choisit l'affichage scientifique avec *n* décimales, 0 ⩽ *n* ⩽ 11.	30
SEED	■ PROB {R} {SEED} Relance la séquence de nombre aléatoire avec la valeur de départ |*x*|.	65
SF *n*	■ FLAGS {SF}*n* *Arme l'indicateur n* (0 ⩽ *n* ⩽ 6), pour indiquer « vrai ».	98
■ SHOW	Montre la mantisse complète (tous les douze chiffres) de *x* (ou le nombre dans la ligne de programme en cours.	31
SIN	SIN *Sinus.* Renvoie sin *x*.	57
SINH	■ HYP SIN *Sinus hyperbolique.* Renvoie sinh *x*.	59
■ SOLVE/∫	Affiche le menu pour la résolution pour une inconnue et pour l'intégration.	111, 127

Nom de la fonction	Séquence et description	Page
SOLVE *variable*	■ SOLVE/∫ {SOLVE} *variable* Résout la fonction en cours pour la *variable*, utilisant les estimations initiales placées dans *variable* et dans *x*.	111
SQRT	√x *Racine carrée* de *x*.	24
■ STAT	Affiche le menu de fonctions statistiques.	156
STO *variable*	STO *variable* *Store.* Copie *x* dans *variable.*	48
STO+ *variable*	STO + *variable* Stocke la *variable* + *x* dans *variable.*	50
STO− *variable*	STO − *variable* Stocke la *variable* − *x* dans *variable.*	50
STO× *variable*	STO × *variable* Stocke la *variable* × *x* dans *variable*.	50
STO÷ *variable*	STO ÷ *variable* Stocke la *variable* ÷ *x* dans *variable*.	50
STOP	R/S Arrête l'exécution du programme et affiche le registre X.	82
sx	■ STAT {s}{sx} Renvoie l'*écart-type* pour les valeurs *x* $\sqrt{\Sigma(x_i - \bar{x})^2 \div (n - 1)}$.	157
sy	■ STAT {s}{sy} Renvoie l'*écart-type* pour les valeurs *y* : $\sqrt{\Sigma(y_i - \bar{y})^2 \div (n - 1)}$.	157
TAN	TAN *Tangente.* Renvoie tan *x*.	57
TANH	■ HYP TAN *Tangente hyperbolique.* Renvoie tanh *x*.	59
■ TESTS	Affiche le menu de tests conditionnels.	96
VIEW *variable*	■ VIEW *variable* Affiche le contenu de la *variable* sans rappeler cette valeur dans la pile.	79

Nom de la fonction	Séquence et description	Page
XEQ *label*	XEQ *label* Exécute le programme *label*.	75, 91
x^2	■ x^2 *Carré* de *x*.	24
$\bar{x}$	■ STAT {$\bar{x}$,$\bar{y}$} {$\bar{x}$} Renvoie la moyenne des valeurs *x* : $\Sigma x_i \div n$.	156
$\hat{x}$	■ STAT {L,R,} {$\hat{x}$} Pour une valeur *y* dans le registre X, renvoie l'estimation *x* basée sur la ligne de régression : $\hat{x} = (y - b) \div m$.	159
x!	■ PROB {x!} *Factorielle* (ou gamma). Renvoie $(x)(x-1)\ldots(2)(1)$ ou $\Gamma(x+1)$.	65
$\bar{x}$w	■ STAT {$\bar{x}$,$\bar{y}$} {$\bar{x}$w} Renvoie la *moyenne pondérée* des valeurs *x* : $(\Sigma y_i x_i) \div \Sigma y_i$.	157
x<>y	x≷y *x permuté avec y*. Place *x* dans le registre Y et *y* dans le registre X.	37
x<0?	■ TESTS {x?0} {<0} Si $x < 0$, exécute la ligne suivante ; si $x \geqslant 0$, saute la ligne suivante.	96
x<y?	■ TESTS {x?y} {<y} Si $x < y$, exécute la ligne suivante ; si $x \geqslant y$, saute la ligne suivante.	96
x=0?	■ TESTS {x?0} {=0} Si $x = 0$, exécute la ligne suivante ; si $x \neq 0$, saute la ligne suivante.	96
x=y?	■ TESTS {x?y} {=y} Si $x = y$, exécute la ligne suivante ; si $x \neq y$, saute la ligne suivante.	96
x>0?	■ TESTS {x?0} {>0} Si $x > 0$, exécute la ligne suivante ; si $x \leqslant 0$, saute la ligne suivante.	96

