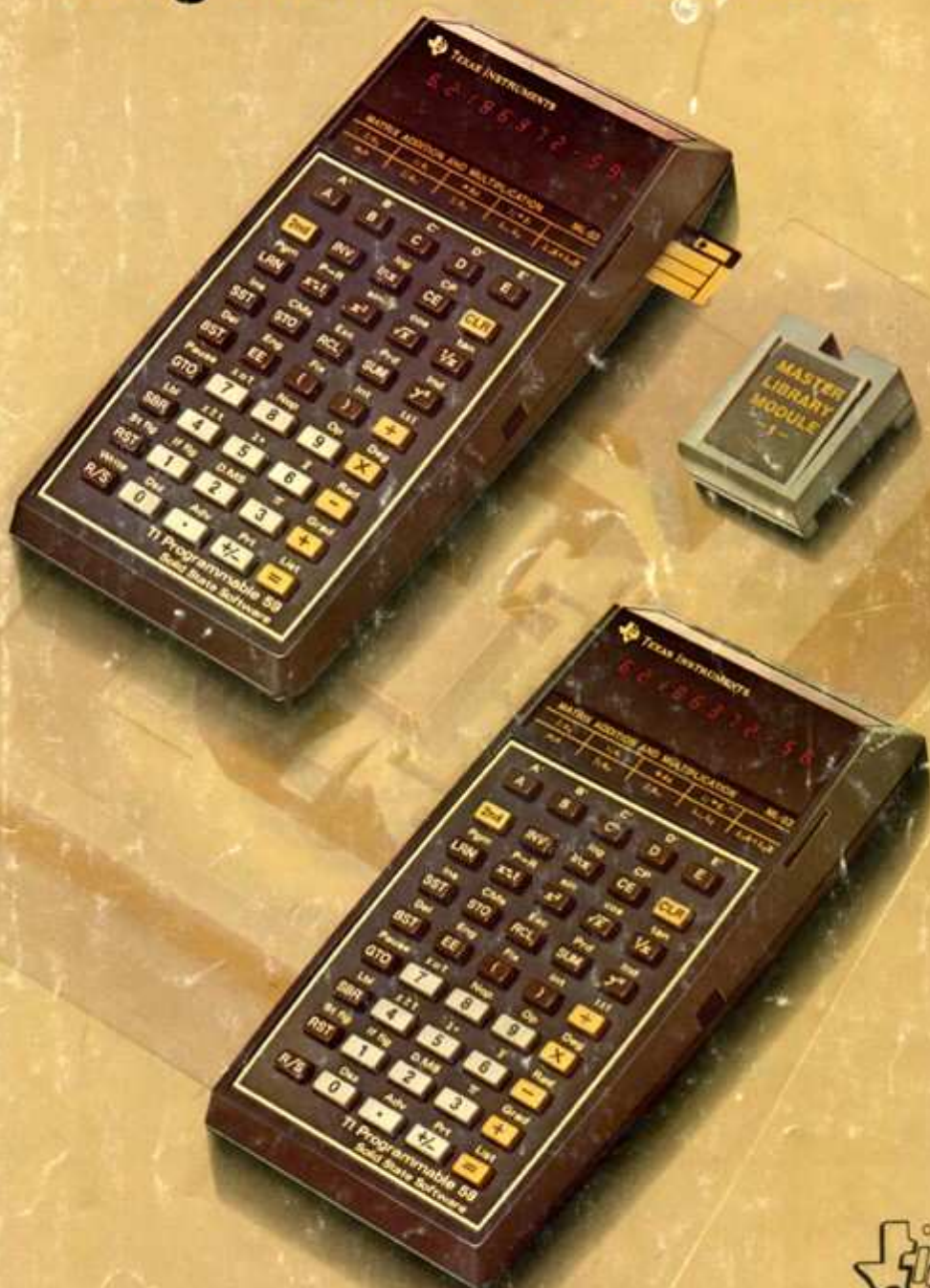


TI PROGRAMMABLES 58/58C/59

Programmer Soi-même



TEXAS INSTRUMENTS



INDEX DES TOUCHES

Cet index vous permet de trouver rapidement la page où sera développée la description de chaque touche.

F	V-55	P	V-55	C	V-55	D	V-55	E	V-55
A	V-55	B	V-55	C	V-55	D	V-55	E	V-55
2nd	V-3	INV	V-3	log	V-16	OP	V-3, 43	CLR	V-3
				Inx	V-16	CE	V-3		
Fn	III-1	P→R	V-30	int	V-17	inv	V-17	tan	V-17
LRN	V-43	x↔t	V-30	x²	V-20	√x	V-20	1/x	V-15
Ins	V-52	(M)	V-3, 23	Exp	V-26	Hz	V-24	ind	V-68
SST	V-48	STO	V-23	RCL	V-23	SUM	V-24	y^x	V-21
Del	V-51	log	V-8	Fix	V-8	log	V-20	 x 	V-20
BST	V-48	EE	V-5	(V-12)	V-12	+	V-10
Pause	V-44	x↔t	V-62	Nrc	V-51	0c	V-27	log	V-16
GTO	V-56	7	V-2	8	V-2	9	V-2	X	V-10
lbl	V-55	x↔t	V-62	Σ+	V-32	x̄	V-33	Rad	V-16
SBR	V-58	4	V-2	5	V-2	6	V-2	-	V-10
St/rg	V-65	llly	V-65	DMS	V-30	π	V-2	Inv	V-16
RST	V-44	1	V-2	2	V-2	3	V-2	+	V-10
Wide	VII-2	Off	V-63	Adv	VI-3	Fit	VI-2	Int	VI-4
R/S	V-43	0	V-2	*	V-2	+/-	V-2	≡	V-10

IMPORTANT

Inscrivez ici le n° de série de votre calculatrice ainsi que sa date d'achat. Le numéro de série est identifié au dos de la calculatrice par les mots «SERIAL NO». Prière de fournir ces informations dans toute correspondance.

TI-PROGRAMMABLE _____
Modèle N°

_____ N° de Série

_____ Date d'achat

TABLE DES MATIERES

Section	Page
I. INTRODUCTION	I-1
Introduction	I-1
Mise en Marche	I-2
Modes de Calcul	I-2
Utilisation d'un programme de la Bibliothèque	I-3
Calculs au clavier	I-4
Ecrire son propre programme - un Exemple	I-4
Possibilités d'impression	I-5
Structure de votre calculatrice	I-6
 II. APERCU GENERAL DES CARACTERISTIQUES ET DES FONCTIONS	 II-1
Fonctions élémentaires	II-2
Effacement de l'affichage - CE , CLR	II-2
Touche d'introduction des données - 0 - 9 - . - +/- - 77	II-2
Opérations de base - + , - , x , ÷ , =	II-2
Le système d'introduction AOS	II-3
Touches Parenthèses - ()	II-4
Touches à double fonctions - 2nd INV	II-5
Touches Mémoires - MC , STO , RCL , Exc	II-6
Opérations algébriques directes en mémoire - SUM Prd	II-7
Affichage	II-8
Affichage Standard	II-8
Touche de notation scientifique - EE	II-8
Touche de notation Ingénieur - Eng	II-9
Décimal fixe - Fix	II-9
Fonctions algébriques	II-10
Carré, Racine carrée, Inverse - x² , √x , 1/x	II-10
Puissances et racines - y^x	II-10
Logarithmes - lnx , log	II-11
Modes Angulaires - Deg , Rad , Grad	II-12
Trigonométrie - sin , cos , tan	II-12
Conversions	II-13
Conversions angulaires - D.MS	II-13
Conversions Polaires/Cartésiennes - P-R	II-14
Fonctions Statistiques	II-16
Moyenne, Variance, Ecart-type	II-16
Régression Linéaire	II-17
 III. UTILISATION DES SOLUTIONS PRE-PROGRAMMEES	 III-1
Bibliothèques de Programmes	III-1
Module préprogrammé enfichable	III-3
Analyse d'un programme de la Bibliothèque (Transfert en mémoire Programme)	III-4

TABLE DES MATIERES (Suite)

Section	Page
IV. LA PROGRAMMATION.	IV-1
Qu'est-ce que la Programmation ?	IV-1
Programmation élémentaire.	IV-2
Placer une variable dans un programme	IV-2
Mécanismes de la programmation	IV-10
Utilisation des touches utilisateur	IV-11
Adressage mémoire simplifié	IV-15
Introduction de votre programme	IV-16
Affichage en mode programmation	IV-17
Calcul du temps écoulé	IV-18
Correction d'un programme	IV-21
Modification du programme de calcul du temps écoulé.	IV-22
Correction d'adresses contractées	IV-26
Applications pratiques de programmes	IV-27
La Programmation est personnelle.	IV-27
Calcul d'un investissement.	IV-27
Contrôle de Prix	IV-32
Conversion de Coordonnées Sphériques	IV-38
Programmation avancée.	IV-43
Complément sur les étiquettes.	IV-43
Instructions de transfert	IV-43
Transferts inconditionnels	IV-44
Instruction GO TO	IV-44
Sous-Programmes	IV-46
Instruction de Sous-Programme — SBR	IV-46
Accéder à ou appeler un Sous-Programme	IV-48
Précautions à prendre dans un Sous-Programme	IV-49
Utilisation des bibliothèques de Prog. en sous-Programmes	IV-52
Biorythme.	IV-53
Transferts Conditionnels (Prise de décision)	IV-57
Registre d'affichage — Registre-T	IV-57
Exemple : Racine Carrée	IV-59
Drapeaux.	IV-61
Fonctions spéciales des Drapeaux	IV-65
Programme de Conversions Métriques	IV-65
Transferts Contrôlés par le Contenu d'une Mémoire	
DSZ	IV-68
Boucles	IV-68
Boucles inconditionnelles	IV-68
Boucles conditionnelles.	IV-70
Boucle contrôlée par l'instruction DSZ	IV-71
Programme de la fonction X!	IV-72
Exemples d'application complémentaires	IV-75
Calcul de la valeur actuelle d'une obligation	IV-75
Résolution de l'équation du second degré	IV-79
Techniques Complémentaires	IV-84
Adressage indirect	IV-84
Adressage indirect des mémoires	IV-84
Transfert indirect	IV-86
Autres caractéristiques	IV-87

TABLE DES MATIERES (suite)

Section	Page
Optimisation d'un programme.	IV-89
Simplifier l'emploi d'un programme	IV-89
Réduire le nombre de pas d'un programme	V-89
Programme de calcul de coût d'exploitation	IV-93
Techniques pour accélérer la vitesse d'exécution d'un programme	IV-98
Code secret (Programme de Jeu)	IV-101
V. Analyse détaillée des caractéristiques et des fonctions	V-1
Opérations de base	V-1
Affichage standard	V-1
Touches d'introduction de données.	V-2
Effacement des opérations	V-3
Touches à double fonction (2nd et INV)	V-3
Formats d'affichage.	V-5
Notation scientifique.	V-5
Notation ingénieur	V-8
Virgule fixe	V-8
Affichage clignotant	V-9
Calculs arithmétiques	V-10
Opérations de base— + , - , x , ÷ , =	V-10
Introduction à la notation algébrique directe	V-11
Parenthèses	V-12
Opérations factices avec les parenthèses	V-15
Fonctions algébriques	V-15
Inverse.	V-15
Logarithmes	V-16
Puissance de 10 et de e	V-16
Calculs angulaires	V-16
Modes angulaires	V-16
Fonctions trigonométriques	V-17
Inverse des fonctions trigonométriques.	V-18
Conversions degrés, radians, grades	V-19
Partie entière et valeur absolue	V-20
Carré et racine carrée.	V-20
Puissances et racines	V-21
Caractéristiques des mémoires.	V-22
Choix de la taille mémoire (partition)	V-22
Effacement des mémoires	V-23
Stockage et rappel des données.	V-23
Arithmétique en mémoire	V-24
Echange Affichage-Mémoire	V-26
Opérations spéciales	V-27
Possibilités d'impression Op 00-08	V-28
Analyse d'un programme de la bibliothèque (transfert est en mémoire prog.) Op 09 ..	V-28
Indicateur de signe Op 10	V-28
Statistiques Op 11-15.	V-28
Partition Op 16-17.	V-29
Indicateur d'erreur Op 18-19	V-29
Incrémentation-Décrémentation de mémoires Op 20-29/30-39	V-29
Test de présence de l'imprimante — Op 40 -- (TI-58C seulement)	V-29

TABLE DES MATIERES (suite)

Section

Page

Conversions et statistiques.....	V-30
Conversions.....	V-30
Conversions angulaires.....	V-30
Conversions de coordonnées polaires en cartésiennes.....	V-30
Statistiques.....	V-32
Introduction des données.....	V-32
Moyenne, variance et écart type.....	V-33
Régression linéaire.....	V-36
Analyse de tendance.....	V-39
Statistiques en cours de calcul.....	V-40
Programmation.....	V-41
Programmer votre calculatrice.....	V-41
Capacité de stockage et partition.....	V-42
Fonctions de base de programmation.....	V-43
Mode programmation.....	V-44
Introduction de votre programme.....	V-45
Exécution de votre programme.....	V-46
Relecture de votre programme.....	V-48
Codes des instructions (codes des touches).....	V-48
Stockage d'un groupe de touches.....	V-51
Correction d'un programme.....	V-51
Remplacement d'une instruction par une autre.....	V-52
Effacement d'une instruction.....	V-52
Insertion d'une instruction.....	V-52
Repérage d'une séquence de programme.....	V-55
Touches-utilisateur.....	V-55
Étiquettes ordinaires.....	V-56
Instructions de transfert.....	V-56
Instructions de transfert inconditionnel (GTO et SBR).....	V-56
Instruction GTO	V-56
Sous-programmes.....	V-58
Utilisation d'un programme de la bibliothèque en tant que sous-programme.....	V-60
Instructions de transfert conditionnel.....	V-62
Comparaison avec le contenu du registre T.....	V-62
Décrément et saut sur zéro.....	V-63
Drapeaux.....	V-65
Drapeaux et conditions d'erreur.....	V-67
Adressage indirect.....	V-68
VI. COMMANDE DE L'IMPRIMANTE.....	VI-1
Impression sélective.....	VI-2
Liste du programme.....	VI-4
Liste du contenu des mémoires.....	VI-4
Impression pas à pas de l'exécution d'un programme.....	VI-5
Opérations spéciales destinées à l'impression.....	VI-6
Impression alphanumérique.....	VI-7
Tracé d'une courbe par points.....	VI-10
Liste des étiquettes utilisées dans un programme.....	VI-11
Procédure de nettoyage de la tête d'écriture.....	VI-12

TABLE DES MATIERES (suite)

Section	Page
VII, CARTES MAGNETIQUES (TI-59 UNIQUEMENT)	VII-1
Enregistrement d'une carte	VII-2
Protection d'un programme	VII-4
Lecture d'une carte	VII-5
Lecture d'une carte au cours d'un programme	VII-5
Précaution à prendre concernant les cartes magnétiques	VII-7
Maniement des cartes magnétiques	VII-7
Nettoyage d'une carte	VII-8
Inscription sur une carte	VII-8
Utilisation de la carte de nettoyage de la tête magnétique	VII-8
Utilisation de la carte de nettoyage du galet d'entraînement	VII-8
Utilisation des cartes de diagnostic	VII-9
APPENDICE A – ENTRETIEN	A-1
Accumulateur et chargeur	A-1
En cas de difficultés	A-3
Vous avez des questions ou besoin d'aide	A-6
APPENDICE B – CONDITIONS D'ERREUR	B-1
Erreurs rencontrées en cours d'exécution de programme	B-2
APPENDICE C – PRECISION DES RESULTATS AFFICHES	C-1
APPENDICE D – DETECTION DES ANOMALIES D'UN PROGRAMME	D-1
Considérations de base	D-1
Diagnostic d'un programme	D-4



INTRODUCTION

Aujourd'hui les calculatrices de poche permettent d'aborder simplement le traitement des problèmes afférent à des domaines d'applications très différents. De nouvelles performances, une précision et une souplesse accrue sont mises à votre disposition pour répondre aux questions quotidiennes relatives à la numération, aux mathématiques tant dans les travaux de routine que pour les calculs les plus complexes.

À l'origine seules les opérations de base - addition, soustraction, multiplication, division - dotaient les calculatrices et à l'époque c'était déjà une révolution. Les calculatrices suivantes possédaient des fonctions mathématiques complexes : carré, racine, logarithmes, trigonométrie, etc.... Non seulement elles permettaient de supprimer les tables mais encore, elles apportaient une rapidité et une précision sans précédent.

Maintenant - une nouvelle dimension ! La calculatrice de poche avec la programmation entre dans un vaste domaine de résolution de nouveaux problèmes - domaine dans lequel seul un ordinateur pouvait jadis entrer. Ce manuel a été spécialement conçu pour vous aider à aborder et à maîtriser la programmation. Vous pourrez ainsi très vite apprécier la facilité avec laquelle vous maîtriserez la puissance de votre calculatrice programmable Texas Instruments.

Ce manuel est écrit pour les TI-58/58C et 59.

Ces calculatrices se distinguent par :

	Mémoires	Pas programmes	Fonctions spéciales
TI-58	Jusqu'à 60	Jusqu'à 480	
TI-58C	Jusqu'à 60	Jusqu'à 480	Mémoire permanente*
TI-59	Jusqu'à 100	Jusqu'à 960	Cartes magnétiques

La majorité des autres fonctions sont analogues sur les trois calculatrices. Lorsqu'il y aura une différence d'une calculatrice à l'autre celle-ci sera soulignée. Le manuel est organisé comme suit :

- Aperçu rapide sur la façon d'utiliser et de programmer la calculatrice.
- Descriptif des différentes fonctions de la calculatrice.
- Introduction à la programmation.
- Descriptif des fonctions de programmation de la calculatrice, assorti d'exemples issue de domaines d'applications variées.
- La dernière section de ce manuel illustre dans le détail les caractéristiques et les limites d'applications de la calculatrice dans différentes situations de calcul. (Si vous êtes déjà familiarisé avec ce type de calculatrice et la programmation et que vous désirez seulement connaître dans le détail ses caractéristiques, vous pouvez vous référer directement à cette section pour avoir une vision claire des caractéristiques de la calculatrice).

* Cette caractéristique assure la conservation des informations contenues en mémoire programme et en mémoire de données lorsque la calculatrice est éteinte ainsi que lorsque les batteries sont enlevées un court instant pour installer la calculatrice sur l'imprimante. (Voir chapitre VI, VII et Appendice A).



MISE EN MARCHÉ

Le bloc batterie fourni avec votre calculatrice est chargé à l'usine avant de vous être livré. Cependant, comme toutes batteries, celles-ci se déchargent d'elles-mêmes aussi est-il recommandé de mettre la calculatrice en charge avant usage. Si vous remarquez à l'utilisation que l'affichage faiblit ou qu'il devient fantaisiste c'est que les batteries ont besoin d'être rechargées. Il faut alors éteindre la machine, brancher le chargeur et attendre quelques minutes avant de la réutiliser. La calculatrice peut être utilisée même lorsqu'elle est en cours de recharge. On notera que la TI-58C possède une mémoire non volatile (Constant Memory™) qui conserve les instructions du programme et le contenu des mémoires si vous prenez soin d'éteindre votre calculatrice dès que les batteries de celle-ci donnent des signes de faiblesse (se reporter à l'appendice A). Du fait de la mémoire permanente, en rechargeant dès que possible les batteries vous pouvez sauvegarder les informations contenues dans votre calculatrice.

Placez l'interrupteur ON/OFF sur la position ON, vous voyez apparaître un zéro sur l'affichage qui vous indique que la calculatrice est chargée, prête à l'emploi. La mise en marche de la calculatrice vide automatiquement les TI-58 et TI-59. Seule la TI-58C qui dispose d'une mémoire non volatile gardera les instructions d'un programme, les données mises en mémoire, la partition et la décimalisation choisies telles qu'elles existaient lorsque la calculatrice fut mise sur "Arrêt". Pour vérifier l'affichage, appuyez sur le point décimal $\boxed{\cdot}$ puis sur le changement de signe $\boxed{+/-}$, introduire ensuite une série de huit. Le huit utilise tous les segments lumineux d'un caractère affiché. Vous pouvez introduire jusqu'à 10 caractères, que le nombre soit positif ou négatif, les caractères suivant le dixième chiffre sont simplement ignorés.

Chaque fois que vous dépasserez les limites de capacité de l'affichage ou que vous demanderez à la machine quelque chose qu'elle ne peut pas faire, la calculatrice vous l'indiquera par un affichage clignotant. Le clignotement est arrêté en appuyant sur la touche \boxed{CE} .

Nous allons effectuer une première présentation de votre machine, mais elle ne saurait se substituer à l'exploration que vous pourrez en faire par vous-même pour en devenir parfaitement maître. Ceci constitue une première approche vous permettant de connaître la souplesse et la puissance qui vous sont offertes. Mieux vous aurez assimilé l'ensemble des caractéristiques de votre calculatrice, plus elle vous rendra de services.

MODES DE CALCUL.

Votre machine peut travailler en trois modes de calcul différents :

Vous pouvez utiliser un des programmes du «Module Préprogrammé Enfichable» (Solid State Software™) contenu dans votre calculatrice pour résoudre des problèmes complexes par simple pression de quelques touches —OU—

Vous pouvez enseigner à votre calculatrice vos propres méthodes de résolution qu'elle peut conserver en mémoire pour une utilisation ultérieure chaque fois que vous en aurez besoin — OU —

Vous pouvez utiliser votre calculatrice à tout moment en tant que puissante calculatrice de poche prête immédiatement à résoudre aussi bien les calculs quotidiens simples que les résolutions mathématiques les plus complexes grâce à ses puissantes possibilités.



Utilisation d'un programme de la bibliothèque.

Sans savoir encore comment programmer vous-même votre calculatrice, vous pouvez néanmoins utiliser différents programmes très utiles. Une bibliothèque de base de programmes pré-enregistrés est contenue dans un petit module enfichable qui se trouve au dos de votre calculatrice. Ce module interchangeable (d'autres modules sont disponibles) contient une variété de programmes d'application générale décrits dans le manuel d'utilisation de la bibliothèque. Leur utilisation se fait à l'aide de la touche **Prog** qui permet de sélectionner l'un d'entre eux utilisable conformément aux indications données dans le manuel. Pour illustrer la simplicité de leur utilisation, prenons pour exemple le jeu du nombre mystérieux.

Le but de ce jeu est de découvrir un nombre secret le plus rapidement possible. La calculatrice choisit un nombre compris entre 1 et 1023 et à chaque tentative du joueur, elle donne une réponse pouvant être : «trop bas», «trop haut» ou «exact». Lorsque le nombre est découvert, la calculatrice vous indique votre résultat (nombre de tentatives). Suivez les instructions et jouez.

MODE D'EMPLOI				
Seq.	Marche à suivre	Introduire	Appuyer	Affichage
1	Sélectionner le programme		2nd Prog 21	
2	Composer un nombre quelconque entre 0 et 1	Votre nombre	A	Votre nombre
3	Générer le nombre secret		B	0.
4	Parier un nombre (1 à 1023) Réponse : -1 pari trop bas 1 pari trop haut clignotant 0 pari exact	Votre pari	C	Réponse
5	Reprendre la séquence 4 tant que le pari n'est pas exact.			
6	Afficher le résultat (nombre de tentatives).		D	Résultat

La plupart des programmes pré-enregistrés sont aussi simples à utiliser que celui-ci. Des dizaines de pas de programme ont été exécutés et ceci de façon entièrement automatique. Les seules manipulations que vous ayez à effectuer, consistent à introduire vos données et à démarrer le programme.

Un point important - La manière d'utiliser les programmes du «Module Pré-programmé Enfichable» est indiquée dans le manuel de la bibliothèque de base. Toutes les entrées, sorties ainsi que les détails dont vous pourriez avoir besoin sont indiqués dans ce manuel. Vous pouvez dès maintenant jeter un coup d'œil sur la bibliothèque de base et prendre connaissance des programmes-mis à votre disposition pour vous aider dans la résolution de vos problèmes.



Calculs au clavier.

Votre calculatrice est équipée de la notation algébrique directe (AOST^M) qui est une méthode d'introduction des différentes fonctions nécessaires à la résolution d'un problème. C'est l'une des méthodes les plus performantes développées actuellement. La résolution d'un problème s'introduit sur la calculatrice exactement dans le même ordre que celui de l'écriture, de la gauche vers la droite. Par exemple, pour convertir 100°C, 36°C, -4°C en degrés Fahrenheit, il faut multiplier les degrés Celsius par 9 diviser par 5 puis ajouter 32. $^{\circ}\text{F} = ^{\circ}\text{C} \times 9/5 + 32$.

Appuyer	Affichage
100 \times	100.
9 \div	900.
5 $+$	180.
32 $=$	212.

Vous pouvez refaire cette séquence d'opérations et trouver : $36^{\circ}\text{C} = 96.8^{\circ}\text{F}$ et $-4^{\circ}\text{C} = 24.8^{\circ}\text{F}$. (Des explications complémentaires sur l'AOS et la puissance de calcul qu'il représente vous seront données plus loin).

Ecrire son propre programme - Un exemple.

Dès qu'une séquence de calcul a été définie pour résoudre un problème et que vous avez à exécuter plusieurs fois cette séquence pour des valeurs différentes, vous pouvez appuyer sur la touche $\overline{\text{LRN}}$ (Learn = apprendre) et enseigner à la calculatrice votre méthode de calcul. Concernant l'exemple précédent appuyez sur $\overline{\text{LRN}}$ et introduisez la séquence de touches suivantes :

$\overline{\text{X}}$
9
 \div
5
 $+$
3 2
 $=$
 $\overline{\text{R/S}}$ (pour arrêter le programme et afficher le résultat)

Appuyer sur $\overline{\text{LRN}}$ à nouveau pour signifier à la calculatrice que «l'apprentissage» de cette séquence de touches est terminé. La calculatrice mémorise alors cette séquence et se trouve prête à exécuter une série de résolutions successives pour chaque valeur introduite (ici, degrés Celsius).



Votre machine est maintenant prête à faire des conversions de degrés Celsius en degrés Fahrenheit pour chaque nombre que vous introduisez.

- Composez sur le clavier votre valeur en Degrés Celsius.
- Appuyez sur **RST** (reset) pour demander à la calculatrice de se placer au début de la séquence mémorisée.
- Appuyez sur **R/S** (run/stop) pour démarrer l'exécution de cette séquence mémorisée (ou programme).

Appuyer	Affichage
100 RST R/S	212.
36 RST R/S	96.8
4 +/- RST R/S	24.8

Ce sont là les seules manipulations que vous ayez à faire pour convertir des degrés Celsius en degrés Fahrenheit. Écrire son propre programme est aussi simple que cela.

La possibilité que vous venez de voir - cette possibilité d'exécuter un programme que vous avez créé vous-même est l'un des aspects les plus performants de votre calculatrice. Dès qu'un programme est mémorisé et que vous avez vérifié son fonctionnement correct, vous pouvez l'utiliser et le réutiliser aussi souvent que vous le désirez, simplement en appuyant sur une touche.

POSSIBILITE D'IMPRESSION.

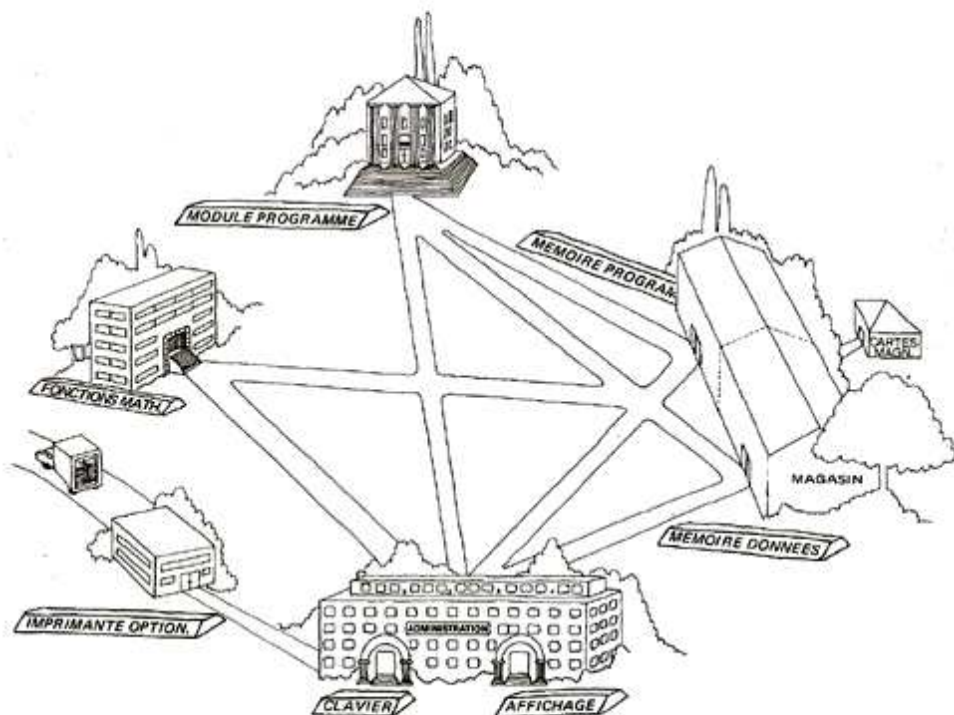
Votre calculatrice est compatible avec le berceau imprimant PC-100A, PC-100B ou PC-100C. Cette imprimante permet de retranscrire sur la bande de papier la valeur affichée, chaque fois que vous le désirez. Quand vous résolvez un problème directement au clavier, vous pouvez choisir d'imprimer tout ou partie des résultats intermédiaires ou encore demander l'édition automatique de l'ensemble des instructions de votre programme. En cours d'exécution du programme, les valeurs à l'affichage peuvent être automatiquement imprimées en intercalant l'instruction d'impression «Prt» (Print) dans le programme. Cette possibilité d'impression vous permet d'exécuter un programme donnant des réponses multiples, en évitant d'arrêter l'exécution de celui-ci pour lire chaque résultat affiché. Le mode «Trace» vous permettra d'obtenir l'exécution du programme en imprimant simultanément l'instruction exécutée à chaque pas ainsi que la valeur numérique correspondante.

Vous pouvez également en utilisant certains codes spéciaux, créer et imprimer un texte pour identifier votre bande ou pour identifier un résultat. Chaque ligne imprimée peut contenir jusqu'à 20 caractères choisis parmi les 64 disponibles.



STRUCTURE DE VOTRE CALCULATRICE.

Votre nouvelle calculatrice dispose d'une structure puissante et modulaire qui vous secondera en étant pour vous un outil très souple susceptible de faire face à toutes sortes de problèmes. Ce chapitre vous a permis de voir rapidement ses différentes possibilités d'utilisation, le chapitre suivant vous donnera un aperçu de la façon dont les différentes parties de la calculatrice, telles que schématisées ci-dessous travaillent ensemble avant d'aborder le vif du sujet c'est-à-dire l'art et la manière de programmer.



II

Aperçu général des caractéristiques et des fonctions



Avant d'aller plus loin dans les explications des caractéristiques les plus complexes de votre calculatrice, il est préférable de passer en revue les caractéristiques principales ainsi que les fonctions du clavier. Ceci est d'autant plus important si cette calculatrice constitue votre première expérience dans le domaine des calculatrices sophistiquées. De nombreux usagers n'utilisent pas toute la puissance de leur calculatrice, uniquement parce qu'ils n'ont pas pris le temps de se pencher sur les possibilités de chaque touche. Dans ce paragraphe, nous allons faire une description de ces différentes touches ce qui ne devrait pas prendre plus de 10 à 15 minutes de votre temps. Vous serez ainsi familiarisé avec les caractéristiques principales du clavier et pourrez par la suite aborder la programmation avec la possibilité de saisir l'ensemble des avantages que cette calculatrice vous offre.

Note aux différents utilisateurs :

Si vous êtes déjà familiarisé avec les calculatrices sophistiquées possédant la Notation Algébrique Directe (AOS), vous pouvez passer directement à la section IV concernant la programmation.

Pour avoir une description plus précise des différentes caractéristiques, référez-vous à la section V, qui vous fournira dans le détail une étude des différents éléments de la machine.

Pour aborder cette section ayez soin d'avoir votre calculatrice à portée de la main. N'hésitez pas à essayer chaque fonction ainsi que les différents exemples proposés. N'oubliez pas de la meilleure façon d'apprendre à utiliser votre calculatrice c'est de vous en servir.



II

FONCTIONS ELEMENTAIRES.

Effacement de l'affichage **CE**, **CLR**.

Il y a deux façons de procéder à l'effacement du registre d'affichage de votre calculatrice en fonction des éléments de votre problème.

CE Clear Entry - Effacement d'une donnée - Cette touche efface le dernier nombre que vous avez introduit sur l'affichage (à condition que vous n'ayez appuyé, ni sur une touche de fonction, ni sur une touche d'opération). L'utilisation de cette touche n'affecte pas les calculs en cours, ainsi, si vous avez appuyé sur 5 à la place d'un 6 en composant un nombre, il vous suffit d'appuyer sur **CE** et de recomposer votre nombre. La touche **CE** permet également d'arrêter le clignotement de l'affichage dû à une condition d'erreur.

CLR Clear - Effacement général - Cette touche efface le contenu de l'affichage ainsi que les différents calculs en cours, elle efface également une condition d'erreur.

Touches d'introduction des données - **0** - **9**, **.**, **+/-**, **TT**.

Les nombres sont introduits à l'aide des touches d'introduction de données suivantes : **0** - **9**, **.**, **+/-**. Lorsque vous comparez un nombre, le point décimal reste à droite des chiffres introduits jusqu'à ce que vous introduisez le point décimal. Tout chiffre introduit après le point décimal déplace celui-ci vers la gauche avec le nombre lui-même. Pour changer le signe d'un nombre affiché, il suffit d'appuyer sur la touche **+/-** (si vous appuyez à nouveau sur cette touche, vous changez à nouveau le signe du nombre).

En appuyant sur **2nd TT** on place les 10 premiers chiffres du nombre π sur l'affichage soit 3.141592654. 13 chiffres sont en fait contenus dans les registres internes soit 3.141592653590, la touche **CE** n'affecte pas cette introduction.

Opérations de base **+**, **-**, **X**, **÷**, **=**.

L'arithmétique de base est traitée avec les cinq opérations de base +, -, x, ÷, =. Votre calculatrice est dotée d'une caractéristique puissante nommée AOS; celle-ci est une méthode d'introduction permettant la résolution de problèmes à l'aide de séquences de touches particulièrement simples. Vous introduisez votre problème exactement dans l'ordre de l'écriture puis en appuyant sur **=** vous obtenez le résultat. L'avantage incomparable de l'AOS réside dans le fait que vous n'avez que votre problème à introduire et que la calculatrice d'elle-même s'organise pour agencer les différents calculs dans un ordre correct pour vous fournir le résultat. (Nous verrons l'AOS plus en détail page suivante).

Quand vous appuyez sur la touche **=**, toutes les opérations en attente (éléments mis en attente dans votre calculatrice) sont déclenchées. Vous obtenez le résultat et la calculatrice se trouve vidée prête à recevoir le problème suivant.

Exemple : Calculez $15 + 7 \times 31 - 4 = ?$

Appuyez : 15 **+** 7 **X** 31 **-** 4 **=** Affichage : 228

NOTE : On observera que l'AOS donne à la calculatrice l'instruction d'interpréter l'expression comme étant $15 + (7 \times 31) - 4$ dans laquelle 7×31 est calculé d'abord, puis 15 est ajouté et enfin 4 est retranché du résultat précédent.



Le système d'introduction AOS.

Les mathématiques sont une science rigoureuse assortie de règles strictes. L'une de ces règles stipule qu'il n'est pas possible d'obtenir deux résultats différents correspondant à une même séquence d'opération. En conséquence de cette règle - une seule réponse pour chaque calcul - les mathématiciens ont établi un ensemble de règles universellement reconnues dans le cas de suites d'opérations utilisées dans un calcul. Par exemple, prenons le problème suivant :

$$3 + 10 - 2 \times 14 \div 7 = ?$$

ce problème possède une seule et unique solution ! (La connaissez-vous? C'est 9).

Vous pouvez introduire ce problème directement, de la gauche vers la droite, dans votre calculatrice et vous obtiendrez le résultat correct. La notation algébrique utilisée par la calculatrice sélectionne les opérations que vous introduisez, ordonnent celles-ci dans l'ordre correct pour vous restituer le seul résultat possible. Regardons ce qui s'est passé, votre calculatrice a exécuté les opérations que vous lui avez indiquées suivant l'ordre universellement reconnu :

1. Fonctions à une variable - qui agissent sur le nombre affiché dès que vous appuyez sur la touche correspondante; nous ferons un descriptif plus détaillé par la suite de ces différentes fonctions: fonctions trigonométriques, logarithmiques et leurs inverses ainsi que les carrés, racines carrées, inverses).
2. Puissances et racines Nième (y^x et $\sqrt[x]{y}$) sont traitées ensuite (voir plus loin le descriptif de ces fonctions).
3. Multiplication et division viennent ensuite suivies des
4. Additions et soustractions.

Cette hiérarchie algébrique s'applique à l'intérieur de chaque niveau de parenthèses.

Enfin, la touche égale termine les opérations.

Certains problèmes nécessitent que ce soit vous qui décidiez de l'ordre dans lequel une expression doit être exécutée. Dans ce cas vous pouvez contrôler cet ordre en utilisant les parenthèses $\square(\square, \square)$ (nous en parlerons plus en détail par la suite). Les parenthèses en mathématique sous-entendent un ordre d'attente et de priorité dans une expression et c'est exactement ce qui sera effectué par votre calculatrice.



Touches Parenthèses ((,)) .

Dans certains problèmes vous pouvez être amené à modifier l'ordre exact dans lequel une expression doit être calculée, ou regrouper des expressions suivant les impératifs du problème. Les parenthèses vous permettent de réaliser l'isolement de ces nombres et de ces opérations. En plaçant une expression entre parenthèses, vous demandez à votre calculatrice «d'exécuter en priorité la résolution de cette expression jusqu'à obtention de la valeur numérique correspondante, laquelle sera ensuite utilisée pour la résolution du reste de votre problème». A l'intérieur d'une parenthèse, votre calculatrice respecte l'ordre de la hiérarchie algébrique. Vous pouvez également utiliser les parenthèses si vous avez un doute sur la façon dont la calculatrice va interpréter une expression. Votre calculatrice permet d'avoir jusqu'à 9 niveaux de parenthèses imbriquées gardant jusqu'à 8 opérations en attente. Ce qui suit est un exemple relatant cette possibilité.

$$(((2 \times (2 \times (2 \times (2 \times (2 \times (2 + 2y^x (2 + .2)) - (2 + 2)))))) \div 2) \div 2)$$

Lorsque vous introduisez cette séquence, vous pouvez remarquer qu'aucun calcul n'est effectué avant que la première fermeture de parenthèse ne soit introduite. Votre calculatrice garde en mémoire, toutes les expressions introduites pour les déclencher au fur et à mesure qu'elles doivent être exécutées.

Note : Une remarque importante pour l'utilisation des parenthèses. Vous rencontrerez certaines expressions écrites avec des parenthèses qui sous-entendent l'opérateur «multiplication» : $(2 + 1)(3 + 2) = 15$. Votre calculatrice ne peut pas interpréter l'opérateur sous-entendu, vous devez nécessairement indiquer celui-ci entre les fermetures et ouvertures de parenthèses :

$$((2 + 1) \times (3 + 2)) = 15.$$

Prenons un exemple utilisant des parenthèses :

$$\text{Calculer : } \frac{8 \times (4 + 9) + 1}{(3 + 6 \div 2) \times 7}$$

Pour un tel problème, vous désirez calculer le numérateur puis diviser celui-ci par le dénominateur. Pour être certain du déroulement des opérations, placez le numérateur entre parenthèses ainsi que le dénominateur lors de l'introduction des touches.

Appuyer	Affichage	Commentaires
CLR	0	Effacement des calculs en cours.
(8 X (4 + 9)	13.	(4 + 9) est calculé.
+ 1	104.	8 x (4 + 9) est calculé.
)	105.	Valeur du numérateur.
÷ ((3 + 6 ÷ 2)	6.	(3 + 6 ÷ 2) est calculé.
X 7)	42.	Valeur du dénominateur.
=	2.5	Résultat final.



Touches à double fonctions $\boxed{2nd}$, \boxed{INV} .

Votre calculatrice possède de nombreuses fonctions destinées à vous faire gagner du temps ainsi qu'à augmenter la puissance de la calculatrice. Pour accéder à la totalité de la puissance de votre calculatrice et pour éviter de multiplier le nombre de touches, certaines d'entre elles possèdent plus d'une fonction. La première fonction est inscrite directement sur la touche elle-même et pour utiliser cette fonction il suffit d'appuyer sur cette touche. Pour utiliser la deuxième fonction de la touche (inscrite juste au-dessus de celle-ci), il suffit d'appuyer sur la touche $\boxed{2nd}$ suivie de la touche placée juste sous l'inscription de la fonction.

Par exemple, pour obtenir le logarithme népérien d'un nombre, appuyez sur $\boxed{\ln x}$. Pour obtenir le logarithme décimal d'un nombre, appuyez sur $\boxed{2nd} \boxed{\ln x}$. Dans la suite de ce manuel, la seconde fonction, d'une touche sera notée $\boxed{2nd} \boxed{\text{fonction}}$; d'une façon générale la première fonction sera indiquée par $\boxed{\text{fonction}}$ et la deuxième fonction par $\boxed{2nd} \boxed{\text{fonction}}$.

La touche inverse \boxed{INV} augmente les possibilités de la calculatrice sans pour autant augmenter le nombre de touches du clavier. Lorsque vous appuyez sur la touche \boxed{INV} avant d'appuyer sur une touche de fonction l'action de cette fonction se trouve inversée et vous obtenez la fonction inverse. Cette touche \boxed{INV} opère sur un très grand nombre de touches.

Les touches $\boxed{2nd}$ et \boxed{INV} vous permettent de disposer de 108 opérations différentes uniquement à partir des 45 touches du clavier. Un descriptif plus détaillé sera fait dans le chapitre **Touches à double fonctions** dans la section V de ce manuel.



Touches Mémoires - **MC**, **STO**, **RCL**, **Exc**.

A la mise en marche de votre calculatrice, 60 mémoires sont disponibles sur le TI-59, 30 pour la TI-58. Concernant la TI-58C le nombre de mémoires sera celui défini avant l'arrêt de la calculatrice. Le nombre de ses mémoires est modifiable en fonction de la place désirée pour le programme. (Voir Ajustement du nombre de mémoires à la section V pour les détails supplémentaires). Les mémoires constituent des emplacements spéciaux dans la calculatrice où vous allez pouvoir mettre un nombre en attente pour pouvoir le réutiliser plus tard. La TI-58C conservera les données mises en mémoire même si la calculatrice est momentanément arrêtée.

Dû au fait que vous avez plus d'une mémoire à votre disposition, vous devrez indiquer par un nombre à deux caractères XX, laquelle des mémoires vous utilisez, par exemple **STO** 08.

Les touches **CE** et **CLR** n'ont pas d'effet sur les mémoires. Cependant vous pouvez effacer l'ensemble de ces mémoires (mettre un 0 dans chaque mémoire) en appuyant sur **2nd** **MC**.

STO XX - STOCKAGE - Cette séquence stocke le nombre affiché dans la mémoire XX (00 à 99) sans modifier le contenu de l'affichage. Tout nombre se trouvant dans cette mémoire se trouve simultanément chassé et remplacé par le nombre contenu dans l'affichage.

RCL XX - RAPPEL - Cette instruction ramène sur l'affichage, le nombre contenu dans la mémoire XX. L'utilisation de cette instruction ne modifie pas le contenu de la mémoire XX.

2nd **Exc** XX - ECHANGE AFFICHAGE MEMOIRE - Cette instruction va simultanément mettre le contenu de l'affichage dans la mémoire XX et ramener le contenu de la mémoire XX dans l'affichage. Une utilisation possible consiste à vérifier le contenu d'une mémoire sans pour autant perdre un résultat affiché.

Exemple : Stocker et rappeler 3.21

Appuyer	Affichage	Commentaires
3.21 STO 08	3.21	Place 3.21 en mémoire 8
CLR	0	Efface l'affichage.
RCL 08	3.21	Rappelle le contenu de la mémoire 8



Exemple : Calculer $(A + 2) + A(A + 2)$ pour $A = 9.3069128$.

Appuyer	Affichage	Commentaires
CLR	0	Effacement des calculs précédents.
9.3069128 STO 12	9.3069128	Stocke A dans la mémoire 12
+ 2 +	11.3069128	Calcule $A + 2$
2nd 1/x 12	9.3069128	Stocke $A + 2$ dans la mémoire 12 et rappelle A sur l'affichage
X RCL 12	11.3069128	Rappelle $A + 2$ sur l'affichage (Notez que la touche X est placée entre A et $A + 2$)
=	116.5393643	Achève l'exécution des opérations en attente et fournit le résultat final.

On notera que la valeur de A n'est introduite qu'une seule fois, ce qui permet de gagner du temps et évite des erreurs possibles.

Opérations algébriques directes en mémoire **SUM**, **Prd**.

Vous avez la possibilité d'effectuer des opérations directement sur le contenu d'une mémoire sans affecter les calculs en cours.

SUM **XX** - SOMMATION EN MEMOIRE - Cette touche vous permet de faire le cumul algébrique du contenu de l'affichage, au contenu de la mémoire **XX**. L'affichage reste inchangé, le résultat de la sommation se trouve en mémoire **XX**. D'une façon analogue, vous pouvez exécuter la séquence suivante **INV** **SUM** **XX** qui aura pour effet de soustraire du contenu de la mémoire **XX**, le contenu de l'affichage.

2nd **Prd** **XX** - PRODUIT EN MEMOIRE - Cette séquence a pour effet d'exécuter le produit du contenu de l'affichage par le contenu de la mémoire **XX**, tout comme **INV** **2nd** **Prd** **XX** exécutera la division du contenu de la mémoire **XX** par le contenu de l'affichage. Dans chaque cas, le contenu de l'affichage reste inchangé. Le résultat de l'opération se trouve en mémoire **XX**.

Ces instructions sont identiques aux quatre opérations de base du clavier et permettent de conduire des calculs sur les mémoires sans affecter le contenu de l'affichage.

Exemple : Calculer le coût total de 2 articles de 28 Frs et 6.60 Frs compte tenu d'une commission de 5%.

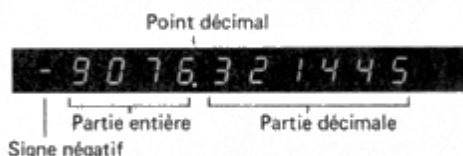
Appuyer	Affichage	Commentaires
28 STO 01	28.	Stocke 28 en mémoire 1.
6.6 SUM 01	6.6.	Additionne 6.6 dans la mémoire 1.
1.05 2nd Prd 01	1.05	Multiplie le contenu de la mémoire 1 par 1.05
RCL 01	36.33	Montant total.



AFFICHAGE

Affichage standard.

L'affichage reçoit les informations numériques complètes avec le signe négatif et le point décimal. Il clignote chaque fois qu'une condition d'erreur ou un excès de capacité aura été généré. (La liste des différentes conditions d'erreurs sera développée dans l'annexe B). Vous pouvez introduire jusqu'à 10 caractères, tout caractère supplémentaire sera simplement ignoré.



Une distinction est faite entre l'affichage et le registre d'affichage : l'affichage est limité au nombre que vous voyez physiquement; le registre d'affichage est un registre interne dans lequel les résultats sont exprimés avec 13 caractères.

Si un résultat est trop petit ou trop grand pour être contenu dans l'affichage, celui-ci est automatiquement converti en notation scientifique; par exemple, si vous multipliez 400.000 par 2.000.000 vous obtenez 800.000.000.000 qui ne peut être représenté tel quel sur l'affichage. Il sera automatiquement converti en $8. \times 10^{11}$.



Touche notation scientifique - \boxed{EE} .

Dans certaines applications, scientifiques notamment, vous pouvez avoir besoin d'utiliser des nombres très petits et très grands. Ces nombres peuvent facilement être introduit avec la notation scientifique.

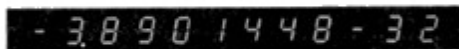
L'expression du nombre contient alors deux parties : la mantisse (ou nombre de base) multipliée par 10 élevé à une certaine puissance ou exposant.

$$\text{Nombre} = \text{mantisse} \times 10^{\text{EXPOSANT}}$$

En pratique, vous devez

- Introduire la mantisse jusqu'à 8 caractères (si besoin utilisez $\boxed{+/-}$).
- Appuyer sur \boxed{EE} (Entrée de l'exposant) - «00» apparaît à droite de l'affichage.
- Introduire la puissance de 10 (si besoin utilisez $\boxed{+/-}$).

Un nombre de la forme $-3.8901448 \times 10^{-32}$ s'écrit ainsi que l'affichage:



En notation scientifique, la puissance de dix indique où devrait se trouver le point décimal si le nombre était écrit en notation ordinaire.

II



Un exposant positif correspond à un décalage du point décimal vers la droite d'un nombre de caractères égal à la valeur de l'exposant; un exposant négatif correspond à un décalage vers la gauche.

Exemple : $2.9979 \times 10^{11} = 299\,790\,000\,000$

Le point décimal est décalé de 11 caractères vers la droite et autant de zéros que nécessaire sont ajoutés.

$1.6021 \times 10^{-9} = 00000000016021$

Le point décimal est décalé de 9 caractères vers la gauche et autant de zéros que nécessaire sont ajoutés.

Lorsque vous mettez la calculatrice en notation scientifique, elle conserve ce mode d'écriture tant que vous ne lui avez pas donné un ordre contraire; la séquence **INV** **EE** replace la calculatrice en notation standard. La touche **CLR** remet également la calculatrice en notation standard.

Touche notation ingénieur **Eng**.

La notation ingénieur est une forme modifiée de la notation scientifique. L'exposant est ajusté pour fournir des multiples de trois (10^{12} , 10^{-9} par exemple). La position du point décimal de la mantisse est modifiée en conséquence donnant un, deux ou trois caractères à gauche du point décimal. Cette notation correspond à des unités souvent employées dans les domaines scientifiques et techniques, par exemple : 10^{-12} pour des picofarads, 10^{-3} pour des millimètres, 10^3 pour des kilogrammes... **CLR** n'efface pas la notation ingénieur.

L'affichage peut être mis en notation ingénieur à tout moment, en appuyant sur **2nd** **Eng** **INV** **2nd** **Eng** remet la calculatrice en notation standard.

Exemple : Calculer $8 \times 98 \times 30$ en notation ingénieur

Appuyer	Affichage
CLR 2nd Eng	0. 00
8 X 98 X	784. 00
30 =	23.52 03
INV 2nd Eng	23520.

Décimal fixe **Fix**.

Vous avez la possibilité de choisir le nombre de caractères affichés après le point décimal en appuyant sur **2nd** **Fix** suivie du nombre de décimales désirées (0 à 8). La calculatrice arrondit tous les résultats à la décimale désirée sans modifier le contenu du registre d'affichage. Vous gardez la possibilité d'introduire des nombres avec autant de caractères que vous désirez, la calculatrice continue à travailler avec ses 13 chiffres. La **TI-58C** conservera la sélection décimale choisie même si la calculatrice est éteinte.

La séquence **INV** **2nd** **Fix** supprime la virgule fixe pour revenir en virgule flottante.

Exemple : $2 \div 3 = .666666667$

Appuyer	Affichage
CLR 2 ÷ 3 =	.666666667
2nd Fix 6	0.666667
2nd Fix 2	0.67
2nd Fix 0	1.
INV 2nd Fix	.666666667



FONCTIONS ALGÈBRIQUES

Carré, Racine Carrée, Inverse - x^2 \sqrt{x} $1/x$.

Ce sont trois fonctions simples souvent utilisées dans de nombreuses applications. Chacune de ces trois fonctions opère directement sur le contenu du registre de l'affichage, sans affecter les autres calculs en cours.

x^2 - CARRE - Calcule le carré du nombre x contenu dans le registre d'affichage.

\sqrt{x} - RACINE CARRÉE - Calcule la racine carrée du nombre x contenu dans le registre d'affichage.

$1/x$ - INVERSE - Divise 1 par le contenu du registre d'affichage x.

Exemple utilisant ces différentes fonctions : $\sqrt{4} \div (1/5)^2 = 50$

Appuyer	Affichage	Commentaires
CLR	0	Efface les calculs précédents.
4 \sqrt{x}	2.	$\sqrt{4}$
\div 5 $1/x$	0.2	1/5
x^2	0.04	$(1/5)^2$
$=$	50.	Résultat

Puissances et Racines y^x .

Cette touche vous permet d'obtenir aussi bien la puissance nième d'un nombre positif, que la racine nième d'un nombre positif.

Puissance (y^x)

- Introduisez le nombre y que vous voulez élever à une puissance nième.
- Appuyez sur y^x .
- Introduisez la puissance x.
- Appuyez sur $=$ (ou sur une touche d'opération).

Exemple : 2^6

Racine ($\sqrt[x]{y}$)

- Introduisez le nombre y dont vous voulez prendre la racine nième.
- Appuyez sur INV y^x .
- Introduisez la racine x.
- Appuyez sur $=$ (ou sur une touche d'opération).

Exemple : $\sqrt[6]{64}$

Appuyer	Affichage	Appuyer	Affichage
CLR	0	CLR	0
2 y^x 6 $=$	64.	64 INV y^x 6 $=$	2.

NOTE : y ne peut être que positif, l'affichage clignotera si vous introduisez une valeur de y négative.

II



Logarithmes - $\ln x$, \log .

Les logarithmes sont des fonctions mathématiques que l'on rencontre dans les applications de calculs techniques ou théoriques. Par définition, si $x = y^2$, alors $\ln x$ (de base y) = 2. Les deux touches que nous allons voir vous donnent immédiatement le logarithme d'un nombre positif sans interrompre les calculs en cours et évitent les reports fastidieux aux tables numériques.

$\ln x$ - LOGARITHME NEPERIEN - Vous donne le logarithme népérien (base $e = 2.718281828459$) du nombre contenu dans le registre d'affichage. (L'affichage clignotera si ce nombre est négatif ou nul). L'antilog du logarithme népérien (e^x) s'obtient en faisant $\text{INV} \ln x$.