Nom de la fonction	Séquence et description	Page
x>y?	■ TESTS {x?y} {>y} Si $x > y$, exécute la ligne de programme suivante ; si $x \leq y$, saute la ligne suivante.	96
x≠0?	■ TESTS {x?0} {≠0} Si $x \neq 0$, exécute la ligne de programme suivante ; si $x = 0$, saute la ligne suivante.	96
x≠y?	■ TESTS {x?y} {≠y} Si $x \neq y$, exécute la ligne de programme suivante ; si $x = y$, saute la ligne suivante.	96
$\bar{y}$	■ STAT {$\bar{x}$,$\bar{y}$} {$\bar{y}$} Renvoie la *moyenne* des valeurs de *y* : $\Sigma y_i \div n$.	156
$\hat{y}$	■ STAT {L.R.} {$\hat{y}$} Pour une valeur *x* dans le registre X, renvoie l'estimation *y* fondée sur la ligne de régression : $\hat{y} = mx + b$.	159
y,x→θ,r	■ P↔RECT {y,x→θ,r} *Rectangulaire* en *polaire.* Convertit (*x*, *y*) en (*r*, θ).	61
y^x	y^x *Puissance.* Renvoie *y* elevée à la puissance *x*.	56

Index

Les nombres de pages en caractères **gras** indiquent une référence primaire. *Pour chercher des noms de fonctions, utilisez l'index des fonctionsplacé avant cet index.*

Caractères spéciaux

↗, **15**
⚠, **32**
▭, **242**
▼, **17**
← →, **150**
▼▲, **49**
←. *Voir* Espace arrière
π, **56**, 57
0 1 2 3, **98**

A

A..Z, 15, **48**
Adressage indirect, **103–106**
Adresse, indirecte, **103–106**
Affichage
  contraste, réglage, **14**
  de la pile, **36**, 40
  format, **29–31**
  format pour intégration, **127**
  mauvais fonctionnement, **245–246**
  temporaire, **31**
Affichage de nombres dans un programme, **79–80**
Ajustement de courbes, **158–160**
  non linéaire, **204–214**
ALL, format, **31**
Angles, conversion de degrés en fractions, **64**
  vecteur, **164**, 171
Angulaire, mode, **56–57**
Arc cosinus, **57**
Arc sinus, **57**
Arc tangente, **57**
Argent, reçu ou versé, **223**
Arithmétique complexe, **139–140**
Arithmétique de rappel, **51–52**
Arithmétique, **24–29**, 38–46
  avec des variables stockées, **50–52**
  complexe, **139–140**
  dans la pile, **38**
  non décimale. *Voir* ar. de base
  vecteur, **164–175**
Arrondi, 24, 30, 49, **67**
Assistance, **240**

B

Base
  arithmétique, **146–148**
  conversions, **144–145**
  modes, programmation, **151–152**
Bessel, fonction de, **128-130**

Binaires, nombres, **144–150**
  importants, 49
  longs, **149**
  positifs, **148**
Bit de sens, **148**
Bit de signe, **148**
Bit le plus élevé, **148**
Blocage du calculateur, **245–246**
Boîte, les dimensions d'une, **113**, 121
Bouclage, **99–102**
  avec (*i*), 106
Boucle (*loop*)
  avec compteur, 95, **101–102**
  conditionnelle, **100**
  de courant, **181**
  infinies, **100**
  nombre de contrôle, **101**
Branchement, **93–94**, 95
  inconditionnel, **94**
  récursif, **99–102**

C

C. *Voir* Cancel.
Calculs d'épargne. *Voir* Calculs sur des flux constants
Calculs de prêts. *Voir* Calculs sur des flux constants
Calculs en chaîne, **26**, **44–46**
Calculs financiers. *Voir* Calcul sur des flux constants
Calculs statistiques, **156–162**
  limitations des, **160–161**
Calculs sur les flux constants, **222–229**
Cancel, touche, **16**, 19, 32, 36, 40, 73
Cartésiennes, coordonnées. *Voir* coordonnées Rectangulaires
Catalogue
  de programmes, **85**
  de variables, **49**
Celsius, conversion, **229–235**
Changement de signe, **21**
Chiffres
  saisie, **23**
  saisie, arrêt de, **23**
  séparateur, **29**
Chiffres significatifs, **22**, **31**, 49
Chiffres, nombre maxiumum de, **21**
Clear, touche. *Voir* Cancel
Coefficient de corrélation, **159**, 204, 211-212
Combinaisons, **65–66**
Complément à deux, **146**, 148
Compteur de boucle, **101**
Compteur, valeur de, **101**
Constante de croissance, **40**
Constante, utilisation, **39–40**, 43
Contraste de l'affichage, **14**
Conversions d'unités, **229–235**
Conversions de surfaces, **229–235**
Conversions,
  angulaires, **64**
  coordonnées, **60–62**
  fractionnaires, **63–64**
Coordonnées sphériques. *Voir* Coordonnées polaires.
Coordonnées, conversions de, **60–62**
Copie de nombres. *Voir* Stockage de nombres
Copie de variables du catalogue, **49**
Correction d'erreurs avec LAST X, 41, **42–43**
Correction d'une fonction erronnée, **42**
Correction de nombres incorrects, **42**
Cosinus, **57**
Courbes de fonctions SOLVE, **123**
Courbes, limitations inhérentes aux, **205**
Courbes, modèles de **204**, 211
Cramer, méthode de, **175**
Curseur, **15–16**, 23

D

Décimal, point et virgule, **29**
Décimales, nombre de, **30**
Décompteur de boucle, **101**
Défaut, réglages par, rétablissement, **255–256**
Défauts, **248**
Défilement, **84**
Défilement vers le bas, **36–37**
DEG, **57**
Degrés, conversion, **64**
Dépassement de capacité supérieure, **24**
  dans un programme, **98**
  en arithmétique non décimale, **146**
  indiqué, **97–98**
Déterminant, méthode du, **175–182**
Discontinuité, fonction SOLVE, **263–264**
Distribution inverse-normale, **215–221**
Distribution normale, **215–221**
Dommages, 250
Données statistiques,
  accumulées, **161**
  correction, **155**
  effacement, 154, **162**
  ensembles, nombre, 154, **161**
  normalisation, **161**
  précision of, **160–161**
  prévisions, **158–160**
  saisie, **153–154**
  suppression, **154**, 161
Données, affichage, **79–80**
Données, saisie dans un programme, **78**
Données stockées, 253
Données à une variable, **154**
Données à deux variables, **154**
DSE, **101–102**