$2^{\text{nd}} \log$ - LOGARITHME DECIMAL - Vous donne le logarithme décimal (ou petit log) (base 10) du nombre contenu dans le registre d'affichage. Ce nombre doit également être positif. L'antilog du logarithme décimal (10^x) s'obtient en faisant $\text{INV} 2^{\text{nd}} \log$.

Exemple : Calculer le logarithme népérien de $(e^{2.7} + 10^{1.2})$.

Appuyer	Affichage	Commentaires
CLR	0	Efface les calculs précédents.
$($ 2.7 $\text{INV} \ln x$	14.87973172	Calcule $e^{2.7}$
$+$ 1.2 $\text{INV} 2^{\text{nd}} \log$	15.84893192	Calcule $10^{1.2}$
$)$	30.72866365	Effectue l'addition qui était en attente.
$\ln x$	3.425195888	Résultat



Modes Angulaires

Votre calculatrice peut résoudre les calculs nécessitant l'utilisation de valeurs angulaires, notamment les fonctions trigonométriques et les conversions de coordonnées polaires en coordonnées cartésiennes. Les angles peuvent être mesurés en degrés, radians ou grades; lorsque vous mettez en marche votre calculatrice, elle est automatiquement en degrés, vous pouvez néanmoins choisir votre mode angulaire par les séquences suivantes :

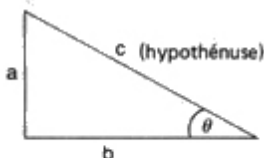
2nd **Deg** - MODE DEGRES - Toutes les introductions et les résultats angulaires sont exprimés en degrés tant qu'un autre mode angulaire n'est pas sélectionné. (Un degré = $1/360$ du cercle; un angle droit = 90°).

2nd **Rad** - MODE RADIAN - Les valeurs angulaires sont exprimées en radians. (Un radian = $1/(2\pi)$ du cercle; un angle droit = $\pi/2$ radians).

2nd **Grad** - MODE GRADE - Les valeurs angulaires sont exprimées en grades (Un grade = $1/400$ du cercle; un angle droit = 100 grades).

Trigonométrie **sin** **cos** **tan**

Ces fonctions donnent immédiatement la valeur des sinus, cosinus, tangente de la valeur d'un angle contenu dans le registre d'affichage. L'angle est mesuré suivant le mode angulaire choisi.



$$\sin \theta = \frac{a}{c}$$

$$\cos \theta = \frac{b}{c}$$

$$\tan \theta = \frac{a}{b}$$

dans lequel a, b et c sont les côtés du triangle rectangle.

Les séquences **INV**, **sin**, **INV**, **cos**, **INV**, **tan** sont utilisées pour calculer l'inverse de ces fonctions (fonctions «Arc...»); les résultats sont alors exprimés suivant le mode angulaire choisi.

NOTE : En mode degrés, les angles sont exprimés en degrés décimaux. (Voir conversions de degrés sexagésimaux en degrés décimaux page suivante).



CONVERSIONS

Conversions Angulaires **0 MS**

Il y a deux façons d'écrire un angle en degrés :

Une manière est d'utiliser les degrés décimaux **DDD.dd** dans lequel **DDD** représente le nombre de degrés et **dd** la partie décimale (jusqu'à 10 caractères possibles).

L'autre façon est d'utiliser la forme degrés, minutes, secondes **DDD.MMSSsss** dans lequel **DDD** représente le nombre de degrés, **MM** le nombre de minutes, **SS** le nombre de secondes, **sss** les centièmes de seconde. Remarquez que le point décimal est placé entre les degrés et les minutes.

Pour convertir des degrés, minutes, secondes en degrés décimaux, introduisez la valeur de l'angle sur l'affichage **DDD.MMSSsss** et appuyez sur **[2nd] [0 MS]** ; en utilisant la séquence **[INV] [2nd] [0 MS]** vous effectuerez la conversion inverse degrés décimaux en degrés, minutes, secondes.

L'écriture suivant le format **DDD.MMSSsss** nécessite obligatoirement l'introduction d'un zéro complémentaire dans le cas où le nombre de minutes (ou secondes) est inférieur à 10; voyez l'exemple ci-dessous.

Exemple : Convertissez $54^{\circ}2'9''$, 6 en degrés décimaux et réciproquement.

Appuyer	Affichage	Commentaires
54.02096 [2nd] [0 MS]	54.036	DD.ddd
[INV] [2nd] [0 MS]	54.02096	DD.MMSSs

Le processus décrit est valable également pour les conversions d'heures, minutes, secondes en heures décimales.

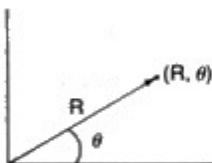


Conversion de coordonnées polaires en coordonnées cartésiennes $\boxed{P \rightarrow R}$

Ceci est une application spécialement utilisée dans les calculs scientifiques. Cette conversion utilise la touche $\boxed{\text{x} \rightarrow \text{t}}$, elle est rapide et facile à utiliser dans un sens comme dans l'autre, en suivant simplement les indications qui suivent :

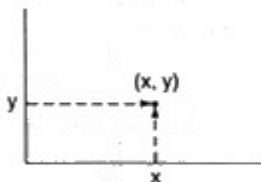
Coordonnées Polaires

DE :



Coordonnées Cartésiennes

EN :



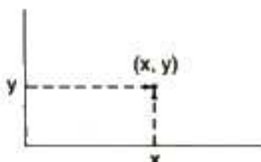
Pour convertir de coordonnées polaires en coordonnées cartésiennes :

- Introduisez la valeur de «R»
- Appuyez sur $\boxed{\text{x} \rightarrow \text{t}}$.
- Introduisez la valeur de « θ » (vérifier le mode angulaire)
- Appuyez sur $\boxed{2\text{nd}} \boxed{P \rightarrow R}$ pour obtenir la valeur de «y»
- Appuyez sur $\boxed{\text{x} \rightarrow \text{t}}$ pour obtenir la valeur de «x»



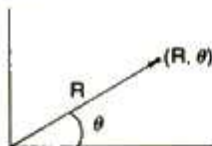
Coordonnées cartésiennes

DE :



Coordonnées polaires

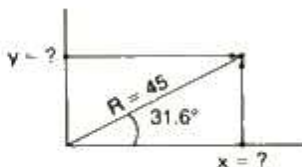
EN :



Pour convertir de coordonnées cartésiennes en coordonnées polaires :

- Introduisez la valeur de «x».
- Appuyez sur $\boxed{x:t}$.
- Introduisez la valeur de «y».
- Appuyez sur \boxed{INV} $\boxed{2nd}$ $\boxed{P \rightarrow R}$ pour afficher la valeur de « θ » exprimée suivant le mode angulaire choisi.
- Appuyez sur $\boxed{x:t}$ pour afficher «R».

Exemple :



Convertissez $R = 45$ mètres, $\theta = 31,6^\circ$ en coordonnées cartésiennes.

Appuyer

 \boxed{CLR} $\boxed{2nd}$ \boxed{Deg}

Affichage

0

45 $\boxed{x:t}$

0.

31.6

31.6

 $\boxed{2nd}$ $\boxed{P \rightarrow R}$

23.57936577

 $\boxed{x:t}$

38.32771204

Commentaires

Efface les calculs précédents et choisi le mode degrés.

Place R dans le registre T*

Introduit « θ » en degrés

Exécute la conversion et affiche y

Affiche x (y se trouve dans le registre T).

* NOTE : Cette conversion utilise un registre particulier T accessible par la touche $\boxed{x:t}$ (x échange t). Nous la reverrons plus loin en détail.



FONCTIONS STATISTIQUES

Moyenne, Variance et Ecart type.

Il peut vous arriver d'avoir à manipuler des ensembles de nombres importants : (ces valeurs peuvent être des résultats d'expériences, des volumes de ventes, etc...). L'ensemble de ces éléments peut être regroupé et synthétisé par quelques nombres représentatifs de l'ensemble; ce sont la moyenne, la variance et l'écart type. La moyenne est une mesure de la valeur moyenne de l'ensemble des éléments. La variance et l'écart type vous donnent une idée sur la dispersion des éléments et une mesure de l'éloignement des éléments par rapport à la valeur moyenne.

Reportez-vous à la partie traitant des statistiques à la section V pour apprendre en détail la façon d'utiliser ces fonctions.

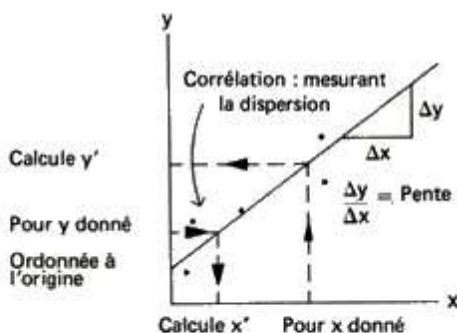


Régression linéaire.

Sous un nom bien mystérieux, la régression linéaire est très simple à utiliser et elle représente une des préoccupations les plus anciennes au monde : tenter de prédire l'avenir.

Dans un problème de régression linéaire, vous vous trouvez devant un certain nombre de paramètres matérialisés sur un graphique par des points représentés par deux valeurs, le plus souvent x et y (x représentant par exemple le nombre de francs investis en publicité et y , le volume des ventes ou encore x un nombre d'essais et y le nombre de coups au but, etc...). Vous désirez savoir pour une valeur de x déterminée qu'elle sera la valeur de y correspondante ou inversement ? Votre calculatrice va pouvoir vous fournir par raisonnement mathématique la « droite de régression » et la valeur que vous cherchez. Cette droite de régression permet de faire des extrapolations.

Dès l'introduction des paramètres, la calculatrice détermine la droite de régression et peut vous donner immédiatement la valeur que vous cherchez ainsi que les éléments suivants :



Les statistiques peuvent nécessiter l'usage de régression non linéaire, la calculatrice permet également de faire des extrapolations suivant des courbes exponentielles par exemple.

L'utilisation de ces possibilités sera développée plus en détail au chapitre « Statistiques » de la section V.



Utilisation des solutions pré-programmées.

Le terme «Logiciel» vous est peut-être familier - différentes définitions lui ont été données. A l'origine, il s'applique aux instructions et aux programmes - c'est-à-dire une suite d'ordres donnés à un ordinateur simplement sur un papier et pour lui indiquer quoi faire et à un utilisateur comme vous pour lui indiquer comment s'en servir. Votre calculatrice dispose d'un assortiment de programmes extrêmement puissants, d'usage simple, qui ont été mis dans la calculatrice prêts à être exploités par simple enfoncement d'une touche. Ces programmes ont été écrits pour répondre aux besoins de l'utilisateur dans différents domaines; ils sont stockés dans un module spécial situé au dos de votre machine. Ce module peut facilement être interchangé avec un module différent. Toutes les instructions de ces différents programmes sont contenues dans un minuscule circuit au silicium identique, de conception, aux circuits intégrés au silicium qui sont le cœur et le cerveau de votre calculatrice. C'est ce que nous appelons un «Module Pré-programmé Enfichable». De nombreux «Logiciels» d'usage courant sont enregistrés dans votre «Module Pré-programmé Enfichable». Les avantages principaux en sont :

- Une grande capacité de programme sous un faible volume - facile à transporter et à utiliser.
- Une bibliothèque de programmes spécialement écrite pour être simple à utiliser même par un utilisateur débutant.

Pour augmenter l'efficacité de chacun des modules, ceux-ci sont décrits dans un manuel donnant toutes les informations nécessaires à l'utilisation des programmes qu'ils contiennent.

BIBLIOTHEQUES DE PROGRAMMES.

La bibliothèque de base, livrée avec votre calculatrice, regroupe des programmes d'utilisation très générale. Vous pouvez obtenir d'autres bibliothèques, auprès de votre revendeur habituel. Chaque bibliothèque contient une sélection de programmes permettant d'utiliser simplement la puissance des solutions mathématiques appliquées à différents domaines. Chaque bibliothèque comprend un «module pré-programmé enfichable», un manuel explicatif détaillant chaque programme, une boîte de rangement et un certain nombre de cartes d'utilisation de ces programmes. Les modules préprogrammés enfichables sont compatibles avec les TI-58, TI-58C et TI-59.

MODULE PRE-PROGRAMME ENFICHABLE.

Les programmes de chaque bibliothèque sont enregistrés dans un module pré-programmé. Un module peut facilement être mis en place au dos de la calculatrice et être immédiatement utilisé. Ces modules sont durables mais doivent être manipulés avec précaution pour éviter toute détérioration.

ATTENTION

Assurez-vous que vous n'êtes pas chargé d'électricité statique avant de toucher un module.

Ceci est également vrai, lorsque le chargeur est branché. Il suffit de toucher un objet métallique pour vous assurer que vous êtes déchargé de toute électricité statique. Le contenu d'un module peut être sévèrement endommagé par une décharge d'électricité statique. Voir Appendice A, pour plus de détails concernant les précautions à prendre.



Le Module de la Bibliothèque de base installé dans votre calculatrice en usine, peut facilement être enlevé et remplacé par un autre. Une bonne précaution à prendre est de ne pas le sortir sauf pour le remplacer par un autre module. Assurez-vous de bien suivre ces conseils pour effectuer le remplacement d'un module par un autre.

1. Eteindre la calculatrice, interrupteur sur OFF. La mise en place ou l'enlèvement du module lorsque la calculatrice est en marche, peut créer des courants parasites qui pourraient endommager le module et/ou la calculatrice elle-même.
2. Retirer le couvercle en plastique recouvrant le module au dos de la calculatrice (voir figure ci-dessous). A nouveau, assurez-vous que vous n'êtes pas chargé d'électricité statique avant de prendre le module.
3. Enlever le module en retournant la calculatrice, le module tombera dans votre main de lui-même.
4. Placer le nouveau module en ayant soin de respecter le sens indiqué de telle sorte que la fente prévue sur le module soit en face de l'ergot prévu sur le boîtier de la calculatrice. Le module doit se mettre en place de lui-même sans forcer.
5. Remettre le couvercle en plastique qui assurera un contact correct du module dans la calculatrice.



Eviter toutes actions pouvant plier, souiller ou causer une détérioration quelconque des contacts.



Utilisation de la Bibliothèque de Programmes.

Le module pré-programmé étant dans la calculatrice, on peut appeler un programme particulier simplement en le désignant par son numéro (chaque programme est identifié dans la bibliothèque par son propre numéro) La séquence utilisée est $\boxed{2nd} \boxed{Pgm} \text{ mm}$ où mm sont les deux caractères d'identification du programme que vous voulez utiliser.

Veillez à suivre avec soin les instructions mentionnées dans le manuel de chaque bibliothèque. Une carte d'utilisation de programme est prévue pour chaque programme. Ces cartes qui ne sont pas magnétisées, servent à identifier les différentes touches utilisateur; elles sont prévues pour être introduites par la fente située sur le côté droit de la calculatrice, fente supérieure seulement.

Exemple : Quelle sera la valeur future d'un montant de 1 000 Frs après 20 ans à 8% de taux d'intérêt annuel, intérêts composés?

Le programme Intérêts Composés de la bibliothèque de base peut résoudre ce problème :

Appuyer	Affichage	Commentaires
$\boxed{CLR} \boxed{2nd} \boxed{Pgm} \boxed{18}$	0.	Sélectionne le programme d'intérêts composés (PGM 18)
$\boxed{2nd} \boxed{f}$	0.00	Initialisation
$\boxed{20} \boxed{A}$	20.00	Introduit le nombre de périodes
$\boxed{8} \boxed{B}$	8.00	Introduit le taux d'intérêts
$\boxed{1000} \boxed{C}$	1000.00	Introduit la valeur présente
$\boxed{0} \boxed{D}$	4660.96	Calcule la valeur future

Ce programme est réutilisable autant de fois que vous le désirez sans avoir à rappeler le numéro du programme à chaque utilisation. Appuyez sur \boxed{RST} ou $\boxed{2nd} \boxed{Pgm} \boxed{00}$ pour revenir au programme utilisateur ou sur $\boxed{Pgm} \text{ mm}$ pour rappeler un autre programme de la bibliothèque. (Par la suite nous distinguerons un «programme utilisateur» d'un «programme» issu d'une bibliothèque; le premier étant conçu et introduit par vous, instruction par instruction, le second étant pré-enregistré dans un module).

Un programme d'une bibliothèque peut également être appelé à partir d'un programme utilisateur. Cette caractéristique qui permet d'augmenter considérablement les possibilités de votre calculatrice sera expliquée en détail à la section IV dans le chapitre concernant les **Sous programmes**.

Pour retrouver l'identification du module placé dans votre calculatrice, il vous suffit d'exécuter la séquence suivante $\boxed{2nd} \boxed{Pgm} \boxed{1} \boxed{SBR} \boxed{2nd} \boxed{Write} (\boxed{2nd} \boxed{Pgm} \boxed{1} \boxed{SBR} \boxed{2nd} \boxed{R/S}$ pour les TI-58 et TI-58C) pour voir affiché le numéro du module préprogrammé. Si vous êtes en possession de l'imprimante, vous verrez s'inscrire le nom du module et son numéro (Master correspond à la bibliothèque de base).



ANALYSE D'UN PROGRAMME DE LA BIBLIOTHEQUE. (Transfert en mémoire programme).

Normalement les programmes des bibliothèques sont contenus dans votre module, prêts à être utilisés dès que vous le désirez. L'utilisation s'effectue directement dans le module lui-même. Si vous désirez avoir accès aux instructions constituant l'un des programmes de la bibliothèque, vous pouvez demander son transfert dans la mémoire programme. Vous pouvez alors utiliser toutes les fonctions de programmation de la calculatrice pour étudier les différentes instructions ou pour les modifier si nécessaire. La mémoire programme contient uniquement une copie du programme de la bibliothèque, le programme contenu dans le module ne peut en aucune manière être modifié. Cette procédure de «Transfert en mémoire programme» est simple à exécuter :

1. Vérifiez que la place disponible en mémoire programme est suffisante pour recevoir le programme.
Voyez le chapitre **Partition** page V-22.
2. Appuyez sur **2nd** **Prog** mm pour sélectionner votre programme.
3. Appuyez sur **2nd** **Op** 09 pour effectuer le transfert en mémoire programme.

Cette procédure place le programme choisi dans la mémoire programme à partir du pas 000.

Le transfert du programme efface toutes les instructions qui se trouvaient contenues préalablement dans la mémoire programme. La procédure inverse - transfert de la mémoire programme vers le module - n'est pas réalisable.

C'est la raison pour laquelle un programme modifié en mémoire programme pour vos besoins spécifiques ne pourra en aucune façon être réintroduit dans le module pré-programmé. Si vous avez besoin de conserver le programme modifié, vous pouvez inscrire les différentes instructions sur une feuille de programmation, l'enregistrer sur une carte magnétique si vous avez une TI-59, ou encore effectuer l'édition des instructions si vous avez le berceau imprimant optionnel. La TI-58C conservera les informations lors de la mise en place de la calculatrice sur le berceau imprimant ou lors de la décharge des batteries pendant un certain temps.

Si les programmes contenus dans un module sont protégés, une tentative de transfert ferait clignoter l'affichage.

Si un module d'une bibliothèque ne fonctionnait pas comme prévu, reportez-vous à l'Appendice A.

IV

La programmation



QU'EST-CE QUE LA PROGRAMMATION ?

L'impact des ordinateurs dans la vie de tous les jours est tel que certains termes nous sont devenus familiers, comme **programmer un ordinateur, programmeur, logiciel...** ou d'une façon plus générale la **programmation**. Ce vocabulaire laisse supposer un matériel super-sophistiqué destiné à des applications très complexes, et la simple idée d'avoir à programmer fait penser qu'il va falloir suivre des stages de formation exceptionnels.

Les habitués de la programmation s'étonnent ! En effet, la programmation d'une calculatrice est simple et la preuve la plus tangible réside dans le fait que tout le monde sera à même de programmer une calculatrice après une initiation rapide. Les calculatrices de Texas Instruments ont été étudiées pour permettre une simplicité maximale dans la programmation, elles ont une souplesse qui vous permettra d'utiliser rapidement toute la puissance offerte par cette programmation, que vous n'utilisiez que les fonctions arithmétiques de base ou que vous soyez dans un domaine où les calculs mathématiques sont extrêmement complexes. La puissance de calcul est à la disposition de chacun, et chacun pourra utiliser ce dont il a besoin pour son application. Vous serez surpris de la rapidité et de la facilité d'utilisation ainsi que du gain de temps, enregistré même sur des applications constituées d'opérations élémentaires.

La **programmation** n'est que logique. En d'autres termes, un programme n'est qu'une suite d'instructions indiquant à une personne ou à une machine ce qui doit être exécuté. Un programme pour une calculatrice lui apprend comment exécuter quelque chose, en l'occurrence comment parvenir au résultat. Si vous souhaitez que votre calculatrice exécute un certain travail, tout ce que vous avez à faire est de lui indiquer ce qu'elle doit effectuer et l'ordre dans lequel elle doit l'effectuer. Un **programme** est une suite d'instructions précises, celles même que vous utilisiez dans l'expression littérale.

Le **langage** constitue le moyen de communiquer avec votre calculatrice. C'est le cas même pour les calculatrices simples à quatre opérations. Appliqué à la programmation, un langage est nécessaire pour établir le trait d'union entre votre programme et votre calculatrice.

Le **langage de la calculatrice** est considérablement simplifié de nos jours pour permettre un accès banalisé à l'usage de l'arithmétique. Si vous savez effectuer un calcul avec un crayon et un papier, vous connaissez la majorité du langage intelligible par votre calculatrice programmable. Les fonctions décrites dans ce manuel utilisées pour les calculs au clavier seront de la même façon utilisées dans le programme.

Une calculatrice (tout comme un ordinateur) effectuera les instructions qui lui sont indiquées et se tiendra rigoureusement à celles-ci. Le résultat sera ce que vous, le programmeur, aurez demandé à la calculatrice d'exécuter et ceci dans l'ordre précis que vous lui aurez indiqué. Vous voyez maintenant l'importance de la programmation et l'énorme domaine d'application que vous, en programmant votre calculatrice, allez pouvoir explorer de la plus simple façon qui soit.



PROGRAMMATION ELEMENTAIRE

Placer une variable dans un programme

Prenons l'exemple simple suivant :

$$A + B = C$$

Lorsque vous effectuez ce calcul au clavier, les valeurs de A et B ne peuvent pas être identifiées par des lettres, il est indispensable de connaître leurs valeurs avant de démarrer le calcul. Une fois l'expression introduite et le résultat obtenu, si vous voulez modifier une valeur ou même les deux, vous devez nécessairement recomposer toute l'expression. Avec une calculatrice programmable, vous pouvez introduire les instructions en gardant un espace pour les valeurs indéfinies et obtenir le résultat uniquement en introduisant les valeurs nécessaires.

Bien sûr pour une expression aussi simple il est tout aussi rapide de la calculer au clavier qu'en la programmant. L'avantage réel de la programmation devient plus explicite avec un exemple comme celui indiqué ci-dessous: Imaginez que vous ayez besoin des résultats correspondant à 10 valeurs différentes de A en supposant que B et C soient constants.

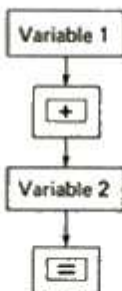
$$A \times (B + (1 + A)^{-C}) = \text{RESULTAT}$$

Vous aimeriez n'introduire cette équation qu'une seule fois et n'avoir qu'à changer la valeur de A pour chaque calcul : c'est précisément ce que vous allez pouvoir faire avec votre calculatrice programmable.

Tout d'abord, revenons sur le premier exemple et voyons de quelle façon il faut fournir les instructions à la calculatrice. En premier lieu, inscrivez les instructions comme si vous enseigniez la façon de procéder à une autre personne, après seulement nous retranscrivons en instructions intelligibles par la calculatrice.



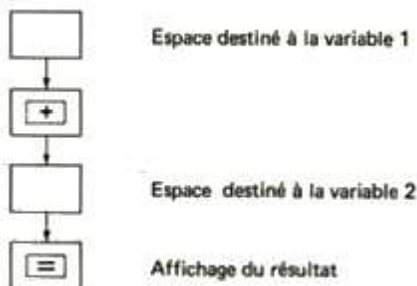
A et B sont des valeurs quelconques pouvant varier. Ces valeurs sont appelées des variables.



De proche en proche on voit se dessiner le langage de la calculatrice ; deux éléments à préciser :

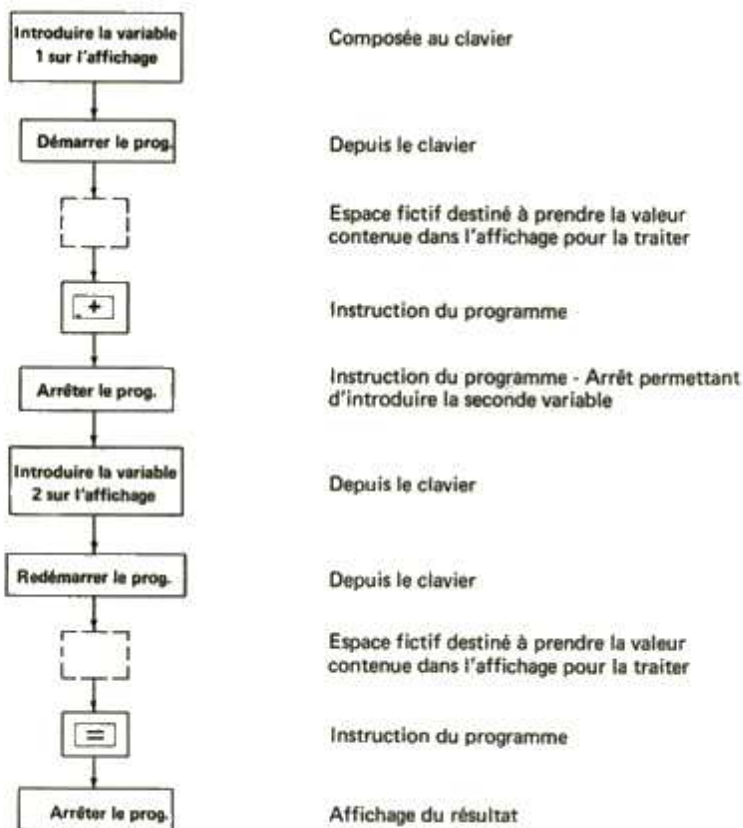
1. Préparez un espace où vous pourrez introduire les variables en temps voulu.
2. Indiquez à la calculatrice l'endroit où elle devra chercher les variables quand elle en aura besoin. Votre calculatrice peut aller chercher une variable, soit dans le registre d'affichage, soit dans une mémoire.

Reécrivons la séquence d'instructions en préparant les espaces nécessaires à l'introduction des variables.



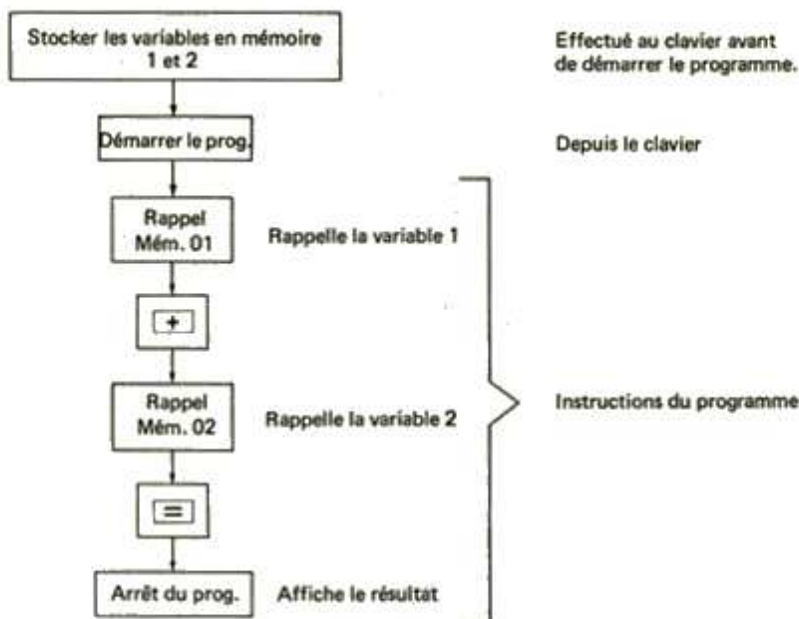
Si la calculatrice utilise le registre d'affichage pour introduire les variables, laissez les espaces vides. Lorsque la calculatrice démarre l'exécution du programme, quelle que soit la valeur affichée celle-ci est placée dans le premier espace. Une valeur doit également être introduite sur l'affichage pour la seconde variable, si bien que nous devons arrêter le déroulement du programme avant l'espace prévu pour pouvoir introduire la valeur de la seconde variable dans l'affichage. Ainsi en faisant redémarrer le déroulement du programme, la calculatrice prendra en compte la valeur affichée pour continuer l'exécution des instructions.

Nous avons laissé des espaces pour l'introduction des variables, car la technique consiste simplement à arrêter l'exécution du programme avec la touche **RS** (Run/Stop - Démarrage/Arrêt). Vous demandez ainsi à la machine d'arrêter toutes exécutions le temps d'introduire la valeur suivante sur l'affichage. Ces espaces sont fictifs de façon à ce qu'il ne soit laissé aucun trou entre les instructions du programme. L'arrêt de l'exécution du programme à un endroit quelconque implique que nous désirions faire quelque chose à cet endroit - introduire une donnée par exemple.



Organigramme pour des variables introduites à partir de l'affichage.

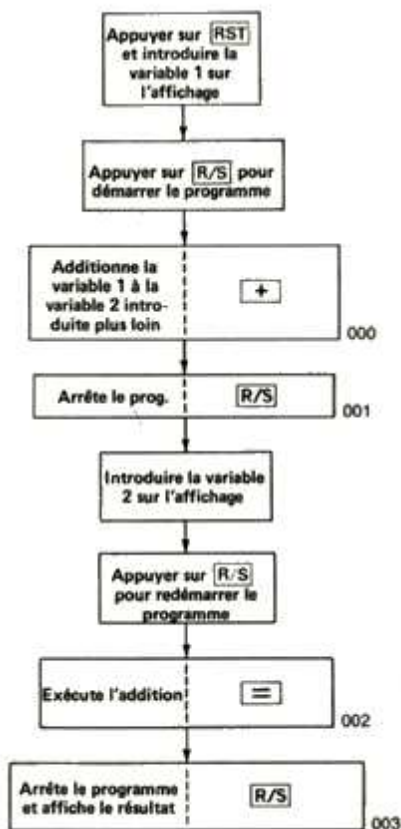
La méthode ci-dessus (arrêt du programme pour introduire les données) est pratique pour introduire un grand nombre de variables à chaque réutilisation du programme. Nous allons voir une technique différente et plus pratique, surtout dans le cas où il n'y a qu'un paramètre qui change. Cette technique utilise les mémoires pour stocker les variables. La calculatrice ira chercher les variables dans les mémoires appropriées au moment où elle en aura besoin, les instructions de rappel de la variable seront intercalées au bon endroit dans le programme. Par exemple, pour rappeler une variable stockée en mémoire 1, nous placerons la séquence **RCL 01**.



Organigramme pour des variables introduites dans les mémoires.

Faisons un rappel rapide de ce que nous venons de faire. En premier, nous avons identifié le problème, puis nous avons envisagé deux méthodes possibles pour introduire les variables (une sans mémoire, l'autre avec les mémoires). En troisième lieu, nous avons développé l'organigramme complet pour chaque méthode. Notons qu'un organigramme a pour objet de dissocier les instructions élémentaires et les actions nécessaires à la résolution du problème.

L'étape suivante consiste à déterminer la suite des instructions en fonction de l'organigramme. On notera dans les deux exemples suivants qu'il est nécessaire d'appuyer sur la touche **RST** avant d'utiliser le programme. Cette touche permet, avant d'appuyer sur **R/S** pour démarrer l'exécution du programme, d'être certain que celui-ci a bien commencé à l'instruction 000. L'exemple suivant montre la séquence de touches à mettre en œuvre pour que la calculatrice résolve le problème en partant de variables introduites dans le registre d'affichage.



Organigramme pour des variables introduites à partir de l'affichage.

- Les carrés centrés de ce diagramme expliquent les opérations que vous avez à exécuter pour utiliser le programme une fois celui-ci enregistré dans la mémoire programme. Les carrés séparés en deux par une ligne pointillée vous donnent les instructions constituant le programme qui doivent être introduites dans la calculatrice une fois celle-ci mise en mode d'enregistrement de programme. Les nombres placés à droite de ces carrés représentent la numérotation des instructions.

IV



Procédure à suivre pour introduire le programme dans votre calculatrice.

1. Appuyer sur **2nd CLR** pour vider le contenu de la mémoire programme.
2. Appuyer sur **LRN** pour mettre la calculatrice en mode programmation reconnaissable par un affichage contenant 000 00.
3. Introduire chaque fonction dans l'ordre indiqué sur l'organigramme en ayant soin de ne pas en oublier. Si vous faites une erreur de touche, revenez au pas précédent en appuyant sur **BST** pour la corriger avant de continuer à introduire votre programme.
4. Appuyer à nouveau sur **LRN** pour revenir du mode programmation au mode calcul reconnaissable par un zéro sur l'affichage. Vous êtes prêt maintenant à utiliser votre programme.

En mode programmation, les trois chiffres de gauche changent au fur et à mesure de l'introduction du programme. Ils vous indiquent le numéro de l'instruction devant laquelle se trouve le pointeur. Ce pointeur est un indicateur utilisé par la machine pour savoir quelle est la prochaine instruction à utiliser. En mode programmation, ce pointeur se place automatiquement sur l'emplacement qui suit immédiatement l'instruction que vous venez d'introduire.

Regardons de quelle façon nous allons introduire le programme étudié précédemment (variables introduites par l'affichage).

1. Mettre en marche la calculatrice et appuyer sur **2nd GP** pour la TI-58C.
2. Appuyer sur **LRN** pour se mettre en mode programmation.
3. Introduire le programme en suivant la séquence de touches indiquée :
+ R/S = R/S
4. Appuyer sur **LRN** pour revenir en mode calcul.

Vous venez de programmer votre calculatrice, vous pouvez maintenant résoudre le problème $227 + 34$ en utilisant ce programme.

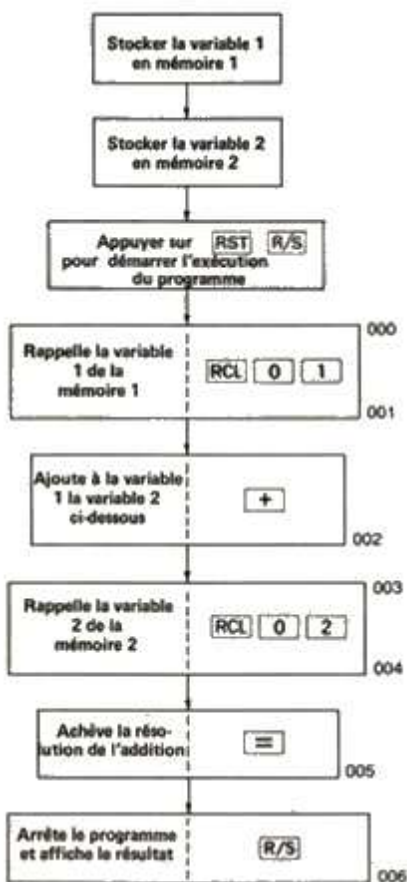
1. Appuyer sur **CLR** pour effacer les calculs précédents.
2. Appuyer sur **RST** et introduire la variable 1 soit 227.
3. Appuyer sur **R/S** 227 reste affiché.
4. Introduire 34, la variable 2, et appuyer à nouveau sur **R/S**, l'affichage vous donne la réponse : 261.

Avant d'utiliser un programme, il est conseillé d'appuyer sur **CLR** pour s'assurer que différents calculs en cours ne viennent pas modifier le résultat. Pour utiliser ce programme il suffit d'appuyer sur **RST** pour remettre le pointeur au pas de départ et d'introduire chaque variables sur l'affichage suivie de **R/S** pour obtenir le résultat. Vous pouvez l'utiliser aussi souvent que vous le désirez.



IV

Voyons maintenant comment écrire le programme en utilisant les mémoires pour stocker les variables.



Organigramme pour des variables stockées en mémoires.

IV



Chargeons le programme dans la calculatrice.

1. Mettez en marche la calculatrice et appuyez sur **2nd** **CP**
2. Appuyez sur **LRN** pour mettre la calculatrice en mode programmation.
3. Introduisez le programme en suivant l'ordre des touches

RCL
0
1
+
RCL
0
2
=
R/S

4. Appuyez à nouveau sur **LRN** pour revenir en mode calcul.

Ce programme utilise les mémoires 01 et 02 dans lesquelles les variables 1 et 2 sont stockées. Mettons par exemple 227 et 34 dans l'une et l'autre des deux mémoires :

Composez **227** puis appuyez sur **STO** **01**

Composez **34** puis appuyez sur **STO** **02**

Comme indiqué page précédente, il ne reste plus qu'à appuyer sur **RST** **R/S** . Le résultat apparaît aussitôt sur l'affichage, la calculatrice a exécuté elle-même le programme et résolu le problème.

Noter que le programme écrit pour des variables introduites par l'affichage nécessite moins d'instructions que lorsque ces variables sont en mémoires. Toutefois l'usage des mémoires permet d'effectuer l'exécution du programme sans arrêt intermédiaire. Vous avez le choix entre l'une et l'autre méthode.

Nous avons vu l'utilisation d'un organigramme et l'aide précieuse qu'il fournit dans la décomposition d'un problème. L'organigramme regroupe toutes les étapes qui se présentent lors du déroulement du programme ainsi que les manipulations préalables nécessaires au démarrage du programme. Les fonctions introduites dans la calculatrice pendant que celle-ci est en mode programmation, constituent le programme proprement dit.



Mécanisme de la programmation

La diversité des applications engendre des programmations simples ou complexes. Si les programmes simples peuvent être introduits, vérifiés et utilisés sans difficultés, il devient nécessaire pour des programmes plus complexes de décomposer en séquences simples la suite des calculs.

Si vous êtes déjà familiarisé avec la programmation, les conseils qui suivent vous sont connus, mais ils vous permettront de retrouver la marche à suivre conseillée pour étudier un problème et programmer votre calculatrice. Si vous abordez la programmation pour la première fois, ce qui suit vous servira de fil conducteur. Dans l'un ou l'autre cas, il ne s'agit que de suggestions à partir desquelles vous réaliserez votre programme avec votre style propre.

1. **Définir le problème clairement et complètement.** Identifier les formules, les variables, les résultats désirés. Qu'est-ce qui est connu ? Qu'est-ce qui est cherché ? Qu'est-ce qui existe comme relations entre ce qui est connu et ce qui est cherché ?
2. **Développer la méthode de calcul** (encore appelée algorithme). Définir la succession des opérations nécessaires pour effectuer une application numérique en fonction des possibilités de calcul et de programmation de votre calculatrice. (Souvenez-vous que rigoureusement parlant, votre calculatrice ne résoud pas les problèmes, c'est vous qui le faites. Votre calculatrice se borne uniquement à exécuter ce que vous lui demandez !).
3. **Développer l'organigramme.** Il est souvent nécessaire de développer un schéma qui vous donnera une idée d'ensemble du déroulement du programme. Vous pourrez ainsi identifier les relations qui peuvent exister entre les variables et les résultats et peut être effectuer des modifications propres à simplifier le programme sur l'organigramme.
4. **Faire l'affectation des mémoires.** Identifiez les mémoires que vous utiliserez et notez ce à quoi est affecté chaque mémoire de façon à éviter en cours de programme de rappeler une mauvaise mémoire ou de venir inopinément modifier le contenu de l'une d'elles.
5. **Transcrire l'organigramme en séquences de touches.** Des feuilles de programmation sont prévues à cet effet. Elles permettent également de noter les étiquettes et les mémoires utilisées. Une colonne réservée aux commentaires permet l'identification des différentes parties du programme.
6. **Introduire le programme.** Appuyez sur **2nd**, **CP**, **LRN** et introduisez l'ensemble des fonctions indiquées sur vos feuilles de programmation. Une fois ce travail achevé, appuyez à nouveau sur **LRN** pour revenir en mode calcul.
7. **Tester le programme.** Utilisez le programme avec des valeurs qui vous donneront un résultat que vous connaissez, pour vous assurer de son bon fonctionnement.
8. **Corriger les erreurs.** Corrigez sur vos feuilles de programmation, les erreurs que vous avez notées en testant le programme.
9. **Modifier le programme.** Mettez la calculatrice en mode programmation, effectuez les différentes corrections et revenez en mode calcul avec la touche **LRN**. (Voir page IV-21).
10. **Tester à nouveau le programme.** Refaites les séquences 7, 8, 9 autant de fois que ce sera nécessaire.
11. **Enregistrer le programme.** Si vous avez une TI-59, enregistrez le programme sur cartes magnétiques. (Voir Section VII)
12. **Rédiger un mode d'emploi.** Il est très important de noter, étape par étape, les instructions nécessaires pour utiliser votre programme. Un programme puissant peut devenir inutilisable si on ne possède plus la marche à suivre détaillée de ce programme.



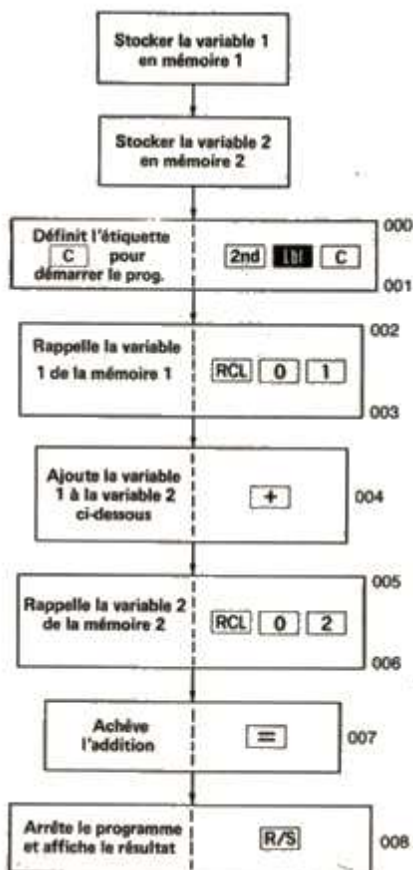
Utilisation des touches-utilisateur.

Dans les exemples précédents nous utilisons les touches **RST** et **R/S** pour exploiter un programme. La touche **RST** ramenait le pointeur à l'instruction de départ 000 et on aurait pu en déduire qu'il était nécessaire qu'un programme commence à l'instruction 000 ce qui n'est pas toujours pratique. Votre calculatrice possède des **Touches-Utilisateur** qui sont utilisées comme **étiquettes** vous permettant de vous adresser à un endroit bien précis dans le programme. Ces **étiquettes** sont des points de repère dans le programme.

Lorsqu'une touche utilisateur est placée dans le programme, le fait d'appuyer sur cette touche en mode calcul vient placer le pointeur à cet endroit dans le programme et automatiquement la calculatrice commence l'exécution du programme à partir de l'instruction suivant cette touche utilisateur. Vous avez 10 touches à votre disposition **A** à **E** et **2nd A** à **2nd E** vous permettant de venir chercher et d'utiliser jusqu'à 10 séquences différentes (programmes ou parties de programmes). Par exemple, les touches **RST** **R/S** utilisées dans les exemples précédents peuvent être remplacées par la touche utilisateur **C** ou une autre, pour démarrer l'exécution du programme ; il faut alors mettre en début de programme la touche **C** qui précédée de **2nd IW** sera utilisée comme une étiquette, le point de repère du début du programme.



IV



En mode programmation, introduisez ce programme, revenez au mode calcul et introduisez les variables : 227 en mémoire 1, 34 en mémoire 2. Appuyez sur **C** vous obtenez le résultat sur l'affichage 261. En effet, la calculatrice est venue chercher l'endroit où se trouvait **LBL C** et automatiquement a démarré l'exécution du programme à cet endroit.

Maintenant que vous avez vu la façon d'utiliser les touches-utilisateur, nous allons reprendre cet exemple en utilisant la touche **A** pour charger la variable 1 en mémoire 1 et la touche **B** pour charger la variable 2 en mémoire 2. La touche **C** nous donnera le résultat.





IV

En mode programmation introduisez les instructions et revenez en mode calcul. Remarquez que les variables 1 et 2 peuvent être introduites dans n'importe quel ordre.

Le programme est plus long mais beaucoup plus facile à utiliser. Le tableau ci-dessous vous donne la comparaison des trois différentes façons d'utiliser les programmes que nous avons écrits précédemment

Première version	Deuxième version	Troisième version
Introduire 227	Introduire 227	Introduire 227
Appuyer sur STO 01	Appuyer sur STO 01	Appuyer sur A
Introduire 34	Introduire 34	Introduire 34
Appuyer sur STO 02	Appuyer sur STO 02	Appuyer sur B
Appuyer sur RST R.S	Appuyer sur C	Appuyer sur C
Affichage : 261	Affichage : 261	Affichage : 261

Une étiquette peut être placée n'importe où dans le programme sans altérer le déroulement de celui-ci. L'exécution du programme se fera en ignorant son existence. Cependant une étiquette ne doit jamais être intercalée au milieu d'une instruction du type **RCL** 14 qui forme un groupe indissociable.

Vous devez inclure les étiquettes au fur et à mesure dans votre programme plutôt que de les rajouter à posteriori; pour cela vous les introduisez de la même façon que les instructions constituant le reste du programme.

Ces étiquettes présentent l'énorme avantage de suivre les modifications éventuellement apportées aux programmes lorsque vous supprimerez ou ajouterez des instructions.

IV



Adressage mémoire simplifié.

Jusqu'à maintenant nous avons adressé les mémoires avec deux caractères, par exemple, pour rappeler le contenu de la mémoire 1 nous avons utilisé la séquence **RCL 01**. Dans certains cas, le zéro n'est pas nécessaire pour l'adressage des mémoires 0 à 9. Ce type d'adressage appelé **adressage simplifié** peut être utilisé à chaque fois qu'une instruction non numérique suit immédiatement l'adresse de la mémoire.

Exemple : Stockez 227 en mémoire 1, 34 en mémoire 2 puis faites la somme des deux mémoires.

Appuyer	Affichage	Commentaires
227 STO 01	227.	} Les instructions qui suivent étant numériques, l'adresse complète est nécessaire. } Adresse réduite dans les trois cas suivants du fait qu'elles sont suivies d'instructions non numériques.
34 STO 2	34.	
RCL 1 +	227.	
RCL 2 =	261.	

On remarquera que l'instruction **rappel mémoire** s'effectue dans le cas d'adressage simplifié, lorsque l'instruction non numérique est utilisée : 227 apparaît lorsque vous appuyez sur **+**. Une étude plus complète sera faite à la section V.



Introduction de votre programme.

La programmation est la technique qui vous permet d'indiquer à la calculatrice par des instructions précises ce qu'elle aura à exécuter. Une fois que vous avez préparé l'ensemble des instructions et que vous êtes prêt à les introduire dans la calculatrice, vous la mettez en mode de programmation. Nous allons maintenant étudier ce mode plus en détails.

Un programme est développé d'abord à partir de la décomposition logique du problème sans nécessiter l'usage de votre calculatrice. La marche à suivre est celle que vous avez vu précédemment sur des exemples. Le mode de programmation permet de franchir le pas entre l'écriture du programme et l'utilisation de ce programme.

La calculatrice en mode programmation peut recevoir les instructions du programme générées depuis le clavier. En pratique, chaque enfoncement de touche est reçu par la calculatrice comme une instruction. Ceci explique pourquoi il faut être très vigilant dans la phase d'introduction du programme et explique également la raison pour laquelle tous les calculs sont impossibles pendant que nous sommes en mode programmation.

Si vous faites une erreur en introduisant les instructions, vous n'êtes pas obligé de recomposer l'ensemble de votre programme depuis le début. Votre calculatrice possède des touches de correction permettant de supprimer, de remplacer ou d'ajouter une instruction. Ces touches seront décrites ultérieurement dans le paragraphe concernant la vérification d'un programme.

Suivez les recommandations suivantes pour introduire un programme.

1. Appuyez sur **[2nd]** **[CF]** pour placer le pointeur au début de la mémoire programme et pour effacer ce qui s'y trouvait auparavant.
2. Appuyez sur **[LRN]** pour mettre la calculatrice en mode programmation (voir page suivante l'affichage en mode programmation).
3. Introduisez votre programme sans oublier aucune touche (**[2nd]** par exemple).
4. Assurez-vous que votre programme ne dépasse pas la capacité de la mémoire programme choisie. Le fait de la dépasser remet la calculatrice en mode calcul en ignorant les instructions superflues.
5. Appuyez sur **[LRN]** pour revenir en mode calcul.
6. Exécutez un exemple de test et si besoin est, faites les modifications nécessaires comme indiqué au paragraphe concernant la **vérification d'un programme**.



Affichage en mode Programmation.

En mode programmation l'affichage vous permet de connaître le numéro de l'instruction où se trouve positionné le pointeur ainsi que l'instruction placée à cet endroit. Allumez votre calculatrice et appuyez sur **[LRN]** pour vous mettre en mode programmation. L'affichage vous montre deux groupes de zéros.

000 00

Le groupe de trois zéros à gauche représente le positionnement du pointeur dans la mémoire programme. Lorsque vous introduisez une instruction, elle se place dans une mémoire programme et le nombre de gauche augmente d'une unité; ceci vous permet non seulement de suivre votre programme mais également indiquera à la calculatrice l'ordre d'exécution des instructions.

La calculatrice n'interprétant que des valeurs numériques, à chaque touche correspond un code numérique à deux caractères que vous pouvez voir sur la droite de l'affichage. Les touches sont codées de la façon suivante :

1. Les touches numériques sont représentées par leur propre valeur, par exemple : **[7]** est codé «07».
2. Les fonctions premières sont codées suivant leur emplacement sur le clavier. Le premier caractère indique le numéro entre 1 et 9 de la ligne (du haut vers le bas). Le second caractère représente le numéro de la colonne entre 1 et 5 (de la gauche vers la droite). Par exemple, l'instruction **[√x]** (ligne 3, colonne 4) placé dans la mémoire programme 073 sera identifiée sur l'affichage par :

073 34

3. Les fonctions secondes sont codées en ajoutant cinq au numéro de la colonne. Le numéro de la ligne reste inchangé. Par exemple, l'instruction **[2nd] [cos]** situé au-dessus de l'instruction **[√x]** (code «34»), sera codée «39». **[2nd] [tan]** situé au-dessus de l'instruction **[1/x]** codée «35» sera codée «30» et non pas «40» de façon à ce que le numéro de la ligne reste inchangé.

Ces codes vous permettent de vérifier votre programme, vous trouverez un tableau détaillé au chapitre **CODES INSTRUCTIONS** de la section V.

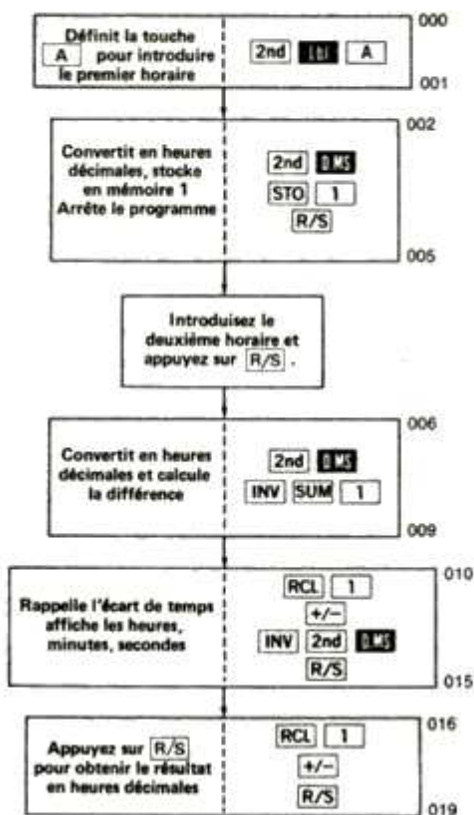
Dans certains cas des instructions regroupées sont introduites dans une mémoire programme unique. Par exemple **[RCL] 12** utilisera seulement deux pas de programme, un pour l'instruction **[RCL]** (code «43»), l'autre pour les instructions **[1]** et **[2]** réunies. Le code est alors «12». Il ne faut pas le confondre avec l'autre signification du code «12» qui correspond à la touche **[B]** ; la confusion n'est pas possible si on se souvient que l'instruction **[RCL]** doit être utilisé avec une adresse numérique. D'autres applications seront explicités chaque fois qu'elles seront employées.

Remarque : A l'écriture du programme le code n'apparaît pas du fait que la calculatrice est en attente d'instructions, pointeur positionné sur la première mémoire programme vide suivant l'instruction déjà introduite.



Calcul du temps écoulé

Ecrivons le programme permettant de calculer le temps écoulé entre deux horaires. Nous introduisons chaque horaire en heures, minutes, secondes (exemple : 3 heures 16' 03" = 3.1603), la calculatrice fera la conversion nécessaire pour faire le calcul (voir section II, chapitre Conversion Angulaire).



IV



Introduisons ce programme dans la calculatrice. Avant toute chose appuyez sur **2nd** **CF** pour effacer la mémoire programme et remettre le pointeur au départ à l'instruction 000 puis introduisez les différentes instructions comme indiqué ci-dessous :

Appuyer	Affichage
LRN	000 00
2nd LSI	001 00
A	002 00
2nd 0.MS	003 00
STO	004 00
1	*004 00
R/S	006 00
2nd 0.MS	007 00
INV	008 00
SUM	009 00
1	*009 00
RCL	011 00
1	*011 00
+/-	013 00
INV	014 00
2nd 0.MS	015 00
R/S	016 00
RCL	017 00
1	*017 00
+/-	019 00
R/S	020 00

Après avoir effacé la mémoire programme au départ (**2nd** **CF**) les codes affichés vous indiquent toujours 00. Ceci est dû au fait que le pointeur se place dès l'introduction d'une instruction, sur le pas suivant qui est vide et non pas sur l'instruction que vous venez d'introduire.