E

E, **22**
e, **55**
Ecart-type, **156–157**
  échantillon, **157**
  population, 219
  vrai, **157**
Effacement de l'affichage, 36
Effacer *x*, 36, **40–41**, 73
Effacer, **15–16**
  données statistiques, 154
  mémoire, **34**, 253, **255–256**
  programmes, **85–86**
  variables du catalogue, **49**
Elévation d'un nombre à une puissance, **56**
Ellipses sur l'affichage, **150**
ENG, format, **30**
Engénierie, format, **30**
ENTER, **23**, 25–27, **39**
Entier positif, le plus grand, **149**
Envoi du calculateur, **251**
Equation cubique, **194**
Equation du second degré, **191–197**
Equations simultanées,
  méthode d'inversion de matrice, **183–190**
  méthode du déterminant, **175–182**
Erreur
  avec une fonction, correction, **42**
  causant un arrêt du programme, **82**
  message, **32**, 82, **281–285**
Erreur d'arrondi en intégration, **131**
Erreurs, intégration, 274
Erreurs, numériques,
  dans les équations du second degré, 191, 197
  dans SOLVE, 272
  en statistiques, 161, 205
  en trigonométrie, 57
Espace arrière, **16**, 19, 23, 32, 40, 73
Estimation pour *x*, **158–159**, 212
Estimation *y*, **158–159**, 212

Estimations initiales, (SOLVE), 111, 118, **120**
choix, **123**
emplacement, **120**
Exponentiation. *Voir* y^x
Exposant,
commun, **55**
courbe, **204–205**, 211
naturel, **55**
Exposant, **22–23**, 30
chiffres de l'—, **21**
saisie de l'—, **22**
Extinction, **14**

F

f(x), **126**
dans l'intégration, 273
dans SOLVE, 259
Factorielle, 19, **65**
Fahrenheit, conversion, **229–235**
Feet (pieds), conversion, **229–235**
Fenêtres, **149–150**
Fix, décimale fixe, format, **30**
Fonction
diadiques, **25**
evaluation (∫FN), **128**
evaluation (SOLVE), **112–113**
noms, **67**
noms dans les programmes, **74**
monadiques, **24–25**
touche, **24**
Fonctions,
index des, **286–298**
numeriques, **54–69**
SOLVE, **112**
SOLVEables, **259–260**
Fonctions agissant sur des parties de nombres, 25, **67**
Fonctions modifiant les nombres **67**
Fonctions non communicatives, 25, **37**, 45
Fonctions non programmables, **87**
FOR-NEXT, boucle, 101
Force, vecteur, 174
Format FIX, **30**
Fractionnelle, part, **67**
arithmétique non décimale, **146**
Fractions, conversion, **63–64**
Fréquence, courbe, **215**
Fréquences, statistiques, **157**

G

Garantie, **248–249**
service, **251**
Go to. *Voir* GTO
GRAD, **57**
Grades, mode, **57**
Grandeur, 24
GTO, 76, 84, **93–94**, **100**
Générateur de nombres premiers, **235–238**

H

Hexadécimaux, nombres, **144–149**
Horner, méthode de, 262
programmation, **87–88**
Humidité, limites, **245**
Hyperboliques, fonctions, 25, **59**

I

i, **103–106**
(*i*), **103–106**
fonctions utilisant, **104**
pour contrôle du programme, **105**
la variable, 53
Index, valeur d'—, **105**
Indicateur
armer, **98**
désarmer, **98**
état, **98–99**
nombres, **97**
test, 95, 97, **98–99**

Indicateurs,
dépassement de capacité, **97–98**
types, **97–98**
INPUT, **77–79**
annulation, **79**
avec intégration, 128
avec nombres non décimaux, **150**
avec SOLVE, 112
effet sur la pile, **257**
Insertion de lignes de programme, **82**
Instructions conditionnelles, **95–99**, 100
∫FN, **134**
SOLVE, **124**
Intégrale définie, **126**
Intégrale, approximation/estimation, **131**
Intégrande, **127**, 131
Intégration et erreur d'arrondi, **131**
Intégration, **126–136**
algorithme, 130, **272–274**
anomalies, **275–277**
approximations, **273–274**
comment elle opère, **273–280**
conditionelle, **134**
création d'un programme, **128**
dans des programmes, **134–135**
échantillonnage, **274**, **277**
emboîtée, **135**
erreurs, 274
fonction, **128**
incertitudes de l'—, 127, **132**, 274
interruption, 127
itérations, **274**
limitations de l'—, **135**
limites, **127**, 130, 134
méthode, **274**
multi-variables, saisie, 128
précision, 127, **131–134**
résultats, **127**, 134, **274–278**
résultats, vérification, **278**
sorties, **134**
temps de calcul, **279–280**
utilisation, **127**
Inverse, complexes, **139**
Inverse, matrice. *Voir* Matrice inverse
ISG, **101–102**

K–L

Kelvin, conversion, **229–235**
Łukasiewicz, **35**
Labels, **71–72**, 73, 77, 85, 86, 94, 95
LAST X, registre, **41–44**
opérations qui l'affectent, **258**
LBL, **71–72**, 73. *Voir aussi* Labels
Lignes de programme, **72**
en modes non décimaux, **151**
insertion, **82**
renumération, 82
suppression, 73, **82**
Linéaire
estimation —. *Voir* Régression linéaire
mouvement, solution, **115**
régression, 156, **158–160**
Logarithmes,
communs, **55**
complexes, **139**
naturels, **55**
Logarithmique
courbe, **204–205**, 211
fonctions, 25, **55**, 139
Longueurs, conversions, **229–235**
LOOP, 99, **101**

M

Mantisse, **22**, 30–31, 49
Matrice
coefficient, **183**
formules, **175–176**, **183**
inverse, **183–190**
inversion, **183–190**
résultat, **183**
Matrices, solutions. *Voir* Equations simultanées

Mauvais fonctionnement du calculateur, **245–247**, 249–250
MEM, **33**, **49**, **85**
Mémoire
allocations, **254**
besoins, **254**
des variables, **50**
disponible, **33**, 49
économie, 51
effacement, **34**, 50, 253
effacement total, **255–256**
gestion, **253–254**
perte à cause de piles faibles, **243**
perte après installation des piles, **245**
programme, 72, 78, **84–87**
stockée, **253–254**
utilisateur, 47
utilisation, **254**
utilisation dans les programmes, **84–85**
utilisation en statistiques, **162**
vérification, **33**
Mémoire permanente, **14**, 243
MEMORY FULL, 85, 162, **253**
Menu, **17**
annulation, **19**
sortie, **19–20**
touches, **16–19**
types de, **18**
utilisation, **16–19**
Messages, **32**, **281–286**
Mètres, conversion, **229–235**
Mise sous tension, **14**
Mode degrés, **57**
MODES, menu, **29**
Moment, 174
Montée de la pile, **38**, 39
activée, 257
inactivée, **257**
neutre, **257**
opérations qui affectent la —, **256–257**
Moyenne, **156–157**
pondérée, **157–158**
population, 219
MEMORY CLEAR, 243, 245, **255**

N

Nombre aléatoire
générateur, **65**
Valeur de départ, **65**
Nombre,
à deux fonctions, **25**
affiché, **30**
arrondi, 30
avec label, 41
grandeur d'un 24, 272
plage, **24**, 272
Nombres,
complexes, **137–143**
correction, **15–16**, 41
dans les lignes de programme, **73**, 151
entier, le plus grand, **149**
justifié à droite, **148**
négatifs, **21**
non décimaux, **144–150**
octaux, **144–149**
représentation interne, 147–148
représentation interne de nombres non décimaux, **147–148**
partiellement cachés, 150
premiers, **235–238**
réels, **54**
saisie, **21**
séparation, **23**, 27, **39**
taille, **21**
trop grands, **21**, 22, 49
trop petits, 22
Nombres complexes, **137–143**
avec intégration, 126
avec SOLVE, 112
saisie, **137**, 138
Nombres imaginaires, **137**

Nombres importants et conversion de base, **149**
Nombres réels, **54**
Notation polonaise inverse, , 25–26, 28, **35**, 44–46.
Numéros de lignes dans un programme, **72**
Numéros de lignes de programme
en modes non décimaux, **151**
mouvement vers, **84**, 94
NO ROOT FND, **119**, 267