* L'affichage n'avance pas d'une unité dans ces différents cas. En effet, nous sommes en présence d'un adressage simplifié. La calculatrice attend l'adresse complète de la mémoire par exemple dans le premier cas, si nous introduisons **STO** **01** nous verrions apparaître 005 00; ici comme l'adresse mémoire est suivie d'une instruction non numérique **R/S**, cette instruction indique à la calculatrice que l'adresse mémoire est complète et qu'elle peut être stockée au pas 004, en même temps que **R/S** est stocké au pas 005. Le pointeur passe automatiquement au pas 006 sans faire apparaître le pas 005.



IV

Pour effectuer une vérification du programme, suivez la procédure qui suit. Si vous possédez le berceau imprimant, il vous suffit de faire **RST** **2nd** **Int** pour obtenir l'édition des instructions. Maintenir appuyée la touche **R/S** pour arrêter l'imprimante.

Appuyer	Affichage	Instructions correspondantes
RST LRN	000 76	2nd Int
SST	001 11	A
SST	002 88	2nd 0.MS
SST	003 42	STO
SST	004 01	1
SST	005 91	R/S
SST	006 88	2nd 0.MS
SST	007 22	INV
SST	008 44	SUM
SST	009 01	1
SST	010 43	RCL
SST	011 01	1
SST	012 94	+/-
SST	013 22	INV
SST	014 88	2nd 0.MS
SST	015 91	R/S
SST	016 43	RCL
SST	017 01	1
SST	018 94	+/-
SST	019 91	R/S
LRN		

Maintenant que vous avez vérifié que les instructions sont exactes, utilisons ce programme pour calculer l'écart entre 2 heures 15' et 3 heures 42' 54".

Introduire	Appuyer	Affichage	Commentaires
2.15	A	2.25	t_1 (H.MMSS) \rightarrow t_1 (H.hh)
3.4254	R/S	1.2754	t_2 (H.MMSS) $\rightarrow \Delta t$ (H.MMSS)
	R/S	1.465	$\rightarrow \Delta t$ (H.hh)

Le temps écoulé est 1 heure 27 mn 54 sec. ou 1.465 heure.

Si vous obtenez un résultat du type 6.396 en Heures Mn. Sec. vous devez l'interpréter comme 6 heures 40 mn. au lieu de 6 heures, 39 mn, 60 sec.



Correction d'un programme

En mode programmation vous avez plusieurs possibilités :

1. Afficher une instruction contenue dans un programme.
2. Remplacer une instruction par une autre.
3. Effacer une instruction et boucher l'emplacement laissé libre.
4. Créer un espace pour ajouter une instruction.
5. Avancer ou reculer pas par pas dans le programme.

Ces possibilités vous permettent de vérifier et de modifier votre programme sans être obligé de le recomposer.

Les quatre touches **SST**, **BST**, **Ins** et **Del** correspondant à ces différentes possibilités ne sont pas interprétées en mode programmation comme des instructions de programme. **SST** et **BST** permettent respectivement d'avancer ou de reculer d'un pas dans le programme. En mode utilisation **SST** permet également de faire une exécution du programme, instruction par instruction. **2nd** **Ins** permet de créer un espace libre dans un programme en décalant toutes les instructions qui suivent d'un pas, cet espace disponible va pouvoir recevoir une instruction. Si la mémoire programme est complètement chargée, le fait d'appuyer sur **2nd** **Ins** à un endroit quelconque du programme, détruira la dernière instruction du programme. **2nd** **Del** permet de détruire une instruction dans le programme en décalant toutes les instructions qui suivent de telle sorte que la place vide ainsi créée se trouve bouchée.

En mode calcul, deux touches sont souvent utilisées pour faciliter la vérification d'un programme **RST** et **GTO** (attention : **RST** a plusieurs fonctions, voyez le chapitre concernant les Fonctions de contrôle à la section V). **RST** remet en outre, le pointeur au pas programme 000. **GTO** suivi d'une adresse à trois caractères, place le pointeur sur le pas programme correspondant (il est possible d'utiliser une adresse contractée; **GTO** suivi d'une étiquette place le pointeur sur le pas suivant l'étiquette. On peut alors passer en mode programmation pour vérifier la séquence de programme correspondante. L'utilisation de ces touches en mode programmation serait interprétée comme des instructions de programme.

Si vous désirez modifier une séquence de votre programme, localisez celle-ci avec les fonctions décrites ci-dessus et recomposez votre nouvelle séquence qui effacera les instructions précédemment introduites; si besoin est, vous pouvez également effacer des instructions.

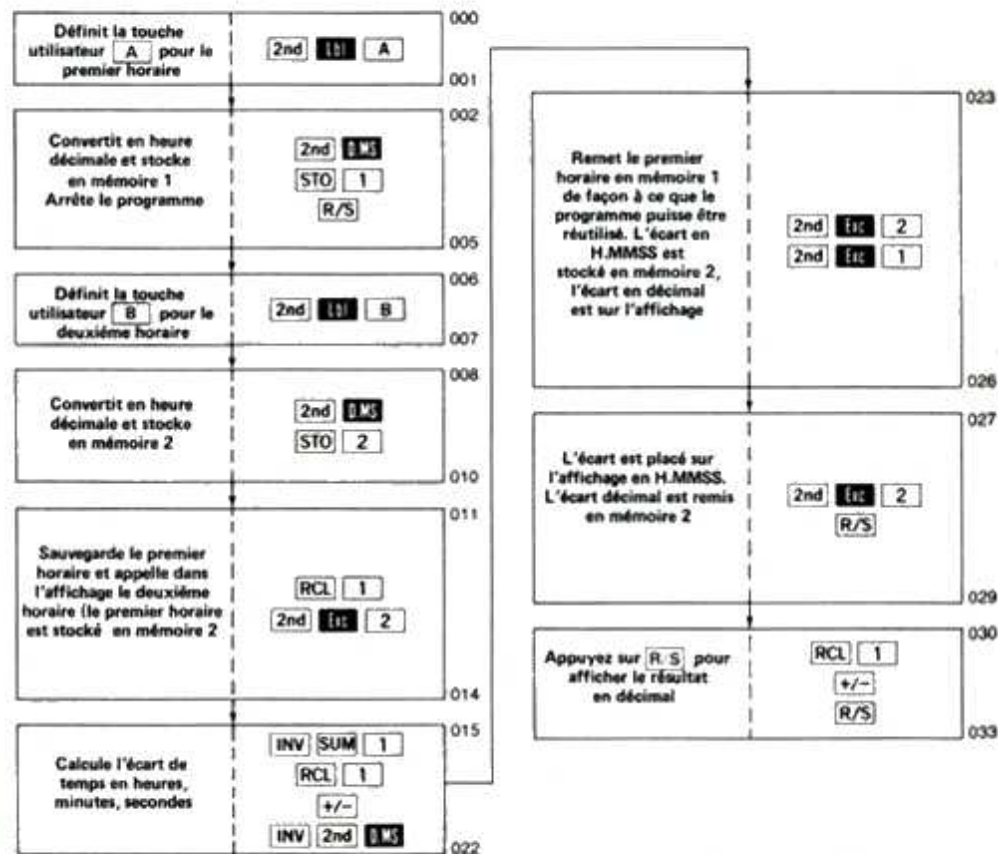


IV

Modification du programme calculant le temps écoulé.

Modifions le programme précédent de telle façon que l'on puisse seulement introduire le second horaire sans être obligé de réintroduire le premier. Faisons les modifications à partir du programme précédent.

Nous devons : premièrement, utiliser une étiquette pour introduire le deuxième horaire de façon à pouvoir accéder directement à cette partie du programme. Deuxièmement, trouver un moyen pour conserver le premier horaire sans qu'il soit détruit par les calculs. Troisièmement, organiser le programme de façon à ce qu'il soit possible après avoir calculé un temps écoulé, de réintroduire un second horaire pour un autre calcul.



Ce programme contient une erreur, à vous de la retrouver. Attention : L'utilisation de la fonction **101** complique ce programme. Etudiez-le avec beaucoup d'attention.

IV



Faites les corrections nécessaires pour modifier le programme écrit à l'origine.

Appuyer	Affichage	Commentaires
GTO 6		Place le pointeur au pas 006
LRN	006 88	Mise en mode programmation
2nd Ins	006 00	Intercalle l'étiquette B .
2nd Ins	006 00	
2nd 1st	007 00	
B	008 88	
SST	009 22	Avance le pointeur d'un pas
2nd Ins	009 00	
2nd Ins	009 00	
STO 2	010 00	
2nd Ins	011 00	
2nd Ins	011 00	Intercalle les instructions
RCL 1	012 00	STO 2 RCL 1 2nd Exc 2
2nd Ins	013 00	
2nd Ins	013 00	
2nd Exc 2	014 00	
SST SST SST	018 43	Avance le pointeur de 8 pas
SST SST SST	021 22	
SST SST	023 91	
2nd Ins	023 00	Intercalle les instructions
2nd Ins	023 00	2nd Exc 2 2nd Exc 1 Ins Exc 2
2nd Exc 2	024 00	
2nd Ins	025 00	
2nd Ins	025 00	
2nd Exc 1	026 00	
2nd Ins	027 00	
2nd Ins	027 00	
2nd Exc 2	028 00	
LRN		Retour en mode calcul.



IV

Suivez cette procédure pour vérifier que les modifications effectuées sont correctes.

Appuyer	Affichage	Instructions correspondantes
RST LRN	000 76	2nd 101
SST	001 11	A
SST	002 88	2nd 0 M1
SST	003 42	STO
SST	004 01	1
SST	005 91	R/S
SST	006 76	2nd 101
SST	007 12	B
SST	008 88	2nd 0 M1
SST	009 42	STO
SST	010 02	2
SST	011 43	RCL
SST	012 01	1
SST	013 48	2nd Esc
SST	014 02	2
SST	015 22	INV
SST	016 44	SUM
SST	017 01	1
SST	018 43	RCL
SST	019 01	1
SST	020 94	+/-
SST	021 22	INV
SST	022 88	2nd 0 M1
SST	023 48	2nd Esc
SST	024 02	2
SST	025 48	2nd Esc
SST	026 01	1
SST	027 48	2nd Esc
SST	028 02	2
SST	029 91	R/S
SST	030 43	RCL
SST	031 01	1
SST	032 94	+/-
SST	033 91	R/S

IV



Utilisez ce programme avec 1h30 comme premier horaire; 2h 13' 57" puis 2h 14' 24" comme deuxième horaire

Introduire	Appuyer	Affichage	Commentaires
1.3	<input type="button" value="A"/>	1.5	t_1 (HH.MMSS) \rightarrow t_1 (HH.hh)
2.1357	<input type="button" value="B"/>	0.4357	t_2 (HH.MMSS) \rightarrow Δt (HH.MMSS)
	<input type="button" value="R/S"/>	-1.5	?

La dernière réponse est manifestement inexacte. Elle correspond à l'expression décimale de l'écart et ne peut être négative puisque l'écart exprimé en heure, mn, sec. est positif. La calculatrice a affiché la valeur, changé de signe du premier horaire. Si vous regardez la séquence aux pas de programmes 023 à 028 nous avons transféré l'information en mémoire 2. Nous devons donc modifier l'adresse utilisée au pas 31 :

Appuyer	Affichage
<input type="button" value="GTO"/> <input type="button" value="3"/> <input type="button" value="1"/>	
<input type="button" value="LRN"/>	031 01
<input type="button" value="2"/>	032 94
<input type="button" value="LRN"/>	

Maintenant vous pouvez refaire l'exemple indiqué plus haut :

Introduire	Appuyer	Affichage	Commentaires
1.3	<input type="button" value="A"/>	1.5	t_1 (HH.MMSS) \rightarrow t_1 (HH.hh)
2.1357	<input type="button" value="B"/>	0.4357	t_2 (HH.MMSS) \rightarrow Δt (HH.MMSS)
	<input type="button" value="R/S"/>	0.7325	\rightarrow Δt (HH.hh)
2.1424	<input type="button" value="B"/>	0.4424	t_2 (HH.MMSS) \rightarrow Δt (HH.MMSS)
	<input type="button" value="R/S"/>	0.74	\rightarrow Δt (HH.hh)



Correction d'adresses contractées.

Supposons que nous ayons utilisé dans l'exemple précédent la mémoire 12 au lieu de la mémoire 2. Dans ce cas, au pas de programme 031 il faut rappeler le contenu de la mémoire 12 en s'assurant que la forme contractée soit correctement introduite lorsqu'on effectue la modification. Une méthode est développée ci-dessous :

Appuyer	Affichage
CLR GTO 30	0
LRN	030 43
RCL	031 01
1 2	032 94
LRN	0

Il est nécessaire de recomposer l'instruction **RCL** pour que l'adresse 12 soit contractée sinon 12 est considéré comme un simple nombre utilisant deux pas programme.

IV



Applications pratiques de programmes

LA PROGRAMMATION EST PERSONNELLE.

Avant de continuer il est important de souligner que la programmation est très personnelle, tant et si bien que deux personnes effectuant le même programme ne développeront pas une programmation identique bien qu'elles arriveront l'une et l'autre à des résultats rigoureusement identiques. Chaque personne abordera un même problème suivant des méthodes différentes suivant son idée propre et en fonction également de sa formation ; un financier n'étudiera pas un problème de la même façon qu'un ingénieur, les programmes n'utiliseront pas le même nombre de pas, mais le résultat restera le même, c'est à chacun de définir son approche.

Votre style va se développer au fur et à mesure des explications qui vous sont fournies et même si vous êtes convaincu de la bonne compréhension de ce que vous venez de voir, n'oubliez pas qu'il ne s'agit que d'exemples explicités que peut-être vous auriez développé vous-même de façon très différente. Faites des expériences sans hésiter à faire des erreurs, ce sont elles qui vous feront maîtriser les possibilités de votre calculatrice. Un ordinateur représente un coût d'utilisation énorme, ce qui n'est pas le cas pour votre calculatrice, alors expérimentez vos développements, faites des tests, faites des essais... Votre calculatrice est à votre disposition.

CALCUL D'UN INVESTISSEMENT

Quel est l'avantage offert par une calculatrice programmable ? Obtenir différentes solutions rapidement avec le minimum de risque d'erreur. Nous allons voir sur un exemple comment votre calculatrice vous fait gagner du temps.

A un moment donné chacun a été ou sera confronté avec des problèmes d'épargne pour lesquels les intérêts acquis sont cumulés avec le capital. Si 5% d'intérêt annuel est fourni, pour un capital de 1000 Frs à la fin de l'année les intérêts représentent 50 Frs qui viendront s'ajouter au capital ; celui-ci deviendra pour l'année suivante 1050 Frs. Les 1000 Frs seront appelés «valeur présente» ; les 1050 Frs, «valeur future». Compte tenu que c'est le nouveau capital qui sert de base de calcul pour le montant de l'intérêt de la période suivante, il est facile d'effectuer le calcul sur une ou deux périodes :

$$1000 \text{ Frs} + 1000 \text{ Frs} \times 0,05 = 1050 \text{ Frs à la fin de la première période.}$$

$$1050 \text{ Frs} + 1050 \text{ Frs} \times 0,05 = 1102,50 \text{ Frs à la fin de la seconde période.}$$

et ainsi de suite pour chaque période.



IV

Le principe est connu mais l'expression qui permet de faire ce type de calcul l'est moins; elle s'exprime par :

«La valeur future est égale à la valeur présente multipliée par le taux d'intérêt plus 1, multiplié par lui-même autant de fois qu'il y a de périodes».

Mathématiquement :

$$FV = PV \times (1 + i)^n$$

Avant d'écrire le programme, il faut auparavant regarder et décomposer ce qui doit être fait. Au départ le taux d'intérêt sera introduit directement en pourcentage, il faut donc prévoir de le diviser par 100. La période peut être l'année mais également le semestre, le trimestre ou le mois, pour plus de souplesse nous allons introduire cette possibilité dans la formule mathématique :

$$FV = PV \times \left[1 + \left(\frac{i}{100} \div c \right) \right]^{cn}$$

Nous choisirons dans cet exemple d'introduire les variables par les mémoires et non directement à l'affichage de façon à ce que l'on puisse réutiliser ce programme, sans être obligé de réintroduire toutes les données du problème.

L'étude étant faite et le mode d'introduction des variables décidé, il nous faut maintenant écrire l'organigramme correspondant.

IV



Associe la touche A à PV	2nd lbl A STO 1 R/S	000
Associe la touche B à i	2nd lbl B STO 2 R/S	
Associe la touche C à c	2nd lbl C STO 3 R/S	
Associe la touche D à n	2nd lbl D STO 4 R/S	019
Associe la touche E au démarrage du calcul	2nd lbl E	020
Convertit i % en décimal	RCL 2 ÷ 1 0 0	022
Ramène le taux d'intérêt par an au nombre de période désiré	÷ RCL 3	028
Calcule le coefficient d'intérêt composé pour c x n périodes	+ 1 = y^x (RCL 3 X RCL 4)	031
Multiplié par PV donne FV	X RCL 1 =	042
Affiche FV arrondi aux centimes	2nd fn 2 R/S	046
		048

Programme d'un calcul d'investissement.



IV

MODE D'EMPLOI				
Séquence	Marche à suivre	Introduire	Appuyer	Affichage
1	Effacer la mémoire programme et placer le pointeur au pas 000		2nd CF	
2	Mise en mode programmation		LRN	000 00
3	Introduire les instructions du programme			
4	Retour en mode calcul		LRN	0
5	Introduire la valeur présente	PV	A	PV
6	Introduire le taux d'int. annuel	i	B	i
7	Introduire le nombre de période de calcul par an	c	C	c
8	Introduire le nombre d'années d'épargne	n	D	n
9	Calculer la valeur future		E	FV
	Les variables peuvent être introduites dans n'importe quel ordre. Il n'est pas nécessaire de réintroduire les paramètres inchangés pour démarrer le calcul d'un nouveau problème.			



Pas et code instruction	Séquence de touches	Pas et code instruction	Séquence de touches
000 76	2nd 1b1	025 01	1
001 11	A	026 00	0
002 42	STO	027 00	0
003 01	1	028 55	÷
004 91	R/S	029 43	RCL
005 76	2nd 1b1	030 03	3
006 12	B	031 85	+
007 42	STO	032 01	1
008 02	2	033 95	=
009 91	R/S	034 45	Y*
010 76	2nd 1b1	035 53	(
011 13	C	036 43	RCL
012 42	STO	037 03	3
013 03	3	038 65	X
014 91	R/S	039 43	RCL
015 76	2nd 1b1	040 04	4
016 14	D	041 54)
017 42	STO	042 65	X
018 04	4	043 43	RCL
019 91	R/S	044 01	1
020 76	2nd 1b1	045 95	=
021 15	E	046 58	2nd 1b1
022 43	RCL	047 02	2
023 02	2	048 91	R/S
024 55	÷		

Liste des instructions du programme d'un calcul d'investissement.



IV

Trouver la valeur future d'un capital de 3 000 Frs investi à 8% par an, sur 5 ans, les intérêts étant composés quotidiennement puis mensuellement.

Introduire	Appuyer	Affichage	Commentaires
3000	<input type="button" value="A"/>	3000.	PV
8	<input type="button" value="B"/>	8.	i
365	<input type="button" value="C"/>	365.	c
5	<input type="button" value="D"/>	5.	n
	<input type="button" value="E"/>	4475.28	FV
12	<input type="button" value="C"/>	12.00	c
	<input type="button" value="E"/>	4469.54	FV

A partir de cette étape, l'affichage se fera avec deux décimales. En ce qui concerne la TI-58C, cette décimalisation sera conservée jusqu'à nouvel ordre même si la calculatrice est momentanément éteinte.

CONTROLE DE PRIX

Jusqu'à-là, nous avons utilisé les mémoires pour stocker les variables. Nous pouvons également effectuer les additions, soustractions, multiplications et divisions directement sur le contenu de ces mémoires sans être obligés de les rappeler. Ce type d'utilisation est appelé «arithmétique en mémoire» et sera l'objet de l'exemple qui suit. Remarquez en outre, qu'un programme sur une application très courante peut être développé uniquement en utilisant les fonctions élémentaires ce qui confirme l'idée que nous avions formulée au départ: même pour la résolution de problèmes très simples vous pouvez gagner du temps avec votre calculatrice.

Le traitement d'une commande nécessite d'effectuer la multiplication pour chaque ligne du nombre d'articles par le prix unitaire de celui-ci pour obtenir le prix par ligne. Celui-ci est ensuite cumulé pour donner le montant total. Vous pouvez également demander d'avoir le nombre d'articles total commandés ainsi que le prix moyen par article. Bien que très simple, ce problème prend du temps.

Numéro de ligne	Quantité	Prix unitaire	Prix par ligne
1	100	Frs 0.25	Frs 25.00
2	200	0.15	30.00
3	50	0.35	17.50
4	150	0.40	60.00
5	300	0.10	30.00
Prix total	800		Frs 162.50
Prix moyen par article		Frs 0.203125	

IV



Nous n'avons que des additions, multiplications et divisions à faire pour résoudre cet exercice. Le problème consiste simplement à l'ordonner pour la calculatrice.

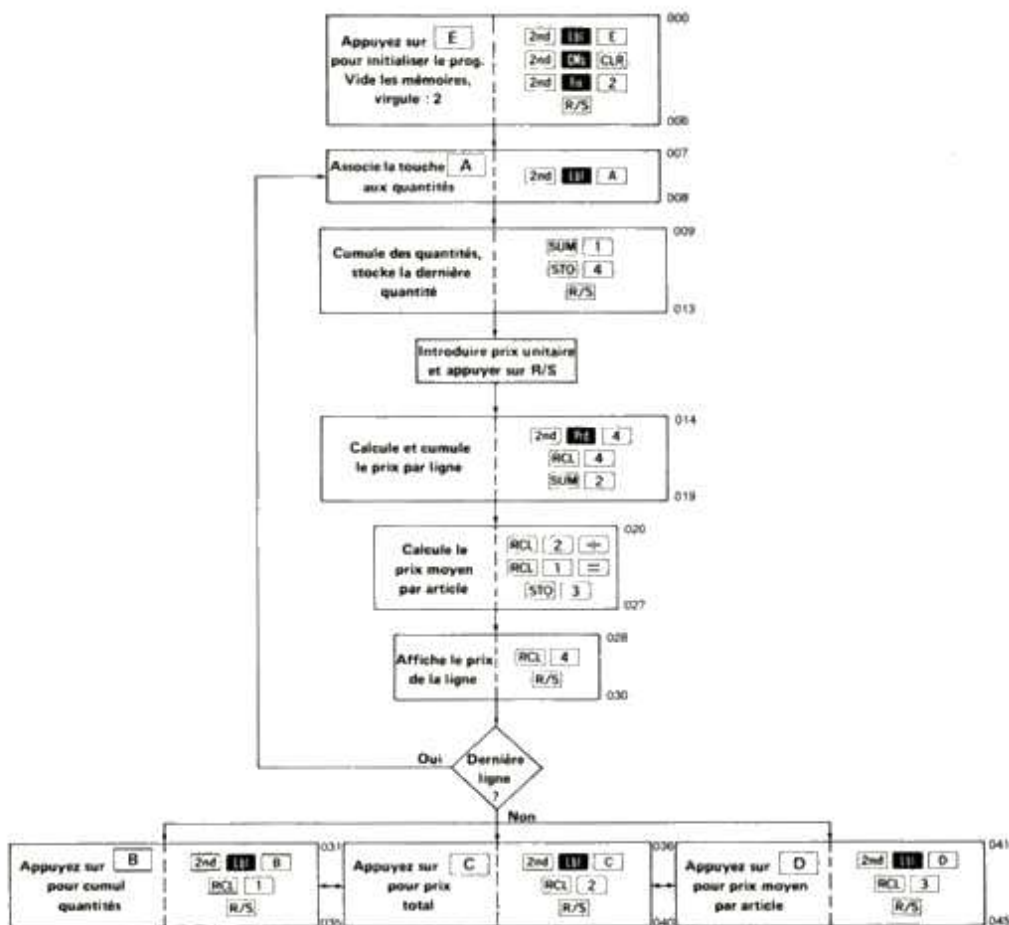
Si vous désirez que votre programme puisse traiter un nombre de ligne illimité, il faut utiliser une autre méthode que celle présentée ci-dessous. Dans cet exemple, les variables seront introduites par l'affichage, il n'est pas nécessaire en effet, de les stocker puisqu'elles n'auront pas à être réutilisées par la suite. Les différents cumuls seront stockés en mémoire. Le cumul des quantités sera stocké en mémoire 1, noté M_1 par la suite; le cumul des totaux par ligne seront stockés en M_2 et le prix moyen par article en M_3 . Après chaque ligne, vous voyez apparaître le montant de la ligne et pouvez appeler l'un des résultats finaux que vous désirerez.

Remarque : Du fait que nous allons faire des cumuls dans les mémoires M_1 et M_2 , il est nécessaire de prévoir une séquence de remise à zéro avant chaque réutilisation du programme.

Cet exemple vous montre à nouveau l'importance de bien définir l'approche du problème. Dans cette première étude, nous traitons un cas limité en nombre de lignes, dans un deuxième temps, vous pouvez envisager la façon de pouvoir introduire un nombre illimité de lignes.



Faisons l'organigramme du problème et définissons les séquences de touches.



Programme du contrôle de prix.



MODE D'EMPLOI				
Séquence	Procédure	Introduire	Appuyez	Affichage
1	Effacer la mémoire programme et remettre le pointeur au pas 000		2nd ☺	
2	Mise en mode programmation		LRN	000 00
3	Introduire les instructions			
4	Retour en mode calcul		LRN	0
5	Initialiser le programme		E	00.00
6	Introduire la quantité de la ligne	Quantité	A	Quantité
7	Introduire le prix unitaire reprendre la séquence 6 et 7 pour chaque ligne. Après chaque ligne les résultats suivants peuvent être obtenus :	Prix unitaire	R/S	Montant de la ligne
	Quantité cumulée		B	Total quantité
	Prix total		C	Prix total
	Prix moyen par article		D	Prix moyen



IV

Pas et code instruction	Séquence de touches	Pas et code instruction	Séquence de touches
000 76	2nd (1)	024 01	1
001 15	E	025 95	=
002 47	2nd (M)	026 42	STO
003 25	CLR	027 03	3
004 58	2nd (F)	028 43	RCL
005 02	2	029 04	4
006 91	R/S	030 91	R/S
007 76	2nd (1)	031 76	2nd (1)
008 11	A	032 12	B
009 44	SUM	033 43	RCL
010 01	1	034 01	1
011 42	STO	035 91	R/S
012 04	4	036 76	2nd (1)
013 91	R/S	037 13	C
014 49	2nd (F)	038 43	RCL
015 04	4	039 02	2
016 43	RCL	040 91	R/S
017 04	4	041 76	2nd (1)
018 44	SUM	042 14	D
019 02	2	043 43	RCL
020 43	RCL	044 03	3
021 02	2	045 91	R/S
022 55	÷		
023 43	RCL		

Liste des instructions du programme de contrôle de prix.

IV



Essayons le programme avec les valeurs indiquées précédemment :

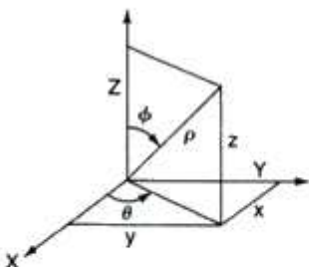
Introduire	Appuyez	Affichage	Commentaires
	<input type="button" value="E"/>	0.00	Initialisation
100	<input type="button" value="A"/>	100.00	Quantité A
.25	<input type="button" value="R/S"/>	25.00	Prix unitaire A
			Total de la ligne
200	<input type="button" value="A"/>	200.00	Quantité B
.15	<input type="button" value="R/S"/>	30.00	Prix unitaire B
			Total de la ligne
50	<input type="button" value="A"/>	50.00	Quantité C
.35	<input type="button" value="R/S"/>	17.50	Prix unitaire C
			Total de la ligne C
150	<input type="button" value="A"/>	150.00	Quantité D
.4	<input type="button" value="R/S"/>	60.00	Prix unitaire D
			Total de la ligne
300	<input type="button" value="A"/>	300.00	Quantité E
.1	<input type="button" value="R/S"/>	30.00	Prix unitaire E
			Total de la ligne
	<input type="button" value="B"/>	800.00	Quantité totale
	<input type="button" value="C"/>	162.50	Prix total
	<input type="button" value="D"/>	0.20	Prix moyen par article (arrondi)
	<input type="button" value="INV"/> <input type="button" value="2nd"/> <input type="button" value="fix"/>	0.203125	Prix moyen par article (exact)



CONVERSION DE COORDONNEES SPHERIQUES.

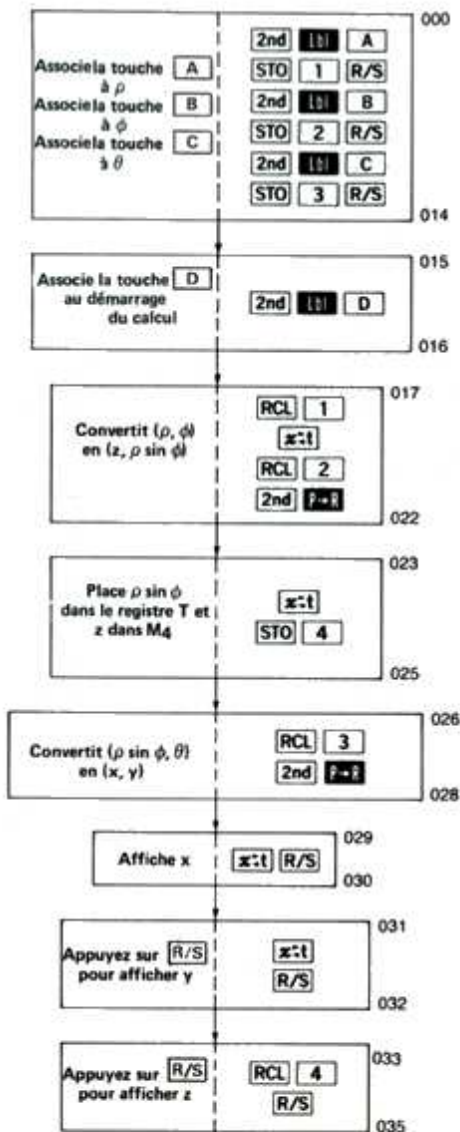
Ecrivons le programme effectuant la conversion de coordonnées sphériques en coordonnées cartésiennes.

$$\begin{aligned}x &= \rho \sin \phi \cos \theta \\y &= \rho \sin \phi \sin \theta \\z &= \rho \cos \phi\end{aligned}$$



Votre calculatrice dispose d'une fonction $\boxed{2^{nd}} \boxed{P \rightarrow R}$ préprogrammée pour convertir les coordonnées polaires en coordonnées cartésiennes. (Voir chapitre **Conversions** à la section V). Cette fonction sera utilisée ici.

Le plus simple pour ce programme est de stocker les paramètres ρ , ϕ , θ en mémoires M1, M2 et M3 respectivement. Placer ensuite ρ dans le registre T et ϕ dans le registre d'affichage, z sera obtenu en utilisant la fonction $\boxed{2^{nd}} \boxed{P \rightarrow R}$. Cette première conversion place $\rho \sin \phi$ dans le registre d'affichage et $z = \rho \cos \phi$ dans le registre T. Nous garderons z en M4, plaçons ensuite $\rho \sin \phi$ dans le registre T et θ dans le registre d'affichage; en utilisant à nouveau la fonction $\boxed{2^{nd}} \boxed{P \rightarrow R}$ nous obtenons y dans le registre d'affichage et x dans le registre T. Le programme sera construit pour obtenir x, y, z en appuyant sur $\boxed{R \rightarrow S}$.



Programme de conversion de coordonnées sphériques.



IV

MODE D'EMPLOI				
Séquence	Procédure	Introduire	Appuyer	Affichage
1	Effacer la mémoire programme et placer le pointeur au pas 000		2nd 09	
2	Mise en mode programmation		LRN	000 00
3	Introduire le programme			
4	Retour en mode calcul		LRN	
5	Introduire ρ	ρ	A	ρ
6	Introduire ϕ	ϕ	B	ϕ
7	Introduire θ	θ	C	θ
8A	Calculer les coordonnées et afficher x		D	x
8B	Afficher y		R/S	y
8C	Afficher z		R/S	z

IV



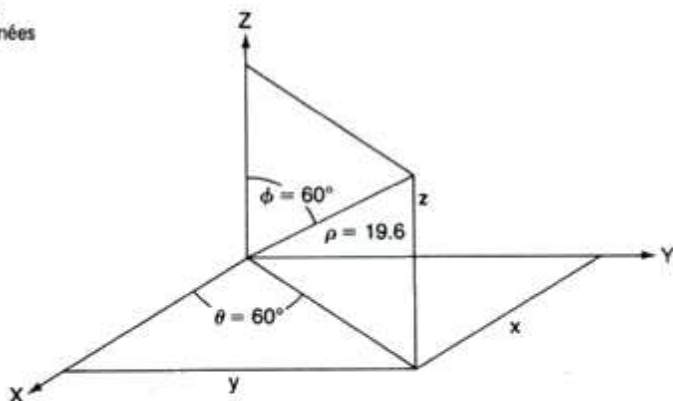
Pas et code instruction	Séquence de touches	Pas et code instruction	Séquence de touches
000 76	2nd ln	019 32	x⁻¹
001 11	A	020 43	RCL
002 42	STO	021 02	2
003 01	1	022 37	2nd F_{→I}
004 91	R/S	023 32	x⁻¹
005 76	2nd ln	024 42	STO
006 12	B	025 04	4
007 42	STO	026 43	RCL
008 02	2	027 03	3
009 91	R/S	028 37	2nd F_{→I}
010 76	2nd ln	029 32	x⁻¹
011 13	C	030 91	R/S
012 42	STO	031 32	x⁻¹
013 03	3	032 91	R/S
014 91	R/S	033 43	RCL
015 76	2nd ln	034 04	4
016 14	D	035 91	R/S
017 43	RCL		
018 01	1		

Liste des instructions du programme de conversion de coordonnées sphériques.



IV

Exemple : Convertir $\rho = 19,6$
 $\phi = 60^\circ$, $\theta = 60^\circ$ en coordonnées
cartésiennes.



Introduire	Appuyer	Affichage	Commentaires
	2nd Deg		Met la calculatrice en mode degrés
19.6	A	19.6	ρ
60	B	60	ϕ
60	C	60	θ
	D	8.487048957	x
	R/S	14.7	y
	R/S	9.8	z



PROGRAMMATION AVANCEE

COMPLEMENT SUR LES ETIQUETTES

Comme nous avons vu, les touches utilisateurs (A-E, A'-E') sont utilisables comme des étiquettes. Lorsqu'une séquence de programme est repérée par une de ces étiquettes, le fait d'appuyer sur cette touche a pour effet de placer le pointeur à l'endroit indiqué et de faire démarrer aussitôt l'exécution du programme.

En plus de ces touches utilisateurs vous avez la possibilité d'utiliser certaines des fonctions premières ou secondes de la calculatrice comme étiquettes. Par exemple $\boxed{2nd} \boxed{sin}$, $\boxed{x^2}$, $\boxed{=}$, \boxed{CLR} , $\boxed{2nd} \boxed{RPN}$, \boxed{EE} , $\boxed{2nd} \boxed{Fix}$ et d'autres encore qui sont appelées des **étiquettes simples**. Seules les touches qui suivent **ne peuvent pas** être utilisées comme étiquettes.

- \boxed{Lbl} $\boxed{2nd} \boxed{SST}$
- \boxed{Int} \boxed{LRN} \boxed{BST}
- \boxed{Del} \boxed{Ind} et les chiffres 0 à 9.

La seule différence existant entre les étiquettes simples et les étiquettes touches-utilisateur réside dans le fait que si vous appuyez sur la touche correspondant à une étiquette simple, la calculatrice ne démarrera pas l'exécution du programme à l'endroit choisi, mais exécutera la fonction appelée. Par exemple, si vous avez un segment de programme repéré par l'étiquette x^2 en appuyant sur $\boxed{x^2}$ vous effectuerez le carré du nombre affiché. Pour démarrer le programme à cet endroit, vous devez faire $\boxed{GTO} \boxed{x^2} \boxed{R/S}$. Vous avez plus de 60 étiquettes supplémentaires à votre disposition.

Les étiquettes simples peuvent être intercalées à n'importe quel endroit dans le programme à condition de ne pas scinder des instructions de la forme $\boxed{STO} \boxed{12}$, $\boxed{2nd} \boxed{Fix} \boxed{6}$ ou $\boxed{INV} \boxed{2nd} \boxed{sin}$.

INSTRUCTIONS DE TRANSFERT

Plusieurs instructions ont une importance capitale et vous offriront une souplesse de programmation considérable, vous permettant de contrôler l'ordre d'exécution de votre programme. Ce sont les **instructions de transfert** appelées **transfert**. Elles permettent de quitter l'ordre séquentiel du déroulement pour aller se brancher à une autre séquence. Ces transferts sont de deux types : **transferts inconditionnels** et **transferts conditionnels**.

Le transfert inconditionnel branche le pointeur à l'adresse qui lui est indiquée inconditionnellement. Il est indépendant des calculs eux-mêmes. Un transfert conditionnel va effectuer un branchement suite à un test, si la valeur testée ne répond pas à la condition requise, il saute l'ordre de branchement pour continuer l'exécution du programme.



TRANSFERT INCONDITIONNEL

RST, **GTO** et **SBR** sont les instructions de transfert inconditionnel. **RST** ramène automatiquement le pointeur au pas 000. **GTO** et **SBR** placent inconditionnellement le pointeur à l'endroit que vous avez défini. Notez que **RST** à d'autres fonctions que nous étudierons page 43.

INSTRUCTION GO TO – **GTO** . (ALLER A).

Au chapitre Correction d'un programme nous avons vu la façon d'utiliser l'instruction Go To (aller à) au clavier. Le principe est identique quand on l'utilise dans un programme. **GTO** suivie d'une adresse numérique ou d'une étiquette provoque un saut du pointeur à l'endroit désigné en maintenant le déroulement du programme à partir de cet endroit.

Il est possible d'utiliser une forme contractée par exemple **GTO 9** est identique à **GTO 009** si les instructions qui suivent ne sont pas numériques. Introduisez ce petit programme de comptage à titre d'exemple :

Appuyez	Affichage	Commentaires
CLR 2nd CP	0.	Préparation pour la programmation
GTO 9 LRN	009 00	Aller au pas 9 et mise en mode programmation.
+	010 00	
1	011 00	
=	012 00	
2nd Pause	013 00	
GTO	014 00	
9	014 00	Le compteur de pas n'avance pas, il attend l'adresse complète
LRN	0.	Retour en mode calcul.
LRN	016 00	Le compteur de pas à avancé quand vous avez appuyé sur le premier LRN indiquant que l'adresse était complète.

Pour utiliser ce programme, appuyez sur **GTO 9 R/S** .

Dans ce programme, le transfert inconditionnel occupe deux pas. La séquence **GTO 136** en occupera trois : **GTO** se trouve dans le premier pas, le nombre des centaines dans le deuxième et les deux derniers nombres contractés dans le troisième pas ce qui s'exprime par les codes «61» «01» «36» en mémoire programme. La contraction est automatique.

IV



Introduisez l'exercice suivant dans votre calculatrice.

Appuyez	Affichage	Commentaires
2nd GP CLR	0.	Efface la mémoire programme et l'affichage.
GTO 136	0.	Place le pointeur au pas 136.
LRN	136 00	
R/S	137 00	Stocke R/S au pas 136.
LRN RST LRN	000 00	Revient au pas 000.
GTO	001 00	Stocke GTO au pas 000.
1	001 00	
3	001 00	
6	003 00	La mémoire programme attend les 3 chiffres avant de stocker l'adresse.
BST	002 36	
BST	001 01	Adresse stockée comme il a été expliqué.
BST	000 61	
LRN	0.	
R/S	0.	Exécute le programme.
LRN	137 00	Le saut est effectué immédiatement, l'instruction R/S au pas 136 arrête le programme au pas 137.

Il est possible de faire la même chose avec une étiquette. L'instruction **GTO** **∞¹** par exemple entraînera un saut à l'étiquette **∞¹** et la poursuite du programme à partir de cet endroit.

Vous remarquerez que l'instruction **GTO** travaille de la même façon lorsqu'elle est utilisée au clavier ou rencontrée dans un programme.

Si vous demandez un transfert à une étiquette qui n'existe pas dans le programme, l'affichage clignotera.



SOUS-PROGRAMMES

Dès que vous commencez à écrire des programmes, vous pouvez vous apercevoir que certaines séquences sont répétitives ; c'est ce que nous appellerons des **sous-programmes**. Plutôt que d'être obligé de réécrire une séquence autant de fois qu'elle se présente, l'utilisation d'un sous-programme vous permettra d'effectuer une seule écriture, une seule séquence d'instructions que vous pourrez venir exécuter autant de fois que ce sera nécessaire à partir d'un endroit quelconque du programme, le pointeur reviendra se positionner après exécution du sous-programme à l'instruction qui suivra celle d'où il est parti pour continuer le déroulement du programme.

Chaque programme peut être écrit de façon à être utilisé comme un sous-programme, il est alors utilisable sans modification à partir d'un autre programme ; ce qui se réalise simplement en utilisant les instructions **INV SBR** à la place de **R/S** pour arrêter l'exécution de celui-ci. Les programmes qui suivront seront écrits de cette façon.

INSTRUCTION DE SOUS-PROGRAMME **SBR**

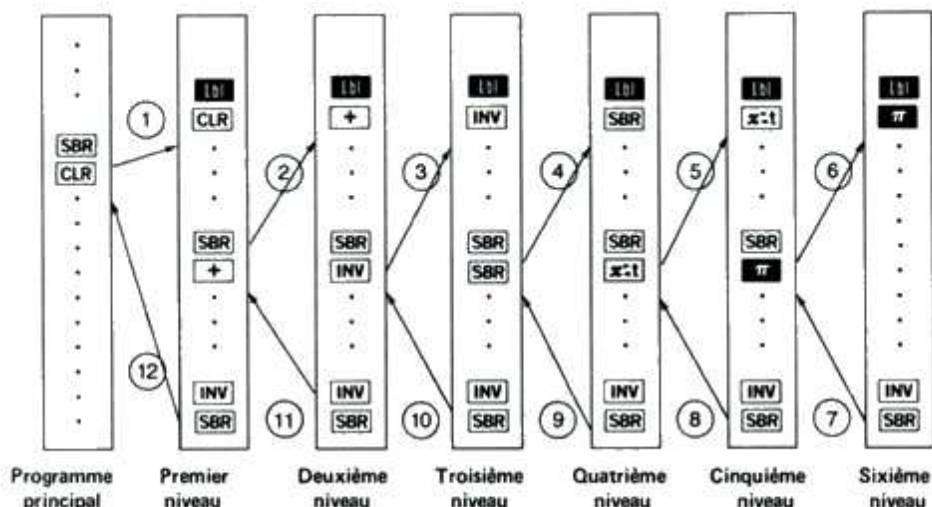
L'instruction de sous-programme diffère du **GTO** par deux caractéristiques : premièrement, utilisée depuis le clavier, elle pourra faire démarrer l'exécution d'un programme ; deuxièmement, elle mémorisera l'adresse de départ du transfert. Au clavier, appuyer sur **SBR 136** place le pointeur instantanément au pas 136 et démarre l'exécution du programme automatiquement à partir de cet endroit. L'effet est équivalent à **GTO 136 R/S**. Il en est de même si vous appuyez sur **SBR** **∞**. le programme démarrera à partir de l'étiquette **∞** où qu'elle soit.

Si l'instruction **SBR 136** remplace aux pas.000-002, l'instruction **GTO 136** dans le programme précédent, elle aura un effet similaire à l'usage, mais en plus elle stockera l'adresse 003 dans une mémoire particulière appelée **mémoire de retour de sous-programme**. Si maintenant au pas 136 nous avons une séquence qui se termine par **INV SBR**, le contenu de la mémoire de retour sera appelée (ici 003) et l'exécution du programme se poursuivra automatiquement à partir de cette adresse.

IV



Dans les mémoires de retour des sous-programmes, il est possible de stocker jusqu'à six adresses de retour en même temps. Ceci signifie qu'un sous-programme peut en appeler un suivant qui peut également en appeler un autre, etc..., ainsi de suite, jusqu'à 6 appels successifs. Nous allons schématiser cette possibilité :



Si une séquence est écrite dans cette forme, l'exécution se fera dans l'ordre indiqué par les numéros 1 à 12. On notera que **INV SBR** termine chaque sous-programme indiquant à la calculatrice de revenir à l'instruction dont l'adresse a été stockée en mémoire de retour des sous-programmes, ceci dans l'ordre inverse où ces adresses ont été stockées. La procédure s'effectue jusqu'au retour au programmé principal là où s'est effectué le premier départ. En outre **INV SBR** est contracté en mémoire programme n'utilisant qu'un pas codé 92. Le code dans ce cas, n'indique pas les numéros de ligne et colonne correspondants.

Lorsqu'une partie du programme est étiquetée avec une touche utilisateur, elle peut être exécutée simplement en appuyant sur la touche correspondante comme nous l'avons vu. Il en est de même si une de ces touches est rencontrée dans la suite des instructions programme, le pointeur ira à cette étiquette et continuera l'exécution. Le fonction **SBR** est sous-entendue dans les touches-utilisateur, si bien que si une séquence de programme est définie par une de ces étiquettes et terminée par **INV SBR**, cette séquence sera interprétée comme un sous-programme sans avoir à utiliser l'instruction **SBR**.

Il est déconseillé d'utiliser une étiquette utilisateur seule pour effectuer un branchement inconditionnel. En effet l'instruction **SBR** est sous entendue, une adresse de retour de sous-programme est enregistrée et ceci peut conduire à des erreurs dans certaines circonstances. Dans ce cas faire précéder l'étiquette utilisateur de l'instruction **GTO**.



ACCEDER A OU APPELER UN SOUS-PROGRAMME.

Pour simplifier ces définitions, on peut dire qu'un sous-programme est une partie d'un programme écrite une seule et unique fois, mais utilisée à plusieurs reprises. Tout sous-programme doit être terminé par **INV** **SBR** pour assurer un retour au point de départ.

Nous avons trois possibilités pour appeler un sous-programme.

- Adresses numériques **SBR** **136**
- Etiquettes simples **SBR** **x²**
- Etiquettes - touches-utilisateur **A**

L'utilisation d'étiquettes clarifie et simplifie les instructions du programme. Elle permet dans la pratique, de donner un nom à chaque séquence en correspondance avec l'étiquette choisie. En outre, vous gardez la possibilité de placer un sous-programme à n'importe quel endroit dans la mémoire programme ainsi que de faire d'éventuelles corrections sans avoir à modifier les adresses.

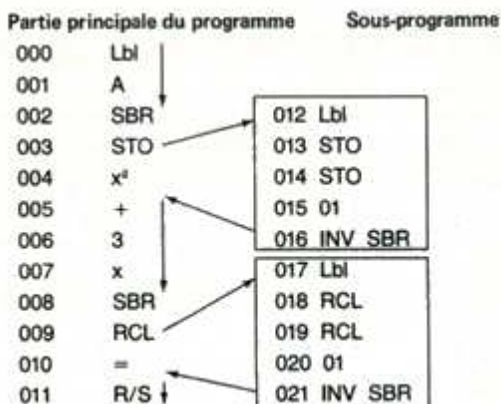
Calculons $x^2 + 3x$ pour différentes valeurs de x , les instructions sont les suivantes :

STO
1
x²
+
3
X
RCL
1
=
R/S

IV



Par exemple, demandons que les stockages et rappels mémoire soient traités par sous-programme, nous aurons :



Composez la valeur de x et appuyez sur **A**, le déroulement du programme s'effectuera suivant les flèches. Le départ du programme se fait avec une touche-utilisateur ce qui est plus pratique pour introduire x. Les sous-programmes sont nommés de façon à identifier leurs fonctions mais toutes autres étiquettes pourraient être utilisées.

PRECAUTION A PRENDRE DANS UN SOUS-PROGRAMME.

RST et **=** sont deux instructions à utiliser avec beaucoup de vigilance dans un sous-programme.

L'instruction **RST** en plus de sa première fonction va également effacer automatiquement le contenu des mémoires de retour des sous-programmes. Si vous désirez ramener le pointeur au pas 000 (première fonction de **RST**) à l'intérieur d'un sous-programme, vous devez utiliser **GTO 000** ou **GTO** une étiquette si l'une d'elle est placée au pas 000.

L'instruction égale provoque l'exécution des différentes opérations en attente, utilisée dans un sous-programme, vous risquez de déclencher également les calculs en attente dans le programme principal.



IV

Par exemple, écrivons le programme calculant $4 + (1 + 2) \times 3$.

```
4
+
SBR x²
X
3
=
R/S
2nd (1) x²
1
+
2
=
INV SBR
```

La touche égale dans le sous-programme x^2 calcule non seulement $1 + 2$, mais ajoute 4 à ce résultat avant de revenir sur $\times 3$. Le résultat est 21 alors qu'il devrait être 13.

Une modification simple peut corriger ceci en utilisant les parenthèses dans le sous-programme.

```
4
+
SBR x²
X
3
=
R/S
2nd (1) x²
(
1
+
2
)
INV SBR
```

Ce programme vous donne la bonne réponse : 13.

IV



Placer une () au début de chaque sous programme et une) juste avant l'instruction **INV** **SBR** est une bonne habitude à prendre. Vous utilisez un pas programme supplémentaire par rapport à l'utilisation de l'instruction **=** , mais vous pouvez éviter de facheuses conséquences. L'avantage essentiel est que ces parenthèses n'affectent pas les opérations en attente dans le programme principal.

L'utilisation des parenthèses dans un cas pareil pour encadrer une séquence traitée dans un sous-programme, n'engendre aucune difficulté supplémentaire par rapport à l'usage de la touche **=** . Néanmoins il y a une particularité que vous devez connaître pour réutiliser le contenu du registre d'affichage dans un sous-programme.

Pour éviter de rappeler le contenu de l'affichage au moment du départ vers le sous-programme en faisant un stockage mémoire préalable puis un rappel, vous avez la possibilité d'utiliser la touche **CE** qui rappellera le contenu du registre d'affichage pour l'introduire dans les parenthèses. Cette particularité est valable aussi bien en cours de programmation qu'au clavier. Exemple :

Appuyez : 2.18 **X** (**CE** + 6) **=**

Affichage : 17.8324

Dans l'exemple ci-dessus la touche **CE** ramène la valeur 2.18 à l'intérieur des parenthèses de telle sorte que la calculatrice exécute : $2.18 \times (2.18 + 6) = 17.8324$.

Dans certains cas, il se peut que vous obteniez un résultat désiré non pas à la fin du programme principal, mais à l'intérieur d'un sous-programme. Il n'est plus nécessaire dans ces conditions de revenir au programme principal pour arrêter le traitement de vos calculs et repartir sur un nouveau cycle d'opérations ou sur un nouveau programme. Néanmoins, avant de redémarrer le nouveau cycle, vous devez vous assurer que les registres de retour des sous-programmes sont correctement effacés pour éviter que les résultats suivants ne soient fantaisistes. Pour cela vous pouvez, soit éteindre votre calculatrice, soit utiliser l'instruction **2nd** **CP** , soit encore l'instruction **RST** . Pour prévenir les erreurs que pourraient entraîner le non-effacement de ces registres de retour de sous-programmes, il est recommandé d'utiliser l'instruction **RST** , soit manuellement, soit de la placer à un endroit correct dans le programme.



UTILISATION DES PROGRAMMES DE BIBLIOTHEQUES COMME SOUS-PROGRAMMES.

Souvent vous voudrez trouver une façon simple d'étendre la puissance d'un programme que vous avez écrit en exploitant l'un de ceux qui existent dans une bibliothèque. Les mêmes instructions que celles que nous avons utilisées au clavier pour appeler un de ces programmes, vont permettre en les insérant dans votre programme d'utiliser les programmes des bibliothèques en sous-programmes du vôtre. Les milliers d'instructions constituant l'éventail des programmes stockés dans chaque module de bibliothèque sont appelables comme sous-programme. Aucun d'eux n'utilisent, ni l'instruction `=`, ni l'instruction `RST` et leurs calculs sont effectués sous parenthèses, achevés par l'instruction `INV` `SBR`.

Au clavier nous faisons `2nd` `Pgm` `mm` pour aller chercher le programme numéro `mm`. Le pointeur restait positionné sur ce programme à la fin de son exécution.

Dans un programme, les instructions `Pgm` et `SBR` sont identiques si ce n'est que `2nd` `Pgm` `mm` appellera le sous programme `mm` contenu dans la bibliothèque et non dans la calculatrice elle-même. Dès que le sous programme `mm` est achevé, le pointeur retourne dans la mémoire programme à l'endroit d'où il était parti et poursuit l'exécution du programme.

Si un segment de la bibliothèque est identifié par une touche utilisateur telle que `A`, pour venir exécuter cette séquence, on utilisera `2nd` `Pgm` `mm` `A`. Si une étiquette est utilisée `tan` par exemple, on utilisera alors `2nd` `Pgm` `mm` `SBR` `2nd` `tan`. Notons que `2nd` `Pgm` `mm` s'il n'est pas suivi de `SBR` ou d'une touche utilisateur est invalide et ne produira pas l'effet souhaité.

IV



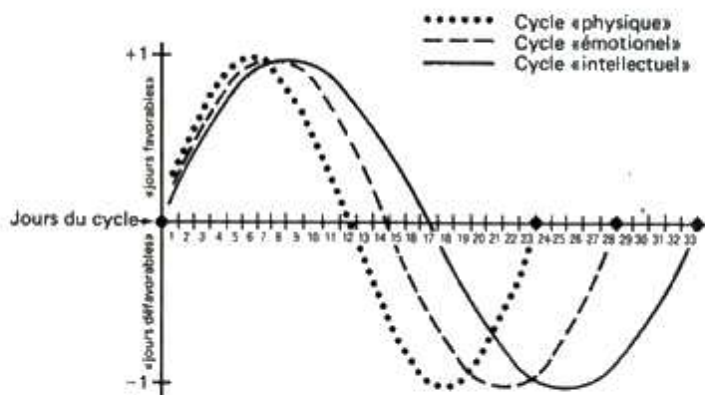
BIORYTHME

Examinons un programme illustrant les différentes possibilités des sous-programmes.