O

Octets dans les programmes, **85**
Opération,
aide, **240**
vérification, **245–247**
Opérations, index, **286–298**
Ordonnée à l'origine, **159**, 204, 211–212
Ordre
des calculs, **26**, 45–46
des nombres, 37
des saisies, 25

P

P↔RECT, **60–62**
Paiement, **226**
Parenthèses, **26**, 28, 45
Partie entière, **67**
en arithmétique non décimale, **146**
PARTS, menu, **67**
Pause, programmée, **82**
Pente, **159**, 204, 211–212
Périodes de composition, **226**
Périodes des nombres, **29**
Permutation de nombres (registres X et Y), 25, **37**
Permutations, **65**
Phasor, forme, complexe, **142**
Pieds (feet), conversion, **229–235**
Pile
complexe, **138**
descente, **38**, 39
mémoire automatique, **35–46**
remplissage par une constante, **39–40**
Pile,
examen, **36**
lecture sans modification, **49**
sous-routine, **92**, 125, 135
Piles faibles, **242–243**
Piles,
dommages causés par les, **248**
installation, **243–245**
types de, **242**
Plage de nombres, **24**, 149
Points de l'affichage, **150**
Polaires
coordonnées, conversion, **60–62**
coordonnées de vecteurs, **170**
forme, complexe, **142**
Pole, fonction SOLVE, **264–265**
Polynômes,
du second degré, **191–197**
expressions, programmation, **87–88**
Pour-cent, **59–60**
Pourcentage de changement, **59–60**
Précision,
de données statistiques, **160–161**
intégration, **132**
numérique, **30–31**
pleine, **31**
SOLVE, **272**
trigonométriques, **57**
Précision interne, **30–31**
Précision, spécifiée pour l'intégration, **132**
PRGM TOP, **72**, 73, 84
mouvement vers, **84**, 94
PRGM, **72**, 73, 75
PROB, menu, **65**
Probabilité, **65–66**
normale, **215–221**, 219

Produit scalaire, vecteur, **164,** 171
Produit vectoriel, vecteur, **164,** 171
Programmation du mode de base, **151–152**
Programmation, **70–89**
Programme
  arrêt, **82**
  création, **71–74**
  exécution, **75**, 76, 85
  lecture ligne par ligne, **76**
  mémoire, 72, **84–87**
  noms. *Voir* Labels
  pointeur, 76, **84**, 94
  reprise du programme, 78, **82**
  retours, (returns), **72**, 73
  test, **75–76**
Programme,
  affichage, via le catalogue, **85**
  catalogue, **85**
  correction, **82**
  exécution, **75**
  exécution via le catalogue, **85**
  exécution pas-à-pas, **76**
  interruption, **82**
  limites, **71–72**
  saisie, **72–73**
  suppression, via le catalogue, **85**
  vérification, **86**
Programmes, effacement, **85–86**
Programmes, labels, **71–72**, 73, 77, 85, 86, 94, 95
  branchement vers, **94**
  dans le catalogue, **85**
  double exemplaire, **72**
  indirect, 103, **104**
Puissance
  consommation, **242**
  courbe, **204–205**, 211
  fonction, **56**
  fonction, complexe, **140**

Q–R

Questions, **240–241**
Raccourci, 246
Raccourcissement en arithmétique non décimale, **146**, 147
Racine cubique, **56**
Racine,
  approximation, 119, 197, **261**
  d'une équation, 110, **116–117**, 119
  du second degré, **191–197**
  fonction, **56**
  maximum, 267
  minimum, 267–268
  pas de —, **267–271**
  trouver la —, **259–260**
Racines complexes, du second degré, **191**
RAD, **57**
Radians mode, **57**
Radians, conversion, **64**
Radius vecteur, 174
Rankine, conversion, **229–235**
Rappel de nombres, **48**
  dans un programme, 78
RCL, **48**
Recherche de variables inconnues, **110–125**
Rectangulaires
  coordonnées, conversion, 58.**60–62**
  coordonnées de vecteur, **170**
  forme, complexe, **142**
Récupération de nombres. *Voir* Rappel de nombres
Référence, fonction, **286–298**
Registre, LAST X, **41–44**
Registres statistiques, **161–162**
  allocatoin, **162**
  effacemrent, **162**
Registres,
  permutation, **37**
  pile, **35–41**, 78
stockage. *Voir* Variables

Régression
coefficients. *Voir* Pente *et* ordonnée à l'origine
non linéaire, **204–214**
Réinitialisation de la mémoire, **254–255**
Remontée de la pile, **38**, 39
Réparation, **245–247**
Réparation, 248. *Voir aussi* Service
Résolution d'équations, **110–125**
Résultats intermédiaires, 26, 28, **35**, 44–46
Réutilisation de nombres avec LAST X, 41, **43–44**
Rotation, coordonnées, **198–203**
RTN, **72**, 73. *Voir aussi* retours de programme
RTN, sous-routine, **91**
Run/stop, 78
R↓, **36–37**

S

Saisie de programmes, **78**
SCI, format, **30**
Scientifique, format, **30**
Service, **249–251**
centres, **250**
contrats, **251**
coût, **250**
international, **250**
SHOW, **31**, 49, 79
nombres non décimaux, **150**
Sinus, **57**
intégrale, **130–131**
Solde, **226**
Sollicitation d'une variable, **77**, 79
Solutions. *Voir* résultats de SOLVE
SOLVE, 259–272
asymptote, 267, **269**
avec des fonctions discontinues, **263–264**
avec une fonction à deux racines, 263
avec une fonction à une seule racine, 262
calcul, interruption, **119**
comment elle opère, **259–260**
conditionnelle, **124**
dans les programmes, **124**
définition de fonctions pour, **112–113**
dépassement de capacité inférieure, **272**
emboîtée, **125**
erreur mathématique, 270
estimations, **261**
itérations, **118**, **259**
limitations, **125**
maximum, 267
méthode, **259–260**
minimum, 267–268
précision, **272**
programmes (fonctions), **112–113**
recherche, **120**, **267–268**
région plate, 267, **271**
restrictions, **259–260**
résultats, 111, **119**, 120, 124, 268, 272
résultats, aucun, **267–271**
résultats, interprétation, **261**
saisie multivariable, 112
sorties, **124**
utilisation, **111–113**
Somme
des carrés, **162**
des produits, **162**
des valeurs de x, **161**
des valeurs de y, **161**

Sommes, positives et négatives, **223**
Sorties de programmes, **78**
Sous-programmes, **90**
Sous-routines, **91**
  embîtées, **92**
STAT, menu, **156**
Statistiques, **153–162**
STO, **48**
Stockage arithmétique, **50–51**
Stockage de nombres, **48**
Support clients, **240**
Suppression de lignes de programme, **82**
Surface d'un cercle, 70, **74**, 78
Surface d'un cylindre, **80–81**

T

T, registre, **35–36,** 38–40, 47
Taille de mot, **149**
Tangente, **57**
Taux d'intérêt, **226**
Témoins, **20–21**
  indicateur, **98**
Températures,
  conversion, **229–235**
  opérations, **245**
  stockage, **245**
Temps, conversions entre minutes et fractions d'heure, **63–64**
Test automatique, calculateur, **246–247**
Test vrai/faux, **95–99**
Tests conditionnels, **95–99**
Tests de comparaison, **95–97**
TESTS, **96**
Totalisation de contrôle, 85, **86–87**
Touche inactive, **32**
Touche préfixe, **15**
Touche préfixe, annulation, **15**
Touches alphabétiques, **15**, 48, 71
Transformations de coordonnées, **198–203**
Translation, coordonnées, **198–203**
Trigonométrie inverse, **57**, 58
Trigonométrie, 25, **56**
  complexe, **139**