La théorie du Biorythme repose sur le fait que trois cycles influent sur votre vie. Ces cycles ayant leurs origines communes au jour de votre naissance :

1. Le cycle physique dure 23 jours.
2. Le cycle émotionnel dure 28 jours.
3. Le cycle intellectuel dure 33 jours.

La première moitié de chaque cycle est considérée comme étant la «partie favorable» et la deuxième moitié, la «partie défavorable».



L'amplitude du Biorythme est déterminée par une sinusoïde d'amplitude comprise en +1 et -1 et répondant à la formule :

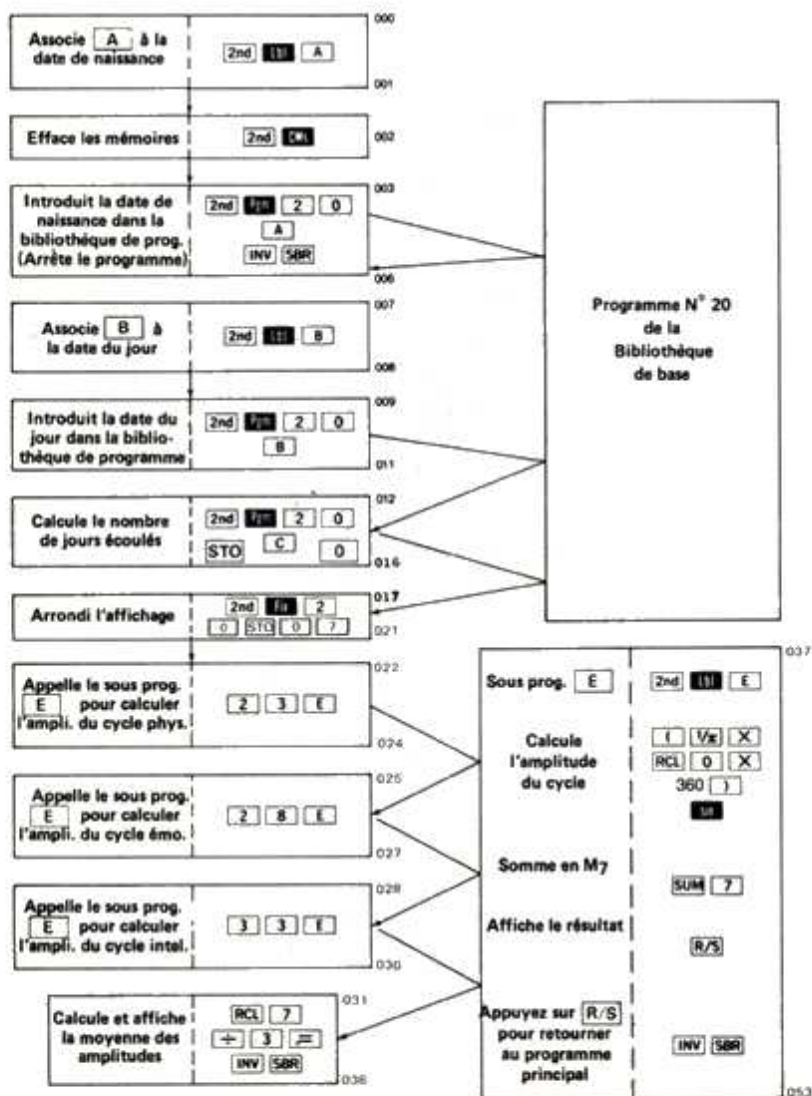
$$\text{Amplitude} = \sin \left(360 \times \frac{\text{Nombre de jours depuis la naissance}}{\text{Durée du cycle}} \right)$$

Ecrivons le programme permettant de déterminer l'amplitude du cycle biorythmique d'une personne. Chaque cycle étant basé sur la même équation, nous simplifierons le programme en utilisant un sous-programme; nous utiliserons également le programme 20 de la bibliothèque de base comme sous-programme pour calculer le nombre de jours écoulés depuis la date de naissance.

Dans l'exemple suivant, l'amplitude de chaque cycle est calculée dans le sous-programme étiqueté **E**. La touche **R/S** permettra d'afficher l'amplitude des cycles physiques, émotionnels et intellectuels dans cet ordre. Le résultat final du programme est la moyenne des trois amplitudes. Les résultats seront exprimés avec un arrondi à deux décimales.



IV



Organigramme du Biorythme.



Pas et codes instructions	Séquence de touches	Pas et codes instructions	Séquence de touches
000 76	2nd 1/x	027 15	E
001 11	A	028 03	3
002 47	2nd CM5	029 03	3
003 36	2nd Pgm	030 15	E
004 20	2 0	031 43	RCL
005 11	A	032 07	7
006 92	INV SBR	033 55	÷
007 76	2nd 1/x	034 03	3
008 12	B	035 95	=
009 36	2nd Pgm	036 92	INV SBR
010 20	2 0	037 76	2nd 1/x
011 12	B	038 15	E
012 36	2nd Pgm	039 53	I
013 20	2 0	040 35	1/x
014 13	C	041 65	X
015 42	STO	042 43	RCL
016 00	0	043 00	0
017 58	2nd 1/x	044 65	X
018 02	2	045 03	3
019 00	0	046 06	6
020 42	STO	047 00	0
021 07	0 7	048 54	I
022 02	2	049 38	2nd 1/x
023 03	3	050 44	SUM
024 15	E	051 07	7
025 02	2	052 91	R/S
026 08	8	053 92	INV SBR

Programme du Biorythme.



IV

MODE D'EMPLOI				
Séquence	Procédure	Introduire	Appuyer	Affichage
1	Effacer la mémoire programme et placer le pointeur au pas de dep.		2nd EP	
2	Mise en mode programmation		LRN	000 00
3	Introduire le prog. du biorythme			
4	Retour en mode calcul		LRN	
5	Introduire la date de naissance.	MMJJ.AAAA	A	0.
6	Introduire la date du jour et le calcul de l'amplitude du cycle physique	MMJJ.AAAA	B	Amplitude du cycle physique.
7	Calcul de l'amplitude du cycle émotionnel.		R/S	Amplitude du cycle émotionnel.
8	Calcul de l'amplitude du cycle intellectuel.		R/S	Amplitude du cycle intellectuel.
9	Calcul de l'amplitude moyenne.		R/S	Amplitude moyenne

Exemple : Paul est né le 2 mai 1944, calculons son biorythme au 1er mars 1977. (Assurez-vous que le programme du biorythme est en mémoire programme).

Appuyer	Affichage	Commentaires
502.1944 A	0.	Date de naissance.
301.1977 B	0.82	Amplitude du cycle physique.
R/S	1.00	Amplitude du cycle émotionnel.
R/S	0.76	Amplitude du cycle intellectuel.
R/S	0.86	Amplitude moyenne

Pour Paul, le biorythme est favorable, regardez pour vous quels seront les résultats.

Lorsque vous utilisez en sous programme un programme de la bibliothèque de base, soyez très prudent sur les mémoires que vous utilisez dans votre programme principal : elles ne doivent pas être les mêmes que celles utilisées par le programme de la bibliothèque de base pour éviter tout résultat erroné.



Transferts conditionnels (Prise de décision).

Une caractéristique très utile dans la résolution de problèmes est de pouvoir prendre des décisions dans le programme. Cette possibilité de prise de décision est fournie par les instructions de transfert conditionnel. A chaque fois que l'une d'elles est placée dans un programme, un test sera fait et du résultat de ce test sera décidé, ou non, un transfert.

Il y a trois types d'instructions de transfert conditionnel suivant ce que vous voulez décider.

1. Comparer le contenu du registre d'affichage avec le registre T **XC=I** , **XC=I** .
2. Tester le contenu d'une des mémoires 0 à 9 **0UI** .
3. Tester l'état d'un drapeau **II II** .

Ces instructions sont suivies d'une adresse de transfert. Si la réponse au test est «oui» un transfert est effectué à cette adresse ; si la réponse est «non» l'instruction de transfert est sautée. Par exemple, considérons **2nd** **XC=I** **A** , si le contenu du registre d'affichage est égal à celui du registre T un transfert est effectué à l'étiquette **A** comme si nous avions l'instruction **GTO** **A** , dans le cas contraire, l'instruction de transfert est ignorée.

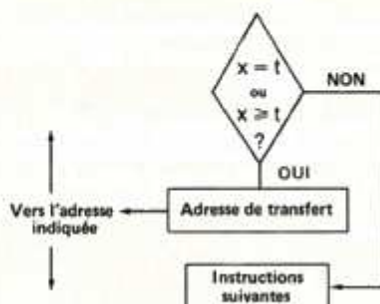
REGISTRE D'AFFICHAGE – REGISTRE T.

Qu'est ce que le registre T ? Il s'agit d'un registre particulier dans lequel nous allons pouvoir stocker, puis rappeler un nombre et le tester par rapport au contenu du registre d'affichage. La touche **XC=I** «x échange t» permute le contenu des deux registres, le contenu de l'affichage «x» se retrouve dans le registre T et le contenu du registre T dans le registre d'affichage. Initialement 0 se trouve dans le registre T, introduisons 5 sur l'affichage et faisons **XC=I** ; on obtient 0 sur l'affichage, si nous réappuyons à nouveau sur **XC=I** on retrouve le 5 sur l'affichage.



IV

Deux instructions permettent de comparer le contenu du registre T et du registre d'affichage : « x égal à t » $\boxed{x=t}$ et « x plus grand ou égal à t » $\boxed{x \geq t}$. Ces instructions doivent être suivies d'une adresse numérique ou d'une étiquette. Quand un test est placé, le transfert s'exécutera pour une réponse « oui » à l'adresse numérique ou à l'étiquette placée juste après l'instruction de test; si la réponse est « non » l'adresse ou l'étiquette est ignorée et l'exécution du programme se poursuit. Ce qui donne graphiquement :



Ces instructions sont conçues pour être utilisées au cours d'un programme. Néanmoins elles peuvent aussi être utilisées à partir du clavier comme indiqué ci-dessous :

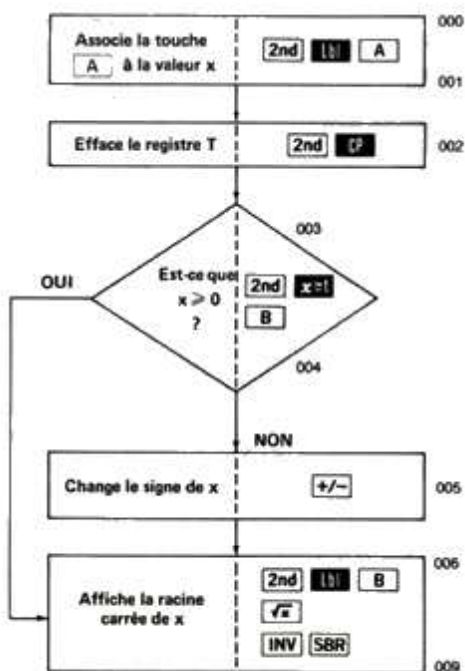
Appuyer	Affichage	Commentaires
5 $\boxed{x=t}$	0.	Place 5 dans le registre T
6 $\boxed{2nd} \boxed{x \geq t}$ 123	6.	Place 6 sur l'affichage et ordonne au pointeur d'aller au pas 123 si $x > t$.
\boxed{LRN}	123 00	Le transfert a été effectué car 6 est supérieur à 5.
\boxed{LRN} 4 $\boxed{2nd} \boxed{x \geq t}$ 111	4.	Retour au mode calcul et test pour $x = 4$.
\boxed{LRN}	123 00	On est resté au pas 123, pas de transfert car 4 n'est pas supérieur à 5.
\boxed{LRN} 5 $\boxed{2nd} \boxed{x \geq t}$ 111	5.	Retour au mode calcul et test pour $x = 5$.
\boxed{LRN}	111 00	Le transfert est effectué car $x = t = 5$.

IV



EXEMPLE : RACINE CARREE.

Problème : Trouver la racine carrée d'un nombre; si celui-ci est négatif, changer son signe et ensuite prendre la racine carrée.





Pas et codes instructions	Séquence de touches
000 76	2nd ↵
001 11	A
002 29	2nd CF
003 77	2nd x=1
004 12	B
005 94	+/-
006 76	2nd b
007 12	B
008 34	√x
009 92	INV SBR

Vous pouvez utiliser cet exemple : introduisez 4, appuyez sur **A** la réponse est 2. Introduisez -4 et appuyez sur **A** la réponse reste toujours 2.

Ces instructions de test peuvent être utilisées avec l'instruction **INV** pour inverser les conditions de transfert comme nous allons le voir:

Séquence d'instructions

2nd **x=1**

INV **2nd** **x≠1**

2nd **x<1**

INV **2nd** **x>1**

Question posée (test effectué)

Le contenu du registre d'affichage est-il exactement égal au contenu du registre T ?

Le contenu du registre d'affichage est-il différent du contenu du registre T ?

Le contenu du registre d'affichage est-il plus grand ou égal au contenu du registre T ?

Le contenu du registre d'affichage est-il plus petit que le contenu du registre T ?

A la réponse «oui» à ces différentes questions, un transfert sera effectué à l'adresse qui suit ces instructions. A la réponse «non» l'adresse sera ignorée et l'exécution poursuivie à l'instruction suivante.



DRAPEAUX

Qu'est-ce qu'un drapeau et comment l'utiliser dans un programme ? Un drapeau est un commutateur qui peut être, soit «ouvert», soit «fermé», un drapeau peut être, soit en position haute, soit en position basse. A l'origine ils sont en position basse et peuvent être mis isolément en position haute ou ramenés en position basse par une instruction: Leur état sera testé par la suite dans le programme. Ces opérations sont indépendantes des mémoires et du registre d'affichage.

Quand utilise-t-on un drapeau ? Les drapeaux ont de nombreuses utilisations dont celles-ci :

- Manuellement, contrôler une option prise sur le déroulement d'un programme.
- Dans un programme, identifier une condition en levant un drapeau qui sera testé par la suite.
- Garder une trace d'une exécution qui indiquera un cheminement spécifique effectué dans le programme pour un cas considéré.

Vous disposez de 10 drapeaux numérotés de 0 à 9 ce qui impose d'indiquer après une instruction concernant les drapeaux, le numéro de celui choisi. Ces instructions sont les suivantes :

- Appuyez sur **2nd** **St Flg** **Y** pour lever le drapeau y (Set Flag Y).
- Appuyez sur **INV** **2nd** **St Flg** **Y** pour baisser le drapeau Y.
- Appuyez sur **2nd** **If Flg** **Y** pour tester et exécuter un transfert si le drapeau Y est en position haute (transfert à l'adresse indiquée juste après l'instruction).
- Appuyez sur **INV** **2nd** **If Flg** **Y** pour tester et exécuter un transfert, si le drapeau Y est en position basse (transfert à l'adresse indiquée juste après l'instruction).



IV

Ces instructions s'utilisent au clavier de la même façon que dans un programme.

Appuyer	Affichage	Commentaires
2nd CP	0.	Efface la mémoire programme, le registre d'affichage, le registre T et remet les drapeaux en position basse.
2nd SI REG 4	0.	Lève le drapeau numéro 4.
2nd II REG 4 136	0.	Si drapeau 4 aller en 136.
LRN	136 00	Transfert effectué.
LRN 2nd II REG 5 111	0.	Si drapeau 5 aller en 111.
LRN	136 00	Pas de transfert.
LRN INV 2nd SI REG 4	0.	Abaisse le drapeau 4.
2nd II REG 4 222	0.	Si drapeau 4 aller en 222.
LRN	136 00	Pas de transfert.
LRN INV 2nd II REG 4 222	0.	Si absence drapeau 4 aller en 222.
LRN	222 00	Transfert effectué.

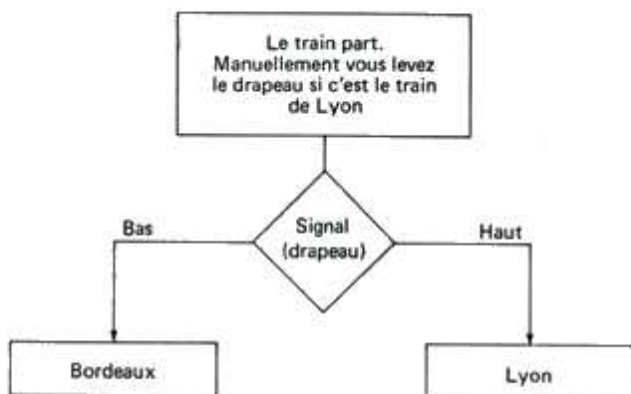
Le test des drapeaux est similaire au test du registre T. La seule différence est que d'un côté on teste l'état d'un drapeau, de l'autre on compare avec un autre nombre. Dans les deux cas l'adresse de transfert suit l'instruction de test et peut être, soit une adresse numérique, soit une étiquette.

Lever un drapeau déjà en position haute, baisser un drapeau déjà en position basse, tester un drapeau, ne change pas l'état du drapeau et n'affecte pas les calculs en cours. Tous les drapeaux peuvent être baissés ensemble avec **RST** ou **2nd** **CP**.

Vous remarquerez qu'il n'est pas possible de visualiser l'état d'un drapeau sur l'affichage si ce n'est par l'effet qu'il produira dans un test.



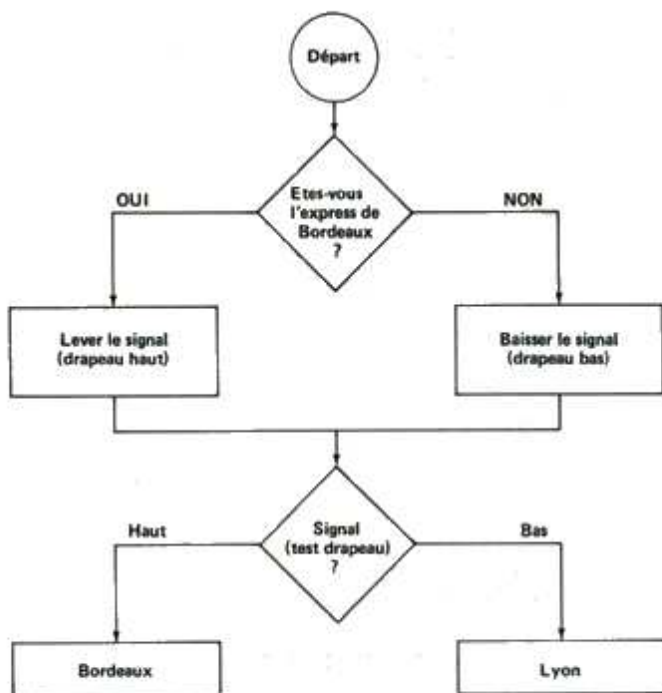
La situation suivante vous montrera la première des trois applications citées pour les drapeaux : l'utilisation manuelle au clavier. Supposons que vous soyez aiguilleur de chemin de fer. Un train démarre, il rencontre un signal précédant un aiguillage. Si le signal est levé le train est dirigé vers Lyon, s'il est baissé le train sera alors placé sur la voie de Bordeaux. En temps qu'aiguilleur vous devez lever ou baisser le signal qui indiquera au train la voie qu'il doit prendre. De la même façon vous pouvez manuellement lever ou baisser un drapeau pour définir les séquences de programme que vous désirez exécuter.



Utilisation manuelle d'un drapeau.

Lorsque vous travaillez avec un programme, vous pouvez manuellement décider de l'état des drapeaux qui président aux choix des séquences utilisées. Imaginons que nous ayons un contrôle de coût à effectuer; c'est-à-dire une série de débits et de crédits à faire digérer par un programme mis en machine. Pour éviter une introduction différente des débits et des crédits, vous pouvez par exemple lever un drapeau pour les débits ce qui permettra à votre programme de faire les imputations convenables de façon automatique suivant les éléments introduits.

Modifions maintenant le programme du train pour montrer comment le train peut lui-même lever ou baisser le drapeau. C'est l'exemple le plus fréquent d'utilisation de ces drapeaux. Voyons comment l'express de Lyon et de Bordeaux seront aiguillés dans la bonne direction.



Déroulement automatique contrôlé par un drapeau.

Le système interroge chaque train : «êtes-vous un express pour Bordeaux ?» si «oui» un drapeau est levé si «non» le drapeau est baissé. Le drapeau est ensuite analysé : en bas le train est envoyé sur Lyon, en haut sur Bordeaux et ceci automatiquement. De la même façon, un résultat de calcul sera identifié : «est-il négatif ?» ou «est-il plus grand que 1000 ?» ou toute autre question. Si la réponse est «oui» nous pourrions identifier cet état en levant un drapeau qui par la suite dans le programme nous permettra de choisir telle ou telle séquence à exécuter.

La troisième possibilité d'utilisation permet d'identifier le chemin pris par le programme pour arriver au résultat. Vous avez remarqué que l'exécution d'un programme s'effectue en aveugle et que par conséquent il est impossible à la vue du résultat de savoir si l'exécution s'est faite par telle ou telle séquence. La seule façon de le savoir est de mettre dans l'une des séquences l'instruction `2nd` `STOn` y et dans l'autre `INV` `2nd` `STOn` y. Le test `2nd` `ITOn` y après exécution vous donnera la réponse. (Il est nécessaire de mettre une instruction dans chaque séquence si on veut pouvoir réutiliser le programme sans risquer de branchement erroné). Ceci pourra vous servir si, par exemple, vous désirez savoir lequel des deux taux d'intérêt possibles a été appliqué dans un programme ou si le signe d'un nombre doit être changé avant de poursuivre les calculs.

IV



FONCTIONS SPECIALES DES DRAPEAUX

Certains drapeaux sont affectés à certaines fonctions particulières.

- Drapeau 7** **2nd** **0p** 18 demande à la calculatrice de lever le drapeau 7 s'il n'y a aucune condition d'erreur. Le drapeau 7 est levé par la séquence **2nd** **0p** 19 lorsqu'une condition d'erreur existe. (Voir page V-27).
En ce qui concerne la TI-58C, **2nd** **0p** 40 lèvera le drapeau 7 si la calculatrice est connectée au PC-100A, PC-100B ou PC-100C. (Voir page V-27 pour le détail des opérations spéciales).
- Drapeau 8** Le drapeau 8 levé demandera à la calculatrice d'arrêter le déroulement du programme dès qu'une condition d'erreur se produira.
- Drapeau 9** Si vous utilisez l'imprimante optionnelle PC-100A/B/C, vous pouvez demander l'impression de la trace de votre programme en levant le drapeau 9. Celui-ci baissé, seuls les calculs suivis de l'instruction «print» seront imprimés. Si vous utilisez la calculatrice sans le berceau imprimant le drapeau 9 peut être utilisé normalement.

PROGRAMME DE CONVERSIONS METRIQUES.

Ecrivons un programme qui convertisse les mètres en pieds et les kilomètres en miles. Nous avons plusieurs moyens de traiter ce problème. La méthode utilisée consiste à convertir l'introduction en pieds et regarder ensuite si la valeur introduite était en mètres ou en kilomètres, de façon à restituer des miles ou des pieds. M1 est utilisée pour stocker le résultat intermédiaire pendant l'exécution du test. Les relations métriques sont :

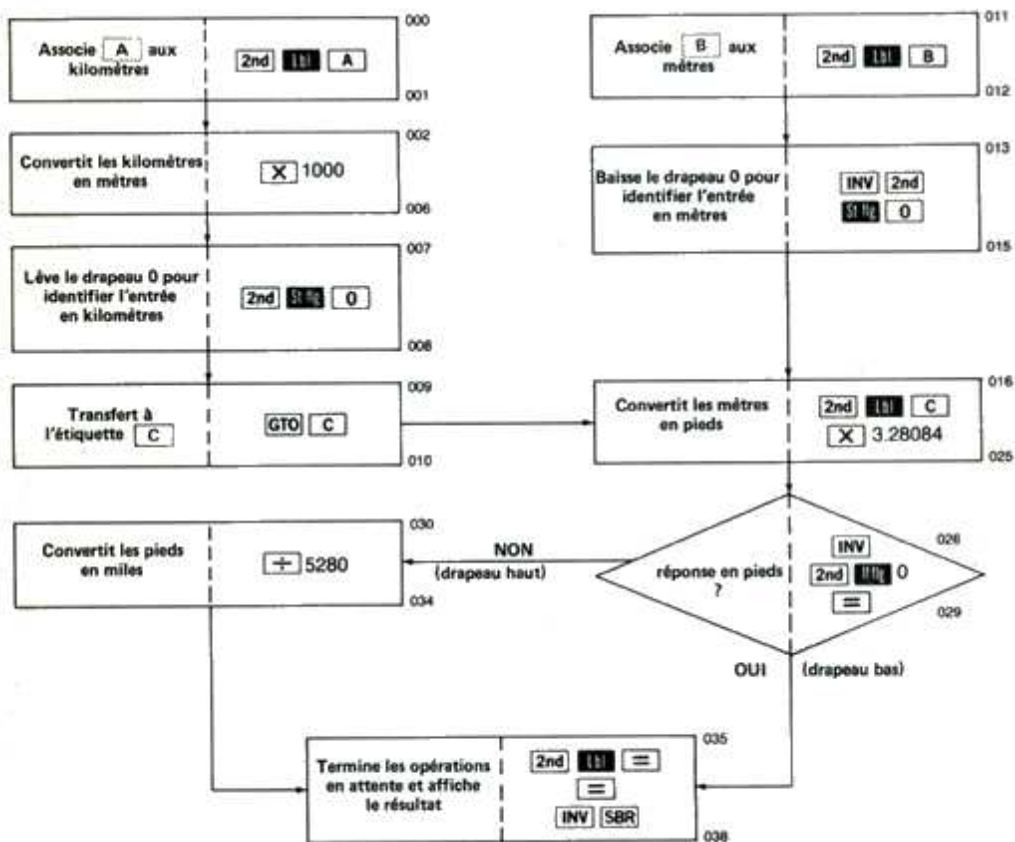
$$1 \text{ km} = 1000 \text{ mètres}$$

$$1 \text{ mètre} = 3.28084 \text{ pieds}$$

$$1 \text{ mile} = 5280 \text{ pieds.}$$



IV



Programme de conversions métriques.



INSTRUCTIONS D'UTILISATION

Séquence	Procédure	Introduire	Appuyer	Affichage
1	Effacer la mémoire programme et placer le pointeur au départ.		2nd CP	
2	Mise en mode programmation.		LRN	000 00
3	Introduire le programme de conversion.			
4	Retour en mode calcul.		LRN	
5	Introduire les kilomètres ou Introduire les mètres		A B	Miles Pieds

Pas et codes
instructions

Séquence
de touches

Pas et codes
instructions

Séquence
de touches

000 76

2nd **(b)**

020 93

.

001 11

A

021 02

2

002 65

X

022 08

8

003 01

1

023 00

0

004 00

0

024 08

8

005 00

0

025 04

4

006 00

0

026 22

INV

007 86

2nd **SI/fg**

027 87

2nd **ll/fg**

008 00

0

028 00

0

009 61

GTO

029 95

=

010 13

C

030 55

+

011 76

2nd **(b)**

031 05

5

012 12

B

032 02

2

013 22

INV

033 08

8

014 86

2nd **SI/fg**

034 00

0

015 00

0

035 76

2nd **(b)**

016 76

2nd **(b)**

036 95

=

017 13

C

037 95

=

018 65

X

038 92

INV **SBR**

019 03

3

Programme de conversions métriques.



Exemple : Introduisez le programme puis convertissez 50 mètres en pieds et 90 kilomètres en miles.

Introduire	Appuyer	Affichage	Commentaires
50	B	164.042	Mètres en pieds
90	A	55.92340909	Kilomètres en miles

TRANSFERTS CONTROLES PAR LE CONTENU D'UNE MEMOIRE **021** .

Cette puissante instruction utilise le contenu des mémoires 0 à 9 pour décider ou non d'un transfert. **021** sert notamment pour contrôler une boucle comme nous allons voir dans le chapitre qui suit.

BOUCLES

Vous aurez dans certains programmes, besoin de répéter une séquence d'opérations un certain nombre de fois pour parvenir au résultat. C'est ce que l'on appelle une «boucle». Pour créer ce type de boucle il suffit de demander en fin de séquence, par une instruction spéciale de revenir se brancher automatiquement au début de cette séquence.

BOUCLES INCONDITIONNELLES.

Il y a deux méthodes pour créer une boucle inconditionnelle.

RST bouclera sur le pas de départ 000.

GTO bouclera sur une adresse de votre choix.

Pour expliciter ceci, créons un programme qui comptera de 4 en 4 :

+ 4 = 2nd PAUSE RST

Cette séquence n'est valable que si le programme est placé à partir du pas 000. Introduisez ce programme en mode programmation, revenez en mode calcul, et remettez le pointeur au pas 000 puis en appuyant sur **RS** faites le démarrer. Vous pouvez ainsi le voir compter. Ce même programme placé à partir du pas 020 en mémoire programme ne pourrait plus utiliser l'instruction **RST**, il faudrait alors constituer la boucle avec **GTO 020** pour obtenir la même chose.

N'oubliez pas en utilisant **RST** que cette instruction remet les drapeaux en position basse et qu'elle efface également les mémoires d'adresses de retour des sous-programmes.

Pour sortir d'une boucle vous pouvez placer un test qui arrêtera l'exécution de celle-ci pour poursuivre le déroulement du programme. Dans l'exemple précédent demandons à la calculatrice de compter à partir de 0 jusqu'à 20.

IV



Pas et codes d'instruction	Séquence de touches	Commentaires
000 02	2	
001 00	0	
002 32	x=t	Stocke 20 dans le registre T.
003 25	CLR	Efface le registre d'affichage.
004 76	2nd 11	
005 85	+	Séquence repérée par l'étiquette +
006 85	+	
007 04	4	
008 95	=	
009 66	2nd Pause	Affiche chaque résultat.
010 67	2nd x=t	Teste la valeur calculée par rapport au registre T
011 00	1	
012 15	5	Saute au pas 015 si x = 20.
013 61	GTO	Si non retourne à l'étiquette +
014 85	+	
015 91	R/S	Arrêt lorsque x = t = 20.

Une fois le programme introduit en mémoire programme, appuyez sur **RST** **R/S** pour le démarrage. Remarquez que le test placé au pas 010 compare chaque résultat avec la valeur 20 et reste inopérant tant que l'égalité n'est pas obtenue. Dès que x = 20 il fait sauter le pointeur au pas 015. La boucle est pilotée par l'instruction **GTO**.



BOUCLES CONDITIONNELLES.

L'exemple précédent est contrôlé par une instruction de transfert conditionnel inclus dans la boucle, il est possible également de remplacer l'instruction de transfert **GTO** par un test, vérifions-le sur le même exemple.

Pas et codes d'instructions	Séquence de touches	Commentaires
000 02	2	
001 00	0	
002 32	x<t	
003 25	CLR	
004 76	2nd (b)	Séquence repérée par l'étiquette A.
005 11	A	
006 85	+	
007 04	4	
008 95	=	
009 66	2nd Plus	
010 22	INV	
011 77	2nd x<=t	Test inversé, saut à l'étiquette A,
012 11	A	si le résultat est inférieur à 20.
013 91	R/S	Arrêt lorsque 20 est atteint.

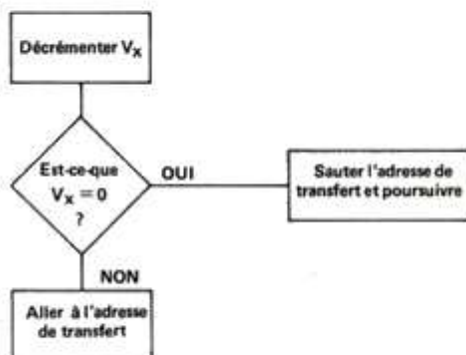
Dans ce cas **INV** **2nd** **x<=t** contrôle la boucle.



BOUCLE CONTROLÉE PAR L'INSTRUCTION **DSZ**

Chaque fois que vous voulez contrôler le nombre de tours effectués dans une boucle, vous pouvez utiliser l'instruction **DSZ** « Décrément et saut sur Zéro ». La séquence est **2nd** **DSZ** X Y où Y est l'adresse de transfert et X le numéro d'une des 10 premières mémoires 0 à 9.

Cette instruction à plusieurs effets : en premier lieu elle décrémente le contenu de la mémoire X de 1 (si le contenu de la mémoire X est compris entre 0 et 1, il se trouve décrémente à 0); en second lieu elle teste la valeur contenue dans la mémoire X (que nous appellerons V_X par la suite); enfin elle sautera l'adresse de transfert si V_X est nul, sinon le transfert à l'adresse Y sera effectué. Graphiquement nous avons :



Tout comme les autres instructions de transfert **DSZ** peut être utilisé tant au clavier que dans un programme. Prenons un exemple :

Appuyer	Affichage	Commentaires
2nd CP	0.	Efface la mémoire programme
2 STO 6	2.	Stocke 2 en mémoire 06
2nd DSZ 6 136		Décrémente V_6 de 1 et teste si « $V_6 = 0 ?$. Réponse « non » il y a transfert au pas 136.
LRN	136 00	Vérifie que le transfert est réalisé.
LRN RCL 6	1.	V_6 valait 2 il est maintenant égal à 1.
2nd DSZ 6 111	1.	Décrément et test.
LRN	136 00	Maintenant $V_6 = 0$ donc il n'y a pas eu de transfert.
LRN RCL 06	0.	Vérifie que $V_6 = 0$.

DSZ représente un compteur du nombre de boucles restant à effectuer.

Pour voir comment il s'utilise dans un programme, reprenons le comptage par 4. Pour arriver au résultat 20 il faut effectuer 5 passages sur les instructions contenues dans la boucle.



IV

Pas et codes instructions	Séquence de touches	Commentaires.
000 47	2nd CMs	Efface toutes les mémoires.
001 05	5	
002 48	2nd Fix	Stocke 5 en mémoire 00 et efface l'affichage.
003 00	0	
004 76	2nd LBL	Séquence repérée par l'étiquette A.
005 11	A	
006 85	+	
007 04	4	
008 95	=	
009 66	2nd Pause	Affiche le résultat à chaque boucle.
010 97	2nd 012	Décrémente la mémoire 00 de 1 et teste si $V_0 = 0$.
011 00	0	
012 11	A	Si V_0 est différent de 0, transfert à l'étiquette A.
013 91	R/S	Arrêt lorsque $V_0 = 0$.

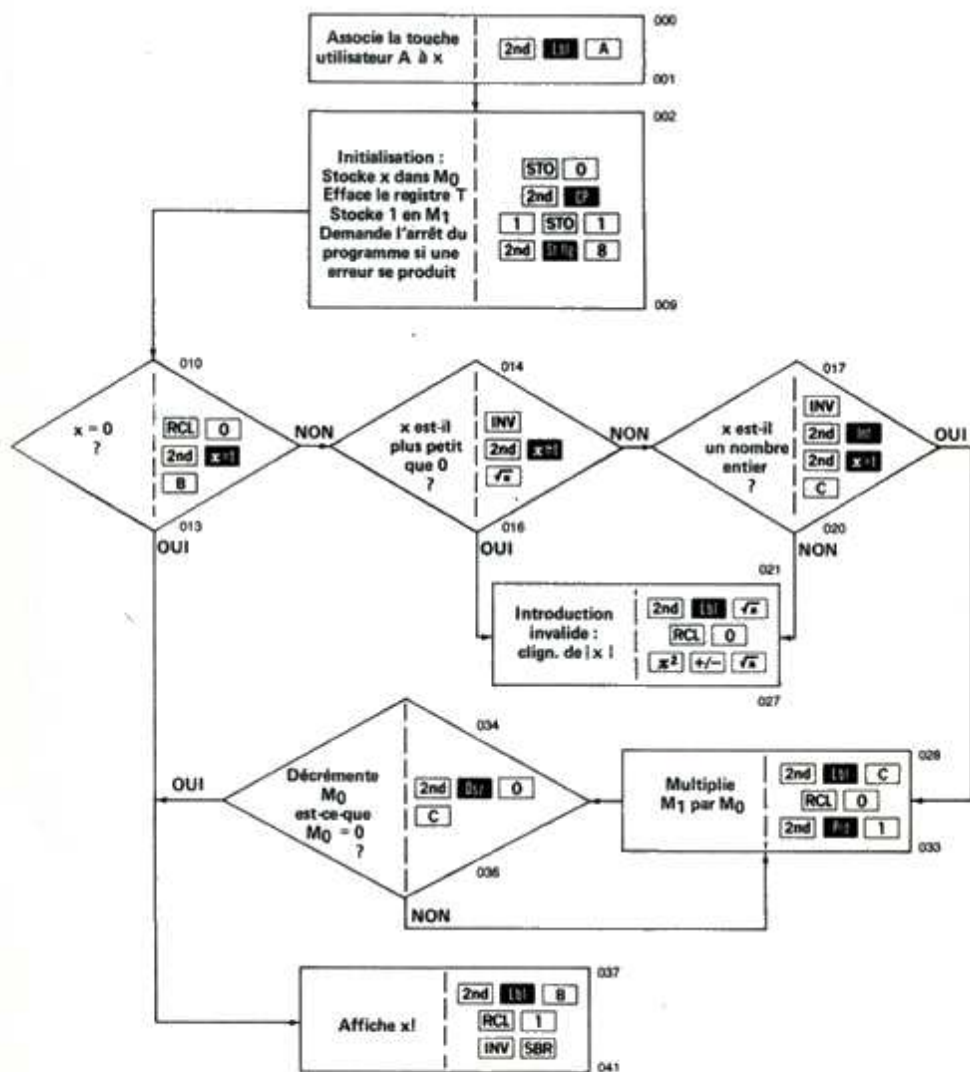
DSZ incrémentera V_x (ajoutera 1 à V_x) si V_x est négatif, dans l'exemple précédent au lieu de 5 nous pouvons mettre -5. De la même façon on pourra incrémenter ou décrémente V_x en utilisant **INV** **2nd** **012** mais le saut de l'instruction de transfert se fera si V_x est différent de zéro et non si V_x égale zéro.

Pour plus de détails voir le chapitre **Décrément et saut sur Zéro** page V-63.

Cette instruction permet de calculer une série de 1 à N; une boucle est définie pour effectuer le calcul. A chaque tour la calculatrice rappelle le contenu de la mémoire décrémentée qui sert, tour après tour, de nouvelle valeur de la variable (notons que la série est calculée de N à 1 pour profiter du décrémente créé par le DSZ).

PROGRAMMER LA FONCTION X!

Ecrivons la boucle susceptible de calculer la factorielle de x soit $x!$
 $x! = x(x-1)(x-2) \dots 2 \times 1$. Par définition x doit être positif et $0! = 1$.



Organigramme de la fonction x!.



IV

Pas et codes d'instructions	Séquence de touches	Pas et codes d'instructions	Séquence de touches
000 76	2nd 1/b1	021 76	2nd 1/b1
001 11	A	022 34	√x
002 42	STO	023 43	RCL
003 00	0	024 00	0
004 29	2nd 0/	025 33	x²
005 01	1	026 94	+/-
006 42	STO	027 34	√x
007 01	1	028 76	2nd 1/b1
008 86	2nd 0/0/	029 13	C
009 08	8	030 43	RCL
010 43	RCL	031 00	0
011 00	0	032 49	2nd 0/0/
012 67	2nd x=1	033 01	1
013 12	B	034 97	2nd 0/0/
014 22	INV	035 00	0
015 77	2nd x=1	036 13	C
016 34	√x	037 76	2nd 1/b1
017 22	INV	038 12	B
018 59	2nd int	039 43	RCL
019 67	2nd x=1	040 01	1
020 13	C	041 92	INV SBR

Programme de la fonction x!

Dans cet exemple nous avons mis 1 en M1 de façon à pouvoir effectuer les produits directement en mémoire. Les trois tests sont effectués au départ du programme pour identifier soit l'introduction d'un zéro, soit les introductions mathématiquement invalides; ces dernières étant signalées par le clignotement de l'affichage et l'arrêt du programme au pas 027 provoqué par le drapeau 8 levé en début de programme. La boucle elle-même se trouve entre les pas 028 et 036.

IV



L'utilisation de ce programme est très simple : introduisez la valeur de x et appuyez sur **A** (x doit être inférieur à 70 sinon la calculatrice se trouve en dépassement de capacité par excès).

Exemple : Calculez 6!; -2!; 0!; 7,3!; 39!.

Introduire	Appuyer	Affichage	Commentaires.
6	A	720	6!
2	+/- A	"2"	Introduction invalide.
	CLR	0	Arrêt du clignotement et effacement.
0	A	1	0!
7.3	A	"7.3"	Introduction invalide.
	CLR	0	Arrêt du clignotement et effacement.
39	A	2.0397882 46	39!

Note : Les guillemets dans la colonne affichage indiquent un affichage clignotant.

EXEMPLES D'APPLICATIONS COMPLEMENTAIRES.

PROGRAMME DE CALCUL DE LA VALEUR ACTUELLE D'UNE OBLIGATION.

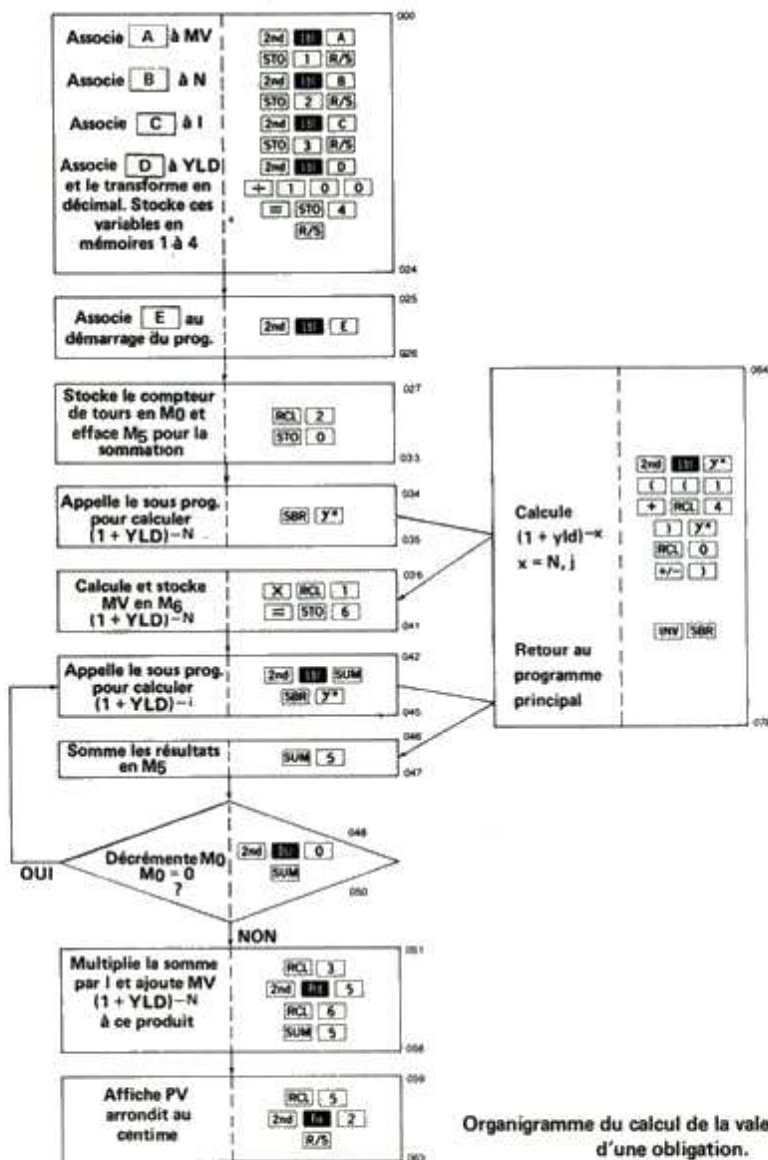
Certains investisseurs ont acheté des obligations pour faire fructifier leur argent; d'autres envisageraient de le faire en étudiant auparavant la rentabilité de l'opération. Le calcul de la valeur actuelle d'une obligation avec des coupons périodiques s'effectue à partir de l'équation suivante :

$$PV = I \sum_{j=1}^N (1 + YLD)^{-j} + MV (1 + YLD)^{-N}$$

dans laquelle :

- MV = valeur à l'échéance.
- N = nombre de période avant l'échéance (j = 1, 2, ... N)
- I = valeur du coupon.
- YLD = rendement de l'obligation à l'échéance (en % par période).
- PV = valeur actuelle de l'obligation (valeur d'achat).

Pour écrire ce programme nous utiliserons une boucle pour calculer la sommation; le nombre de tours à effectuer étant connu, nous utiliserons le DSZ, d'autant que le contenu de la mémoire décrétementée nous servira chaque fois que nous aurons besoin de la valeur j. Pour économiser des pas programmes, nous calculerons $(1 + YLD)^{-x}$ pour x = j et N par appel d'un sous-programme.



Organigramme du calcul de la valeur actuelle d'une obligation.



MODE D'EMPLOI

Séquence	Procédure	Introduire	Appuyer	Affichage
1	Effacer la mémoire programme et remettre le pointeur au pas 000.		2nd CP	
2	Mise en mode programmation.		LRN	000 00
3	Introduire le programme.			
4	Retour en mode calcul.		LRN	0
5	Introduire la valeur à l'échéance	MV	A	MV
6	Introduire le nombre de périod.	N	B	N
7	Introduire la valeur du coupon.	I	C	I
8	Introduire le rendement de l'obligation.	YLD	D	YLD/100
9	Calculer la valeur actuelle. Les variables qui ne changent pas, n'ont pas à être réintroduites pour effectuer d'autres calculs.		E	PV



IV

Pas et codes d'instructions	Séquence de touches	Pas et codes d'instructions	Séquence de touches	Pas et codes d'instructions	Séquence de touches
000 76	2nd (b)	027 71	RCL	054 05	5
001 11	A	028 45	2	055 43	RCL
002 42	STO	029 65	STO	056 06	6
003 01	1	030 43	0	057 44	SUM
004 91	R/S	031 01	CLR	058 05	5
005 76	2nd (b)	032 95	STO	059 43	RCL
006 12	B	033 42	5	060 05	5
007 42	STO	034 06	SBR	061 58	2nd (b)
008 02	2	035 43	y*	062 02	2
009 91	R/S	036 02	X	063 91	R/S
010 76	2nd (b)	037 42	RCL	064 76	2nd (b)
011 13	C	038 00	1	065 45	y*
012 42	STO	039 20	=	066 53	(
013 03	3	040 42	STO	067 53	(
014 91	R/S	041 05	6	068 01	1
015 76	2nd (b)	042 76	2nd (b)	069 85	+
016 14	D	043 44	SUM	070 43	RCL
017 55	÷	044 71	SBR	071 04	4
018 01	1	045 45	y*	072 54)
019 00	0	046 44	SUM	073 45	y*
020 00	0	047 05	5	074 43	RCL
021 95	=	048 97	2nd (b)	075 00	0
022 42	STO	049 00	0	076 94	+/-
023 04	4	050 44	SUM	077 54)
024 91	R/S	051 43	RCL	078 92	INV SBR
025 76	2nd (b)	052 03	3		
026 15	E	053 49	2nd (b)		

Programme du calcul de la valeur actuelle d'une obligation.

IV



Exemple : Trouver la valeur actuelle d'une obligation de 20 000 Frs arrivant à échéance dans 12 ans avec un coupon annuel de 1 400 Frs et un rendement souhaité de 8%.

Introduire	Appuyer	Affichage	Commentaires
20000	<input type="button" value="A"/>	20000	MV
12	<input type="button" value="B"/>	12	N
1400	<input type="button" value="C"/>	1400	I
8	<input type="button" value="D"/>	.08	YLD
	<input type="button" value="E"/>	18492.78	→ PV

Ces conditions procurent un rendement annuel de 8% pour un prix d'achat de 18 492.78 Frs. Le profit réalisé sur une telle opération est $12 \times 1400 + (20000 - 18492.78) = 18307.22$ Frs.

RESOLUTION DE L'EQUATION DU SECOND DEGRE.

La résolution de l'équation du second degré est un exemple qui va nous permettre de revoir les différentes possibilités vues précédemment. Par ailleurs, vous aurez sous la main le programme pour le cas où vous auriez à résoudre un problème comprenant une telle équation.

Ecrire le programme consiste à calculer les parties réelles et imaginaires de l'équation :

$$ax^2 + bx + c = 0 \quad (a \neq 0)$$

Les racines x_1 et x_2 sont données par :

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Si $b^2 - 4ac$ est positif ou nul, les racines sont réelles et calculées avec les formules ci-dessus; par contre si $b^2 - 4ac$ est négatif x_1 et x_2 sont des racines complexes dont l'expression est donnée ci-dessous :

$$x_1 = R + (j \cdot I)$$

$$x_2 = R - (j \cdot I)$$

dans lesquelles :

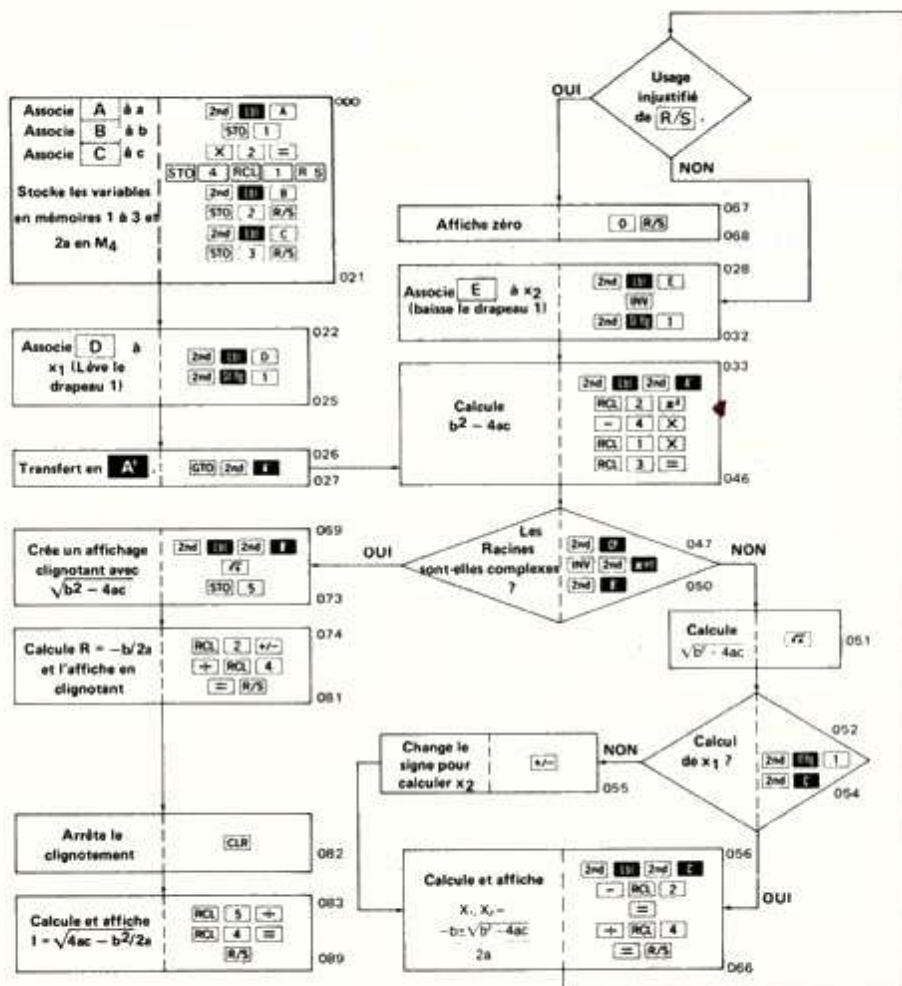
$$R = \frac{-b}{2a}$$
$$j = \sqrt{-1}$$
$$I = \frac{\sqrt{4ac - b^2}}{2a}$$

Puisque les racines x_1 et x_2 sont calculées à partir d'une même formule de base nous économiserons des pas programme en combinant les différents calculs, l'emploi d'un drapeau nous indiquera la racine calculée. Pour les racines complexes, nous devons séparer les parties imaginaires et réelles, Remarquez que pour les racines complexes, les parties réelles restent les mêmes, les parties imaginaires égales sont égales au signe près.



IV

Pour visualiser le cas où les racines sont complexes, nous ferons clignoter la partie réelle des racines sur l'affichage. (Notez que lorsque $b^2 - 4ac$ est négatif $\sqrt{b^2 - 4ac}$ se calcule par l'expression $\sqrt{4ac - b^2}$ ce résultat sera stocké pour être utilisé par la suite dans le calcul de la partie imaginaire des racines complexes). Dans cet exemple, pour les racines complexes, la partie réelle sera affichée en clignotant et la partie imaginaire sera affichée après avoir appuyé sur $\boxed{R/S}$. A titre de sauvegarde un zéro sera affiché pour une utilisation erronée de $\boxed{R/S}$ dans le cas de racines réelles. Ce programme contenant les instructions $\boxed{=}$ et $\boxed{R/S}$ ne pourra être utilisé sous cette forme comme sous-programme.



Organigramme de l'équation du second degré.



MODE D'EMPLOI				
Séquence	Procédure	Introduire	Appuyer	Affichage
1	Effacer la mémoire programme et remettre le pointeur au pas 000		2nd CP	
2	Mise en mode programmation.		LRN	000 00
3	Introduire le programme de l'équation du second degré.			
4	Retour en mode calcul.		LRN	0
5	Introduire a ($a \neq 0$).	a	A	a
6	Introduire b.	b	B	b
7	Introduire c.	c	C	c
8	Calculer x_1 Si la partie réelle clignote, les racines sont imaginaires. Calculer la partie imaginaire.		D R/S	x_1 (réelle) x_1 (imaginaire)
9	Calculer x_2 Si la partie réelle clignote, les racines sont imaginaires. Calculer la partie imaginaire.		E R/S	x_2 (réelle) x_2 (imaginaire)



IV

Pas et codes d'instructions

000 76
001 11
002 42
003 01
004 65
005 02
006 95
007 42
008 04
009 43
010 01
011 91
012 76
013 12
014 42
015 02
016 91
017 76
018 13
019 42
020 03
021 91
022 76
023 14
024 86
025 01
026 61
027 16
028 76
029 15
030 22
031 86

Séquence de touches

2nd **LBL**
A
STO
1
X
2
=
STO
4
RCL
1
R/S
2nd **LBL**
B
STO
2
R/S
2nd **LBL**
C
STO
3
R/S
2nd **LBL**
D
2nd **St/Flg**
1
GTO
2nd **I**
2nd **LBL**
E
INV
2nd **St/Flg**

Pas et codes d'instructions.