U–V

Underflow, dépassement de capacité inférieure, 24
  SOLVE, **272**
Valeur absolue, **67**
Valeur de compteur, **101**
Valeur future, **226**
Valeurs de sommation, statistiques, 156, **161–162**
Variables, **47–53**
  affichage, **49**
  catalogue, **49**
  copie, **49**
  dans les programmes, 77
  dans les programmes, affichage, **79–80**
  dans les programmes, copie, **79–80**
  effacement, **49–50**
  inconnue, **110–112**, 120
  indirectes, **103**
  intégration, **128**
  listage, **49**
  noms, **47–48**, 77
  SOLVE, **112**
  valeur courante, **77**
Variable dépendante, **154**
Variable indépendante, **154**
Variable, visualisation, **49**
Vecteur
  addition, **142**
  composants, **171**
  conversion en coordonnées rectangulaires, **62**
  opérations, **164–175**
Vecteur colonne, 189
VIEW, **49**, **79–80**
  avec des nombres non décimaux, **150**
Virgule comme séparateur décimal, **29**
Volume d'un cylindre, **80–81**

X, Y, Z

X, registre, **35–40**, 47
  avec SOLVE, **113**, **120**
  comparaison avec le registre Y, **95–96**
  comparaison avec zéro, **96**
  echange avec le registre Y, **37**
  effacement, **40–41**
  effacement dans un programme, 73
  en programation, **70**
  et intégration, **128**
  pour données statistiques, **154**
  test, **95–96**
XEQ, **75**
  sous-routine, **91**
y^x, **56**
Y, registre, **35–37**, 47
  et l'intégration, **132**
  pour les données statistiques, **154**
Z, registre, **35–36**, 47
Zéro, **40**
Zéro dans une variable, **50**

Comment contacter Hewlett-Packard

Renseignements sur l'utilisation du calculateur. Si vous avez des questions relatives au fonctionnement du calculateur et que vous ne pouvez trouver de réponse dans ce manuel (après avoir consulté **Réponses à des questions fréquemment posées**, l'index et la table des matières) consultez votre **distributeur Hewlett-Packard** ou bien adressez-vous directement à :

Pour la France :

Appelez le numéro d'assistance
téléphonique
(1) 43 37 63 50

Pour la Belgique :

Hewlett-Packard Belgique S.A./N.V.
100, boulevard de la Woluwe
B-1200 Bruxelles
(02) 762-32 00

Pour la Suisse francophone :

Hewlett-Packard (Suisse) S.A.
7, rue du Bois-du-Lan
CH-1217 Meyrin 1-Genève
(22) 83 11 11

Pour le Canada francophone :

Hewlett-Packard Limitée (Canada)
17500 Route Transcanadienne
Voie de Service Sud
Kirkland, Québec H9J 2M5
(514) 697-4232

L'annexe A indique comment déterminer si le calculateur nécessite réellement une réparation. Ele vous indique également comment procéder pour le faire réparer si cela est nécessaire.

Table des matières

Page 13 Première partie : fonctionnement de base
Introduction • La pile automatique • Comment stocker des données dans des variables • Fonctions sur les nombres réels

69 Deuxième partie : programmation
Programmation élémentaire • Techniques de programmation

109 Troisième partie : fonctionnement de haut niveau
Résolution d'une équation à une inconnue • Intégration numérique • Opérations sur les nombres complexes • Conversions de bases et arithmétique • Opérations statistiques

163 Quatrième partie programmes d'application
Programmes mathématiques • Programmes statistiques • Programmes divers

239 Cinquième partie : annexes et matériel de référence
Assistance, piles et service après-vente • Mémoire utilisateur et pile des résultats • Résolution d'équations : notions approfondies • Notions complémentaires sur l'intégration • Messages • Index des fonctions • Index

HEWLETT PACKARD

Numéro de référence
00032-90043

00032-90044 French—Français
Printed in West Germany 6/88