032 01
033 76
034 16
035 43
036 02
037 33
038 75
039 04
040 65
041 43
042 01
043 65
044 43
045 03
046 95
047 29
048 22
049 77
050 17
051 34
052 87
053 01
054 18
055 94
056 76
057 18
058 75
059 43
060 02

Séquence de touches

1
2nd **LBL**
2nd **I**
RCL
2
x²
-
4
X
RCL
1
X
RCL
3
=
2nd **CP**
INV
2nd **x≠1**
2nd **B**
√x
2nd **Flg**
1
2nd **C**
+/-
2nd **LBL**
2nd **C**
-
RCL
2

Pas et codes d'instructions

061 95
063 55
064 43
065 04
066 95
067 91
068 00
069 91
070 76
070 17
072 34
073 42
074 05
075 43
076 02
077 94
078 55
079 43
080 04
080 95
082 91
083 25
084 43
085 05
086 55
087 43
088 04
089 95
090 91

Séquence de touches

=
÷
RCL
4
=
R/S
0
R/S
2nd **LBL**
2nd **B**
√x
STO
5
RCL
2
+/-
+
RCL
4
=
R/S
CLR
RCL
5
+
RCL
4
=
R/S

Programme de l'équation du second degré.

IV



Exemple : Calculer les racines de l'équation :

$$1.5x^2 + 3.7x + 2.25 = 0$$

Introduire	Appuyer	Affichage	Commentaires
1.5	<input type="button" value="A"/>	1.5	a
3.7	<input type="button" value="B"/>	3.7	b
2.25	<input type="button" value="C"/>	2.25	c
	<input type="button" value="D"/>	- 1.088036702	Calcule x_1 (l'affichage fixe indique que les racines sont réelles).
	<input type="button" value="E"/>	- 1.378629965	Calcule x_2 .

Calculer les racines de l'équation :

$$x^2 + 2x + 17 = 0.$$

Introduire	Appuyer	Affichage	Commentaires
1	<input type="button" value="A"/>	1.	a
2	<input type="button" value="B"/>	2.	b
17	<input type="button" value="C"/>	17.	c
	<input type="button" value="D"/>	"-1."	Calcule une racine (l'affichage clignotant annonce des racines complexes - R est affiché).
	<input type="button" value="R/S"/>	4.	Calcule I

Les racines complexes sont :

$$x_1 = 1 + 4i$$

$$x_2 = 1 - 4i$$



TECHNIQUES COMPLEMENTAIRES.

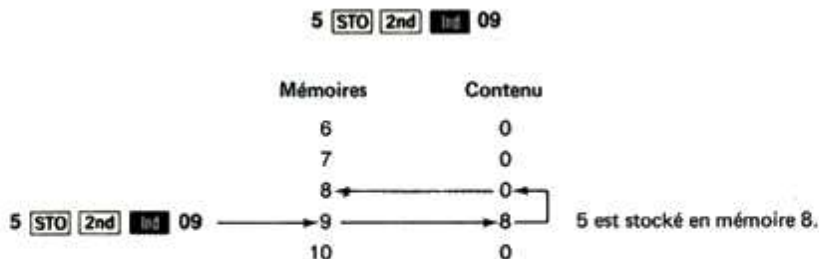
ADRESSAGE INDIRECT.

Une technique très intéressante va accroître considérablement les possibilités des opérations en mémoire, des instructions de transfert ou de contrôle, ainsi que des accès aux bibliothèques de programmes. Il s'agit de l'adressage indirect **2nd Ind**. Le principe de base est simple : vous utilisez une mémoire non pas pour l'information qu'elle contient mais pour qu'elle vous indique où se trouve l'information cherchée. Exactement comme si vous disiez à une personne «va demander à Paul où se trouve François» au lieu de lui dire «va trouver François». Si Paul sait où est François, la personne que vous interrogez saura immédiatement où se trouve François alors qu'elle aurait pu passer des heures à le chercher elle-même. Dans certains cas, il est presque impossible de trouver François directement et la méthode indirecte est le seul moyen possible d'y parvenir. Les instructions sont utilisées indirectement en accolant **2nd Ind** suivi d'une adresse mémoire. Dans cette mémoire se trouve l'information nécessaire pour compléter l'instruction.

ADRESSAGE INDIRECT DES MEMOIRES.

Toutes les instructions relatives aux mémoires (stockage, rappel, échange, sommation, produit) peuvent utiliser l'adressage indirect.

Prenons la séquence :



IV



Ecrivons un programme permettant d'effacer un certain nombre de mémoires. Par exemple, effaçons les mémoires 1 à X, X étant introduit suivant vos besoins.

Pas et codes d'instructions	Séquence de touches	Commentaires
000 76	2nd 1st	Introduit X par l'intermédiaire de A
001 11	A	
002 42	STO	Stocke X en mémoire 00.
003 00	0	
004 76	2nd 1st	
005 12	B	
006 25	CLR	
007 72	STO 2nd 1st	M_0 indique la mémoire à effacer
008 00	0	
009 97	2nd DSZ	M_0 contrôle l'exécution de la boucle.
010 00	0	
011 12	B	Si M_0 différent de zéro, transfert à B.
012 92	INV SBR	Arrête le programme quand M_0 est nul.

Au premier passage dans la boucle, X est en mémoire 00 ; la séquence **CLR** **STO** **2nd** **1st** 00 stocke 0 en mémoire X. DSZ décrémente le contenu de la mémoire 00 qui devient (X - 1). Le deuxième passage stocke 0 en mémoire (X - 1). Les mémoires sont effacées dans l'ordre décroissant. Pouvez-vous écrire le programme qui effacera dans l'ordre croissant ?

Notez qu'il y a ici un code spécial 72 pour **STO** **2nd** **1st** . Certaines instructions indirectes sont contractées comme celle-ci pour économiser des pas de programme. Pour connaître la liste complète, voyez le chapitre Codes des Instructions (codes des touches) page V-48.



TRANSFERT INDIRECT.

La pratique de l'adressage indirect peut être étendue aux instructions de transfert. Rappelez-vous qu'il y a deux possibilités pour indiquer une adresse de transfert : en utilisant soit une adresse numérique, soit une étiquette placée en mémoire programme. Le transfert indirect donne une possibilité supplémentaire plus souple. Vous indiquez seulement la mémoire dans laquelle se trouve l'adresse numérique que vous voulez utiliser. Une étiquette ne peut être stockée en mémoire.

Une instruction de transfert indirect se constitue en plaçant **2nd** **Ind** juste après un transfert inconditionnel (**GTO** , **SBR**) ou un transfert conditionnel (**2nd** **≠=1** , **2nd** **0sr** , etc...). L'instruction est complétée par le numéro de la mémoire contenant l'adresse numérique où vous désirez effectuer le transfert. Essayez cette séquence au clavier :

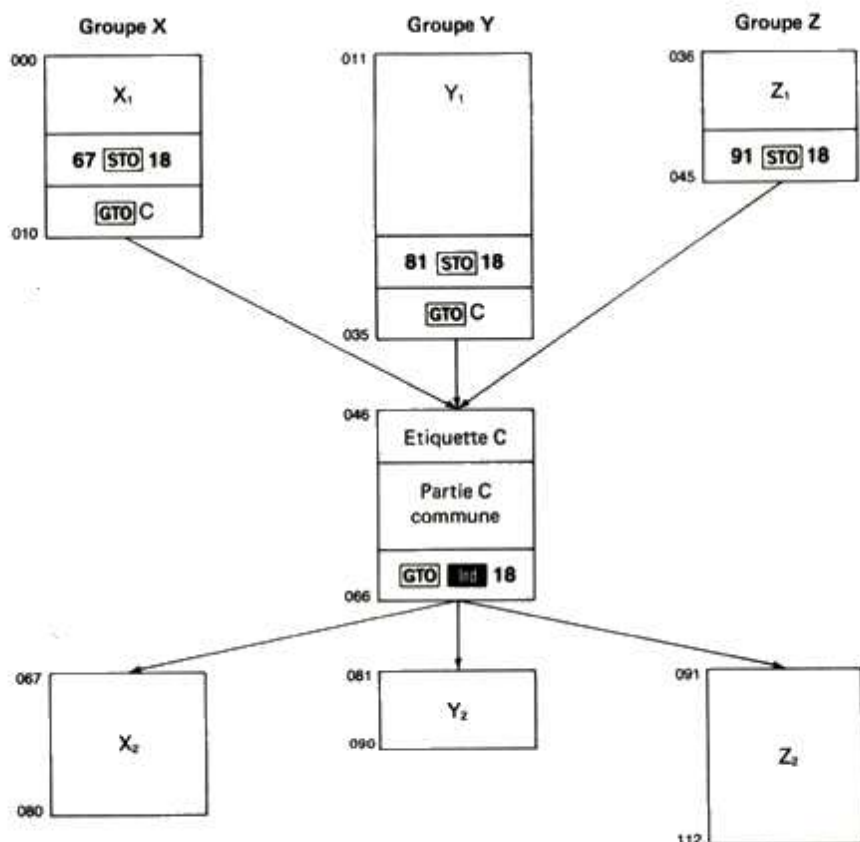
Séquence de touches	Affichage	Commentaires
35 STO 18	35.	Stocke 35 en mémoire 18
GTO 2nd Ind 18 LRN	035 00	Le pointeur est placé au pas 035.

Pour l'instruction DSZ, il est possible d'utiliser l'adressage indirect pour désigner la mémoire à décrémenter. Il est également possible d'écrire la séquence **2nd** **0sr** **2nd** **Ind** **12** **2nd** **Ind** **14** dans laquelle d'une part la mémoire décrémentée sera désignée par le contenu de la mémoire 12, d'autre part l'adresse de transfert sera désignée par le contenu de la mémoire 14.

Pour expliciter cette méthode de transfert prenons un exemple qui montrera comment elle peut être utilisée. Considérons trois groupes d'instructions devant se trouver dans un même programme.

Groupe X	Groupe Y	Groupe Z
X ₁	Y ₁	Z ₁
C	C	C
X ₂	Y ₂	Z ₂

La séquence centrale (C) de chaque groupe d'instructions est identique., il est donc logique de ne l'écrire qu'une seule fois. En fin de séquence x₁ et y₁, il est possible d'utiliser l'instruction **GTO** (z₁ continuant directement sur C) pour aller chercher la séquence C. Comment se fera le transfert aux séquences appropriées x₂, y₂ et z₂ ? Ce problème peut être résolu en utilisant le transfert indirect, il faut pour cela, en fin de première séquence, stocker l'adresse de la troisième séquence en mémoire de façon à ce qu'en fin d'exécution de la séquence C, l'instruction **GTO** **2nd** **Ind** fasse le transfert à l'adresse appropriée. Dans l'organigramme qui suit, les numéros des pas de programme sont arbitraires. ; ils permettent de localiser les différents groupes entre eux.



AUTRES CARACTERISTIQUES.

2nd **Ind** peut être également utilisé pour lever ou abaisser indirectement un drapeau ou pour fixer indirectement le nombre de chiffres après le point décimal.

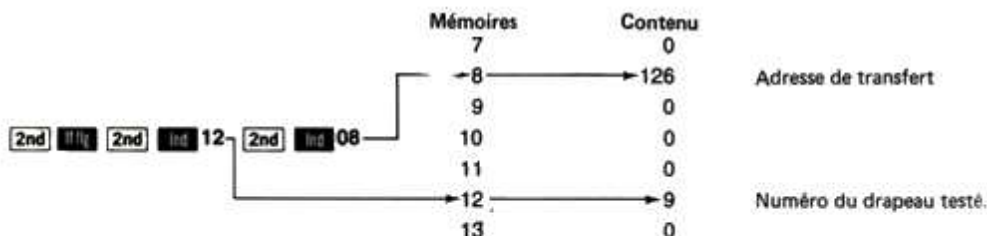
D'une façon analogue aux précédentes instructions, le contrôle indirect d'un drapeau s'effectue en stockant le numéro du drapeau dans une mémoire. Par exemple, mettons 6 en M_{12} , la séquence **2nd** **St 6** **2nd** **Ind**

12 lèvera le drapeau 6 alors que **2nd** **Ind** **2nd** **Ind** **12** **A** fera un transfert à l'étiquette **A** suivant l'état du drapeau 6.



IV

De la même façon, $\boxed{2nd} \boxed{f/f}$ étant une instruction de branchement, nous allons pouvoir l'utiliser indirectement. La séquence $\boxed{2nd} \boxed{f/f} \boxed{2nd} \boxed{ind} \boxed{12} \boxed{2nd} \boxed{ind} \boxed{08}$ est également possible :



La séquence est équivalente à $\boxed{2nd} \boxed{f/f} \boxed{9} \boxed{126}$. Introduisez l'exemple qui suit pour visualiser le fonctionnement.

Appuyer	Affichage	Commentaires.
126 $\boxed{STO} \boxed{08}$	126.	Stocke 126 en mémoire 08.
9 $\boxed{STO} \boxed{12}$	9.	Stocke 9 en mémoire 12.
$\boxed{2nd} \boxed{ST} \boxed{9}$	9.	Lève le drapeau 9.
$\boxed{2nd} \boxed{f/f} \boxed{2nd} \boxed{ind} \boxed{12}$		
$\boxed{2nd} \boxed{ind} \boxed{8} \boxed{LRN}$	126 00	Transfert effectué au pas 126.

De la même façon, le point décimal peut être indirectement fixé comme l'indique l'exemple qui suit :

Appuyer	Affichage	Commentaires.
2 $\boxed{STO} \boxed{12}$	2.	Stocke 2 en mémoire 12.
$\boxed{2nd} \boxed{fix} \boxed{2nd} \boxed{ind} \boxed{12}$	2.00	Affichage avec 2 chiffres après le point décimal.

Pour plus de détails sur l'adressage indirect, reportez-vous au chapitre **Adressage indirect**, page V-68.

IV



Optimiser un Programme.

Des différentes raisons nécessitant d'optimiser un programme, deux sont souvent rencontrées. La première est de rendre un programme plus simple à utiliser ; la seconde, de réduire sa taille pour la rendre compatible avec la taille mémoire dont on dispose.

SIMPLIFIER L'EMPLOI D'UN PROGRAMME.

La simplicité d'emploi d'un programme dépend surtout de vos besoins et de vos préférences. En règle générale, toutefois, un programme correctement écrit doit être utilisable avec un minimum de touches à enfoncer (même par une personne différente de celle qui a écrit le programme).

Pour beaucoup de programmes, une erreur d'introduction ou une erreur de touche nécessite de reprendre toute la phase d'introduction des données du programme. Ceci peut être évité et un gain de temps sera enregistré surtout si vous travaillez sur des programmes longs et complexes. Simplifier les procédures permettant de récupérer une erreur est une des façons de rendre un programme plus simple à utiliser. En pratique, vous pouvez résoudre ce genre de problème en stockant et en sauvegardant les données de départ. D'autre part, au début des séquences utilisant l'arithmétique en mémoire une bonne précaution consiste à utiliser l'instruction **STO** de façon à pouvoir réutiliser le programme sans être obligé d'effacer préalablement les mémoires.

DIMINUER LE NOMBRE DE PAS DE PROGRAMME.

Réduire le nombre de pas d'un programme prend du temps, c'est pourquoi nous vous déconseillons de chercher à le faire, si votre programme tient dans la mémoire programme disponible, à moins que ce ne soit pour votre satisfaction personnelle.

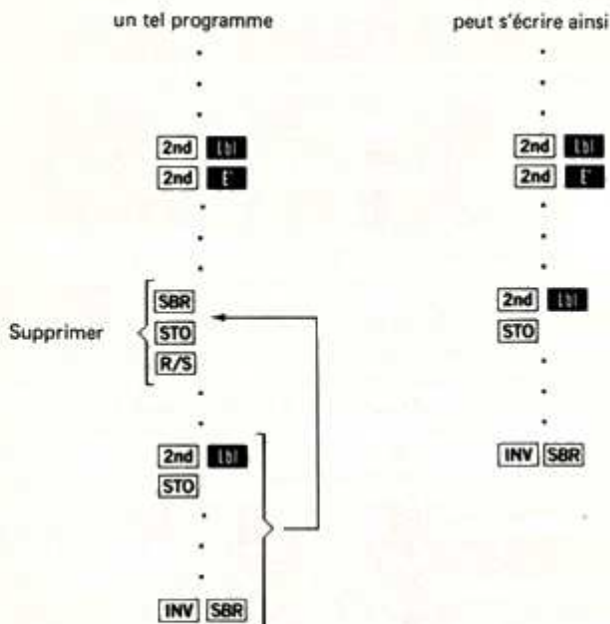
Lorsque vous essayez de réduire un programme, regardez en premier lieu les séquences qui apparaissent à plusieurs reprises. Si ces séquences sont longues ou répétitives, en les traitant sous forme de sous-programmes vous gagnerez de la place.

Un programme utilisant plusieurs sous-programmes peut également dépasser les limites de la mémoire programme ; l'optimisation des sous-programmes devient importante.

Il y a plusieurs méthodes qui peuvent réduire un programme en combinant les différentes parties constituant ce programme. Par exemple, si l'appel d'un sous-programme se trouve à la fin d'une séquence, il est possible de placer ce sous-programme en fin de la séquence considérée, comme le montre l'exemple suivant :



IV



Non seulement nous avons économisé quelques pas, mais nous avons également libéré un niveau dans les mémoires d'adresses de retour des sous-programmes. **INV SBR** agit maintenant comme **R/S** car la mémoire d'adresse de retour des sous-programmes est vide.

IV



Une autre méthode est illustrée dans l'exemple qui suit :

un tel programme

·
·
·
2nd | x | 1 |
D
·
1
STO
4
INV | SBR
2nd | 1 | 1 |
D
1
STO
4
INV | SBR

peut s'écrire ainsi

·
·
·
2nd | x | 1 |
D
·
2nd | 1 | 1 |
D
1
STO
4
INV | SBR

Le but du programme ici est de stocker 0,1 ou 1 suivant le résultat d'un test. Les deux séquences sont correctes, la seconde économise les quatre pas encadrés.



IV

En plus des différentes façons de combiner les séquences d'un programme, vous avez également quelques astuces de programmation qui sont très rentables. Dans l'exemple qui suit nous voulons utiliser seulement la valeur exprimée avec deux décimales d'un résultat. Si nous mettons la calculatrice en virgule fixe à deux décimales, l'arrondi se fera uniquement sur l'affichage et les calculs seront toujours effectués avec l'ensemble des décimales.

Un tel programme

.
.
.
X
1
0
0
=
2nd f/x
+
1
0
0
=
.
.
.

peut s'écrire ainsi

.
.
.
.
2nd f/x
2
EE
INV
EE
2nd f/x
9
.
.
.

La méthode employée à gauche est simple; le cheminement utilisé dans celle de droite est plus rentable mais aussi plus obscur. L'instruction **EE** n'opérant que sur le nombre affiché, va permettre de chasser les caractères superflus, l'affichage ayant été mis en virgule fixe auparavant. Les instructions qui suivent remettront la calculatrice dans les conditions initiales et le calcul se poursuivra avec la valeur arrondie.

L'exemple qui suit donne trois méthodes pour résoudre le même problème : Ajouter 10 000 au contenu du registre d'affichage.



.	.	.
.	.	.
.	.	.
+	+	+
1	1	4
0	EE	INV
0	4	2nd lg
0	=	=
0	.	.
=	.	.
.	.	.
.	.	.

Les deux dernières méthodes occupent le même nombre de pas de programme. Toutefois la deuxième méthode n'est pratique que si vous souhaitez conserver un affichage en notation scientifique.

Au fur et à mesure de la progression de vos connaissances concernant les possibilités de votre calculatrice, vous trouverez sûrement des procédés d'optimisation qui conviendront à vos besoins. Assurez-vous d'avoir bien enregistré les séquences précédentes de façon à pouvoir les réutiliser lorsque vous aurez à optimiser un programme. En attendant vous pouvez également utiliser toutes les fonctions susceptibles de réduire le nombre de pas et qui sont déjà à votre disposition dans votre calculatrice telles que les opérations arithmétiques en mémoire **SUM** et **2nd Prt**, l'adressage indirect ainsi que les différentes fonctions de contrôle.

Si malgré tout vous ne parvenez pas à faire tenir votre programme dans la place disponible, vous allez être obligé de couper votre programme en segments, chacun d'eux étant exécuté séparément : les résultats partiels du premier seront stockés en mémoire avant d'introduire le segment de programme suivant qui viendra réutiliser les informations mémorisées et poursuivre l'exécution du programme jusqu'au résultat final. Parfois il y a plusieurs façons de traiter un problème et, suivant la façon utilisée pour le résoudre, vous pouvez faire une économie de pas substantielle comme nous allons voir.

Programme de calcul de coût d'exploitation

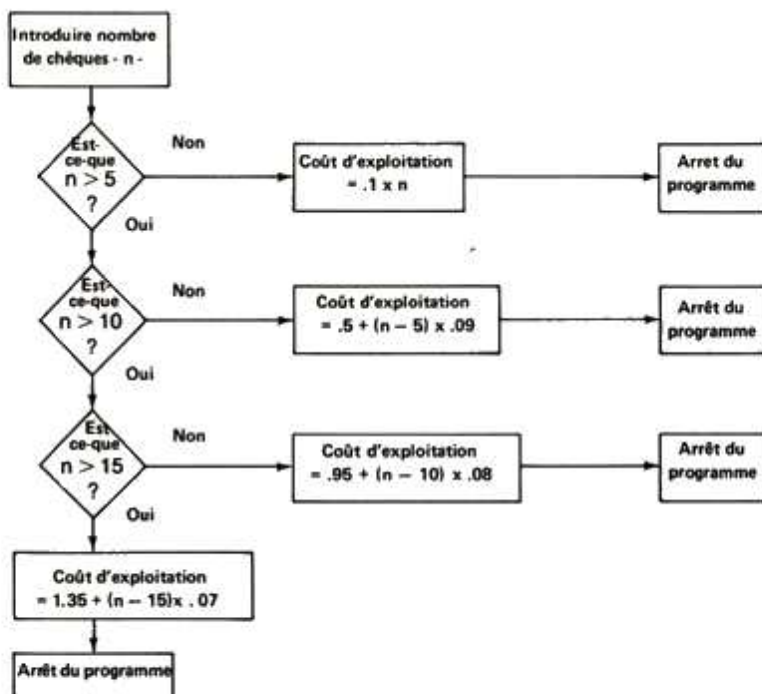
En tant que responsable d'une importante banque locale, vous désirez avoir un moyen simple et rapide pour calculer chaque mois le coût d'exploitation des chèques utilisés par vos clients.

Les coûts unitaires sont les suivants :

0.10 Frs par chèque pour les cinq premiers	(1 à 5)
0.09 Frs par chèque pour les cinq suivants	(6 à 10)
0.08 Frs par chèque pour les cinq suivants	(11 à 15)
0.07 Frs par chèque au-delà du quinzième	



L'organigramme ci-dessous vous donne une méthode simple pour résoudre le problème :



Organigramme du calcul du coût d'exploitation
(Approche simplifiée)

IV

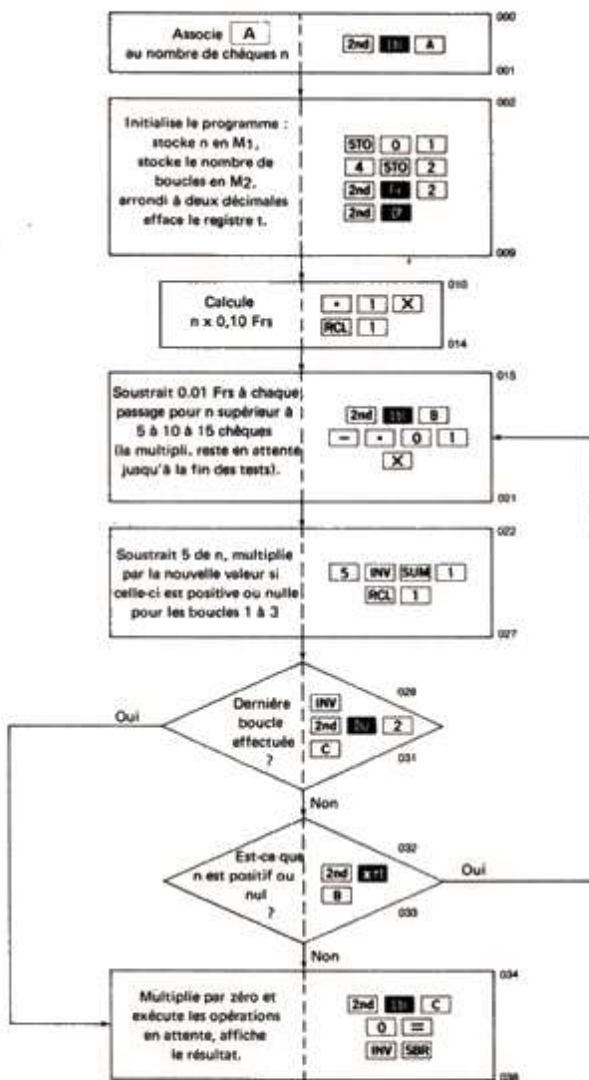


En suivant l'organigramme précédent, vous utiliserez environ quatre vingt à quatre vingt dix pas de programme. Un tel programme tient sans difficulté en mémoire programme; par contre si ce programme doit être utilisé en tant que sous-programme dans une application plus générale, il est possible que la place vous manque pour introduire la totalité de votre application. Il devient alors nécessaire de réduire le nombre de pas de programme. L'organigramme suivant vous donne une autre méthode de résolution plus élaborée mais aussi plus économique.



Organigramme du calcul du coût d'exploitation (Approche élaborée)

A première vue un programme ainsi conçu sera plus facile à loger en mémoire programme; toutefois, examinez de près le cheminement logique utilisé car le raisonnement suivi n'est pas immédiat à saisir.



Organigramme du calcul de coût d'exploitation.

IV



Jusqu'au pas 022, le programme est simple, au pas 021, la multiplication est mise en attente pendant qu'un ajustement du coût est effectué en fonction de la valeur de n . La boucle permet de réduire à chaque tour le coût de chaque chèque (plus de 5 chèques, coût réduit à 0.09 Frs; plus de 10, 0.08 Frs; plus de 15, 0.07 Frs...) L'instruction `2nd 0n` permet de savoir qu'elle est la boucle en cours d'exécution. Avant chaque boucle (1, 2 et 3), la valeur de n modifiée au pas 023 est testée; lorsqu'elle devient négative, un zéro est placé sur l'affichage et les opérations en attente sont exécutées. Le programme s'arrête en affichant le coût total d'exploitation.

Lorsque la quatrième boucle est effectuée, la multiplication restée en attente se termine par un résultat nul car autrement le coût de chaque chèque au-dessus du vingtième aurait été réduit à 0.06 Frs. Le programme, calcule ensuite les charges totales et s'arrête. La dernière boucle n'est pas nécessaire pour le calcul; cependant sa suppression demanderait de rajouter des instructions supplémentaires alors que le but est de réduire au maximum le nombre de pas de programme.

Nous venons d'envisager deux méthodes différentes pour résoudre ce problème de coût d'exploitation. Vous pouvez vous rendre compte que les chemins envisageables pour parvenir au même résultat sont multiples et peuvent être très différents. Ils sont bien évidemment exploitables l'un et l'autre. Dans le cas présent, la seconde méthode demande moins de la moitié de la place requise en mémoire programme par la première méthode. Par contre son exécution sera un peu plus longue. Si l'approche envisagée pour résoudre un problème ne nécessite pas une vigilance particulière, soyez convaincu que la meilleure méthode reste celle qui vous donne les résultats exacts.



IV

Pas et codes instructions	Séquence de touches	Pas et codes instructions	Séquence de touches
000 76	2nd 1/1	020 01	1
001 11	A	021 65	X
002 42	STO	022 05	5
003 01	0 1	023 22	INV
004 04	4	024 44	SUM
005 42	STO	025 01	1
006 02	2	026 43	RCL
007 58	2nd fix	027 01	1
008 02	2	028 22	INV
009 29	2nd DP	029 97	2nd DSZ
010 93	.	030 02	2
011 01	1	031 13	C
012 65	X	032 77	2nd ≠1
013 43	RCL	033 12	B
014 01	1	034 76	2nd 1/1
015 76	2nd 1/1	035 13	C
016 12	B	036 00	0
017 75	-	037 95	=
018 93	.	038 92	INV SBR
019 00	0		

Programme de calcul du coût d'exploitation

Pour utiliser ce programme, introduisez simplement le nombre de chèques et appuyez sur **A**. Vous trouverez ainsi que 1 chèque coûte 0.10 frs; 6 chèques coûtent 0.59 frs et 63 chèques coûtent 4.71 frs.

Techniques pour accélérer la vitesse d'exécution d'un programme

Dans certains cas, il est possible d'optimiser le temps de traitement d'un programme. Ceci est particulièrement intéressant dans le cas d'un programme dont la durée d'exécution est longue et l'utilisation fréquente.

Quand un programme s'exécute, ce sont les transferts qui sont les plus longs à effectuer et réduire leur nombre rendra l'exécution du programme plus rapide. En particulier chaque fois que la place disponible en mémoire programme le permet, vous pouvez accélérer l'exécution de votre programme en remplaçant un sous-programme par une séquence d'instructions linéaire.

IV



Souvenez-vous que l'identification d'un pas de programme peut être faite soit par une adresse numérique à trois caractères, soit par une étiquette. Dans le premier cas le pointeur se positionne au pas programme désigné, dans l'autre cas la calculatrice doit rechercher quelle adresse correspond à l'étiquette utilisée. Cette recherche s'effectue à partir du pas de départ 000 par exploration systématique de la mémoire programme jusqu'à l'endroit où se trouve l'étiquette; l'exécution du programme se poursuit alors à partir de cette étiquette.

Lorsque vous introduisez les instructions d'un programme dans la calculatrice, il est souvent difficile de prévoir la valeur de certaines adresses numériques. De surcroît, si ce programme doit subir quelques modifications le fait d'insérer ou de supprimer des instructions va venir modifier ces adresses numériques. La meilleure méthode consiste donc à écrire une première version de programme avec des étiquettes et de les transformer par la suite en adresses numériques après s'être assuré qu'il n'y a plus de corrections à effectuer. Le remplacement des étiquettes par des adresses numériques nécessite l'emploi de l'instruction **2nd** **Nop** comme nous allons le voir.

2nd **Nop** est une opération blanche, elle n'aura donc aucun effet sur les calculs effectués ni sur le déroulement du programme (excepté si elle sert d'étiquette) et permettra de combler les espaces vides dans un programme. Cette technique est illustrée dans l'exemple ci-dessous.

	.		.
	.		.
	.		.
027	SBR	027	SBR
028	Inx	028	0
029	2nd Nop	029	7 5
	.		.
	.		.
	.		.
073	2nd lb	073	2nd Nop
074	Inx	074	2nd Nop
	.		.
	.		.
	.		.
099	GTO	099	GTO
100	Inx	100	0
101	2nd Nop	101	7 5

Transformation d'un adressage par étiquette en adressage numérique



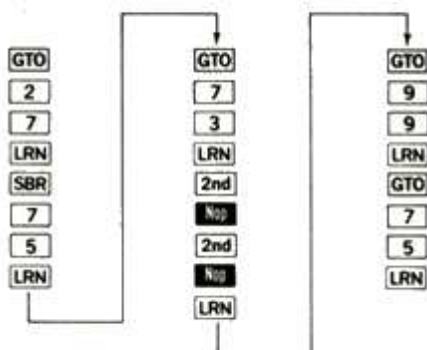
IV

Notez que le pas 075 est utilisé comme adresse numérique du fait que lors d'un transfert à une étiquette le pointeur se place au pas qui suit immédiatement cette étiquette.

Souvenez-vous également qu'une adresse numérique telle que 075 occupe deux pas de programme et non pas trois. Le chiffre des centaines (0) est enregistré dans le premier pas suivant immédiatement l'instruction de transfert; le chiffre des dizaines et celui des unités sont contractés et enregistrés dans le second pas (75 dans le cas présent).

L'instruction de transfert elle-même doit nécessairement être réintroduite pour indiquer à la calculatrice que l'adresse doit être contractée.

Pour convertir l'exemple précédent en adresses numériques, suivez la séquence de touches suivantes.





Le code secret (Programme de jeu)

Ce dernier exemple de programme n'est pas destiné aux débutants. Toutes les techniques de programmation disponibles ont été utilisées de façon à rendre le programme aussi performant que possible. Le jeu est amusant en lui-même et le programme très instructif à étudier.

Le "Code secret" est un jeu dans lequel la calculatrice génère un nombre secret de quatre chiffres, que vous devez découvrir. Le zéro n'est pas autorisé et les 4 chiffres doivent être tous différents. Malgré ces restrictions vous avez 3024 codes possibles qui vous laissent très peu de chances de trouver le "Code secret" du premier coup. Le résultat de la partie est immédiatement affiché sous la forme "N", "R". N indique le nombre de chiffres exacts et situés dans la bonne position du nombre que vous avez essayé et R le nombre de chiffres incorrectement positionnés. Par exemple, si le "Code secret" généré est 8261 et votre essai 6285, le résultat sera "1.2", un chiffre est bon et correctement placé (2), deux chiffres sont bons mais mal positionnés (8 et 6). Le résultat final doit être 4.0 indiquant que tous les chiffres sont bons et bien placés.

Essayez de développer votre propre programme pour ce jeu. Ensuite étudiez l'exemple donné dans les pages suivantes. Optimisez votre programme de façon à utiliser le minimum d'instructions.

Le déroulement du programme dans cet exemple est simple, toutefois pour simplifier la compréhension nous avons placé simultanément sur l'organigramme, les explications et les instructions.

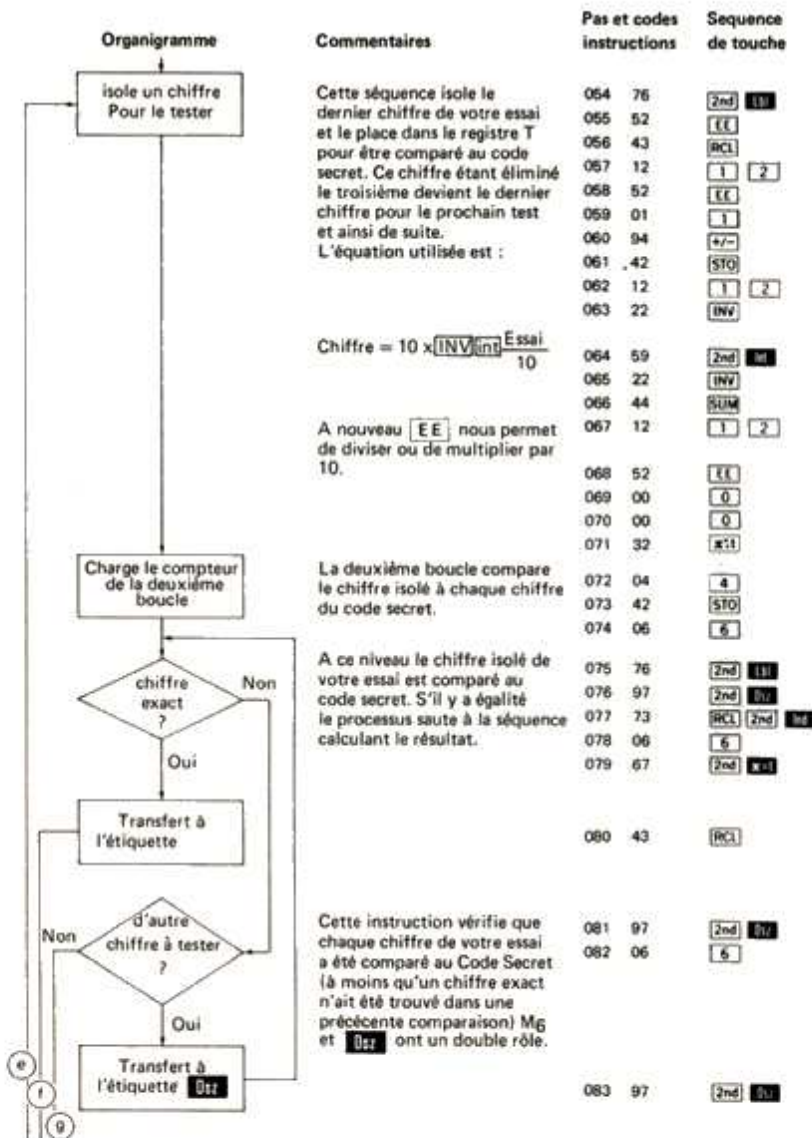
Pour démarrer, le programme doit générer un "Code secret" qui sera dérivé d'un nombre source introduit par la touche .



Organigramme	Commentaires	Pas et codes instructions	Séquence de touche
Associe A au nombre source	Cette séquence va déterminer le Code Secret	000 76 001 11 002 22 003 58 004 47 005 36 006 15 007 15	2nd 10 A INV 2nd 10 2nd 0M 2nd FIN 1 5 E
Efface les mémoires et appelle le programme 15	Le programme 15 de la Bibliothèque de base va générer un nombre aléatoire.	008 76 009 44 010 69 011 25	2nd 10 SUM 2nd 10 2 5
Incrémente indirectement le pointeur et le compteur de boucles	Le nombre généré est indirectement stocké en fonction du contenu de M5 ; Op 25 permet d'incrémenter M5 de 1.	012 76 013 42 014 43 015 05 016 42 017 06	2nd 10 STO RCL 5 STO 6
Restaure le compteur de boucles	Boucle destinée à vérifier qu'il n'y a ni zéro ni chiffre doublé. Pour réduire le temps d'exécution le nombre de boucles est réduit au minimum.	018 36 019 15 020 71 021 88 022 52 023 01 024 59 025 32 026 25	2nd FIN 1 5 SBR 2nd 0M EE 1 2nd 10 X:1 CLR
Génère le nombre aléatoire	Le sous-programme 0MS du Programme 15 génère un nombre x tel que $0 < x < 1$, x est ensuite multiplié par 10 puis la partie entière est placée dans le registre T . En utilisant la séquence C du programme 15 pour générer ce nombre, nous aurions utilisé des pas supplémentaires pour définir les bornes limites de sortie ; en plus, cette séquence utilise les Mémoires M1 à M5 que n'utilise pas 0MS . Remarquez la technique pour multiplier par 10 : elle économise un pas. De même CLR est utilisé pour supprimer la notation scientifique à la place de INV EE .		



Organigramme	Commentaires	Pas et codes instructions	Séquence de touches
	Puisque les mémoires ont été effacées au début, la première boucle teste si un zéro est généré. Les boucles suivantes testent si le chiffre est répété. Si un chiffre est rejeté, un nouveau est généré sans incrémenter M5. M5 sert à la fois de compteur de tours pour les boucles et de pointeur pour l'adressage indirect. L'instruction OSR sert également de compteur de boucles et de pointeur pour l'adressage indirect.	027 76	[2nd] [11]
		028 67	[2nd] [x=1]
	Si un chiffre est accepté, il est indirectement stocké en utilisant le pointeur M5	029 73	[RCL] [2nd] [1st]
		030 06	[5]
	Lorsque les quatre chiffres ont été acceptés, le code est trouvé. Le test porte sur 3 pour gagner 1 pas programme car en utilisant 4 il aurait fallu écrire 4 INV [2nd] [x=1] .	031 67	[2nd] [x=1]
		032 42	[STO]
	Cette séquence détermine le résultat de chaque essai.	033 97	[2nd] [0]
		034 06	[5]
	Deux boucles sont utilisées. la boucle principale traite chaque chiffre de votre essai.	035 67	[2nd] [x=1]
		036 32	[x=1]
	Arrête le programme et affiche zéro	037 72	[STO] [2nd] [1st]
		038 05	[5]
	Associe [B] aux différents essais	039 43	[RCL]
		040 05	[5]
	Mémorise l'essai et charge le compteur de la boucle principale	041 32	[x=1]
		042 03	[3]
	Cette séquence détermine le résultat de chaque essai.	043 77	[2nd] [x=1]
		044 44	[SUM]
	Arrête le programme et affiche zéro	045 00	[0]
		046 91	[R/S]
	Cette séquence détermine le résultat de chaque essai.	047 76	[2nd] [11]
		048 12	[0]
	Deux boucles sont utilisées. la boucle principale traite chaque chiffre de votre essai.	049 42	[STO]
		050 12	[1] [2]
		051 04	[4]
		052 42	[STO]
		053 06	[5]





Organigramme	Commentaires	Pas et codes instructions	Séquences de touches
	Si aucun chiffre n'est correct nous sautons la séquence de calcul du résultat.	084 61 085 25	GTD CLR
	Le chiffre est correctement positionné lorsque les compteurs de boucles sont au même niveau.	086 76 087 43 088 43 089 06 090 32 091 43 092 06 093 67 094 24	2nd 10 RCL RCL 5 x-1 RCL 5 2nd x-1 CE
	Si le chiffre n'est pas bien placé accorde 0,1 point au résultat ; si le chiffre est bien placé accorde 1 point au résultat. Regardez la façon dont sont combinées les deux opérations.	095 93 096 76 097 24 098 01 099 44 100 13	. 2nd 10 CE 1 SUM 1 3
	Utilise CLR à la place de INV EE ce qui économise deux pas du fait que cette instruction place aussi dans l'affichage un zéro dont nous allons avoir besoin.	101 76 102 25 103 25	2nd 10 CLR CLR
	Si la boucle principale a été utilisée pour chaque chiffre de votre essai, le programme est fini et le résultat affiché.	104 97 106 06	2nd 10 5
		106 52	EE
	En plus d'afficher le résultat Esc place 0 en M13 (Voir pas 103) pour préparer l'introduction de l'essai suivant. Fix 1 permet d'afficher le résultat N.R. même si R est nul.	107 48 108 13 109 58 110 01 111 91	2nd 10 1 3 2nd 10 1 R/S



IV

Le programme demande avant de démarrer l'introduction par d'un nombre compris en 0 et 1 utilisé comme source pour la génération du "Code secret". Une fois ceci accompli un zéro apparait sur l'affichage. Le jeu commence en introduisant vos essais par . L'exactitude de chaque essai est affichée sous forme d'un nombre "N.R" explicité dans l'introduction. Faisons une partie :

Introduisez	Appuyer	Affichage	Commentaires
.258	<input type="text" value="A"/>	0.	Introduit le nombre source générant le code secret
1234	<input type="text" value="B"/>	0.1	Premier essai
5678	<input type="text" value="B"/>	2.1	Deuxième essai
9238	<input type="text" value="B"/>	1.0	Troisième essai
5694	<input type="text" value="B"/>	1.0	Quatrième essai
5198	<input type="text" value="B"/>	2.1	Cinquième essai
5718	<input type="text" value="B"/>	4.0	Sixième essai correct

Un bon joueur ne doit pas dépasser six essais.

V

Analyse détaillée des caractéristiques et des fonctions



Nous allons entreprendre un descriptif détaillé de votre calculatrice. Cette section est spécialement destinée à fournir le moindre détail utile, dès lors que les fonctions de base sont correctement assimilées. Si vous ne possédez pas dès maintenant une compréhension suffisante de votre calculatrice, revenez aux sections précédentes pour compléter vos connaissances.

Dans cette section, les explications données sur les opérations et les fonctions du clavier s'appliquent à la fois au calcul manuel (opération par opération) et aux séquences de programme utilisant ces mêmes opérations.

OPERATIONS DE BASE

Affichage Standard.

En plus d'indiquer la mise en marche de la calculatrice, l'affichage reçoit les informations numériques ainsi que le signe négatif, le point décimal, et signale en clignotant les dépassements de capacité ou les conditions d'erreurs. Une introduction peut contenir jusqu'à 10 caractères, tout caractère suivant le dixième est ignoré.



Le signe d'un nombre négatif est immédiatement accolé à gauche du nombre lui-même.



Voir **Appendice C** pour la précision des résultats affichés.



Touches d'introduction de données.

Les touches ont été positionnées sur le clavier de façon à simplifier les opérations. Certaines sont inscrites en clair, d'autres non. Les instructions et les exemples qui suivent peuvent vous aider à développer votre assurance et votre dextérité dans l'emploi de votre calculatrice.

TOUCHES **0** à **9** - Introduisent les nombres.

. **POINT DECIMAL** - Introduit le point décimal. Le point décimal peut être introduit où vous le désirez. Si vous ne l'introduisez pas, il se placera à droite du nombre introduit et apparaîtra dès l'enfoncement d'une touche d'opération ou de fonction. Un zéro précède le point décimal pour les nombres plus petit que 1 à moins que les 10 caractères de l'affichage soient utilisés. Les zéros non significatifs de la partie décimale d'un nombre ne sont pas affichés. Seul le premier point décimal introduit est pris en compte, toute autre introduction est ignorée. Le fait d'appuyer sur le point décimal après l'introduction d'un exposant, vous permet de revenir à la mantisse.

2nd **π** **PI** - Introduit la valeur de **π** avec 13 caractères (3.141592653590) pour les calculs; l'affichage donne la valeur arrondie. **CE** n'efface pas π , toutefois cette valeur peut être effacée par l'introduction d'un autre nombre.

+/- **CHANGEMENT DE SIGNE** - Indique à la calculatrice de changer le signe du nombre affiché. Utilisée après **EE**, change le signe de l'exposant.

La procédure pour introduire un nombre positif consiste simplement à appuyer sur les touches dans l'ordre exact de l'écriture du nombre de la gauche vers la droite. Chaque caractère introduit décale d'un caractère vers la gauche le nombre affiché. Seul le premier point décimal introduit est accepté.

Exemple : $7.892 - \pi + (-2) = 2.750407346$

Appuyer	Affichage
7.892 =	7.892
2nd π	3.141592654
+	4.750407346
2 +/-	-2
=	2.750407346



Effacement des Opérations.

CE **EFFACEMENT D'UNE INTRODUCTION** - Efface l'introduction d'un nombre, du point décimal et du changement de signe pour autant qu'une touche de fonction n'ait pas été appuyée. Cette touche n'efface pas un résultat de calcul, un nombre rappelé d'une mémoire ou π . **CE** arrête le clignotement de l'affichage. Si le clignotement est dû à l'enfoncement successif de deux touches de fonction (condition d'erreur n° 5, page B-1), **CE** efface la dernière fonction introduite. L'utilisation de cette touche n'affecte pas les opérations en attente.

CLR **EFFACEMENT GENERAL** - Efface les calculs en cours et l'affichage. Change la notation scientifique en notation standard et arrête le clignotement de l'affichage. Cette touche n'a pas d'effet sur le contenu des mémoires programme, le registre T, le mode angulaire, la notation ingénieur, la virgule fixe ou la partition de la mémoire.

La calculatrice s'efface d'elle-même après la plupart des calculs, en effet lorsque vous appuyez sur **=** pour achever une opération, la réponse est affichée et la calculatrice est prête pour commencer un nouveau problème sans que vous soyez obligé d'appuyer sur une touche d'effacement. Le contenu des mémoires n'est pas automatiquement effacé.

2nd **CP** **EFFACEMENT MEMOIRE PROGRAMME** - Efface chaque pas de la mémoire programme, le registre d'adresses de retour des sous-programmes, le registre T, remet les drapeaux en position basse et ramène le pointeur au pas de départ 000 lorsque qu'utilisé à partir du clavier. Placé dans un programme, efface uniquement le registre T.

2nd **CNR** **EFFACEMENT DU CONTENU DES MEMOIRES** - Demande à calculatrice d'effacer toutes les mémoires définies par la partition en cours d'utilisation.

Touches à double fonction (**2nd** et **INV**).

La plupart des touches de votre calculatrice possèdent deux fonctions. La première est inscrite sur la touche et la seconde juste au-dessus. Pour exécuter une fonction inscrite sur la touche, vous appuyez simplement dessus. Pour utiliser la deuxième fonction de la touche, appuyez sur **2nd** avant d'appuyer sur la touche placée juste en-dessous de la fonction désirée. Par exemple, pour obtenir le logarithme népérien d'un nombre, appuyez sur **ln x** ; pour obtenir le logarithme décimal d'un nombre appuyez sur **2nd** **ln x** . Pour distinguer la deuxième fonction, nous écrivons dans le manuel **2nd** **ln** . La première fonction sera indiquée par **ln** et la seconde par **2nd** **ln** . Si vous appuyez deux fois de suite sur **2nd** ou si la touche n'a pas de deuxième fonction, la calculatrice revient d'elle-même en première fonction.



La touche inverse **INV** permet d'augmenter les possibilités de calcul sans augmenter le nombre de touches présentes sur le clavier, tout comme la touche **2nd**. En faisant précéder une touche par **INV** vous inversez l'application de la fonction indiquée sur cette touche. La touche inverse peut être utilisée avec les fonctions suivantes pour obtenir les fonctions indiquées.

Fonction	Fonction inverse
EE	Annule EE
ENG	Annule ENG et EE
Fix	Annule Fix
log	10^x
ln x	e^x
y^x	$\sqrt[y]{x}$
Int	Partie décimale
sin	Arc sin
cos	Arc cos
tang	Arc tan
Prod	Divise en mémoire
SUM	Soustrait de la mémoire
D.MS	Décimal en D.MS
P \rightarrow R	R \rightarrow P.
$\Sigma +$	$\Sigma -$
$\frac{\square}{x}$	Ecart type
list	Listage du contenu des mémoires
SBR	Retour de sous programme
$x = t$	$x \neq t$
$x \geq t$	$x < t$
if flg	Le drapeau est-il baissé?
st flg	Baisse un drapeau
Dsz	Saut si différent de zéro
Write	Lire

L'emploi programmé de la touche **INV** sera explicité dans la section concernant la programmation.

Si aucune fonction n'a été appuyée, la fonction inverse peut être annulée en appuyant une seconde fois sur **INV**. Certaines touches n'ayant pas de fonctions inverses, par exemple **LCI**, **LRN** etc..., le fait d'appuyer sur la touche inverse sera sans effet. Dans le cas des deuxièmes fonctions, vous pouvez indifféremment appuyer sur **INV** **2nd** **log** ou sur **2nd** **INV** **log**; ceci n'est vrai qu'au clavier; dans un programme, la touche **INV** doit nécessairement venir en premier. Pour des exemples d'utilisation de la fonction **INV**, reportez-vous au chapitre relatif à cette touche.



Formats d'affichage

Pour augmenter la souplesse d'utilisation de l'affichage standard à 10 caractères, plusieurs possibilités vous permettent d'accroître la capacité et les possibilités de votre calculatrice.

Même si un maximum de 10 caractères peuvent être affichés ou introduits, le registre interne d'affichage reçoit les résultats sur 13 caractères; c'est seulement lorsque le résultat est affiché qu'il est arrondi. Ces caractères supplémentaires sont conservés de façon à augmenter la précision des calculs, non pour fournir des chiffres significatifs supplémentaires. L'usage de ces caractères supplémentaires ne peut se faire qu'avec beaucoup de précaution. Voir **Appendice C** les explications concernant la précision.

NOTATION SCIENTIFIQUE

[EE] INTRODUCTION DE L'EXPOSANT - Indique à la calculatrice que le nombre introduit juste après est un exposant de 10. Après avoir appuyé sur **[EE]**, tous les résultats qui suivent sont affichés en notation scientifique jusqu'à ce que vous appuyez sur **[CLR]** ou que vous éteignez la calculatrice. De même **[INV]** **[EE]** ou **[INV]** **[2nd]** **[Eng]** peuvent annuler ce type d'affichage si le nombre affiché est compris entre $\pm 5 \times 10^{-11}$ et $\pm 1 \times 10^{10}$. Lorsque vous appuyez sur **[EE]** après un résultat (partiel ou final), les caractères internes 11, 12 et 13 sont éliminés et seule la valeur affichée est prise en compte pour la suite des calculs.

Tout nombre peut être interprété comme le produit d'une valeur (mantisse) et de 10 élevé à une certaine puissance (exposant). Introduisez la mantisse (jusqu'à 8 caractères), appuyez sur **[EE]** et introduisez l'exposant (2 caractères).



Cette possibilité vous permet de travailler avec des nombres aussi petits que $\pm 1 \times 10^{-99}$ ou aussi grand que $\pm 9,9999999 \times 10^{99}$. Les nombres plus petits que ,0000000001 ou plus grands que 9999999999 doivent être introduits en notation scientifique. Lorsque les résultats de calcul dépassent ces limites, la calculatrice passe automatiquement en notation scientifique. La procédure consiste à introduire la mantisse (jusqu'à 8 caractères) et son signe, à appuyer sur **[EE]** puis à introduire la puissance de 10 et son signe.

Par exemple, le nombre 320000000000 peut s'inscrire $3,2 \times 10^{11}$ et peut être introduit dans la calculatrice de la façon suivante:

Appuyer	Affichage
[CLR]	0
3.2	3.2
[EE]	3.2 00
11	3.2 11



Vous pouvez introduire plus de deux caractères après avoir appuyé sur **EE**, mais seulement les deux derniers seront retenus comme exposant. Cette possibilité permet de corriger une valeur erronée sans être obligé de recomposer l'ensemble du nombre introduit.

En notation scientifique, un exposant positif indique de combien de caractères il faut décaler le point décimal vers la droite; un exposant négatif indique de combien de caractères le point décimal doit être décalé vers la gauche.

La calculatrice normalise le nombre exprimé en notation scientifique en affichant un seul caractère à gauche du point décimal dès qu'une touche de fonction est utilisée.

Exemple : Introduisez 6025×10^{20} .

Appuyer	Affichage
CLR	0
6025	6025
EE	6025 00
20	6025 20
+	6.025 23

En notation scientifique, la mantisse est limitée à 8 caractères pour permettre d'afficher l'exposant. La mantisse résultant d'un calcul est également affichée sur 8 caractères, alors qu'en interne elle occupe 13 caractères. Cette valeur sur 13 caractères est la seule utilisée pour la suite des calculs. Reportez-vous à l'Appendice C pour plus de détails sur les caractères supplémentaires.

Note : Vous ne pouvez pas introduire un nombre en notation scientifique, même en appuyant sur **EE** si la mantisse contient plus de 8 caractères. Si vous appuyez sur **EE** et qu'il y a plus de 8 caractères sur l'affichage, celui-ci ne se mettra en notation scientifique qu'après que vous ayez appuyé sur une touche de fonction ou d'opération.

La touche de changement de signe permet de changer le signe de la mantisse ou de l'exposant. Il suffit d'appuyer sur **+/-** après l'introduction de la mantisse ou de l'exposant. Pour changer le signe de la mantisse ou compléter la partie décimale de celle-ci après avoir appuyé sur **EE**, appuyez sur **.** puis changez le signe de la mantisse ou complétez la partie décimale de celle-ci.

Exemple : Introduisez -4.962×10^{-12} puis complétez l'expression de façon à obtenir -4.96236×10^{12} .

Appuyer	Affichage	Commentaires
CLR	0	
4.962 +/-	-4.962	Introduit la mantisse et son signe
EE	-4.962 00	
12 +/-	-4.962-12	Introduit l'exposant et son signe
+/-	-4.962 12	Change le signe de l'exposant
. +/-	4.962 12	Change le signe de la mantisse
36 +/-	-4.96236 12	Complète la mantisse.



Des données en notation scientifique peuvent être mélangées avec des données au format standard. La calculatrice convertit dans le format adéquat pour les calculs. Après avoir appuyé sur **EE** la calculatrice affiche tous les résultats en notation scientifique jusqu'à ce que **CLR**, **INV** **EE** soient utilisés, ou, encore que la calculatrice soit éteinte. **CE** efface une introduction, mais conserve le format choisi pour l'affichage.

Exemple : $1,816 \times 10^3 - 581,432191 = 1,2345678 \times 10^3 = 1234,567809$

Appuyer	Affichage
CLR	0
1.816 EE	1.816 00
3 -	1.816 03
581.432191 =	1.2345678 03
INV EE	1234.567809

Si un résultat n'est pas compris entre $\pm 1 \times 10^{10}$ et $\pm 5 \times 10^{-11}$ en appuyant sur **INV** **EE**, la calculatrice restera en notation scientifique jusqu'à ce qu'un résultat compris entre ces bornes apparaisse.

Exemple : $(7 \times 10^{11} + 5 \times 10^{10}) \div 25 + 25 = 1200000000$

Appuyer	Affichage
7 EE	7 00
11 +	7. 11
5 EE	5 00
10 = INV EE	7.5 11
+	7.5 11
25 = +	3. 10
25 =	1200000000.

Si un calcul dépasse 999999999 ou est plus petit que .0000000001, l'affichage se met automatiquement en notation scientifique. Si **EE** n'a pas été utilisé durant la séquence de calcul, l'affichage reviendra automatiquement en format standard dès que cela sera numériquement possible.

Pour convertir un résultat en notation scientifique, il y a deux possibilités. La première est **X** 1 **EE** **=** qui multiplie le nombre contenu dans le registre d'affichage par 1×10^0 et convertit l'affichage en notation scientifique; le nombre complet avec ses 13 caractères est toujours présent. La seconde méthode est **EE** **=**. Soyez prudent avec cette seconde méthode qui générera le nombre ARRondi, tel qu'il est affiché pour la suite des calculs.

Vous devez éviter l'emploi d'une méthode utilisant **=** au milieu d'un calcul car cette touche exécute toutes les opérations en attente. Pour l'éviter, si vous n'êtes pas en fin de calcul, utilisez **X** 1 **EE** suivi d'un opérateur.



NOTATION INGENIEUR

Cette variante de la notation scientifique s'obtient en appuyant sur **[2nd]** **[Eng]**. La valeur affichée avec cette notation comprend une mantisse et un exposant, tous deux sont ajustés de sorte que l'exposant soit un multiple de trois (10^{12} , 10^{-6} , etc...). La mantisse présentera donc 1,2 ou 3 caractères à gauche du point décimal. Ceci permet à la calculatrice de fournir des résultats directement dans un système d'unités fréquemment employé tel que 10^{-12} pour picofarads, 10^{-3} pour millimètres, 10^6 pour megohms ou 10^{-9} pour nanosecondes, par exemple.

Exemple : Quel est le diamètre d'une fibre en microns (1 micron = 10^{-6} mètre) dont la circonférence est 3×10^{-3} mètres ?

Appuyer	Affichage
[CLR] [2nd] [Eng]	0. 00
3 [EE]	3 00
3 [+/-] [+]	3.-03
[2nd] [π] [=]	954.92966-06

Appuyez sur **[INV]** **[2nd]** **[Eng]** pour annuler la notation ingénieur ; les fonctions d'effacement et **[INV]** **[EE]** n'affectent pas cette notation.

VIRGULE FIXE

En affichage standard, notation scientifique ou notation ingénieur, vous pouvez sélectionner le nombre de chiffres après le point décimal que vous désirez en utilisant **[2nd]** **[Fix]** suivi du nombre de décimales choisies (0 à 8). La calculatrice arrondira tous les résultats affichés au nombre de décimales désirées. Cet arrondi affecte uniquement l'affichage à l'exclusion du registre d'affichage, si bien que les calculs sont poursuivis avec la précision maximale. Les séquences **[2nd]** **[Fix]** 9 ou **[INV]** **[2nd]** **[Fix]** replacent la calculatrice en affichage standard (virgule flottante).

Les données introduites peuvent avoir 10 caractères (8 en notation scientifique); l'exécution des calculs conserve ses 13 caractères de précision, excepté pour la conversion DMS-DD qui utilise uniquement la valeur affichée. Seul l'affichage est modifié pour donner le nombre de décimales désirées à condition de ne pas utiliser **[EE]** **[INV]** **[EE]** qui détruit les caractères non affichés.

La "mémoire permanente" de la TI-58C conserve la décimalisation choisie. Cette sélection est conservée jusqu'à ce qu'elle soit modifiée ou si l'alimentation électrique est coupée pendant une période trop longue.



Assurez-vous d'avoir annulé le format d'affichage en notation ingénieur avant de considérer l'exemple suivant :

Exemple : $2/3 = 0.666666667$

Appuyer	Affichage
2 \div	2.
3 $=$.666666667
2nd Fix 5	0.66667
2nd Fix 2	0.67
2nd Fix 0	1.
INV 2nd Fix	.666666667

Rappelez-vous que seul l'affichage est arrondi au nombre de décimales désirées.

Exemple : $1 \times 10^{-3} \div 2 = 0.0005$

Appuyer	Affichage
1 EE	1 00
3 $+/-$ \div	1. -03
2 $=$	5. -04
2nd Fix 2	5.00 -04
INV EE	0.00
2nd Fix 3	0.001
2nd Fix 4	0.0005
2nd Fix 5	0.00050

Vous remarquerez que le zéro au milieu de l'exemple ne correspond pas à un zéro dans le registre d'affichage, il est dû à l'arrondi à deux décimales demandé. N'oubliez pas que le registre d'affichage conserve toujours sa précision maximale même lorsque l'affichage est arrondi.

AFFICHAGE CLIGNOTANT.

L'affichage clignote chaque fois que les limites de la calculatrice sont dépassées ou qu'une opération mathématiquement impropre est demandée. Vous arrêtez le clignotement en appuyant sur CE sans pour autant perturber les calculs en cours; ceux-ci peuvent même être poursuivis dans la mesure où le nombre affiché est utilisable. CLR arrête le clignotement de l'affichage, mais il efface également l'affichage et les calculs en cours. Reportez-vous à l'Appendice B pour avoir la liste complète des erreurs et des dépassements de capacité ainsi que la signification des valeurs affichées en cas de clignotement de l'affichage.



CALCULS ARITHMETIQUES

La méthode d'introduction des nombres et des opérations avec cette calculatrice permet d'introduire la plupart des problèmes simplement, en respectant l'écriture mathématique. La calculatrice se souvient de chaque opération et si besoin est, la stocke jusqu'au moment où elle sera rappelée conformément aux principes de base de l'algèbre.

Opérations de base

Pour exécuter une simple addition, soustraction, multiplication ou division, la calculatrice avec la notation algébrique directe vous permet d'introduire le problème simplement dans l'ordre où il est écrit.

Exemple : $1.6 \times 10^{-19} \times 6.025 \times 10^{23} = 9.64 \times 10^4$

Appuyer	Affichage
CLR	0
1.6 EE	1.6 00
19 +/- X	1.6 -19
6.025 EE	6.025 00
23 =	9.64 04

Notez que la touche **=** exécute les opérations arithmétiques et affiche le résultat final.

En appuyant sur **CLR** au départ d'un calcul, vous effacez les opérations en cours et vous vous assurez ainsi qu'il n'y a plus dans la calculatrice d'opérations en attente dues à un calcul précédent. Cette manipulation n'est pas nécessaire si le résultat du précédent calcul a été obtenu avec la touche **=**. Cette touche a le même effet que la touche **CLR** si ce n'est que **=** laisse affiché le résultat, conserve le format d'affichage et n'arrête pas le clignotement.

Appuyer successivement sur deux des touches opérations **+** **-** **X** **÷** **y^x** provoque un clignotement de l'affichage. De même si vous appuyez après l'une d'elles sur **=** ou **)** ou avant l'une d'elles sur **(** vous provoquerez le clignotement de l'affichage. Reportez-vous à l'Appendice B pour plus de détails sur les conditions d'erreur.

Une fois obtenu le résultat d'un calcul vous pouvez utiliser celui-ci comme premier terme d'un second calcul, sans être obligé de le recomposer au clavier.

Exemple : $1.84 + 0.39 = 2.23$ $(1.84 + 0.39)/365 = 0.006109589$

Appuyer	Affichage	Commentaires
1.84 +	1.84	
.39 =	2.23	1.84 + 0.39
÷	2.23	
365 =	0,006109589	2.23 ÷ 365



Introduction à la notation algébrique directe. (A.O.S. ou Algebraic Operating System)

La hiérarchie algébrique est une caractéristique essentielle de la méthode d'introduction en notation algébrique directe. Pour rendre plus simple les combinaisons d'opérations, les règles élémentaires de la hiérarchie algébrique ont été spécialement programmées dans la calculatrice.

Les principes de l'algèbre donnent des priorités aux différentes fonctions mathématiques. En l'absence d'une liste de priorités, remarquez que l'expression $5 \times 4 + 3 \times 2$ pourrait avoir plusieurs interprétations :

$$\begin{aligned} & 5 \times (4 + 3) \times 2 = 70 \\ \text{ou } & (5 \times 4) + (3 \times 2) = 26 \\ \text{ou } & ((5 \times 4) + 3) \times 2 = 46 \\ \text{ou } & 5 \times (4 + (3 \times 2)) = 50 \end{aligned}$$

Les règles de l'algèbre donnent une priorité à la multiplication devant l'addition, donc algébriquement l'ambiguïté est levée; la réponse correcte est $(5 \times 4) + (3 \times 2) = 26$. La liste complète des priorités pour interpréter une expression est :

1. Fonctions mathématiques
2. Puissance (y^x) et racine ($\sqrt[x]{y}$)
3. Multiplication et division
4. Addition et soustraction
5. Egalité

1. Les fonctions mathématiques (trigonométrie, logarithme, carré, racine carrée, e^x , 10^x , partie entière, valeur absolue, inverse et conversions) remplacent instantanément la valeur affichée par la valeur de la fonction.
2. L'élevation à une puissance (y^x) et l'extraction de racine ($\sqrt[x]{y}$) sont effectuées ensuite.
3. La multiplication et la division sont exécutées après les fonctions mathématiques, la puissance et les racines.
4. L'addition et la soustraction sont exécutées seulement après les fonctions mathématiques, les puissances, les racines, les multiplications et les divisions.
5. L'égalité déclenche la résolution des opérations non encore effectuées dans l'ordre indiqué ci-dessus.



Une opération déclenche l'exécution d'une autre opération d'un niveau de priorité équivalent ou supérieur. Votre calculatrice connaît cette règle et l'applique à chaque problème au fur et à mesure de son introduction. Certaines opérations sont exécutées instantanément pendant que d'autres sont mises en attente jusqu'au moment où la règle définie commandera leur exécution. Pour illustrer ceci prenons un exemple.

Exemple : $4 \div 5^2 \times 7 + 3 \times 0.5^{\cos 60^\circ} = 3.241320344$

Appuyer	Affichage	Commentaires
4 \div	4	$(4 \div)$ est mis en attente
5 x^y	25	(5^2) , x^y donne le résultat immédiatement
x	0.16	$(4 \div 5^2)$, calcule la division puisque x a le même niveau de priorité que \div
7 $+$	1.12	x priorité supérieure devant $+$, ainsi $(4 \times 5^2 \times 7)$ est calculé, 1.12+ mis en attente
3 x	3	$(3 \times)$ est mis en attente
.5 y^x	0.5	$0.5 y^x$ est mis en attente
60 2^{nd} \cos	0.5	$\cos 60^\circ$ est calculé immédiatement
$=$	3.241320344	Exécute les opérations en attente : $0.5^{\cos 60^\circ}$, puis $3 \times 0.5^{\cos 60^\circ}$ et enfin additionne 1.12.

En introduisant cette expression comme elle est écrite, la calculatrice l'interprète correctement de la façon suivante :

$$\{[(4 + 5^2) \times 7] + (3 \times 0.5^{\cos 60^\circ})\}$$

L'élément important à ne pas oublier est que les opérations sont interprétées suivant la règle stricte des priorités. La calculatrice se souvient des opérations en attente et rappelle chacune d'elles avec son nombre associé au moment et à l'endroit précis où elles doivent être exécutées. Quand vous aurez assimilé ce principe, vous trouverez que la plupart des problèmes sont extrêmement simple à résoudre du fait de la facilité de leur introduction dans la calculatrice. L'usage des parenthèses permet en plus de contrôler l'ordre des exécutions.

Parenthèses

Dans certaines séquences d'opérations vous pouvez être amené à indiquer à la calculatrice la façon dont elle doit traiter un problème pour arriver au résultat. Les parenthèses vous donnent la possibilité « d'isoler » des nombres avec leurs opérateurs associés. En mettant entre parenthèses une série de nombres et d'opérations, vous indiquez à la calculatrice d'exécuter en priorité ce qui est compris entre les parenthèses avant de traiter les autres opérations.

V



Pour illustrer l'intérêt de ces parenthèses, introduisez (5×7) et constatez que la valeur affichée est 35. La calculatrice a évalué l'expression 5×7 et l'a remplacée par sa valeur numérique 35 sans que vous ayez eu à appuyer sur $=$. Les règles de l'algèbre s'appliquent à l'intérieur de chaque niveau de parenthèses. Leur utilisation vous donne la confirmation que votre problème peut être introduit exactement dans l'ordre où il est écrit. La calculatrice se souviendra de chaque opération et traitera chaque élément de l'expression dès que l'ensemble des informations lui sera fourni. Une fermeture de parenthèses déclenchera les opérations introduites depuis l'ouverture de parenthèse correspondante. Ces parenthèses peuvent être utilisées dès que vous avez le moindre doute sur la façon dont les calculs seront entrepris par la calculatrice.

Même si une expression s'écrit habituellement sous la forme $(3 + 2)(4 + 5)$ sous-entendant la multiplication entre les deux parenthèses, vous devez nécessairement introduire l'opérateur multiplication. **Votre calculatrice ne sait pas interpréter une multiplication sous-entendue.**

Exemple : $4 \times (5 + 9) \div (7 - 4)^{(2+3)} = 0,2304526749$ introduisez cette expression et suivez son déroulement.

Appuyer	Affichage	Commentaires
4 \times (4.	(4 x) mis en attente pendant l'exécution de la parenthèse.
5 +	5.	(5 +) mis en attente.
9)	14.	(5 + 9) calculé
\div	56.	Suivant la règle, (4 x 14) est calculé.
(56.	(56 \div) mis en attente pendant l'exécution de la parenthèse.
7 -	7.	(7 -) mis en attente
4)	3.	(7 - 4) calculé
y^x (3.	(3 y^x) mis en attente
2 +	2.	(2 +) mis en attente
3)	5.	(2 + 3) calculé
=	.2304526749	(7 - 4) ⁽²⁺³⁾ est calculé et divise 4 x (5 + 9).

Il y a une limite au nombre d'opérations et de nombres associés pouvant être mis en attente. Vous pouvez en fait ouvrir jusqu'à 9 niveaux de parenthèses imbriqués correspondant à 8 opérations et nombres associés en attente. Cette limite est rarement atteinte même avec des calculs très complexes. Si vous essayez d'ouvrir plus de 9 niveaux de parenthèses imbriqués ou d'introduire plus de 8 opérations en attente, l'affichage clignotera.



Exemple : $5 + 8/[9 - (2/3)] = 5.96$

Appuyer	Affichage	Commentaires
5 + (5.	
8 + (8.	
9 - (9.	
2 ÷ 3)	.666666667	(2/3) est calculé
)	8.333333333	[9 - (2/3)] est calculé
)	0.96	8/[9 - (2/3)]
=	5.96	$5 + \{8/[9 - (2/3)]\}$

Du fait que la touche $\boxed{=}$ déclenche l'exécution des calculs en attente, elle peut être utilisée à la place des trois fermetures de parenthèses. Reprenez cet exemple en appuyant sur $\boxed{=}$ à la place de la première fermeture de parenthèse.

Exemple : $3 \times [4^{2(-4\sqrt{7})}] = 4.700043401$

Appuyer	Affichage	Commentaires
CLR (0.	
3 X (3.	
4 y ^x (4.	
2 y ^x (2.	
7 INV y ^x	7.	
4)	1.626576562	$\sqrt[4]{7}$
+/-	-1.626576562	$-(\sqrt[4]{7})$
)	.3238557891	$2^{-1.626576562}$
)	1.566681134	$4^{.3238557891}$
)	4.700043401	$3 \times 4^{.3238557891}$

Dès qu'une parenthèse est fermée, les calculs indiqués depuis la plus proche ouverture de parenthèse sont effectués, le niveau de parenthèses correspondant est remplacé par sa valeur numérique. Ceci étant connu, il vous est possible de structurer l'ordre du déroulement des opérations de façon à atteindre votre but. Vous pouvez également obtenir ainsi des résultats intermédiaires.



OPERATIONS FACTICES AVEC LES PARENTHÈSES.

L'opération factice **CE** utilisée avec des parenthèses permet de réintroduire la valeur affichée à l'intérieur de celles-ci sans avoir à la recomposer, ou plus particulièrement, elle permet d'éviter la réintroduction d'une valeur figurant deux fois consécutives dans une expression, comme le montre l'exemple suivant :

Exemple : $3.296214 + (3.296214 \times 6) = 23.073498$

Appuyer	Affichage	Commentaires
CLR 3.296214 +	3.296214	
(CE X	3.296214	CE réintroduit 3,296214
6)	19.777284	
=	23.073498	

FONCTIONS ALGÈBRIQUES.

Les fonctions d'une variable sont les plus simples à décrire et à comprendre. Ces fonctions opèrent sur le contenu du registre d'affichage et remplace celui-ci par la valeur de la fonction correspondante. Ces fonctions n'ont aucun effet sur les calculs en cours et peuvent ainsi être utilisées à n'importe quel endroit dans un calcul. La précision de ces fonctions sera abordée dans l'Appendice C ou au cours des explications.

INVERSE

$\frac{1}{x}$ **INVERSE** -Calcule l'inverse du nombre contenu dans le registre d'affichage x en divisant 1 par x. x = 0 provoque le clignotement de l'affichage.

Appuyer	Affichage
3.2 $\frac{1}{x}$	0.3125

Remarquez que dès qu'une fonction mathématique est utilisée la valeur correspondante de la fonction est affichée.



Logarithmes

[Inx] LOGARITHME NEPERIEN - Calcule le logarithme népérien (base e) du nombre contenu dans le registre d'affichage x. $x \leq 0$ provoque le clignotement de l'affichage.

[2nd] [log] LOGARITHME DECIMAL - Calcule le logarithme décimal (base 10) du nombre contenu dans le registre d'affichage x. $x \leq 0$ provoque le clignotement de l'affichage.

Exemple : $\log(1 + \ln 1.7) = 0.1848697249$

Appuyer	Affichage
[CLR]	0
[(] 1 [+]	1.
1.7 [Inx])	1.530628251
[2nd] [log]	.1848697249

Puissance de 10 et de e

[INV] [Inx] EXPONENTIELLE - Calcule l'exponentielle, e^x , du nombre contenu dans le registre d'affichage x. $x \geq 230.2585093$ ou $x \leq -227.9559243$ provoque le clignotement de l'affichage.

[INV] [2nd] [log] ANTILOG DECIMAL - Calcule l'élévation de 10 à la puissance x, x étant le nombre contenu dans le registre d'affichage. $x < -99$ ou $x > 99.99999999$ provoque le clignotement de l'affichage.

Exemple : $e^{(3+10^{0.3})} = 147.7116873$

Appuyer	Affichage
[CLR] [(] 3 [+]	3.
.3 [INV] [2nd] [log])	4.995262315
[INV] [Inx]	147.7116873

Calculs angulaires

Votre calculatrice vous offre un maximum de souplesse dans les calculs relatifs aux angles.

MODES ANGULAIRES

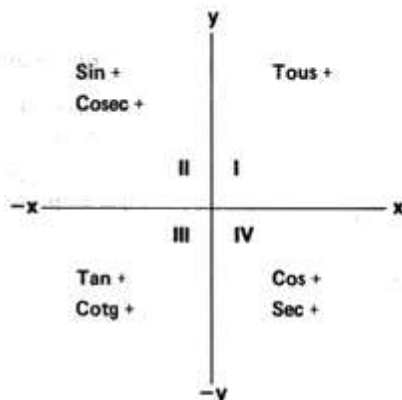
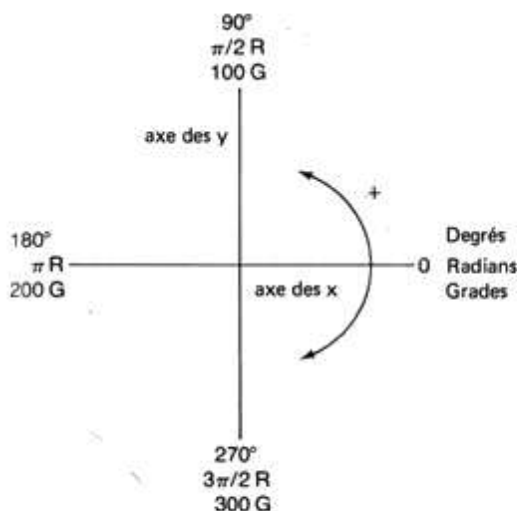
Un angle peut être mesuré en degrés décimaux, radians ou grades (angle droit = $90^\circ = \pi/2$ radians = 100 grades). A la mise sous tension la calculatrice est en degrés et y reste tant qu'un autre choix n'a pas été fait. Une fois le mode angulaire sélectionné en appuyant sur **[2nd] [Deg]**, **[2nd] [Rad]** ou **[2nd] [Grad]**, tous les calculs d'angles ou les introductions seront interprétés suivant le mode choisi jusqu'à la sélection d'un mode différent ou l'extinction de la calculatrice. **[CE]** et **[CLR]** n'ont pas d'effet sur le mode angulaire choisi.

Le mode angulaire n'a absolument aucun effet sur les calculs à l'exception des fonctions trigonométriques et de la conversion de coordonnées polaires en coordonnées cartésiennes. Si le choix du mode angulaire est simple à effectuer il est SIMPLE A OUBLIER. Son oubli explique un grand nombre des erreurs rencontrées dans les calculs angulaires.



FONCTIONS TRIGONOMETRIQUES.

2nd **sin**, **2nd** **cos**, **2nd** **tan** – SINUS, COSINUS, TANGENTE – Calculent le sinus, le cosinus ou la tangente du nombre contenu dans le registre d'affichage. Tous les angles sont mesurés à partir de l'axe des X, positifs dans le sens inverse des aiguilles d'une montre, négatifs dans le sens des aiguilles d'une montre.



Le diagramme de droite indique en fonction des quadrants I à IV les fonctions trigonométriques positives. Les fonctions non indiquées sont négatives.

Lorsque vous mesurez un angle rappelez-vous que chaque angle a un équivalent de signe opposé. Par exemple -45° a pour équivalent 315° .

Si vous avez un angle exprimé en degrés, minutes, secondes, la touche **DMs** effectue la conversion en degrés décimaux. (Voir page V-30). Assurez-vous que la calculatrice est en mode degrés, si vous avez un doute appuyez sur **2nd** **0/gp**.

Exemple : $\sin 30^\circ 13' 48'' + \tan 315^\circ = 0.4965275891$

Appuyer	Affichage
30.1348 2nd DMs	30.23
2nd sin +	.5034724109
315 2nd tan	-1.
=	-.4965275891



Il est possible d'obtenir la valeur des fonctions trigonométriques d'angles supérieurs à une révolution. Tant que la valeur des fonctions trigonométriques n'est pas affichée en notation scientifique, celle-ci, affichée en format standard, est exacte à $\pm 1 \times 10^{-9}$ près pour les angles compris dans la gamme $\pm 36\,000$ degrés, $\pm 200\pi$ radians ou $\pm 40\,000$ grades. En général, la précision diminue d'un chiffre à chaque dizaine hors de cette gamme. Pour un argument x supérieur à $\pm 3.6 \times 10^{14}$ degrés, $\pm 4 \times 10^{14}$ grades ou $\pm 2\pi \times 10^{12}$ radians, une rotation partielle n'est plus identifiable.

Les autres fonctions trigonométriques peuvent se calculer simplement

cotg	=	2nd	tan	1/x
sec	=	2nd	sin	1/x
cosec	=	2nd	cos	1/x

INVERSES DES FONCTIONS TRIGONOMETRIQUES

INV – INVERSE – Lorsque placée devant une autre touche, donne la fonction inverse de cette touche. Par exemple Arcsin (Sin^{-1}) s'obtient en appuyant sur **INV** **2nd** **sin**.

L'inverse des fonctions trigonométriques calcule l'angle dont la valeur de la fonction correspondante est affichée. Le plus grand angle exprimé résultant du calcul d'une fonction Arc est 180 degrés, π radians ou 200 grades. Compte tenu du fait qu'à une valeur correspond un infini d'angles équivalents, par exemple $\text{Arcsin } 0.5 = 30^\circ, 150^\circ, 390^\circ$ etc., l'angle exprimé par chaque fonction sera défini suivant le tableau ci-dessous :

FONCTION ARC	ANGLE RESULTANT
Arcsin x	entre 0 et 90° , $\pi/2$ ou 100 G
Arcsin $(-x)$	entre 0 et -90° , $-\pi/2$ ou -100 G
Arccos x	entre 0 et -90° , $-\pi/2$ ou -100 G
Arccos $(-x)$	entre 90 et 180° , $-\pi/2$ et π ou 100 et 200 G
Arctg (x)	entre 0 et 90° , $-\pi/2$ ou 100 G
Arctg $(-x)$	entre 0 et -90° , $-\pi/2$ ou -100 G

Pour arcsin x et arccos x , $x > 1$ ou $x < -1$ provoque le clignotement de l'affichage.

Exemple : $\pi/4 + \text{tg}^{-1}(0.2\pi) = 1.34638028$

Appuyer	Affichage
2nd Rad	0
2nd π +	3.141592654
4 +	.785391634
(.2 X 2nd π)	.6283185307
INV 2nd tan	.5609821161
=	1.34638028

La sélection du mode angulaire radian peut se faire à n'importe quel moment avant la fonction **INV** **2nd** **tan** ; il est préférable cependant de le sélectionner en début de calcul. Le choix du mode angulaire n'affecte que l'interprétation des mesures angulaires.



L'inverse des autres fonctions trigonométriques peut se calculer ainsi :

$$\begin{aligned} \text{Arcotg} &= \boxed{1/x} \boxed{\text{INV}} \boxed{2\text{nd}} \boxed{\pi} \\ \text{Arcsec} &= \boxed{1/x} \boxed{\text{INV}} \boxed{2\text{nd}} \boxed{\pi} \\ \text{Arcosec} &= \boxed{1/x} \boxed{\text{INV}} \boxed{2\text{nd}} \boxed{\pi} \end{aligned}$$

CONVERSIONS DE DEGRES, RADIANS ET GRADES

Il est souvent nécessaire de pouvoir convertir la valeur d'un angle d'un système d'unités à un autre. Le tableau ci-dessous vous donnera les coefficients nécessaires pour traiter ces conversions.

DE \ A	DEGRES	RADIANS	GRADES
DEGRES		$\times \frac{\pi}{180}$	$\div 0.9$
RADIANS	$\times \frac{180}{\pi}$		$\times \frac{200}{\pi}$
GRADES	$\times 0.9$	$\times \frac{\pi}{200}$	

Ces opérations peuvent être effectuées quelque soit le mode angulaire sélectionné.

Exemple : Convertir 120° en radians puis en grades.

Appuyer	Affichage	Commentaires
120 $\boxed{\times}$ $\boxed{2\text{nd}}$ $\boxed{\pi}$ $\boxed{\div}$	376.9911184	
180 $\boxed{=}$ $\boxed{\times}$	2.094395102	Radians
200 $\boxed{\div}$ $\boxed{2\text{nd}}$ $\boxed{\pi}$ $\boxed{=}$	133.3333333	Grades
$\boxed{\times}$	133.3333333	
.9 $\boxed{=}$	120.	Degrés

Si le résultat d'une de ces conversions est utilisé pour un calcul de fonction angulaire, Le mode angulaire doit être sélectionné correctement pour que le résultat soit cohérent.



Partie entière et valeur absolue

2nd **int** — **PARTIE ENTIERE** — Supprime la partie fractionnaire du nombre contenu dans le registre d'affichage. **INV** **2nd** **int** supprime la partie entière du nombre contenu dans le registre d'affichage.

2nd **|x|** — **VALEUR ABSOLUE** — Rend positif le contenu du registre d'affichage.

Exemple : Trouver la valeur absolue de la partie entière de $-13/5$.

Appuyer	Affichage
13 +/- ÷	-13.
5 =	-2.6
2nd int	-2.
2nd x 	2.

Ces fonctions sont particulièrement utiles en programmation.

Notez que la partie entière opère sur le registre d'affichage et non sur l'affichage. Pour cette raison si vous avez 4.999999999999 dans le registre d'affichage (affichage arrondi à 5), en appuyant sur **2nd** **int** vous obtenez 4 et non pas 5. Pour conserver ce 5 il est nécessaire de faire **EE** **INV** **EE** pour éliminer les caractères non affichés avant d'appuyer sur **2nd** **int**.

En notation scientifique, seules les décimales réelles sont supprimées, par exemple : 1.2345×10^3 **2nd** **int** devient 1.234×10^3 et 1234.5 devient 1234.

CARRE ET RACINE CARREE.

x² — **CARRE** — Elève au carré le nombre contenu dans le registre d'affichage. $|x| > 10^{5.0}$ provoque le clignotement de l'affichage.

√x — **RACINE CARREE** — Calcule la racine du nombre contenu dans le registre d'affichage. x négatif provoque le clignotement de l'affichage.

Exemple : $\sqrt{3.1452} - 7 + (3.2)^2 |^{1/2} = 2.239078197$

Appuyer	Affichage.
CLR (0.
3.1452 √x -	1.773471173
7 +	-5.226528827
3.2 x²	10.24
)	5.013471173
√x	2.239078197



Puissances et racines

[y^x] – **PUISSANCE** – élève le nombre y contenu dans le registre d’affichage à la puissance x. La séquence d’introduction est y **[y^x]** x suivie d’une opération de niveau inférieur ou équivalent (voir hiérarchie algébrique page V-11) ou de la touche **[=]**. Y < 0 provoque le clignotement de l’affichage.

[INV] [y^x] – **RACINE** – ($\sqrt[x]{y}$ ou $y^{1/x}$) – calcule la racine x ième du nombre contenu dans le registre d’affichage y. La séquence d’introduction est y **[INV] [y^x]** x suivie d’une opération de niveau inférieur ou équivalent ou de la touche **[=]**. Y < 0 ou x = 0 ou y = 0 et x < 0 provoquent le clignotement de l’affichage.

Exemple : $\sqrt[3]{2.36^{-0.23}} = 0.9362893421$

Appuyer	Affichage	Commentaires
2.36 [y^x]	2.36	Introduit y de y ^x
.23 [+/-]	-0.23	Introduit x de y ^x
[INV] [y^x]	.8207865654	Calcule y de $\sqrt[y]{x}$
3 [=]	.9362893421	Introduit x de $\sqrt[x]{y}$ et calcule le résultat

La fonction y^x est calculée par l’intermédiaire des logarithmes; le tableau ci-dessous vous indique ce qui se produit en fonction des valeurs de x et de y suivant les règles fondamentales des mathématiques. Les guillemets symbolisent le clignotement de l’affichage.

y	x	y ^x	$\sqrt[x]{y}$
0	0	1.	"1."
0	-x	"9.9999999 99"	"9.9999999 99"
0	x	0.	0.
1	0	1.	"1."
y	0	1.	"9.9999999 99"
-1	0	"1."	"1."
-y	0	"1."	"9.9999999 99"
-y	±x	" y ^{±x} "	" $\sqrt[±x]{ y }$ "



CARACTERISTIQUES DES MEMOIRES

Des mémoires de données accessibles à l'utilisateur vous permettent de stocker ou d'accumuler des données pour les utiliser ultérieurement. Cette zone de stockage est couramment appelée mémoires de données ou mémoires par opposition à la mémoire programme où est stocké le programme. Les touches mémoire sont utilisables à tout moment dans un calcul, celles-ci n'ayant aucun effet sur les calculs en cours. Le contenu des mémoires est conservé dans la TI-58C si celle-ci est momentanément arrêtée ainsi que si elle se trouve non alimentée pendant une courte période (passage sur l'imprimante ou batteries déchargées).

Le choix de la mémoire utilisée est absolument arbitraire sauf si vous effectuez des calculs statistiques qui utilisent les mémoires 01-06. Si vous utilisez un grand nombre de mémoires, il est recommandé de noter la désignation de chaque donnée stockée en regard du numéro de chaque mémoire.

Choix de la taille mémoire (Partition)

Dans ce chapitre, les informations relatives à la TI-58 ou TI-58C seront indiquées entre parenthèses après celles données pour la TI-59.

A la mise en marche, vous avez 60 (30) mémoires de données. En ce qui concerne la TI-58C la partition choisie est conservée dans la **mémoire permanente** même si la calculatrice est momentanément arrêtée.

Les mémoires de données représentent une partie de l'ensemble des mémoires de stockage, l'autre partie étant affectée au stockage du programme. Cet ensemble de mémoires peut être ajusté en fonction de vos besoins en modifiant l'un et l'autre de ces domaines par groupe de 10 mémoires. Il suffit pour cela d'introduire le nombre de groupes de 10 mémoires dont vous avez besoin et d'appuyer sur [2nd] [Op] 17. Par exemple pour avoir 20 mémoires, appuyez sur 2 [2nd] [Op] 17, l'affichage vous indique ceci :



Dans ce cas vous disposez des 20 mémoires numérotées 00 à 19 et de 800 (320) pas de programme numérotés 000 à 799 (000 à 319). Avant de changer une partition, annuler le format virgule fixe, notation scientifique ou notation ingénieur. Vous avez en pratique une correspondance de 8 pas de programme par mémoire.

En appuyant sur [2nd] [Op] 16 vous pouvez à tout moment vérifier la partition choisie, elle se trouve affichée comme indiqué ci-dessus. Pour plus de détails reportez-vous page V-42.

Du fait que vous pouvez utiliser jusqu'à 100 (60) mémoires il est nécessaire de préciser la mémoire que vous choisissez par une adresse à deux chiffres XX qui sera accolée immédiatement après la touche mémoire utilisée. L'absence de cette indication prévoque le clignotement de l'affichage. Vous pouvez utiliser une adresse simplifiée à un chiffre si le numéro de la mémoire est inférieur à 10 et si cette adresse est immédiatement suivie d'une touche non numérique. La souplesse de ce système de mémoire permet de manipuler des données de façon très variée.



Effacement des mémoires

2nd CLR — **EFFACEMENT DES MEMOIRES** — indique à la calculatrice d'effacer l'ensemble des mémoires disponibles suivant la partition.

L'usage de cette touche n'a pas d'effet sur le registre T, la mémoire programme, la partition, l'affichage ou sur les calculs en cours.

Stockage et rappel des données

STO XX — **STOCKAGE** — stocke le nombre contenu dans le registre d'affichage en mémoire **XX**. Une donnée contenue précédemment dans cette mémoire se trouve éliminée. Le nombre contenu dans le registre d'affichage est conservé.

RCL XX — **RAPPEL** — rappelle sur l'affichage le nombre contenu en mémoire **XX**, ce nombre reste en mémoire **XX**. Le rappel d'un nombre peut être utilisé pour introduire un nombre dans un calcul.

Exemple : stocker 3.012 en mémoire **22** et le rappeler.

Appuyer	Affichage
3.012 STO 22	3.012
CLR	0
RCL 22	3.012

Une telle séquence sert souvent lorsque les nombres sont longs et utilisés plusieurs fois.

Exemple : Calculer $3x^2 - x - 7.1$ pour $x = 2.9467281$

Appuyer	Affichage
CLR 3 X	3.
2.9467281 STO 12	2.9467281
x² -	26.04961949
RCL 12	2.9467281
- 7.1 =	16.00289139

Il suffit d'introduire x une seule fois, le stockage et le rappel n'ont pas d'effet sur les calculs en cours.



Vous pouvez également utiliser une mémoire pour stocker un résultat intermédiaire à réutiliser plus tard.

Exemple : calculer $\frac{\sin(3x/2) - \cos(3x/2)}{x}$ pour $x = 20.6821776$ degrés

Appuyer	Affichage	Commentaires
2nd [MC] () () ³ [X]	3.	
20.6821776 [STO] 14	20.6821776	Stocke x en mémoire 14
[+] 2 [)] [STO] 17	31.0232664	Stocke 3x/2 en mémoire 17
2nd [sin] [-]	.5153861069	
[RCL] 17	31.0232664	Rappelle 3x/2 de la mémoire 17
2nd [cos] [)] [+]	-.3415719789	
[RCL] 14	20.6821776	Rappelle x de la mémoire 14
[=]	-.0165152812	Résultat

Tenter d'utiliser une mémoire située au delà des limites de la partition choisie provoque le clignotement de l'affichage.

Arithmétique en mémoire

Vous pouvez stocker un nombre affiché à n'importe quel moment pendant un calcul sans affecter le déroulement de celui-ci. Vous pouvez également additionner, soustraire, multiplier et diviser directement en mémoire, sans modifier le contenu du registre d'affichage. Si vous provoquez un dépassement de capacité en mémoire l'affichage clignote, de même si vous appelez une mémoire non définie par la partition.

[SUM] XX – SOMME EN MEMOIRE – ajoute le nombre contenu dans le registre d'affichage au contenu de la mémoire XX; le résultat est stocké en mémoire XX.

[INV] [SUM] XX – SOUSTRACTION EN MEMOIRE – soustrait le nombre contenu dans le registre d'affichage de celui contenu en mémoire XX, le résultat est stocké en mémoire XX.

2nd [Prf] XX – PRODUIT EN MEMOIRE – multiplie le nombre contenu dans le registre d'affichage par celui contenu en mémoire XX, le résultat est stocké en mémoire XX.

[INV] 2nd [Prf] XX – DIVISION EN MEMOIRE – divise le nombre contenu en mémoire XX par celui contenu dans le registre d'affichage, le résultat est stocké en mémoire XX.

Ces possibilités évitent un rappel et un stockage ultérieurs qui allongeraient les séquences de calcul.

V



Exemple : calculer $x^2 + 9$ pour $x = -1, 2$ et 3 et cumuler les résultats.

Appuyer	Affichage	Contenu de la mémoire 3
1 +/- x^2 +	1.	0
9 = STO 03	10.	10
2 x^2 +	4.	10
9 = SUM 03	13.	23
3 x^2 +	9.	23
9 +	18.	23
RCL 3	23.	23
=	41.	23

Le premier résultat est placé en mémoire 3 en utilisant la touche **STO** ce qui permet de s'assurer que son précédent contenu est bien éliminé, le second résultat est cumulé directement en mémoire.

Exemple : le pourcentage des étudiants terminant les différentes années d'une certaine école est 76.8 % pour les élèves de première année; 81.3 % pour ceux de la deuxième année; 92.2 % pour ceux de la troisième année; 95.9 % pour ceux de la dernière année. Quel est le pourcentage des étudiants diplômés et quel est le pourcentage de ceux qui accomplissent la troisième et la quatrième année ?

Appuyer	Affichage
.768 X	0.768
.813 X	0.624384
.922 STO 11 X	0.575682048
.959 2nd F11 11 =	0.552079084
RCL 11	0.884196

Environ 55 % des élèves sortant sont diplômés, plus de 88 % de ceux qui sont passés en troisième année sortent diplômés.



Echange affichage – Mémoire

2nd **MC** **XX** – ECHANGE MEMOIRE —échange le nombre contenu dans le registre d'affichage avec celui contenu dans la mémoire **XX**. Le nombre contenu dans le registre d'affichage est stocké en mémoire **XX** et, simultanément, celui contenu en mémoire est affiché.

Cette touche permet d'économiser des manipulations de touches supplémentaires : vous pouvez l'utiliser pour examiner deux résultats sans perdre l'un des deux. Vous pouvez aussi stocker momentanément un nombre et le réutiliser quand vous en aurez besoin.

Exemple : calculer $A^2 + AB + 2B^2$ pour $A = 0.258963$ et $B = 1.255632$

Appuyer	Affichage	Commentaires
.258963 STO 13	0.258963	Stocke A en mémoire 13
x² + 1.255632 X	1.255632	Introduit B
2nd MC 13	0.258963	Stocke B, rappelle A
+ 2 X	2.	
RCL 13	1.255632	Rappelle B de la mémoire 13
x² =	3.545447504	Résultat



OPERATIONS SPECIALES

Une série d'opérations spéciales sont accessibles par la touche **Op** qui augmente considérablement les possibilités de votre calculatrice. Certaines de ces opérations peuvent être utilisées quelque soit le mode de calcul employé alors que d'autres sont destinées à un mode d'utilisation particulier ou à l'emploi de l'imprimante optionnelle PC-100A, PC-100B ou PC-100C.

Pour utiliser une de ces opérations il suffit d'appuyer sur **2nd Op nn**, nn étant au code à deux chiffres identifiant l'opération en question (l'usage du code adresse contracté est possible). Une description de ces différentes opérations est donnée ci-dessous.

Code nn	Fonction correspondante
00*	Initialise le registre d'impression
01*	Interprète les 10 chiffres affichées comme 5 codes alphanumériques à placer à l'extrême gauche de la ligne à imprimer.
02*	Interprète les 10 chiffres affichés comme 5 codes alphanumériques placés au centre gauche de la ligne à imprimer.
03*	Interprète les 10 chiffres affichés comme 5 codes alphanumériques placés au centre droit de la ligne à imprimer.
04*	Interprète les 10 chiffres affichés comme 5 codes alphanumériques placés à l'extrême droite de la ligne à imprimer.
05*	Imprime le contenu du registre d'impression.
06*	Imprime les 4 derniers caractères de OP 04 ainsi que le contenu de l'affichage.
07*	Imprime un * dans la colonne spécifiée par l'affichage.
08*	Liste des étiquettes utilisées par le programme situé en mémoire programme.
09	Transfère un programme déterminé de la bibliothèque de programmes (module) en mémoire programme.
10	Indicateur de signe du contenu du registre d'affichage.
11	Calcule la variance.
12	Calcule la pente et l'ordonnée à l'origine.
13	Calcule le coefficient de corrélation
14	Calcule la valeur de y (y') pour x introduit sur l'affichage.
15	Calcule la valeur de x (x') pour y introduit sur l'affichage.
16	Affiche la partition mémoire courante.
17	Change la partition mémoire.
18	Lève le drapeau 7 si aucune condition d'erreur n'est rencontrée.
19	Lève le drapeau 7 si une condition d'erreur est rencontrée.
20-29	Incrémente les mémoires 0 à 9 de 1.
30-39	Décrompte les mémoires 0 à 9 de 1.
40*	Lève le drapeau 7 si l'imprimante est connectée (TI-58C seulement)

*Utilisable seulement avec l'imprimante optionnelle PC-100A, PC-100B ou PC-100C.



Possibilités d'impression - Op 00-08

Ces opérations sont spécifiquement prévues pour être utilisées avec l'imprimante optionnelle. Cette imprimante accroît la souplesse de votre calculatrice, en vous donnant la possibilité d'avoir une «trace écrite» des différents éléments de vos calculs. Les opérations de contrôle permettent d'ajouter aux caractéristiques propres de l'imprimante, la possibilité d'imprimer des messages alphanumériques, de tracer des courbes par points et d'éditer la liste des étiquettes présentes dans un programme avec leurs adresses. La section **Contrôle de l'imprimante** page VI-7 vous donnera dans le détail la façon d'utiliser ces différentes possibilités.

Analyse d'un programme de la bibliothèque (Transfert en mémoire-programme) - Op 09.

En appuyant sur [2nd] [Pgm] mm [2nd] [Op] 09 vous placez une copie du programme mm en mémoire programme. Vous pouvez utiliser alors toutes les fonctions de vérification du clavier et également adapter ce programme à vos besoins particuliers. Soyez prudent en effectuant des modifications de ne pas détériorer les adresses numériques ou les codes contractés. Le transfert en mémoire programme chasse les éventuelles instructions qui s'y seraient trouvées sur une longueur égale à celle du programme transféré. Le programme se place à partir du pas 000. Le transfert inverse est impossible à réaliser. Si la place laissée disponible par la partition n'est pas suffisante, le transfert ne se fait pas et l'affichage clignote. Il clignote également si le code mm ne correspond pas à un programme existant ou si le module n'est pas placé dans la calculatrice.

Indicateur de signe - Op 10.

L'opération Op 10 indique l'état du nombre x contenu dans le registre d'affichage d'après le principe suivant :

Contenu x du registre d'affichage	Indication affichée
$x > 0$	1.
$x = 0$	0.
$x < 0$	-1.

Statistiques - Op 11-15.

Les opérations spéciales 11-15 permettent d'effectuer des analyses statistiques, elle seront développés en détail au chapitre **Conversions et statistiques**.

**Partition mémoire - Op 16-17**

En appuyant sur **2nd** **Op 16** vous affichez instantanément l'état actuel de la partition mémoire. L'affichage vous indique, séparés par un point décimal, le numéro du dernier pas de programme et celui de la dernière mémoire utilisable. A la mise en marche cette partition est 479.59 pour la TI-59, 239.29 pour la TI-58 et la TI-58C. La TI-58C conserve la partition indiquée même si celle-ci est arrêtée et remise en marche, ou si les batteries sont retirées un court instant.

Pour changer la partition mémoire, introduisez le nombre de groupes de 10 mémoires désirées (0 à 10 pour la TI-59; 0 à 6 pour la TI-58 et la TI-58C et appuyez sur **2nd** **Op 17**. La nouvelle partition s'affiche comme précédemment. Avant d'effectuer une partition, annulez la virgule fixe, la notation scientifique ou la notation ingénieur. Voir le chapitre **Capacité de stockage et partition** à la section **Programmation**.

Indicateur d'erreur Op 18-19

Les opérations 18 et 19 servent à déceler une condition d'erreur rencontrée dans le déroulement d'un programme. Placée dans un programme **2nd** **Op 18** lève le drapeau 7 si aucune condition d'erreur ne s'est produite. **2nd** **Op 19** lève le drapeau 7 si une condition d'erreur s'est produite. Le drapeau 7 peut ensuite être testé pour permettre de prendre les décisions qui s'imposent suivant le résultat du test. Voir le chapitre concernant les drapeaux page V-65.

Incréméntation - Décréméntation de mémoires - Op 20-29/30-39.

La fonction Op vous permet également d'incrémenter ou de décrémenter de 1 le contenu d'une mémoire 0 à 9.

Pour incrémenter la mémoire n de 1, appuyez sur **2nd** **Op 2n** où n est le numéro d'une des mémoires comprises entre 0 et 9.

Pour décrémenter la mémoire n de 1, appuyez sur **2nd** **Op 3n** où n est le numéro d'une des mémoires comprises entre 0 et 9.

A chaque fois qu'une telle séquence est composée au clavier ou rencontrée dans un programme le contenu de la mémoire 0n est modifié d'une unité.

Exemple : **2nd** **Op 34** soustrait 1 au contenu de la mémoire 04.

DETECTION DE PRESENCE DE L'IMPRIMANTE — Op 40 (TI-58C SEULEMENT).

Si la TI-58C est connectée sur le berceau imprimant PC-100A, PC-100B ou PC-100C, la séquence **2nd** **Op 40** lèvera le drapeau 7. Cette opération sera sans effet sur le drapeau 7 si la TI-58C n'est pas connectée.



CONVERSIONS ET STATISTIQUES

Certaines séquences de calcul d'usage courant ont été programmées dans la calculatrice pour fournir une puissance de calcul maximum tout en minimisant le nombre de touches nécessaires pour les mettre en œuvre.

Conversions.

Votre calculatrice peut effectuer la conversion entre les systèmes de coordonnées polaires et cartésiennes. Elle peut également convertir un angle exprimé en degrés, minutes, secondes en degrés décimaux et vice-versa.

Conversions sexagésimales.

[2nd] [DMS] - DEGRES, MINUTES, SECONDES EN DEGRES DECIMAUX. - Convertit un angle mesuré en degrés, minutes, secondes en degrés décimaux. **[INV] [2nd] [DMS]** exécute la conversion inverse. Minutes et secondes sont exprimées par deux nombres de deux chiffres. Cette conversion opère uniquement sur la valeur affichée.

Le format d'introduction pour les degrés, minutes, secondes est: DD.MMSSsss, le point décimal séparant les degrés des minutes. Une conversion peut dans certains cas, du fait de l'arrondi, vous restituer un nombre de minutes (ou de secondes) égal à 60. Il convient alors d'augmenter le nombre de degrés (ou de minutes) d'une unité.

Appuyer	Affichage	Commentaires
47.131272 [2nd] [DMS]	47.2202	DD.dddd
[INV] [2nd] [DMS]	47.131272	DD. MMSSsss

DD représente les degrés et dd les fractions de degrés. MM représente les minutes et SSsss les secondes et fractions de secondes. Assurez-vous que deux chiffres sont introduits pour les minutes et les secondes. Par exemple : 5° 4' 30" doit être introduit de la façon suivante 5.0403 pour être correctement interprété.

Cette conversion peut être utilisé également pour les calculs horaires.

CONVERSION DE COORDONNEES POLAIRES EN COORDONNEES CARTESIENNES.

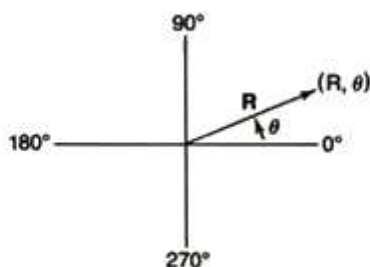
[x↔t] - x ECHANGE t - Echange le nombre x contenu dans le registre d'affichage avec le nombre t contenu dans le registre T. Cette touche est utilisée pour introduire certaines données dans les conversions de coordonnées, les calculs statistiques et pour certains tests en programmation.

Le registre T est indépendant de l'ensemble des mémoires. Il est accessible par la touche **[x↔t]** qui place le contenu du registre d'affichage dans le registre T et simultanément place le contenu du registre T dans le registre d'affichage.

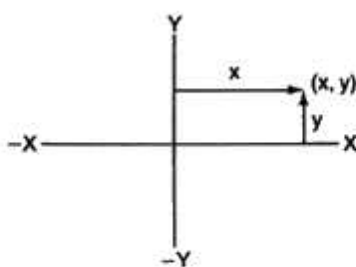
[2nd] [P-R] - POLAIRES/CARTESIENNES - Convertit les coordonnées polaires en coordonnées cartésiennes; **[INV] [2nd] [P-R]** effectue la conversion inverse de coordonnées cartésiennes en coordonnées polaires. Cette opération utilise 4 niveaux d'opérations en attente et un niveau de sous-programme.



Les coordonnées polaires définissent un point par le couple «R» rayon et « θ » l'angle mesuré à partir de l'axe des x. Ce même point peut être défini par des coordonnées cartésiennes utilisant deux vecteurs «x» et «y» respectivement parallèles aux axes des x et des y.



Coordonnées Polaires



Coordonnées cartésiennes

Conversion de coordonnées.

Polaires en cartésiennes

- Introduire «R»
- Appuyer sur $\boxed{\times \div}$
- Introduire « θ ».
- Appuyer sur $\boxed{2nd} \boxed{P \rightarrow R}$ pour afficher «y».
- Appuyer sur $\boxed{\times \div}$ pour afficher «x»

Cartésiennes en polaires

- Introduire «x»
- Appuyer sur $\boxed{\times \div}$
- Introduire «y»
- Appuyer sur $\boxed{INV} \boxed{2nd} \boxed{P \rightarrow R}$ pour afficher « θ »
- Appuyer sur $\boxed{\times \div}$ pour afficher «R»

Assurez-vous que le mode angulaire sélectionné est correct

Suivant le mode angulaire choisi, vous obtenez la valeur de θ dans les limites suivantes :

$$\left. \begin{array}{l} -90^\circ \\ -\pi/2 \text{ rad} \\ -100 \text{ grad} \end{array} \right\} \leq \theta < \left\{ \begin{array}{l} 270^\circ \\ 3\pi/2 \text{ rad} \\ 300 \text{ grad} \end{array} \right.$$

Ceci signifie que les angles du quatrième quadrant sont négatifs.

Le mode angulaire influe tant sur l'interprétation de l'introduction que sur la restitution de la valeur de l'angle.

Exemple : Convertir en coordonnées cartésiennes :

$R = 5, \theta = 30^\circ$.

Sélectionner le mode degrés

Appuyer	Affichage	Commentaires
5 $\boxed{\times \div}$	0.	Introduit «R»
30 $\boxed{2nd} \boxed{P \rightarrow R}$	2.5	Introduit « θ », affiche «y»
$\boxed{\times \div}$	4.330127019	Affiche «x».



Exemple: Convertire en coordonnées polaires (radians) :

$$x = 3, y = 4$$

Appuyer	Affichage	Commentaires
CLR 2nd Rd	0	Selectionne le mode radians
3 x:t	0.	Stocke "x"
4 INV 2nd F-3	0.927295218	Introduit "y", affiche "θ" en radians
x:t	5.	Affiche "R"

Les deux conversions suivantes $P \rightarrow R$ et $D.MS \rightarrow DD.dd$, utilisent 1 niveau de sous programme et 4 niveaux d'opérations en attente.

La conversion polaires cartésiennes est très utile pour les calculs relatifs aux vecteurs.

STATISTIQUES

Ils est souvent utile de pouvoir déterminer un nombre qui soit représentatif d'un ensemble d'éléments indépendant les uns des autres. Ces variables peuvent graphiquement être représentées sur une système d'axes x et y . L'ensemble des points peut ensuite être analysé en déterminant la moyenne, la variance, l'écart type ainsi que la droite de tendance (régression linéaire). Partant de là, il est également possible de faire une extrapolation sur cette droite et d'obtenir le coefficient de corrélation entre l'ensemble d'éléments et la droite qui est censée les représenter.

INTRODUCTION DES DONNÉES

2nd **Pgm** **1** **SBR** **CLR** - initialise la calculatrice en effaçant les mémoires 01 à 06 ainsi que le registre T qui sont utilisés par les fonctions statistiques. Bien sûr, le module de la bibliothèque de base doit être dans la calculatrice pour effectuer cette séquence. Sinon, il vous faut effacer ces mémoires 01 à 06 et le registre T manuellement.

x:t – x EXCHANGE t – Exchange le contenu t du registre T avec le contenu x du registre d'affichage. Dans les calculs statistiques, cette touche permet d'introduire la variable x dans la calculatrice.

2nd **$\Sigma+$** – SOMME STATISTIQUE – Introduit la variable x du couple (x_i, y_i) .

INV **2nd** **$\Sigma+$** Permet d'annuler l'introduction d'un point (couple de valeurs).

Pour introduire un ensemble de valeurs, à une seule variable (x_i) utilisez seulement **2nd** **$\Sigma+$** ; l'annulation d'une valeur s'effectue également par **INV** **2nd** **$\Sigma+$** .

Après chaque introduction (ou annulation) le nombre de couples (ou de valeurs) introduits apparait sur l'affichage.



Les statistiques à deux variables utilisent pour introduire les données correspondantes à chaque point $(x_i, y_i$ pour $i = 1, 2, 3, \dots, N$) la séquence de touches suivante.

x $\boxed{x \div t}$ y $\boxed{2nd}$ $\boxed{\Sigma+}$

i est affiché après chaque couple de valeurs introduit. Pour annuler un point erroné, il faut nécessairement réintroduire le couple de valeurs à annuler x et y en appuyant sur \boxed{INV} juste avant $\boxed{2nd}$ $\boxed{\Sigma+}$. Le nombre de couples introduits se trouve automatiquement diminué d'une unité.

Au fur et à mesure des introductions les paramètres sont ventilés et cumulés dans les mémoires 01 à 06.

Mémoires	Contenu
01	Σy } variable dépendante
02	Σy^2 }
03	N
04	Σx } variable indépendante
05	Σx^2 }
06	Σxy

Si ces paramètres sont précalculés il est possible de les introduire directement dans les mémoires correspondantes avant de démarrer l'analyse statistique.

La calculatrice cumule les différents paramètres de chaque couple introduit, par conséquent, si les mémoires 01 à 06 et le registre T n'étaient pas correctement effacées au préalable, les résultats seraient erronés, c'est pour cette raison qu'il faut initialiser par la séquence $\boxed{2nd}$ \boxed{Pg} $\boxed{1}$ \boxed{SBR} \boxed{CLR} .

MOYENNE, VARIANCE ET ECART TYPE

Après avoir introduit les différents couples (ou après l'un d'eux pour autant que plus de deux couples aient été introduits), il est possible d'obtenir la moyenne, la variance et l'écart type de chacune des deux variables x ou y .

$\boxed{2nd}$ $\boxed{\bar{x}}$ – Calcule et affiche la moyenne des différentes valeurs de y . $\boxed{x \div t}$ affiche ensuite la moyenne des différentes valeurs de x .

\boxed{INV} $\boxed{2nd}$ $\boxed{\sigma}$ – Calcule et affiche l'écart type des différentes valeurs de y . $\boxed{x \div t}$ affiche ensuite l'écart type des différentes valeurs de x .

$\boxed{2nd}$ $\boxed{0p}$ $\boxed{11}$ – Calcule et affiche la variance des différentes valeurs de y . $\boxed{x \div t}$ affiche ensuite la variance des différentes valeurs de x .



La calculatrice utilise les équations suivantes :

$$\text{Moyenne des } x = \bar{x} = \frac{\sum x}{N} \qquad \text{Moyenne des } y = \bar{y} = \frac{\sum y}{N}$$

dans lequel N est le nombre total de couples introduits.

$$\text{Ecart type des } x = \sigma_x = \left[\frac{\sum x^2 - \frac{(\sum x)^2}{N}}{N-1} \right]^{1/2}$$

$$\text{Ecart type des } y = \sigma_y = \left[\frac{\sum y^2 - \frac{(\sum y)^2}{N}}{N-1} \right]^{1/2}$$

$$\text{Variance des } x = \sigma_x^2 = \frac{\sum x^2}{N} - \bar{x}^2$$

$$\text{Variance des } y = \sigma_y^2 = \frac{\sum y^2}{N} - \bar{y}^2$$

Pour plus de commodité, l'option a été prise de sélectionner les deux pondérations N et N-1 pour le calcul des écarts types et des variances. Un résultat pondéré par N est utilisé généralement pour représenter une population, et un échantillonnage est généralement pondéré par N-1.

La variance est calculée avec une pondération N alors que l'écart type est pondéré par N-1. La variance étant le carré de l'écart type par définition, vous obtiendrez la variance pondérée par N-1 en appuyant sur $\text{INV } 2^{\text{nd}} \text{ X } \text{x}^2$ et $\text{x}^2 \text{ x}^2$, l'écart type pondéré à N en appuyant sur $2^{\text{nd}} \text{ OP } 11 \sqrt{\text{x}}$ et $\text{x}^2 \sqrt{\text{x}}$. Le tableau cidessous vous donne les différentes séquences de touches:

Séquences de Touches

Fonctions	Ponderation	Y	X
Moyenne		$2^{\text{nd}} \bar{x}$	$\text{x}^2 \text{t}$
Ecart type	N	$2^{\text{nd}} \text{ OP } 11 \sqrt{\text{x}}$	$\text{x}^2 \text{t} \sqrt{\text{x}}$
Variance	N	$2^{\text{nd}} \text{ OP } 11$	$\text{x}^2 \text{t}$
Ecart type	N-1	$\text{INV } 2^{\text{nd}} \bar{x}$	$\text{x}^2 \text{t}$
Variance	N-1	$\text{INV } 2^{\text{nd}} \bar{x} \text{ x}^2$	$\text{x}^2 \text{t} \text{ x}^2$



Pour les statistiques à une seule variable, vous n'avez pas besoin d'utiliser la touche Σx^2 puisqu'elle n'est utile que pour introduire la seconde variable. Les mémoires 01 à 06 et le registre T restent néanmoins tous utilisés.

Exemple: Analyser les données suivantes : 96, 81, 87, 70, 93, 77.

Appuyer	Affichage	Commentaires
Σ 2nd Σ 1 SBR CLR	0	Initialisation
96 2nd Σ +	1.	1ère introduction
81 2nd Σ +	2.	2ème introduction
97 2nd Σ +	3.	3ème introduction (incorrecte)
97 INV 2nd Σ +	2.	Annule la 3ème introduction
87 2nd Σ +	3.	Corrige la 3ème introduction
70 2nd Σ +	4.	4ème introduction
93 2nd Σ +	5.	5ème introduction
77 2nd Σ +	6.	6ème introduction
INV 2nd \bar{x}	9.879271228	Ecart type
2nd \bar{x}	84.	Moyenne
2nd σ^2 11	81.33333333	Variance
RCL 01	504.	Somme des données

Remarquez que l'écart type peut être calculé en premier, bien qu'il fasse appel au calcul de la moyenne.

Exemple: Suite à une commande, vous avez découpé des tubes en tronçons de 100 cm. de long. Un test de qualité est nécessaire pour répondre à l'uniformité requise $6g/cm \pm 0,01$. 6 échantillons ont été prélevés.

Echantillon	1	2	3	4	5	6
Longeur (cm)	101.3	103.7	98.6	99.9	97.2	100.1
Poids (g)	609	626	586	594	579	605

Quel est le poids moyen des échantillons prélevés ? Quel est la précision de la machine à découper ? Quelle est l'uniformité des échantillons ?



Appuyer	Affichage	Commentaires
2nd Fig 1 SBR CLR	0	Initialisation
101.3 x\divt	0.	Introduit x_1
609 2nd Σ+	1.	Introduit y_1
103.7 x\divt	102.3	Introduit x_2
626 2nd Σ+	2.	Introduit y_2
98.6 x\divt	104.7	Introduit x_3
586 2nd Σ+	3.	Introduit y_3
99.9 x\divt	99.6	Introduit x_4
594 2nd Σ+	4.	Introduit y_4
97.2 x\divt	100.9	Introduit x_5
579 2nd Σ+	5.	Introduit y_5
100.1 x\divt	98.2	Introduit x_6
605 2nd Σ+	6.	Introduit y_6
2nd x	599.8333333	Moyenne des y (poids)
\div x\divt	100.1333333	Moyenne des x (longueurs)
=	5.990346205	Uniformité moyenne (g/cm)
INV 2nd x	17.05774507	Ecart type des poids
x\divt	2.240238083	Ecart type des longueurs

Le poids moyen des échantillons est environ de 599.8g. La machine coupe à une longueur moyenne de 100.1 cm. L'uniformité moyenne est supérieure à 5.99g/cm, donc acceptable. En plus l'écart type des poids et des longueurs est respectivement de 17g et 2.24 cm par rapport à la moyenne.

REGRESSION LINEAIRE

2nd **0p** **12** – Calcule et affiche l'ordonnée du point d'intersection de la droite de régression avec l'axe des y (valeur de y pour $x = 0$); en appuyant ensuite sur **x \div t** vous obtenez sur l'affichage la valeur de la pente de cette droite. Pour le cas particulier d'une droite parallèle à l'axe des y, l'opération sera considérée comme invalide par la calculatrice (intersection non définie, pente infinie). **2nd** **0p** **13** qui calcule le coefficient de corrélation donne un résultat équivalent dans le cas d'une droite parallèle à l'axe des x ou à l'axe des y.

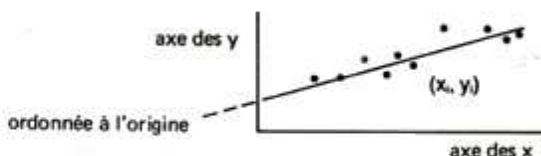
2nd **0p** **13** – Calcule et affiche le coefficient de corrélation des différents points introduits par rapport à la droite de régression. La valeur affichée est comprise entre +1 et -1. Une corrélation parfaite donnerait comme résultat ± 1 . Si la pente de la droite est 0 ou l'infini, l'affichage clignote; cette condition peut être identifiée dans un programme en utilisant l'une des opérations **2nd** **0p** **18** ou **19** et le drapeau 7.

2nd **0p** **14** – Calcule et affiche la valeur y' d'un point de la droite de régression en fonction d'une valeur de x introduite au clavier. Si la droite de régression est parallèle à l'axe des y (pente infinie), la valeur de y' est indéterminée et l'affichage clignote.

2nd **0p** **15** – Calcule et affiche la valeur x' d'un point de la droite de régression en fonction d'une valeur de y introduite au clavier. Si la pente de la droite est nulle, la valeur de x' est indéterminée et l'affichage clignote.



Le mode d'introduction des données pour une régression linéaire est le même que celui utilisé pour les calculs des moyennes, variances et écarts types. En pratique, dès qu'un groupe de données est introduit, toutes les fonctions statistiques peuvent être utilisées pour analyser ces données. La régression linéaire vous permet d'établir une relation entre les différentes valeurs. La droite de régression est définie par la méthode des moindres carrés.



La droite est définie par $y = mx + b$ avec m représentant la pente de la droite et b l'intersection de la droite avec l'axe des y . (Ordonnée à l'origine)

Du fait que les points sont rarement alignés, vous avez le moyen de déterminer la dispersion des points introduits par rapport à la droite de régression. Cette mesure est appelée coefficient de corrélation, et peut être calculé à partir des coefficients de la droite de régression et des données.

La pente et l'ordonnée à l'origine sont calculées d'après les formules :

$$\text{pente} = m = \frac{\sum xy - \frac{\sum x \sum y}{N}}{\sum x^2 - \frac{(\sum x)^2}{N}}$$

$$\text{ordonnée à l'origine} = b = \frac{\sum y - m \sum x}{N}$$

$$\text{Coefficient de Corrélation} = R = \frac{m \sigma_x}{\sigma_y}$$

Une fois la droite de régression définie, il est possible pour une valeur de x (ou de y) déterminée, de demander la valeur correspondante y' (ou x') sur la droite de régression.



Exemple : Une compagnie d'assurances sur la vie a remarqué que le volume de ses ventes variait suivant le nombre de ses démarcheurs de la façon suivante:

Nombre de démarcheurs	7	12	3	5	11	8
Ventes en milliers de Frs. par mois.	99	152	81	98	151	112

Combien de démarcheurs faut-il pour parvenir à un volume de vente de 100 000 Frs. par mois? Quel serait le volume des ventes avec 15 démarcheurs?

Appuyer	Affichage	Commentaires
2nd Fix 1 SBR CLR	0	Initialise
7 x^{-t}	0.	Première valeur de x
99 2nd Σ+	1.	Premier couple de valeurs
12 x^{-t}	8.	Deuxième x
152 2nd Σ+	2.	Deuxième couple
3 x^{-t}	13.	etc.
81 2nd Σ+	3.	
5 x^{-t}	4.	
98 2nd Σ+	4.	
22 x^{-t}	6.	Introduction incorrecte
151 2nd Σ+	5.	
22 x^{-t}	23.	Correction de la mauvaise introduction
151 INV 2nd Σ+	4.	
11 x^{-t}	21.	
151 2nd Σ+	5.	
8 x^{-t}	12.	
112 2nd Σ+	6.	
200 2nd Op 15	17.81578947	Nombre de démarcheurs nécessaire pour un volume de vente de 200 000 F/mois
15 2nd Op 14	176.5561798	Volume de vente en milliers de F pour 15 démarcheurs
2nd Op 12	51.66853933	Ordonnée à l'origine
x^{-t}	8.325842697	Pente de la droite

La pente et l'ordonnée à l'origine peuvent être calculées dès que les points ont été introduits. Ici la pente représente le volume de vente en milliers de Frs par démarcheur et l'ordonnée à l'origine, le volume des ventes sans démarcheur.

Il est possible d'obtenir des régressions non linéaires en appliquant à l'une des variables une fonction mathématique du type logarithme, exponentielle, racine ou puissance ou encore leurs inverses.



Avant d'introduire les couples de données, vous devez déterminer le type de courbe correspondant le mieux à l'analyse que vous désirez effectuer.

Exemple : Une ville publie les résultats de ses recensements. Déterminer la population en 1980 et en combien de temps celle-ci atteindra 50 000 habitants.

Année	1930	1940	1950	1960	1970
Population	3221	5361	9212	15410	27612

Dans cet exemple la population augmente exponentiellement suivant une équation de la forme $y = ae^{bx}$. Prenons le logarithme des deux membres de l'expression : $\ln y = \ln a + bx$. De cette façon en introduisant chaque couple en x et en $\ln y$, nous pouvons étudier l'analyse de la tendance. (Echelle semi-logarithmique).

Appuyer	Affichage
2nd $\frac{\square}{\square}$ 1 SBR CLR	0.
1930 $\ln x$	0.
3221 $\ln x$ 2nd Σ^+	1.
1940 $\ln x$	1931.
5361 $\ln x$ 2nd Σ^+	2.
1950 $\ln x$	1941.
9212 $\ln x$ 2nd Σ^+	3.
1960 $\ln x$	1951.
15410 $\ln x$ 2nd Σ^+	4.
1970 $\ln x$	1961.
27612 $\ln x$ 2nd Σ^+	5.
1980 2nd $\frac{\square}{\square}$ 14 INV $\ln x$	46081.80979
50000 $\ln x$ 2nd $\frac{\square}{\square}$ 15	1981.524472

En 1980 la population sera d'environ 46 080 personnes et la ville aura 50 000 résidents en 1981.

ANALYSE DE TENDANCE

Pour des couples de données répartis à intervalles réguliers tels qu'années, mois, jours, la calculatrice peut automatiquement incrémenter la valeur de x d'une unité sans avoir à introduire pour chaque couple la nouvelle valeur de x . Pour cela il suffit d'introduire la première valeur de x dans le registre T, l'incrémementation de 1 en 1 se fera au fur et à mesure de l'introduction du second paramètre sur les touches $\frac{\square}{\square}$ 2nd Σ^+ : x_1 $\frac{\square}{\square}$ $\ln x$ $\frac{\square}{\square}$ Σ^+ , y_1 $\frac{\square}{\square}$ 2nd Σ^+ , y_2 $\frac{\square}{\square}$ Σ^+ , y_3 $\frac{\square}{\square}$ 2nd Σ^+ , etc.

Pour supprimer une introduction incorrecte, faire $\frac{\square}{\square}$ $\frac{\square}{\square}$ 1 $\frac{\square}{\square}$ suivi de la valeur de y invalide.



Exemple : Une société de service présente l'historique de rentabilité suivant, après la mise en service d'un ordinateur :

année	1962	1963	1964	1965-1970	1971	1972	1973	1974
rentabilité en millions	-2.1	-0.3	0.8	inactive	2.9	2.8	3.6	4.0

Quelle est la rentabilité prévisible en 1980 et à quel moment sera atteint la barre des 10 millions de Frs.

Appuyer	Affichage	Commentaires
2nd Pgn 1 SBR CLR	0.	Initialisation
1962 x:t	0.	Initialise x
2.1 +/- 2nd Σ+	1.	Perte en 1962
.3 +/- 2nd Σ+	2.	Perte en 1963
.8 2nd Σ+	3.	Gain en 1964
1971 x:t	1965.	Réinitialise x
2.9 2nd Σ+	4.	Gain en 1971
2.8 2nd Σ+	5.	Gain en 1972
3.6 2nd Σ+	6.	Gain en 1973
4 2nd Σ+	7.	Gain en 1974
1980 2nd 14	6.52181966	
10 2nd 15	1988.297788	

En 1980 la compagnie peut espérer un profit de 6.5 millions de Frs et le passage aux 10 millions de Frs devrait se faire en 1988.

Pour chacun des calculs statistiques, vous pouvez utiliser l'ensemble des fonctions statistiques que vous avez vues dans les précédents exemples : moyenne, variance, écart-type, pente et ordonnée à l'origine.

Statistiques en cours de calcul

Les opérations statistiques peuvent être utilisées au cours d'un calcul complexe. Vous pouvez avoir jusqu'à 4 opérations en attente et simultanément utiliser les calculs statistiques. (Les statistiques utilisent 4 niveaux d'opérations en attente). Par exemple, certaines compagnies d'assurances utilisent pour calculer leurs frais généraux la formule $3 + 2 \times 1.2^{(4+N)}$ où N est le nombre de démarcheurs pour un volume de vente de 200 000 F. Il suffit d'introduire l'expression $3 + 2 \times 1.2^{(4+N)}$ puis comme nous l'avons fait page V-38 de calculer x' pour y = 200 et en fin d'appuyer sur la touche $\boxed{=}$ pour obtenir le résultat. Remarquez que quatre opérations étaient en attente pendant l'exécution des calculs statistiques.

Les calculs statistiques utilisent également un niveau de sous-programme. Nous en reparlerons dans le chapitre concernant les sous programmes.



LA PROGRAMMATION

Programmer votre calculatrice

Pour résoudre un problème au clavier, vous déterminez la séquence d'opérations et de fonctions nécessaires à l'obtention du résultat puis vous composez touche après touche représente une **instruction de programme**, calculatrice. La programmation, ce n'est rien d'autre que d'introduire une séquence de touches en mode programmation et de demander à la calculatrice de s'en souvenir. La succession des touches (instructions) est stockée en **mémoire programme** ; chaque enfoncement de touches représente une **instruction de programme** l'ensemble constituant le **programme**. Lorsque les instructions du programme sont exécutées, elles produisent le même effet et fournissent le même résultat que lorsque vous effectuez le calcul au clavier. Tant qu'il est stocké, ce programme peut être utilisé autant de fois que vous le désirez avec des variables différentes sans avoir à répéter à chaque fois la même séquence de touches. Non seulement vous gagnez du temps mais vous diminuez également considérablement les risques d'erreurs dus par exemple à l'utilisation erronée d'une touche.

Le programme reste en mémoire programme tant qu'il n'est pas remplacé par un autre programme ou qu'il n'est pas effacé en appuyant sur [2nd] [DP] ou en éteignant la calculatrice (TI-58 et TI-59 uniquement). Entre temps vous pouvez utiliser votre programme autant de fois que vous le désirez. Par exemple si pendant que vous effectuez une série d'opérations manuelles, vous avez besoin d'un résultat provenant de l'exécution d'un programme, il vous suffit d'exécuter celui-ci puis de continuer vos calculs avec le résultat fourni par le programme.

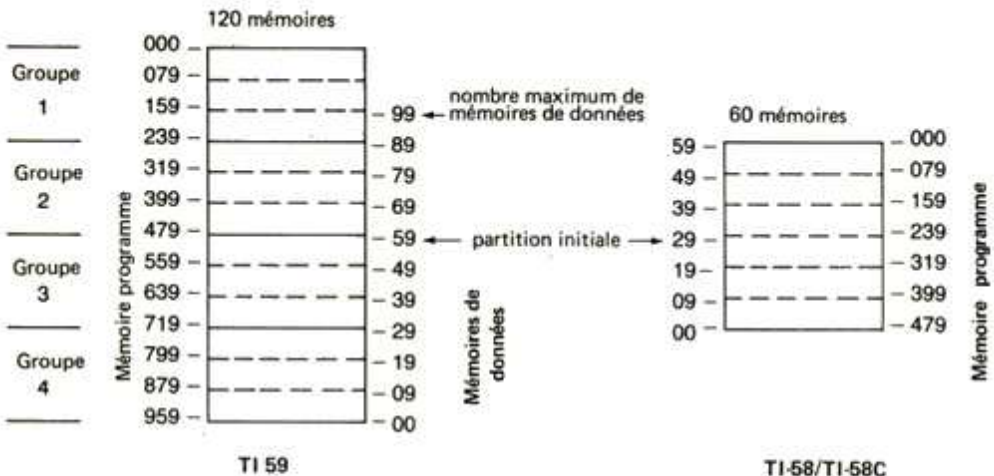
Cette calculatrice possède le système de programmation le plus perfectionné qui soit et également le plus simple à utiliser. Une bonne compréhension des structures de la calculatrice vous permettra de disposer d'un outil à résoudre les problèmes d'une puissance comparable à celle d'un mini-ordinateur.

La **mémoire permanente** de la TI-58C conserve vos instructions en mémoire programme même si la calculatrice est mise sur arrêt ou si le bloc accumulateur est enlevé (ou déchargé) pendant un court instant. Les données en mémoire, la partition et la décimalisation sont également sauvegardées. Si vous interrompez l'exécution de votre programme en mettant sur «Arrêt» votre TI-58C, les opérations en attente, position des drapeaux, adresses des sous-programmes, contenu du registre T et mode angulaire sont effacés. **Toujours arrêter l'exécution d'un programme avant de mettre votre TI-58C sur «Arrêt».**



Capacité de stockage et partition

Il y a dans votre calculatrice une zone affectée aux mémoires de stockage. Cette zone contient à la fois les mémoires de données et la mémoire programme.



ZONE AFFECTEE AUX MEMOIRES

Dans ce qui suit, les informations inscrites entre parenthèses concernent la TI-58 et la TI-58C.

Il y a 120 (60) mémoires disponibles dans la calculatrice. A la mise en marche, il y a une répartition égale entre les mémoires de données et les mémoires programme, 60 (30). Chaque mémoire de données peut contenir l'équivalent de huit instructions de programme. Ainsi vous pouvez avoir $8 \times 60 = 480$ ($8 \times 30 = 240$) instructions de programme avec 60 (30) mémoires de données.

Vous pouvez changer la répartition de cette zone de mémoires suivant vos besoins par groupe de 10 mémoires. Par exemple vous pouvez utiliser les 120 (60) mémoires pour un programme, soit $120 \times 8 = 960$ ($60 \times 8 = 480$) pas de programme et zéro mémoire de données ou vous pouvez les répartir de façon à avoir 100 (60) mémoires de données et $20 \times 8 = 160$ ($0 \times 8 = 0$) pas de programme ou tout autre combinaison. La seule restriction à ce système est le maximum de 100 (60) mémoires pouvant être utilisées, du fait que les codes adresse de mémoire ne peuvent avoir que deux caractères compris entre 00 et 99.

La totalité des 60 registres de la TI-58C sont sauvegardés dans la mémoire permanente, y compris lorsque celle-ci est mise sur «Arrêt».



Pour repartir la zone de mémoire, introduisez le nombre de groupes de mémoires de données que vous désirez, 0 à 10 (0-6), et appuyez sur **[2nd] [0]** 17. Annulez auparavant le format d'affichage virgule fixe, notation scientifique ou notation ingénieur. Pour obtenir 20 mémoires de données appuyez sur **2 [2nd] [0]** 17, l'affichage vous indique :



Ceci montre que vous disposez de 20 mémoires, 00 à 19, disponibles pour stocker des données et 800 (320) pas de programme 000 à 799 (000 à 319), destinés à stocker le programme.

A chaque instant vous pouvez vérifier la partition qui est en machine en appuyant sur **[2nd] [0]** 16, le mode d'affichage est identique à celui ci-dessus.

Fonctions de base de Programmation

La compréhension de ces quelques fonctions de base vous permettra de commencer à programmer votre calculatrice.

[LRN] – (LEARN) – APPRENDRE – Appuyer sur cette touche permet instantanément de mettre la calculatrice en mode d'enregistrement de programme appelé aussi mode programmation. Il vous donne accès à la mémoire programme et vous permet d'enregistrer votre programme. En appuyant sur **[LRN]** à nouveau, vous revenez en mode calcul avec un format d'affichage standard. L'usage de la touche **[LRN]** ne donne pas une valeur affichée, par contre il arrête le clignotement de l'affichage. Il n'est pas possible de passer en mode programmation si un programme protégé se trouve en mémoire programme (T1-59) ou si la partition ne donne pas d'instruction de programme.

[2nd] [CP] (CLEAR PROGRAMME) – EFFACEMENT DU PROGRAMME – Utilisé à partir du clavier, il efface l'ensemble de la mémoire programme, les mémoires d'adresses de retour des sous-programmes, le registre T et remet le pointeur au pas de départ 000. Il supprime également la condition de protection d'un programme. Placé comme instruction dans un programme, efface uniquement le contenu du registre T.

[R/S] (RUN/STOP) – DEMARRAGE/ARRET – Inverse le processus en cours. Si le programme s'exécute appuyer sur **[R/S]** l'arrête ; si le programme est arrêté, **[R/S]** démarre son exécution à partir de l'endroit où se trouve le pointeur. Cette touche permet d'introduire des données ou de faire apparaître des résultats intermédiaires en cours de traitement.



RST —**RESET** —Ramène le pointeur au pas programme 000 (même depuis un programme de la bibliothèque), efface le contenu des mémoires d'adresses de retour des sous programmes et remet les drapeaux en position basse. Egalement employé pour sortir d'un programme de la bibliothèque dans le cas où ce dernier "bouclerait" sur lui-même; dans ce cas, les résultats sont perdus à l'exception de ceux contenus dans les mémoires.

2nd **Pause** —**PAUSE**— Placé dans un programme, permet de maintenir le contenu du registre d'affichage affiché pendant 1/2 seconde environ. Cette instruction peut être placée où vous le désirez et répétée à volonté. Maintenu enfoncée au clavier pendant le déroulement d'un programme, provoque l'affichage du résultat de chaque instruction du programme. Cette touche est inopérante pendant l'exécution d'un programme de la bibliothèque ou d'un programme protégé.

Mode programmation

Dès qu'une séquence de calcul a été établie, passez en mode programmation en appuyant sur **2nd** **CP** **LRN** et introduisez la séquence de touches correspondante en mémoire programme. **2nd** **CP** permet de s'assurer que le programme est introduit depuis le premier pas de programme 000. Lorsque vous passez en mode programmation l'affichage a le format suivant :



Le programme se place à partir du pas 000 et jusqu'à une limite définie par la partition que vous utilisez. Au départ la mémoire programme a été effacée et chaque pas contient un zéro.

Chaque pas peut recevoir une touche de fonction, une adresse ou encore un simple chiffre.

Un code de touche instruction (ou simplement un code de touche) est composé de deux chiffres définissant chaque touche suivant son emplacement sur le clavier. La calculatrice indique normalement 00 comme code de touche lorsque vous introduisez votre programme. La raison est que lorsqu'une instruction est introduite la calculatrice saute immédiatement au pas programme suivant qui est vide. Dans les pages qui suivent vous trouverez un tableau des codes des différentes touches.

Garder la trace de l'endroit exact où vous vous trouvez en mémoire programme est le rôle du pointeur. En mode programmation, cet indicateur avance dans la mémoire programme au fur et à mesure des instructions affichant toujours le numéro du pas suivant celui utilisé.

Après avoir introduit un programme en mémoire programme appuyez sur **LRN** à nouveau pour ramener de contrôle de la calculatrice au clavier (mode calcul) où l'introduction des variables pourra être faite et l'exécution du programme commencer.



INTRODUCTION DE VOTRE PROGRAMME

L'ordre des opérations pour introduire votre programme est le suivant :

1. Au clavier appuyez sur **RST** ou **2nd CP** . L'un et l'autre ramènent le pointeur au pas de départ 000 de la mémoire programme. **2nd CP** efface également la mémoire programme.
2. Appuyez sur **LRN** pour mettre la calculatrice en mode programmation. L'affichage particulier avec 5 chiffres identifie ce mode d'utilisation.
3. Introduisez la totalité de votre programme, instruction après instruction sans oublier les préfixes du type **2nd** ou **INV** . L'affichage vous indique toujours le premier pas disponible suivant celui que vous venez de charger.
4. Assurez-vous que votre programme ne dépasse pas la place maximum disponible dans la mémoire programme. Si le dernier pas est dépassé, la calculatrice revient automatiquement en mode calcul ignorant les instructions excédentaires.
5. Revenez en mode calcul en appuyant à nouveau sur **LRN** .
6. Exécutez un calcul de test avec des valeurs résultat connues, pour vous assurez du bon fonctionnement du programme. Si nécessaire effectuez les corrections.

L'exemple suivant illustre ces différentes étapes.

Exemple : Créer un programme calculant le volume d'un cylindre droit de rayon «r» et de hauteur «h».

Equation utile : $V = \pi r^2 h$

Opérations nécessaires : Introduire «r»
 Démarrer l'exécution du programme
 Arrêter l'exécution pour introduire «h»
 Redémarrer l'exécution jusqu'à l'obtention du résultat sur l'affichage.



Séquence de touches	Affichage	Commentaires
2nd EP	0	place le pointeur au pas 000 et efface la mémoire programme
LRN	000 00	Mise en mode programmation pour commencer l'introduction du programme
x²	001 00	calculer r ² (r ayant été introduit sur l'affichage.)
X	002 00	
2nd π	003 00	Occupe un seul pas programme
X	004 00	πr ² dans le registre d'affichage
R/S	005 00	Arrêt pour l'introduction de h
=	006 00	Calcule le volume
R/S	007 00	Arrêt pour afficher le volume
RST	008 00	Retour au pas 000
LRN	0	Retour au mode calcul

Les deux dernières fonctions ont un rôle particulier. **R/S** arrête l'exécution du programme et affiche la réponse finale, **RST** provoque un retour au pas 000 si bien qu'en introduisant une nouvelle valeur de «r» et en appuyant sur **R/S** la calculatrice va effectuer un nouveau calcul à partir du début du programme (pas 000).

La touche **R/S** a pour effet d'inverser le processus en cours. Pendant le déroulement du programme, si vous appuyez sur **R/S** ou si cette instruction fait partie du programme, l'exécution est interrompue. En appuyant sur **R/S** au clavier, le déroulement du programme redémarre et les calculs reprennent là où ils avaient été interrompus. Le pointeur est maintenu à l'endroit où l'arrêt s'est produit, ce qui permet de reprendre le déroulement sans rien changer aux calculs en cours, pour autant que, pendant l'interruption, vous n'ayiez rien modifié à partir du clavier.

Exécution de votre Programme

Pendant le déroulement du programme, les instructions sont exécutées séquentiellement dans l'ordre où elles avaient été introduites en mémoire programme en commençant à partir de l'adresse courante du pointeur. (Il y a des exceptions dont nous reparlerons plus tard). Pour démarrer le processus, appuyez simplement sur la touche **R/S**. Le pointeur suit rigoureusement les instructions du programme. Si la calculatrice tente de dépasser la limite supérieure de la mémoire programme, le déroulement s'arrête provoquant le clignotement de l'affichage; c'est la raison pour laquelle un programme doit se terminer par une instruction **R/S** ou par **INV** **SBR** ou par un transfert; nous verrons ces deux possibilités plus loin.)

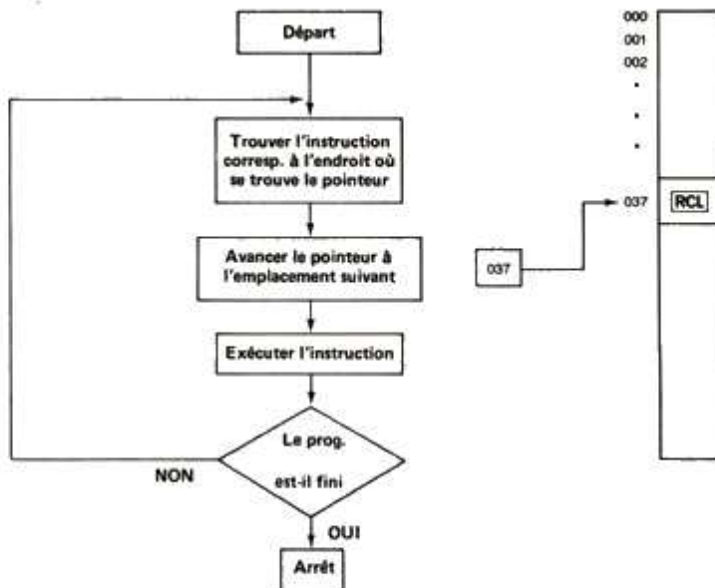
Le programme du volume du cylindre une fois introduit, calculons le volume correspondant à $r = 3$ et $h = 4$



Appuyer	Affichage	Commentaires
RST	0	Positionner le pointeur au pas 000
3	3	Introduit «r»
R/S	Aveugle	Commence l'exécution du programme
	28.27433388	rr? – valeur contenue dans le registre d'affichage au moment de l'arrêt du programme
9	9	Introduit «h»
R/S	Aveugle	Achève l'exécution du programme
	254.4690049	Arrêt du déroulement, affiche «V»

Remarquez que, pendant l'exécution du programme, l'affichage est aveugle. La durée de cet état est fonction du programme utilisé et du type d'alimentation.

Pendant l'exécution d'une séquence le pointeur contrôle le déroulement du processus en s'immobilisant sur chaque instruction jusqu'à ce que celle-ci ait été exécutée.



Organigramme du déplacement du pointeur.

Le mode de déplacement du pointeur sera étudié plus en détail dès que d'autres possibilités de programmation auront été explicitées.



Relecture de votre Programme

SST – (SINGLE-STEP) – **AVANCE PAS A PAS** – Avance le pointeur d'un pas de programme. En mode programmation, cette touche permet d'afficher le pas de programme suivant. En mode calcul, elle permet d'exécuter le programme, instruction par instruction, en affichant à chaque stade la valeur résultat de chaque instruction.

BST – (BACK-STEP) – **RECUK PAS A PAS** – En mode programmation, cette touche permet de faire reculer le pointeur d'un pas et d'afficher le pas programme précédent. Cette touche est inopérente au clavier lors du déroulement du programme.

Ces deux touches sont inopérentes pendant l'exécution d'un programme de la Bibliothèque.

Dans la phase d'introduction ou de vérification d'un programme, deux instructions permettent d'explorer instruction par instruction ce qu'il y a dans le programme. En mode programmation, appuyer sur **SST** fait avancer le pointeur d'un pas et affiche le numéro du pas suivant ainsi que le code de l'instruction correspondant sans affecter l'instruction stockée. Son emploi répétitif permet d'explorer le contenu des différents pas du programme et donc, de «relire» votre programme.

Il en est de même pour la touche **BST** qui fait reculer le pointeur d'un pas à chaque fois qu'elle est utilisée. Elle permet en outre de vérifier une instruction qui vient juste d'être introduite (il faut appuyer sur **SST** ensuite pour revenir au pas suivant). Ces deux touches combinées permettent de vérifier efficacement le programme.

La touche **SST** est utilisable également en mode calcul et provoque l'exécution du programme, instruction par instruction. Dès que vous appuyez sur **SST** la calculatrice se place sur le pas suivant et exécute l'instruction qui s'y trouve contenue, l'affichage donnant la valeur résultant de l'exécution de cette instruction. Dans certains cas, il est nécessaire d'avancer de plusieurs pas pour voir le code affiché changer, ceci est dû au fait que certaines fonctions utilisent plusieurs pas de programme. Par exemple, la séquence $+ RCL 09$ occupe trois pas de programme et le contenu de la mémoire 09 n'apparaîtra qu'après avoir positionné le pointeur au pas contenant l'adresse 09.

En mode programmation l'usage de ces deux touches fait apparaître le code des différentes instructions stockées à chaque pas.

Codes des instructions (Codes des Touches)

Une instruction est représentée par un code à deux chiffres qui représente en général la position de la touche correspondante sur le clavier. Le premier caractère représente le numéro de la ligne (1 à 9 compté de haut en bas) où se trouve la touche, le second caractère désigne la colonne où elle se trouve (numéroté de 1 à 5 de gauche à droite). Par exemple, la touche **STO** est codée 42 car elle se trouve sur la ligne 4, colonne 2. Pour les secondes fonctions il suffit d'ajouter 5 au code de la fonction correspondante, par exemple : **2nd sin** est codée 38 car la touche **sin** est codée 33. Pour la dernière colonne à droite, le code d'une fonction seconde s'obtient toujours en ajoutant 5 au code de la fonction correspondante mais sans changer le numéro de la ligne, par exemple : **2nd E** est codée $15 + 5 = 10$ et non pas 20. Les chiffres 0 à 9 sont représentés par les codes 00 à 09.



Codes des touches et des instructions.

Touches	Codes des touches	Touches	Codes des touches	Touches	Codes des touches	Touches	Codes des touches	Touches	Codes des touches
A	16	F	17	C	18	D	19	E	10
A	11	B	12	C	13	D	14	E	15
		INV	27	log	28	CP	29	CLR	20
2nd	Contracté	INV	22	lnx	23	CE	24	CLR	25
Pgm	36*	P→R	37	sin	38	cos	39	tan	30
LRN	Néant	x↔t	32	x²	33	√x	34	1/x	35
Ins	Néant	CMs	47	Exc	48*	Fix	49*	Ind	40 (ou contracté)
SST	Néant	STO	42*	RCL	43*	SUM	44*	y^x	45
Del	Néant	log	57	inv	58*	int	59	 x 	50
BST	Néant	EE	52	(53)	54	÷	55
Pause	66	x↔t	67*	Ndg	68	0_o	69*	Deg	60
GTO	61*	7	07	8	08	9	09	X	65
 b 	76*	x↔t	77*	Σ+	78	x̄	79	Rad	70
SBR	71*	4	04	5	05	6	06	-	75
St log	86*	log	87*	0 MS	88	π	89	Grad	80
RST	81	1	01	2	02	3	03	+	85
Write	96	0_o	97*	Ad_o	98	Fit	99	List	90
R/S	91	0	00	*	93	+/-	94	=	95

* Touches devant être suivies d'une adresse ou d'une autre instruction pour être complètes.

Note : Le code de l'instruction **Ind** est parfois contracté avec le code de l'instruction sur laquelle elle opère.

Vous trouverez, fourni avec votre calculatrice, un clavier amovible en plastique qui placé sur la calculatrice vous indique les codes des touches.



Codes des touches par ordre numérique

Codes des touches	Touches	Codes des touches	Touches	Codes des touches	Touches
00	0	39	2nd cos	72	STO 2nd Ind
↓	↓	40	2nd Ind	73	RCL 2nd Ind
09	9	42	STO	74	SUM 2nd Ind
10	2nd E	43	RCL	75	-
11	A	44	SUM	76	2nd b
12	B	45	y*	77	2nd x=
13	C	47	2nd CMs	78	2nd Σ+
14	D	48	2nd Tac	79	2nd Σ-
15	E	49	2nd P/d	80	2nd Grad
16	2nd A	50	2nd x	81	RST
17	2nd B	52	EE	83	GTO 2nd Ind
18	2nd C	53	(84	2nd Op 2nd Ind
19	2nd D	54)	85	+
20	2nd CLR	55	±	86	2nd St. Rtg
22	INV	57	2nd Eng	87	2nd W. Rtg
23	Inx	58	2nd Fix	88	2nd D. MS
24	CE	59	2nd Int	89	2nd W
25	CLR	60	2nd Deg	90	2nd List
27	2nd INV	61	GTO	91	R/S
28	2nd log	62	2nd Pgm 2nd Ind	92	INV SBR
29	2nd CP	63	2nd Tac 2nd Ind	93	.
30	2nd Ln	64	2nd P/d 2nd Ind	94	+/-
32	x ^{1/t}	65	X	95	=
33	x ²	66	2nd Pause	96	2nd Write
34	√x	67	2nd x=	97	2nd On/Off
35	1/x	68	2nd Nop	98	2nd Adv
36	2nd Pgm	69	2nd Op	99	2nd Prt
37	2nd P→R	70	2nd R/d		
38	2nd St	71	SBR		

À l'usage, certains de ces codes vous deviendront familiers; ceci évitera d'avoir à consulter cette table trop fréquemment. Le clavier amovible vous permettra également d'identifier rapidement les différents codes.



Stockage de groupes de touches.

La majorité des instructions occupe un pas de programme. Certaines instructions composées de plusieurs touches, occupent également un seul pas. La touche **2nd** est combinée avec l'instruction qui suit pour ne prendre qu'un seul pas comme nous l'avons vu. Les deux chiffres composant l'adresse d'une mémoire de données, le numéro d'un programme de la bibliothèque ou d'une opération spéciale sont combinés de façon à ne prendre qu'un seul pas. Par exemple **RCL 16** ne prendra que deux pas dans la mémoire programme tout comme **2nd Pgm 12** pour lequel **2nd Pgm** utilisera un pas et 12 en second. La calculatrice effectue ces regroupements d'elle-même.

Les adresses de transfert inconditionnel (décrit page V-56) du type **GTO 123** sont stockées en plaçant **GTO** dans un premier pas, 01 dans le suivant et 23 dans le troisième. A mesure que vous composez l'instruction, la calculatrice identifie la séquence de touches et les place correctement dans les différents pas de programme sans que vous ayez à intervenir.

Dans certains cas, lorsque la touche **Ind** est utilisée avec une autre instruction, l'ensemble n'occupe qu'un seul pas de programme et se trouve représenté par un code spécial. Ce nouveau code n'est plus représentatif de la disposition des touches sur le clavier; il utilise les codes laissés disponibles par les touches numériques; ainsi **STO 2nd Ind** occupera un seul pas et sera codé 72. Cette affectation de code est automatique. La liste de ces codes a été donnée précédemment. Les instructions indirectes suivant ce principe sont :

Séquence de touches	Code des touches
2nd Pgm 2nd Ind	62
2nd Exc 2nd Ind	63
2nd Prt 2nd Ind	64
STO 2nd Ind	72
RCL 2nd Ind	73
SUM 2nd Ind	74
GTO 2nd Ind	83
2nd Op 2nd Ind	84

Les instructions indirectes ne suivant pas ce principe de contraction seront identifiées dans le programme par un code 40 correspondant à **2nd Ind** suivi du code de l'instruction sur laquelle il s'applique. Ainsi **2nd x<=1 2nd Ind** sera stocké en plaçant le code 67 (**2nd x<=1**) dans un premier pas, suivi de 40 (**2nd Ind**) dans le pas suivant. L'usage de l'instruction indirecte sera explicité par la suite.

CORRECTION D'UN PROGRAMME

2nd No — (NO OPERATION) — OPERATION BLANCHE — En mode programmation, l'introduction de cette instruction permet d'annuler l'effet d'une instruction ou de ménager un espace entre deux séquences de programme pour le compléter ultérieurement. L'exécution du programme se poursuit sans que rien ne soit modifié. L'usage des touches n'a aucun effet, ni sur l'exécution d'une séquence, ni même sur les introductions de données (sauf lorsqu'elle est utilisée comme étiquette).

2nd Del — (DELETE) — SUPPRESSION D'UNE INSTRUCTION — En mode programmation, supprime l'instruction affichée et déplace toutes les instructions qui suivent d'un pas, de telle sorte que l'emplacement ainsi libéré se trouve immédiatement réutilisé. Le pointeur garde sa position initiale.



2nd **Ins** –(INSERT) – Insertion d'une instruction – En mode programmation, cette instruction décale d'un pas toutes les instructions qui suivent à partir de celle affichée, vers la fin de la mémoire programme. Elle permet de créer un espace disponible pour introduire une instruction supplémentaire.

Lorsque vous analysez un programme situé en mémoire programme vous pouvez :

1. Avancer ou reculer pas à pas en affichant les codes des instructions contenues dans les différents pas.
2. Remplacer une instruction par une autre.
3. Supprimer une instruction et contracter l'ensemble du programme.
4. Créer un espace disponible pour venir insérer une instruction nouvelle.

Ces différentes possibilités vous permettent de corriger ou de modifier un programme avec le minimum d'effort.

Remplacement d'une instruction par une autre

En mode programmation, l'enfoncement d'une touche à un endroit quelconque du programme remplace instantanément l'instruction qui y était stockée. Si dans un programme vous découvrez une instruction erronée, il vous suffit, pour la remplacer, d'appuyer sur la touche correspondant à la bonne instruction pour qu'instantanément l'une remplace l'autre.

Suppression d'une instruction

En mode programmation, une instruction affichée peut être supprimée simplement en appuyant sur **2nd** **Del**. Toutes les instructions qui suivent sont automatiquement déplacées d'un pas vers le haut pour combler l'espace ainsi créé. Un zéro se trouve placé à la place de la dernière instruction. Si besoin est, on peut utiliser cette touche plusieurs fois de suite.

Insertion d'une instruction

En mode programmation, si vous avez à insérer une instruction à un endroit quelconque de votre programme, vous pouvez, en appuyant sur **2nd** **Ins**, créer un espace libre tout en décalant toutes les instructions qui suivent, y compris celle qui était affichée, d'un pas vers le bas. Il suffit alors d'introduire l'instruction nouvelle. Cette opération peut être répétée plusieurs fois de suite pour introduire une série d'instructions. Chaque fois que vous utilisez cette touche vous perdrez la dernière instruction de votre programme si celui-ci occupait la totalité de la mémoire programme disponible.

Il est possible pour ne pas modifier la longueur d'un programme de remplacer une instruction par l'instruction blanche **2nd** **Nil** plutôt que d'utiliser l'effacement; on peut aussi prévoir des instructions blanches permettant de rajouter, à posteriori, des instructions utiles sans avoir à utiliser l'insertion.



Exemple : Introduire l'équation $x^2 + 3x - 2$, vérifier le programme et l'utiliser pour différentes valeurs de x .
 x sera introduit sur l'affichage avant le démarrage du programme.

Appuyer	Affichage	Commentaires
LRN	000 00	Mise en mode programmation
STO	001 00	Stocke x en mémoire 01
1	001 00	
x^2	003 00	
-	004 00	Introduction incorrecte
BST	003 75	Reculé au pas 003
+	004 00	Remplace - par +
4	005 00	Introduction incorrecte
X	006 00	
BST	005 85	Retourne à l'instruction incorrecte
BST	004 04	
3	005 85	Remplace 4 par 3
SST	006 00	Avance au pas 006
X	007 00	
RCL	008 00	
1	008 00	
BST	008 01	X est introduit deux fois de suite
BST	007 43	Reculé au pas 006
BST	006 65	
2nd Del	006 43	Efface X , place RCL de 007 en 006, etc...
SST	007 01	Avance d'un pas pour sauter RCL
SST	008 00	Avance d'un pas pour sauter 1
2	009 00	
=	010 00	
BST	009 95	- a été oublié
BST	008 02	Reculé au pas 008
2nd Inv	008 00	Libère 008 et décale 2 =
-	009 02	Inverse -
SST	010 95	Avance d'un pas pour sauter 2
SST	011 00	Avance d'un pas pour sauter =
R/S	012 00	Arrête le programme, affiche le résultat.
RST	013 00	



Vérifions maintenant que le programme a été correctement introduit en avançant pas à pas. Le programme devrait se présenter de la façon suivante:

Pas et Codes des Touches	Sequence de Touches
000 42	STO
001 01	0 1
002 33	x²
003 85	+
004 03	3
005 65	X
006 43	RCL
007 01	0 1
008 75	-
009 02	2
010 95	=
011 91	R/S
012 81	RST

Soyez prudent en interprétant les codes ou en modifiant un programme comme vous le montre l'exemple suivant:

Pas et Codes des Touches	Sequence de Touches
.	
.	
.	
019 95	=
020 42	STO
021 12	1 2
022 61	GTO
.	
.	
.	

Si vous effacez l'instruction **STO**, l'adresse de la mémoire devient la touche **B** (code 12). Si vous voulez remplacer l'adresse 12 par 13 en plaçant le pointeur au pas 021 et en introduisant 13 vous obtenez un 1 au pas 021 et le 3 au pas 022 au lieu de 13 au pas 021. Pour effectuer ceci il faut nécessairement réintroduire l'ensemble **STO** 13 depuis le pas 020 ou encore introduire **C** (code 13) au pas 021. Cette astuce consistant à introduire **C** pour créer un code 13 au pas 021 est très employée pour modifier un programme.



REPERAGE DE SEQUENCES DE PROGRAMME

2nd **LBL** – (LABEL) – ÉTIQUETTE – Utilisée uniquement en mode programmation, une étiquette permet de repérer une section d'un programme. Cette touche indique à la calculatrice que l'instruction qui suit doit être interprétée comme un point de repère et non comme une instruction à exécuter.

L'usage des étiquettes est analogue à celui d'onglets dans un livre; elles permettent d'identifier des séquences pouvant être exécutées isolément ou suivant un ordre déterminé. Il y a deux sortes d'étiquettes : les étiquettes des touches utilisateur et les étiquettes ordinaires.

Touches-utilisateur

Les touches situées en haut du clavier **A** à **E** et **2nd** **A'** à **2nd** **E'** sont appelées touches-utilisateur. Repérer un segment de votre programme avec l'une de ces touches vous permet en mode calcul, si vous appuyez sur la touche correspondante, de démarrer instantanément l'exécution du segment de programme correspondant. Vous donnez ainsi à cette touche une fonction particulière correspondant à votre calcul, elle devient une instruction au même titre que celles qui sont sur le clavier. La séquence peut occuper l'ensemble de la mémoire programme ou être très brève comme dans l'exemple précédent. Si cet exemple est toujours dans votre calculatrice, effectuez les modifications indiquées ci-dessous.

Appuyer	Affichage	Commentaires
LRN	0	Retour en mode calcul
RST	0	Retour au pas de départ 000
LRN	000 42	Mise en mode programmation
2nd INS	000 00	Crée deux places vides
2nd INS	000 00	
2nd LBL	001 00	Place 2nd LBL au pas 000
A	002 42	Place A au pas 001

Le programme est maintenant repéré par la touche utilisateur **A** et peut être utilisé en introduisant x sur l'affichage et en appuyant sur **A**.

1 A	2.	Valeur de l'expression pour $x = 1$
1 +/- A	-4.	Pour $x = -1$
6 A	52.	Pour $x = 6$
3.2 EE	3.2 00	
12 A	1.024 25	Pour $x = 3.2 \times 10^{12}$
2nd π +	3.1415927 00	
.03 = A	1.1278386 04	Pour $x = \pi/0.03$

Chaque touche utilisateur peut être affectée à une séquence de programme et être utilisée comme nous venons de le voir. Chaque fois qu'une touche utilisateur est utilisée au clavier ou rencontrée dans le déroulement d'un programme, le pointeur est immédiatement déplacé à l'endroit où se trouve l'étiquette et le déroulement du programme exécuté à partir de cet endroit.



Étiquettes ordinaires.

Presque toutes les touches du clavier (y compris les fonctions secondes) peuvent être utilisées comme étiquettes à l'exception des touches suivantes : **2nd**, **LRN**, **Ms**, **Del**, **SST**, **BST**, **Inv** et des chiffres de 0 à 9. Vous devez éviter également l'emploi de **R/S** comme étiquette dû à son rôle qui est de faire démarrer l'exécution du programme.

Ces étiquettes ont un rôle équivalent aux touches-utilisateur, si ce n'est qu'un segment de programme défini par une de ces étiquettes ne sera pas aussi accessible qu'avec une touche utilisateur. Si par exemple un segment est repéré par l'étiquette **cos**, en appuyant en mode calcul sur **2nd** **cos** vous ne ferez pas démarrer l'exécution du programme à partir de cet endroit, mais simplement vous obtiendrez le cosinus de la valeur affichée. Ces étiquettes sont plus destinées à l'usage des transferts internes qu'à une utilisation directe au clavier.

Appuyez sur **GTO** **2nd** **cos** par exemple place le pointeur au pas suivant immédiatement l'étiquette «cos». **SBR** **2nd** **cos** fera de même, et en plus démarrera l'exécution du programme tout comme le ferait **GTO** **2nd** **cos** **R/S**.

En pratique vous pouvez utiliser autant d'étiquettes que vous désirez (il y en a 72 disponibles) dans un programme. **Aucune étiquette ne peut être utilisée pour repérer plus d'un segment de programme**, mais une étiquette peut être appelée aussi souvent que nécessaire.

Toutes les étiquettes contenues dans un programme peuvent être listées sur l'imprimante avec l'adresse de leur emplacement en mémoire programme. Il suffit pour cela d'appuyer sur **GTO** 0 ou **RST** pour placer le pointeur au pas de départ 000 et d'appuyer ensuite sur **2nd** **0** **08** pour voir la table des étiquettes s'imprimer.

INSTRUCTIONS DE TRANSFERT.

Jusqu'à maintenant, pour l'exécution d'un programme, il fallait se positionner au pas de départ 000 pour démarrer l'exécution du programme, lequel s'arrêterait au premier **R/S** rencontré. Un certain nombre de touches appelées instructions de transfert, permettent de modifier le déroulement séquentiel d'un programme. Ces transferts sont de deux types : les transferts inconditionnels et les transferts conditionnels. Les transferts inconditionnels placent le pointeur immédiatement à l'adresse indiquée; les transferts conditionnels testent une condition particulière et décident, en fonction de la réponse obtenue, si le transfert sera exécuté ou non.

Instructions de transfert inconditionnel **GTO** et **SBR**.

INSTRUCTION "GO TO"

GTO N ou nnn – (Go To) – **ALLER A** – Utilisée dans un programme, **GTO** transfère le déroulement de l'exécution à l'endroit où se trouve l'étiquette N ou à l'adresse nnn. N peut être une touche utilisateur ou une étiquette ordinaire. Au clavier **GTO** place le pointeur à la séquence de programme repérée par l'étiquette N ou par nnn, mais ne déclenche pas le démarrage du programme. Au clavier cette instruction est souvent utilisée pour effectuer une vérification à un endroit déterminé du programme.



Un transfert à un pas de programme précis (nnn) est appelé adressage absolu.

Exemple : Ecrire un programme pour compter de quatre en quatre.

Séquence de touches

2nd CP
 LRN
 2ndLbl SUM
 + 4 =
 2nd Pause
 GTO SUM
 LRN
 GTO SUM

Commentaires

Efface la mémoire programme, et place le pointeur au pas 000
 Mise en mode programmation
 Séquence de programme appelé «SUM»
 Ajoute 4 à chaque résultat
 Affiche le résultat pendant 1/2 seconde.
 Sauter à l'étiquette «SUM» et réexécute la séquence
 Retour au mode calcul
 Positionne le pointeur à l'étiquette «SUM»

Maintenant il suffit d'appuyer sur **R/S** pour voir apparaître 4, puis 8, puis 12, etc... La calculatrice effectue cette séquence jusqu'à ce que vous appuyiez sur **R/S** pour l'arrêter.

Ce programme peut s'écrire de différentes façons. Supposons que chacune d'elles commence au pas 000.

Séquence de touches

2ndLbl
 SUM
 +
 4
 =
 2nd Pause
 GTO
 SUM

Etiquette ordinaire

Séquence de touches

+
 4
 =
 2nd Pause
 GTO
 0
 0
 0

Adressage absolu

Séquence de touches

2ndLbl
 C
 +
 4
 =
 2nd Pause
 GTO *
 C

Touche utilisateur

Séquence de touches

+
 4
 =
 2nd Pause
 RST

Reset

*Note : Si le programme appelé est considéré comme un sous-programme, l'instruction **C** seule est nécessaire et sous-entend l'instruction **SBR**. Attention : **C** est équivalent à **SBR C**, **C** n'est pas équivalent à **GTO C** (**C** ou toute autre touche utilisateur).



La première méthode utilisait une étiquette ordinaire. Vous pouvez remarquer que l'adressage n'utilise pas d'étiquette. Maintenant si cette séquence était déplacée dans la mémoire programme, il serait nécessaire de modifier cette adresse absolue. La touche utilisateur effectue automatiquement le transfert à l'étiquette C. Nous retrouverons plus loin ces principes qui concernent également les sous-programmes.

Une adresse absolue à trois chiffres est regroupée pour être stockée dans deux pas de programme. Ainsi, **GTO 126** est stockée dans trois pas de programme : **GTO 01 26**. Si vous utilisez des adresses inférieures à 100, il suffit d'introduire la partie significative de l'adresse. Appuyez sur **GTO 7**, cette forme contractée sera automatiquement stockée dans trois pas de programme **GTO 00 07** pour autant que l'instruction qui suit soit non numérique. Un minimum de un chiffre doit être introduit sous peine de voir la calculatrice interpréter l'instruction suivante comme un label.

SOUS-PROGRAMMES

SBR N ou **nnn** — (SUBROUTINE) — SOUS-PROGRAMME — Un sous programme est une suite d'instructions qui peuvent être écrites pour définir une opération mathématique ou logique indépendamment du programme principal. Le programme principal, ou un autre sous-programme, peut à tout moment demander l'exécution de cette séquence. Les sous-programmes sont conçus pour être utilisés chaque fois qu'une série d'opérations doit être traitée de façon répétitive. Plutôt que d'écrire la séquence concernée autant de fois qu'elle est rencontrée dans la résolution du problème, elle peut être développée une seule fois et appelée dès qu'elle sera nécessaire. Après exécution d'un sous-programme le déroulement se poursuivra depuis l'instruction qui suit celle ayant servi à appeler le sous-programme.

L'instruction **SBR** est suivie d'une étiquette **N** ou d'une adresse **nnn**. Dans le déroulement du programme l'exécution sera poursuivie à partir de l'endroit où se trouve l'étiquette ou au pas indiqué par l'adresse. Automatiquement l'adresse du pas suivant celui du transfert est stockée dans un des **registres d'adresses de retour des sous-programmes**. L'instruction **INV SBR** qui termine chaque sous-programme ordonnera le transfert au pas dont l'adresse a été stockée précédemment pour poursuivre l'exécution du programme. Si **INV SBR** n'est pas utilisé pour achever un sous-programme il agit comme **R/S**. Les registres des adresses de retour peuvent contenir jusqu'à six adresses permettant de passer d'un sous-programme à un autre. Dès qu'un sous-programme est achevé, l'adresse de retour correspondante est supprimée libérant ainsi le registre qu'elle occupait.

Utilisée au clavier cette instruction permet de démarrer l'exécution d'un programme automatiquement depuis l'endroit spécifié.



Exemple : Calculer $x^2 - 3x - 2$ pour x_1 contenu en R_{01} et pour x_2 contenu en R_{02} ; cumuler les deux résultats en R_{03} .

Vu qu'il va falloir exécuter deux fois la même séquence nous allons l'écrire sous forme de sous programme. Arbitrairement plaçons cette séquence au pas 030 en appuyant sur **GTO 30** **LRN** :

Pas et codes des touches	Séquence de touches	Commentaires
030 76	2nd CE	Séquence repérée par l'étiquette CE
031 24	CE	
032 53	(
033 42	STO	Stocke la valeur de x pour usage ultérieur
034 05	5	
035 33	x²	
036 75	-	
037 03	3	
038 65	X	
039 43	RCL	Rappelle x
040 05	5	
041 75	-	
042 02	2	
043 54)	Evalue l'expression
044 44	SUM	Cumule en R_{03}
045 03	3	
046 92	INV SBR	Retour au programme principal

Remarquez l'emploi des parenthèses pour évaluer l'expression. La touche égal aurait pu être utilisée, mais elle aurait déclenché l'exécution de toutes les opérations en attente tant dans le sous programme que dans le programme principal. Soyez donc prudent et utilisez les parenthèses. Nous pouvons maintenant écrire le programme principal qui permettra d'introduire chaque x et d'afficher le résultat.



Pas et Codes des Touches	Séquence de Touches	Commentaires
000 43	RCL	Rappelle x_1
001 01	1	
002 71	SBR	Exécute le sous-programme CE
003 24	CE	
004 43	RCL	Rappelle x_2
005 02	2	
006 71	SBR	Exécute le sous-programme CE
007 24	CE	
008 43	RCL	Rappelle le résultat
009 03	3	
010 91	R/S	Arrête et affiche le résultat
011 81	RST	Ramène le pointeur au pas 000 et efface les registres des adresses de retour des sous-programmes

Quand le sous-programme est appelé pour la première fois, la calculatrice effectue le transfert à l'étiquette CE (pas 030), exécute le sous-programme et revient au pas 004 continuer le déroulement. Le deuxième appel du sous-programme exécutera un retour au pas 008. Le retour s'effectue toujours au pas qui suit le pas de départ.

A partir d'un sous-programme appelé par le programme principal il est également possible de partir vers un autre sous-programme. Les registres d'adresses de retour mémorisent automatiquement les adresses où chaque sous-programme doit venir se brancher après avoir été exécuté.

Après avoir introduit le programme, appuyez sur **RST** pour revenir au pas 000 et pour effacer les registres d'adresses de retour.

UTILISATION D'UN PROGRAMME DE LA BIBLIOTHEQUE EN TANT QUE SOUS-PROGRAMME

Chaque programme du module d'une bibliothèque peut être utilisé comme un sous-programme. Les programmes d'un module sont, en effet, spécialement prévus et élaborés de façon à pouvoir servir de sous-programmes. Chaque partie est repérée par une étiquette et terminée par **INV** **SBR**. La plupart d'entre eux n'utilise pas l'instruction **R/S**. Voyez le manuel de chaque Bibliothèque pour connaître les étiquettes utilisées.



Nous avons vu comment accéder depuis le clavier à un programme de la bibliothèque. Dans le déroulement d'un programme, vous pouvez appeler l'un d'eux ou utiliser une partie de l'un d'eux comme sous programme. Si la séquence est repérée par une touche utilisateur, utilisez $\boxed{2nd} \boxed{Pgm} \text{ mm, N}$ qui ira traiter dans le programme **mm** la séquence identifiée par **N**. Si la séquence est repérée par une étiquette ordinaire, vous devez utiliser $\boxed{2nd} \boxed{Pgm} \text{ mm} \boxed{SBR} \text{ N}$. Dans les deux cas lorsque l'exécution de la séquence est achevée, le retour se fait automatiquement au pas suivant le pas de départ dans votre programme et l'exécution se poursuit à partir de là.

Mémoire programme	Module de bibliothèque
.	.
.	.
.	.
.	.
x^2	PGM 05
STO	.
06	.
+	.
Pgm	.
05	Etiquette
A	A
+	.
2	.
.	.
.	.
.	.
.	INV SBR
	.
	.
	.
	.

Dans le programme l'instruction $\boxed{2nd} \boxed{Pgm} \text{ mm, N}$ travaille de la même façon que $\boxed{SBR} \text{ N}$ en ce qui concerne la mémorisation dans les registres d'adresses de retour, du pas qui suit le pas de départ. La séquence $\boxed{2nd} \boxed{Pgm} \text{ mm}$ suivie par autre chose que \boxed{SBR} où une touche utilisateur est invalide et peut conduire à des résultats inattendus.



L'usage de programmes de bibliothèque en tant que sous-programmes augmente considérablement les capacités de calcul de votre calculatrice. Cependant certains aspects de programmation deviennent critiques lorsque vous appelez deux programmes indépendants en même temps. Les programmes ne doivent pas interférer l'un avec l'autre, par exemple en utilisant des mémoires ou des drapeaux communs. Un programme d'une bibliothèque peut utiliser également différents sous-programmes, en conséquence assurez-vous qu'au total il n'y aura pas plus de 6 niveaux de sous programme utilisés. En général il n'y a pas de problème si vous ne dépassez pas trois niveaux de sous-programme dans votre propre programme. Un programme d'une bibliothèque peut également utiliser un format d'affichage particulier : virgule fixe, notation scientifique ou notation ingénieur. Les différentes façons d'utiliser des sous-programmes sont résumées ci-dessous.

INSTRUCTIONS D'APPEL DE SOUS PROGRAMME

De	Vers	Séquence de touches
Mémoire programme ou d'un programme bibliothèque vers un autre	Programme bibliothèque	2nd F mm, N (touche utilisateur) ou 2nd F mm SBR N (étiquette ordinaire) ou 2nd F mm SBR nnn 2nd F mm R/S SBR nnn or N 2nd F 00, N (ou nnn) ou 2nd F 00 SBR N ou RST
Mémoire programme	Mémoire programme	
Programme bibliothèque	Mémoire programme	

Transfert conditionnel (Instruction de test)

Ces instructions provoquent un transfert uniquement si une condition particulière est réalisée : comparaison du contenu du registre d'affichage avec celui du registre T, du nombre de boucles effectuées ou encore de l'état d'un drapeau.

COMPARAISON AVEC LE CONTENU DU REGISTRE T

x=t	Echange le contenu du registre d'affichage x avec celui du registre T, t.
2nd x=1 N ou nnn	Demande "si le contenu du registre d'affichage est strictement égal à celui du registre T".
INV 2nd x≠1 N ou nnn	Demande "si les contenus des deux registres sont différents".
2nd x>1 N ou nnn	Demande "si le contenu du registre d'affichage est supérieur ou strictement égal à celui du registre T".
INV 2nd x<1 N ou nnn	Demande "si le contenu du registre d'affichage est plus petit que celui du registre T".

Si la réponse à l'une de ces questions est "oui" le déroulement du programme se poursuit à l'adresse indiquée juste après l'instruction. Si la réponse est "non" le déroulement se poursuit en ignorant l'adresse de transfert.



Ces tests n'effectuent pas les opérations en attente et peuvent ainsi être utilisés à n'importe quel endroit dans un programme.

Exemple :

Pas et code des touches	Séquence de touches
022 43	RCL
023 01	1
024 95	=
025 77	2nd ≥
026 12	B
027 61	GTO
028 00	3
029 03	
030 76	2nd
031 12	B
032 91	R/S

REMARQUE : Les branchements conditionnels ne mémorisent pas l'adresse de départ du transfert comme c'était le cas pour les sous-programmes. Si vous voulez appeler un sous-programme en sortant d'un test, il vous faut placer ce test en début de sous-programme. Ainsi **2nd** **sc=1** **∞** ne mémorisera pas l'adresse de retour mais **SBR** **∞** puis **2nd** **|||** **∞** **2nd** **sc=1** mémorisera l'adresse de retour. Faites attention au fait que le retour au programme principal s'effectue, quelque soit l'option prise par le test, par l'intermédiaire de l'instruction **INV** **SBR** .

Dans cette séquence, le résultat obtenu au pas 024 est testé au pas 025 pour voir si il est plus grand que ou égal au contenu du registre T . Si la réponse est oui, il y a un transfert à l'étiquette B et arrêt du déroulement au pas 032. Si la réponse est non l'étiquette de transfert B est sautée et l'exécution se poursuit avec un transfert au pas 003.

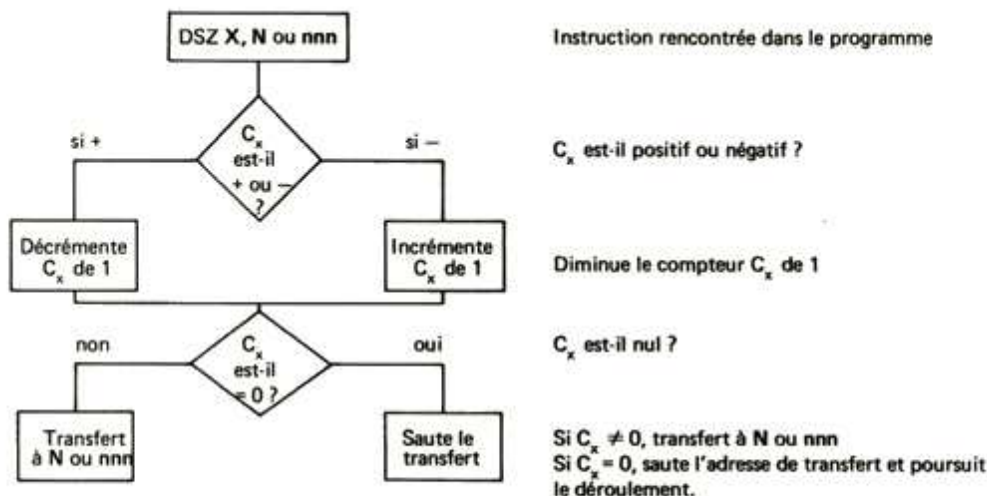
DECREMENT ET SAUT SUR ZERO

2nd **00** **X**, N ou nnn – DECREMENT ET SAUT SUR ZERO – Diminue le contenu d'une mémoire X (0 à 9) d'une unité et procède au transfert à l'étiquette N ou pas nnn tant que le contenu de cette mémoire (C_x) est différent de zéro. L'instruction de transfert est sautée lorsque C_x est nul.

INV **2nd** **00** **x**, N ou nn transfert à l'étiquette N ou au pas nn lorsque $R_x = 0$.



Cette puissante instruction regroupe à la fois un compteur et un test. Si vous voulez effectuer une séquence un nombre y de fois, il vous suffit de mémoriser y dans une mémoire (0 à 9) et de placer l'instruction **DSZ** dans la séquence. Après y passages, la boucle s'arrête et le programme continue. Cette instruction opère de la façon suivante :



Ceci, montre que si cette instruction est placée en début de séquence elle va compter avant l'exécution de la séquence, si elle est placée en fin de séquence l'exécution se fera avant le comptage; de sa position peut dépendre un passage supplémentaire : pour effectuer y passages, si l'instruction DSZ est en fin de séquence il faut au départ que $C_x = y$ et si elle est placée en début de séquence il faut que $C_x = y + 1$.



Exemple: Ecrire le programme calculant $F!$ (factorielle F) $F! = 1 \times 2 \times 3 \times \dots \times F$ avec par définition $0! = 1$
 F par définition doit être un nombre entier.

Pas et Codes des Touches	Sequence de Touches	Commentaires
000 76	2nd 101	
001 15	E	
002 42	STO	Stocke F en M_{00}
003 00	0	
004 29	2nd CP	Efface le registre T
005 67	2nd *-1	Teste si $F = 0$?
006 11	A	Si oui, transfert à A
007 76	2nd 101	
008 12	B	
009 43	RCL	Rappelle F
010 00	0	
011 65	X	
012 97	2nd 812	Décrémente C_{00} de 1
013 00	0	
014 12	B	Si C_{00} est différent de zéro, transfert vers B
015 76	2nd 101	Si C_{00} égale zéro, achève le programme
016 11	A	
017 01	1	
018 95	=	
019 91	R/S	Arrêt et affichage de $F!$

Introduisez F et appuyer sur **E**. F doit être plus petit que 70 sinon l'affichage clignotte car $70! > 9.9999999 \times 10^{99}$.

DRAPEAUX

2nd **514** y – (SET FLAG) – LEVER LE DRAPEAU – Lève le drapeau numéro y (0 à 9). **INV** **2nd** **515** y baisse le drapeau y .

2nd **514** y , N ou nnn – (IF FLAG SET) – LE DRAPEAU EST-IL LEVE ? – Teste si le drapeau y est levé, si oui il y a transfert à l'étiquette N ou à l'adresse nnn .

INV **2nd** **515** y , N ou nnn – (IF FLAG NOT SET) – LE DRAPEAU EST-IL BAISSÉ ? – Teste si le drapeau y est baissé, si oui il y a transfert à l'étiquette N ou à l'adresse nnn .



Il y a 10 drapeaux numérotés de 0 à 9 pouvant être utilisés dans un programme. A l'origine tous les drapeaux sont en position basse. Ils peuvent servir à identifier ce qui s'est produit à un endroit du programme et permettre ainsi de faire un choix ultérieurement. Ils permettent également de diriger l'exécution d'un programme en fonction de l'option choisie. Tous les drapeaux sont baissés en appuyant sur **RST** ou **2nd GP**. En cours de programme **RST** remet également les drapeaux en position basse.

Exemple : établir un programme pour cumuler les nombres introduits en imprimant uniquement ceux qui sont positifs. Afficher le cumul après chaque introduction.

Pas et Codes des Touches	Séquences des Touches	Commentaires
018 76	2nd LBL	
019 11	A	
020 22	INV	
021 86	2nd STO	Baisse le drapeau 3 pour préparer une nouvelle introduction.
022 03	3	
023 77	2nd x=1	Le nombre est-il positif ou nul ?
024 12	B	Si oui, transfère à l'étiquette B
025 86	2nd STO	Si non, lève le drapeau 3
026 03	3	
027 76	2nd LBL	
028 12	B	
029 44	SUM	Cumule les nombres introduits
030 12	1 2	
031 87	2nd IF	Le drapeau 3 est-il levé ?
032 03	3	
033 13	C	Si oui, transfère à l'étiquette C
034 99	2nd PR	Si non, imprime le nombre
035 76	2nd LBL	
036 13	C	
037 43	RCL	
038 12	1 2	
039 91	R/S	

Assurez-vous que la mémoire 12 est vide avant de commencer à introduire une série de nombres. L'introduction d'un nombre se fait en appuyant sur **A**.



DRAPEAUX ET CONDITIONS D'ERREUR

Le drapeau 7 est affecté à l'identification de l'existence d'une condition d'erreur dans un programme. Normalement le déroulement d'un programme continue même si une condition d'erreur est apparue. Si le drapeau 8 est levé par programme ou au clavier, l'exécution du programme s'arrête dès qu'une condition d'erreur se produit.

2nd **0g** 18 lève le drapeau 7 si aucune condition d'erreur n'a été rencontrée.

2nd **0g** 19 lève le drapeau 7 dès qu'une condition d'erreur est rencontrée. Le drapeau 7 peut identifier une condition d'erreur et permet ainsi de prendre la décision nécessaire.

Si l'un ou l'autre de ces tests est négatif la position du drapeau 7 n'est pas modifiée.

2nd **0g** 40 (TI-58C uniquement) indique si le berceau imprimant est connecté en levant le drapeau 7. En testant le drapeau 7 on peut aussi détecter la présence du berceau imprimant. Il n'y aura aucune action sur le drapeau 7 si le berceau imprimant n'est pas connecté.



Adressage indirect.

XX – SUFFIXE D'ADRESSAGE INDIRECT – Utilisé après l'une des instructions qui suivent, il rappelle le contenu de la mémoire **XX** et l'utilise comme adresse de transfert ou comme numéro de la mémoire à rappeler.

Séquence de touches

STO	2nd	Ind	XX					
RCL	2nd	Ind	XX					
2nd	Exc	2nd	Ind	XX				
SUM	2nd	Ind	XX					
INV	SUM	2nd	Ind	XX				
2nd	Prs	2nd	Ind	XX				
INV	2nd	Prs	2nd	Ind	XX			
GTO	2nd	Ind	XX					
2nd	Pgm	2nd	Ind	XX				
2nd	Op	2nd	Ind	XX				
SBR	2nd	Ind	XX					
2nd	x>t	2nd	Ind	XX				
INV	2nd	x>t	2nd	Ind	XX			
2nd	x<t	2nd	Ind	XX				
INV	2nd	x<t	2nd	Ind	XX			
2nd	f ix	2nd	Ind	XX				
2nd	St Hg	2nd	Ind	XX				
2nd	DsZ	2nd	Ind	XX, N ou nnn				
2nd	DsZ	X	2nd	Ind	XX			
2nd	DsZ	2nd	Ind	XX	2nd	Ind	yy	
INV	2nd	DsZ	2nd	Ind	XX, N ou nnn			
INV	2nd	DsZ	X	2nd	Ind	XX		
INV	2nd	DsZ	2nd	Ind	XX	2nd	Ind	yy
2nd	H Hg	2nd	Ind	XX, N ou nnn				
2nd	H Hg	X	2nd	Ind	yy			
2nd	H Hg	2nd	Ind	XX	2nd	Ind	yy	
INV	2nd	H Hg	2nd	Ind	XX, N ou nnn			
INV	2nd	H Hg	X	2nd	Ind	XX		
INV	2nd	H Hg	2nd	Ind	XX	2nd	Ind	yy

Codes des touches

72 XX
73 XX
63 XX
74 XX
22 74 XX
64 XX
22 64 XX
83 XX
62 XX
84 XX
71 40 XX
77 40 XX
22 77 40 XX
67 40 XX
22 67 40 XX
58 40 XX
86 40 XX
97 40 XX
97 X 40 XX
97 40 XX 40 yy
22 97 40 XX
22 97 X 40 XX
22 97 40 XX 40 yy
87 40 XX
87 X 40 yy
87 40 XX 40 yy
22 87 40 XX
22 87 X 40 XX
22 87 40 XX 40 yy

Fonctions

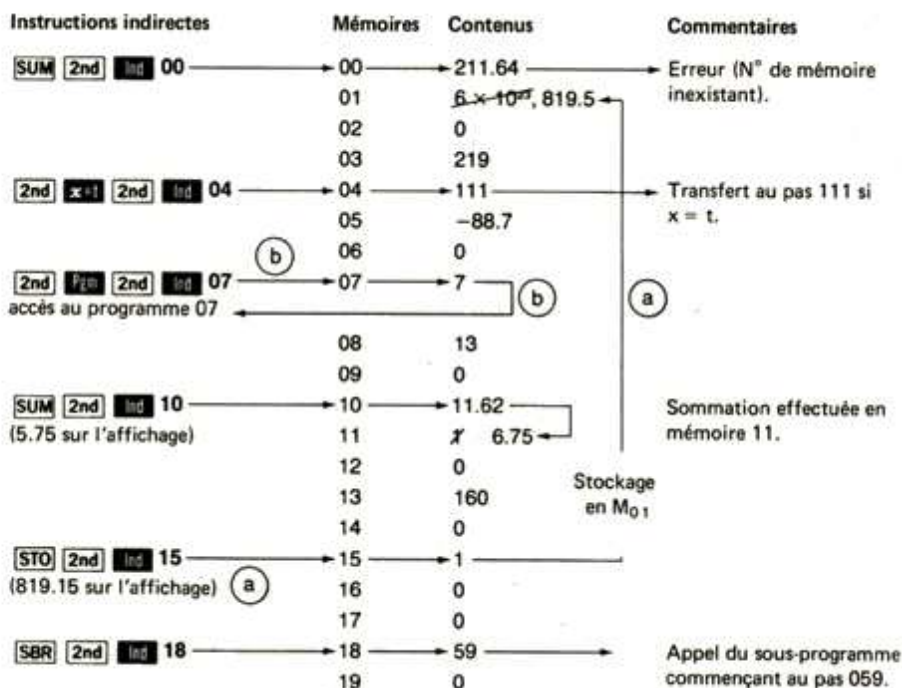
Stockage indirect		
Rappel indirect		
Echange indirect		
Somme indirecte en mémoire		
Soustraction indirecte en mémoire		
Multiplication indirecte en mémoire		
Division indirecte en mémoire		
Transfert inconditionnel indirect		
Appel indirect d'un programme		
Opération spéciale indirecte		
Sous programme indirect		
x > t transfert conditionnel indirect		
x < t transfert conditionnel indirect		
x = t transfert conditionnel indirect		
x ≠ t transfert conditionnel indirect		
Virgule fixe indirecte		
Lever un drapeau indirectement.		
Test DSZ avec	[mémoire indirecte adresse indirecte mémoire et adresse indirectes	
Saut si différent de zéro avec		[mémoire indirecte adresse indirecte mémoire et adresse indirectes
Tester si le drapeau est levé avec		
Tester si le drapeau est baissé avec	[numéro indirect du drapeau adresse indirecte numéro et adresse numéro	

Remarquez que les opérations indirectes en mémoire et les instructions PGM, OP et GTO ont des codes particuliers indépendants de leur position sur le clavier. Ces codes ont été créés de façon à économiser de la place en mémoire programme.



Les instructions indirectes utilisent le contenu de la mémoire XX pour définir l'adresse propre de l'instruction. En appuyant sur **RCL** **2nd** **Ind** **04** vous ne rappelez pas le contenu de la mémoire 04, mais le contenu de cette mémoire vous indique le numéro de la mémoire à rappeler. (Le contenu de la mémoire XX sera noté : C_{XX} de M_{XX}). Si 111 est stocké en M_{04} alors 111 devient l'adresse de l'instruction.

Le schéma ci-dessous illustre graphiquement ce principe :



Dans le cas de l'instruction **2nd** **bc=1** **2nd** **Ind** **04** l'adresse de transfert se trouve en mémoire 04 (111). L'instruction **STO** **2nd** **Ind** **15** va stocker 819.5 en M_{01} puisque le nombre 1 est contenu en M_{15} .

Si une adresse ne correspond pas aux limites de la machine telles que définies par la partition, il y a présence d'une condition d'erreur. L'instruction **SUM** **2nd** **Ind** **00** fait appel à la mémoire 211 qui n'existe pas. Il faut noter que si le contenu d'une mémoire est décimal, l'adresse prise en compte correspond à la partie entière de ce nombre; si le contenu est négatif, l'adresse correspondante sera 00.



Le berceau imprimant optionnel PC-100A, PC-100B ou PC-100C peut être utilisé avec votre calculatrice programmable pour exécuter différentes sortes d'impressions ou fonctions diverses.

Il vous permet en particulier :

1. D'imprimer le contenu de l'affichage à tout moment.
2. D'imprimer la liste des instructions contenues dans votre programme ainsi que le contenu des différentes mémoires.
3. D'imprimer un résultat à un endroit donné pendant l'exécution de votre programme.
4. D'imprimer chaque instruction exécutée par programme ou manuellement et la valeur numérique correspondante.
5. D'imprimer la liste de toutes les étiquettes contenues dans un programme ainsi que les adresses correspondantes.
6. D'imprimer un message alphanumérique chaque fois que nécessaire.
7. D'imprimer une courbe par points, par programme ou manuellement.
8. De verrouiller votre calculatrice sur le berceau imprimant, la mettant ainsi à l'abri du vol.
9. De recharger le bloc accumulateur de votre calculatrice pendant que l'imprimante est en fonctionnement.

L'inverseur situé dans la cavité prévue pour recevoir et pour recharger le bloc batterie doit être placé en regard de la mention «OTHER» pour permettre l'utilisation du PC-100A avec la TI-58, TI-58C ou TI-59.

L'interprétation des symboles et le programme de nettoyage de la tête d'écriture sont particuliers pour ces deux calculatrices et seront explicités plus loin.

Un manuel d'utilisation est fourni avec cette imprimante à multiples fonctions, cependant certains détails vous seront utiles à connaître dès maintenant.

Les instructions de base pour mettre en service le berceau imprimant, mettre en place la calculatrice, changer le rouleau de papier, sont fournies dans le manuel d'utilisation livré avec l'imprimante. Ce paragraphe décrit les différentes opérations possibles. Si vous utilisez l'imprimante, reportez vous au programme indiqué page VI-12 de ce manuel pour nettoyer la tête d'impression.

IMPORTANT : La mémoire permanente de la TI-58C permet de conserver les informations contenues dans la calculatrice lorsque vous désirez placer celle-ci sur l'imprimante. **S'assurer que l'exécution du programme soit arrêtée chargeur débranché et la calculatrice sur arrêt avant toute opération de mise en place.** Si vous utilisez un PC-100A/B la sauvegarde des informations ne se fera que si l'imprimante reste en position «Marche» et que la clef est correctement verrouillée. Pour le PC-100C la position de l'interrupteur n'a pas d'importance, toutefois veillez à ce que la clef soit correctement verrouillée.

*Note : Les TI-58, TI-58C et TI-59 ne peuvent pas fonctionner sur le PC-100.



IMPRESSION SELECTIVE.

Chaque fois que **2nd** **Pt** est utilisée manuellement ou rencontrée dans un programme, le contenu de l'affichage est imprimé. La touche **PRINT** de l'imprimante a le même effet que **2nd** **Pt** depuis le clavier. Si une condition d'erreur existe au moment où l'instruction d'impression a été rencontrée, un point d'interrogation est imprimé à droite du nombre imprimé.

Prenons le programme suivant qui imprimera les multiples de 4.

Pas et codes des touches	Séquence de touches
000 85	+
001 04	4
002 95	=
003 99	2nd Pt
004 81	RST

Introduisez un nombre de départ, appuyez sur **RST** **R/S** ; vous obtenez ceci :

Résultats



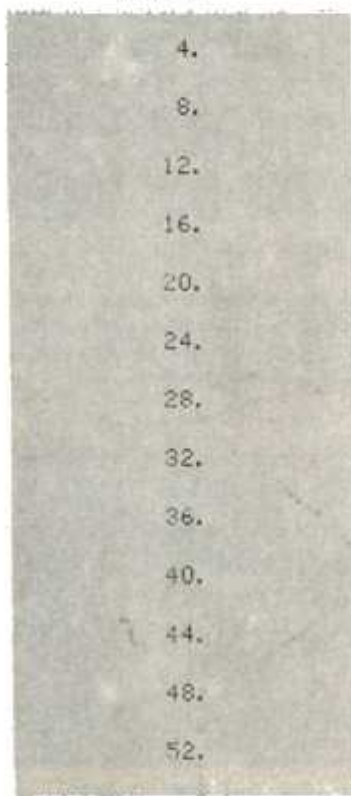
VI



Un certain nombre d'instructions d'impression ont été placées dans les programmes des Bibliothèques de façon à ce que les données et les résultats soient automatiquement imprimés si vous possédez l'imprimante.

La touche **2nd Adv** permet de séparer des groupes de nombres imprimés en faisant une avance de papier. Elle peut être utilisée manuellement ou placée dans un programme. Par exemple dans le programme précédent, intercallons l'instruction **2nd Adv** entre l'instruction d'impression et **RST** de façon à séparer chaque résultat.

Résultats



Cette touche peut être répétée plusieurs fois pour obtenir des espacements plus grands.

La touche **ADV*** sur l'imprimante fait avancer le papier de façon continue jusqu'à ce que la touche soit relâchée.



LISTE DU PROGRAMME.

Pour lister un programme il suffit d'appuyer sur **2nd** **List** à partir du clavier. Le programme se trouve listé à partir de l'endroit où se trouve le pointeur jusqu'à la fin de la mémoire programme. Pour arrêter cette liste vous appuyez quand vous le désirez sur **R/S**. Pour obtenir la liste complète de votre programme appuyez sur **RST** **2nd** **List**. La liste du programme précédent vous donne :

Pas de Progr.	Codes de touches	Symboles des touches
---------------	------------------	----------------------

000	85	+
001	04	4
002	95	=
003	99	PRT
004	98	ADV
005	81	RST
006	00	0
007	00	0

LISTE DU CONTENU DES MEMOIRES.

La séquence **INV** **2nd** **List** au clavier ou dans un programme liste le contenu de chaque mémoire en commençant par la mémoire dont le numéro est affiché jusqu'à la dernière mémoire définie par la partition ou jusqu'à l'utilisation de **R/S**. Ainsi la liste des mémoires M_{50} à la dernière définie par la partition M_{59} , par exemple, s'imprime de la façon suivante :

Contenu de la mémoire	N° de la mémoire
-----------------------	------------------

14.13181818	50
665.8568182	51
110.9761364	52
0.	53
0.	54
-5.5498068-12	55
0.	56
0.	57
1.	58
0.085106883	59

Cette liste a été obtenu en faisant **50** **INV** **2nd** **List**. Lorsqu'une telle séquence est utilisée dans un programme, les calculs cessent, la liste des mémoires s'imprime puis le programme s'arrête en retournant le contrôle au clavier.



IMPRESSION PAS A PAS DE L'EXECUTION D'UN PROGRAMME.

En appuyant sur la touche **TRACE** de l'imprimante vous provoquez l'impression pas à pas des calculs effectués au clavier ou dans un programme. La valeur calculée et l'instruction qui a été effectuée sont imprimées.

La touche **TRACE** est une touche à cliquet qui maintient le mode «Trace» tant que vous ne réappuyez pas sur cette touche. En mode Trace toute nouvelle fonction et tout nouveau résultat sont automatiquement imprimés. En cas de condition d'erreur un point d'interrogation est imprimé à droite du nombre.

Si votre programme «multiple de 4» est toujours en mémoire programme, appuyez sur **TRACE** et démarrez son exécution en appuyant sur **CLR** **RST** **R/S** ; vous obtenez ce qui suit :

Registre d'affichage	Symbole des fonctions
0.	+
4.	=
4.	
4.	PRT
	RST
4.	+
4.	=
8.	
8.	PRT
	RST
8.	+
4.	=
12.	
12.	PRT
	RST
12.	+
4.	=
16.	
16.	PRT

Pour arrêter l'exécution, appuyez sur **R/S**.

INTERPRETATION DES SYMBOLES EN MODE TRACE.

La plupart des symboles imprimés sont facilement identifiables, certains autres le sont moins. Page suivante vous trouverez un tableau complet des différents symboles et des fonctions qu'ils représentent.



VI

Symboles Instructions

A - E	A - E
A' - E'	2nd I - 2nd I'
ADV	2nd ADV
BST	Voir note
CE	CE
CLR	CLR
CP	2nd CP
CMS	2nd CMS
COS	2nd COS
DEG	2nd DEG
DEL	Voir note
DMS	2nd DMS
DSZ	2nd DSZ
EE	EE
ENG	2nd ENG
EQ	2nd x=1
EX*	2nd EX 2nd ENG
EXC	2nd EXC
FIX	2nd FIX
GE	2nd x>1
GO*	GTO 2nd ENG
GRD	2nd GRD
GTO	GTO
I EQ	INV 2nd x=1 †
I GE	INV 2nd x>1 †
I Σ+	INV 2nd Σ+ †
ICOS	INV 2nd ICOS †
IDMS	INV 2nd DMS †
IDSZ	INV 2nd DSZ †
IFF	2nd IFF
IFIX	INV 2nd FIX †
IFF	INV 2nd IFF †
IINT	INV 2nd INT †
ILNX	INV INX †

Symboles Instructions

ILOG	INV 2nd LOG †
IND	2nd IND
INS	Voir note
INT	2nd INT
INV	INV
IPD*	INV 2nd P1 2nd ENG †
IP/R	INV 2nd P+R †
IPRD	INV 2nd P1 †
ISBR	INV SBR †
ISIN	INV 2nd SIN †
ISM*	INV SUM 2nd ENG †
ISTF	INV 2nd STF †
ISUM	INV SUM †
ITAN	INV 2nd TAN †
Ix	INV 2nd x †
IXI	2nd IXI
IYX	INV Y* †
LBL	2nd LBL
LNX	INX
LOG	2nd LOG
LRN	Voir note
LST	2nd LST
NOP	2nd NOP
OP	2nd OP
OP*	2nd OP 2nd ENG
PAU	2nd PAU
PD*	2nd PD 2nd ENG
PG*	2nd PGR 2nd ENG
PGM	2nd PGM
P/R	2nd P+R
PRD	2nd PRD
PRT	2nd PRT
RAD	2nd RAD
RC*	RCL 2nd ENG

Symboles Instructions

RCL	RCL
R/S	R/S
RST	RST
RTN	INV SBR
SBR	SBR
SIN	2nd SIN
SM*	SUM 2nd ENG
SST	Voir note
ST*	STO 2nd ENG
STF	2nd STF
STO	STO
SUM	SUM
TAN	2nd TAN
WRT	2nd WRT
X ≥ T	x>t
X²	x²
x	2nd x
IXI	2nd IXI
1/X	1/x
√x	√x
YX	y*

SYMBLES

Σ+
π
)
(
-
+
×
÷
=
.
+/-

NOTE : Cette instruction ne peut apparaître qu'au cours du listage d'un programme, mais du fait que ce code ne peut être placé en mémoire programme par simple pression de la touche correspondante, il ne peut avoir été généré, accidentellement, qu'à l'occasion de la modification d'une instruction.

† Imprimé en mode trace seulement.

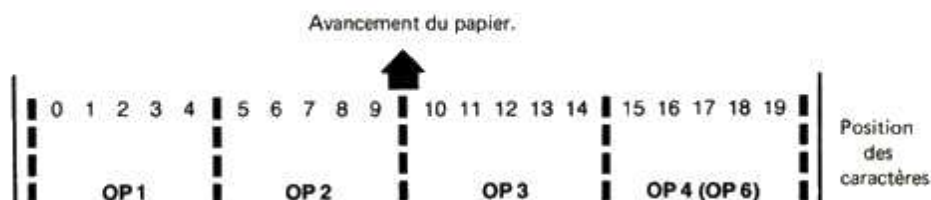
■ Uniquement pour la TI-58C : Le symbole «▲» s'imprime en mode trace lorsqu'un branchement conditionnel s'effectue.

OPERATIONS D'IMPRESSION SPECIALES.

Les opérations de contrôle 00 à 08 sont spécialement destinées à l'utilisation de l'imprimante.

IMPRESSION ALPHANUMERIQUE — Op 00-06

Les sept premières opérations de contrôle vous permettent de créer et d'imprimer des messages alphanumériques. Vingt caractères peuvent être imprimés sur chaque ligne. Ils sont assemblés et stockés par groupe de 5 caractères à la fois comme indiqué ci-dessous :



Chaque caractère imprimé correspond à un code à deux chiffres déterminé par le tableau suivant :

		CHIFFRES DES UNITES							
		0	1	2	3	4	5	6	7
CHIFFRES DES DIZAINES	0	blanc	0	1	2	3	4	5	6
	1	7	8	9	A	B	C	D	E
	2	-	F	G	H	I	J	K	L
	3	M	N	O	P	Q	R	S	T
	4	.	U	V	W	X	Y	Z	+
	5	×	+	!	@	#	\$	%	&
	6	↑	↓	;	'	=	*	>	<
	7	~	?	+	0	I	J	T	Σ

Par exemple, A est codé 13, + est codé 47 ... Les codes correspondant aux cinq caractères (10 chiffres) peuvent être introduits sur l'affichage en une seule fois. Si vous n'introduisez pas les caractères, les caractères manquants sont interprétés comme des zéros (chaque couple de zéros représente un espace). Pour obtenir un espace entre deux caractères, il suffit donc d'intercaler deux zéros.



Dès qu'une série de codes sont placés sur l'affichage, appuyez sur **2nd** **0p** 01, 02, 03 ou 04 pour indiquer l'endroit où doivent être imprimés ces caractères sur la ligne.

- 2nd** **0p** 01 — premier quart de la ligne, extrême gauche
- 2nd** **0p** 02 — deuxième quart de la ligne, milieu gauche
- 2nd** **0p** 03 — troisième quart de la ligne, milieu droit
- 2nd** **0p** 04 — quatrième quart de la ligne, extrême droit

En appuyant sur **2nd** **0p** 00 vous effacez le registre d'impression. L'instruction **2nd** **0p** 05 déclenche l'impression de ce registre. Il est ainsi possible en mode calcul ou dans un programme d'imprimer des annotations permettant l'identification des données ou des résultats qui seront imprimés.¹

Exemple : Imprimer le titre suivant « π^2 VS X% TESTS 3/22»

Symboles	π	2	V	S	X	%	T	E	S	T	S	3	/	2	2				
Codes	53	70	00	42	36	00	44	61	00	37	17	36	37	36	00	04	63	03	03

Appuyer	Affichage	Commentaires
CLR 2nd 0p 00	0.	Efface le registre d'impression.
5370004236 2nd 0p 01	5370004236.	« π^2 VS» pour imp. à l'extrême gauche
44610037 2nd 0p 02	44610037.	«X% T» pour imp. au milieu gauche
1736373600 2nd 0p 03	1736373600.	«ESTS» pour imp. au milieu droit
463030300 2nd 0p 04	463030300.	«3/22» pour imp. à l'extrême droite
2nd 0p 05	463030300.	Imprime le contenu du registre d'impression.

Les espaces sont créés par un couple de zéros; lorsque ceux-ci sont non-significatifs, il n'est pas utile de les introduire comme le montre l'exemple ci-dessus. Les différents quarts de la ligne peuvent être introduits dans n'importe quel ordre. Le stockage de ces différentes annotations utilise les registres d'opérations en attente 5, 6, 7 et 8; pour cette raison n'utilisez pas plus de 4 niveaux d'opérations en attente lorsque vous composez un message.

Assurez-vous que le format virgule fixe ou notation scientifique ou notation ingénieur soit annulé avant de commencer à composer un message.

On notera pour la TI-58 et la TI-59 que les opérations **2nd** **0p** 00 à **2nd** **0p** 05 provoquent la suppression de la partie fractionnaire d'un nombre dans le registre d'affichage ou comme le ferait la fonction «partie entière». Pour la TI-58C, seules les opérations **2nd** **0p** 01 à **2nd** **0p** 04 auront la même conséquence sur le registre d'affichage.

VI



Une opération spéciale permet d'affecter un symbole alphanumérique de 4 caractères maximum sur la même ligne qu'un nombre imprimé. **2nd** **06** imprime simultanément le contenu du registre d'affichage et les 4 caractères situés le plus à droite dans le registre d'impression.

Exemple : Ecrire le programme qui imprimera sur la même ligne, Pi et son approximation soit 22/7.

Pas et codes des touches	Séquence de touches		
000 03	3	}	
001 03	3		P
002 02	2		}
003 04	4	I	
004 69	2nd 06	Stocke PI pour l'imprimer à l'extrême droite.	
005 04	0 4		
006 02	2		
007 02	2		
008 55	+		
009 07	7		
010 95	=	22 ÷ 7	
011 69	2nd 06	Impression.	
012 06	6		
013 91	R/S		

Lors de l'exécution, l'imprimante vous indiquera «3.142857143 PI».

Tracé d'une courbe par points – Op 07

L'opération spéciale 07 imprime une astérisque (*) dans une des colonnes 0 à 19 suivant la valeur contenue dans le registre d'affichage (0 à 19); 0 correspond à la position d'extrême gauche. Spécialement destinée à être utilisée dans un programme, cette opération vous permet de tracer une courbe point par point. Une seule astérisque est imprimée sur chaque ligne correspondant à la partie entière du nombre x affiché, $0 \leq x < 20$. Toute valeur n'appartenant pas à cet intervalle provoque le clignotement de l'affichage. Seule la partie entière du nombre affiché est prise en compte.

Exemple : Ecrire le programme qui tracera une sinusoïde par points, de 18 degrés en 18 degrés.

Séquence de touches

2nd **LN**
 A
 RCL
 1
 2nd **SIN**
 +
 1
 =
 X
 9
 -
 9
 =
 2nd **STO**
 0 **7**
 1
 8
 SUM
 1
 A

Résultat

0 1 3 5 7 9 11 13 15 17 19 Valeurs

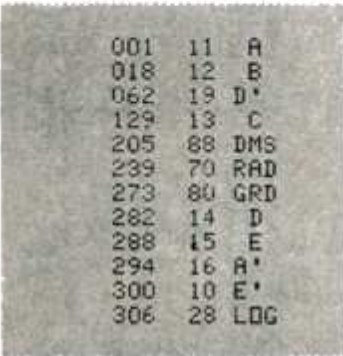


Le sinus est rendu positif en ajoutant 1, les valeurs sont donc comprises entre 0 et 2; en multipliant par 9.9 nous transformons le résultat de façon à ce qu'il soit compris entre 0 et 19.8, occupant ainsi toute la largeur de la bande de papier. Pour démarrer l'exécution, placez une valeur de départ en mémoire 01 puis appuyez sur **A**. L'arrêt s'effectue en appuyant sur **R/S**.

Liste des étiquettes utilisées dans un programme — **0p** 08

Pour obtenir la liste des étiquettes et le numéro du pas où elles sont situées en mémoire programme, il suffit simplement d'appuyer sur **2nd** **0p** 08. La liste débute à partir de l'endroit où se trouve le pointeur, aussi appuyez au préalable sur **GTO** 0 ou **RST** avant **2nd** **0p** 08, pour obtenir la liste de toutes les étiquettes.

Pas de prog.	Codes touches	Symboles
-----------------	------------------	----------



001	11	A
018	12	B
062	19	D'
129	13	C
205	88	DMS
239	70	RAD
273	80	GRD
282	14	D
288	15	E
294	16	A'
300	10	E'
306	28	LOG



PROCEDURE DE NETTOYAGE DE LA TETE D'IMPRESSION.

La procédure de nettoyage de la tête d'impression est donnée dans le manuel d'utilisation du berceau imprimant. Le programme qui suit peut être utilisé avec les TI-58, TI-58C, et TI-59.

Pas et codes des touches	Séquence de touches
000 04	4
001 42	STO
002 00	0 0
003 09	9
004 42	STO
005 06	6
006 52	EE
007 01	1
008 00	0
009 94	+/-
010 22	INV
011 52	EE
012 35	1/x
013 76	2nd 1bl
014 11	A
015 84	2nd Op 2nd Ind
016 00	0
017 97	2nd Op
018 00	0
019 11	A
020 76	2nd 1bl
021 12	B
022 69	2nd Op
023 05	5
024 97	2nd Op
025 06	6
026 12	B
027 91	R/S

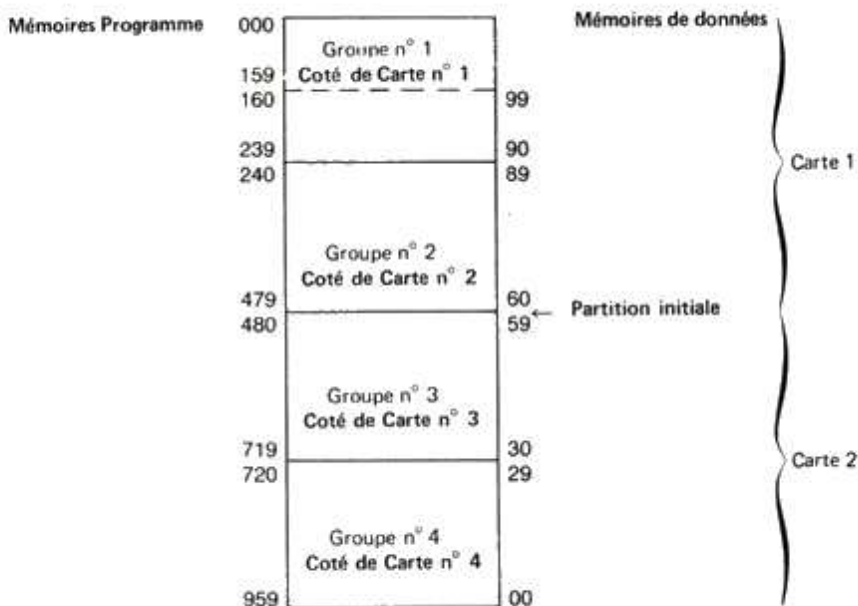
Pour démarrer l'exécution, appuyez sur **RST** **R/S** . Eventuellement répétez l'exécution si nécessaire.

VII

Cartes magnétiques (TI-59 uniquement)



Vous pouvez conserver un programme ou des données se trouvant dans votre calculatrice en les enregistrant sur l'une des cartes magnétiques vierges fournies avec votre machine. Comme vous le savez maintenant, votre calculatrice possède 120 mémoires de stockage réparties entre les mémoires de données et la mémoire-programme. Ces mémoires sont réparties en quatre groupes de trente mémoires chacun. Une carte magnétique peut recevoir deux de ces groupes de mémoires, un par côté de carte.



Note : Pour faire lire une carte magnétique, vous devez nécessairement remettre à zéro l'affichage avec la touche CLR, sélectionner le mode décimale flottante en appuyant sur [2nd] [Fix] 9 et introduire la carte (piste 1, 2, 3 ou 4)* dans la fente centrale située sur le côté droit de la calculatrice. Le lecteur se met en marche automatiquement et procède à la lecture de la carte (une piste). Pour plus de détail reportez-vous à la page VII-5.

- *. Chaque piste peut enregistrer les données d'un groupe.
- Une carte contient deux pistes.
- Deux cartes sont nécessaires (quatre pistes) pour enregistrer la totalité de la capacité de la TI-59 (mémoires et programmes).



VII

ENREGISTREMENT D'UNE CARTE

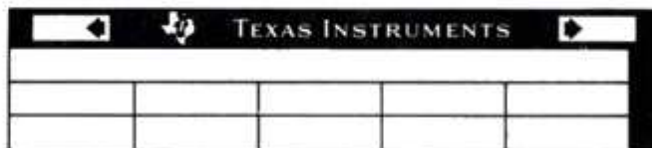
L'enregistrement d'une carte se fait par l'instruction $\boxed{2nd} \boxed{Write}$. Pour enregistrer le contenu du groupe n ($n = 1, 2, 3$ ou 4) sur le côté de carte n , appuyez sur n $\boxed{2nd} \boxed{Write}$ et introduisez la carte (côté imprimé vers le haut) dans la fente la plus basse située sur le côté droit de la calculatrice. Notez que $Fix\ 0$ est la seule option du format virgule fixe qui vous permettra d'enregistrer une carte. Si vous n'êtes pas certain du format d'affichage, appuyez sur $\boxed{INV} \boxed{2nd} \boxed{Fix}$ avant d'entreprendre la séquence d'enregistrement.



Le contenu de ce groupe de mémoires, qu'il contienne un programme, des données ou les deux, se trouve enregistré sur la carte. Le côté de carte contient les informations du groupe ainsi que son numéro. Seule la partie entière du nombre affiché est prise en compte lorsque vous appuyez sur $\boxed{2nd} \boxed{Write}$. Ainsi 2.31 $\boxed{2nd} \boxed{Write}$ enregistre le contenu du groupe numéro 2 sur la carte. Tout nombre inférieur à 1 ou supérieur à 5 provoque le clignotement de l'affichage ; l'enregistrement est alors impossible. Aussi, assurez vous avant tout enregistrement que l'affichage contienne bien 1, 2, 3 ou 4

Lorsque vous introduisez la carte dans la calculatrice, ne tentez pas de la retenir à partir du moment où elle est entraînée par le moteur électrique. L'affichage reste aveugle tant que l'enregistrement n'est pas achevé, puis le numéro du groupe apparaît. Si ce numéro clignote après l'enregistrement, effacez l'affichage et recommencez la procédure d'enregistrement. Si le clignotement persiste, la carte est peut-être en cause, prenez en une autre.

Pour enregistrer des données, souvenez vous que la première mémoire de données 00 est située à la fin du groupe 4 (voir page précédente).



Vous pouvez inscrire sur la carte les renseignements nécessaires à son identification. Les coins supérieurs permettent de noter le numéro du groupe enregistré ; les flèches indiquent le sens d'introduction de la carte. Au centre de la carte vous pouvez inscrire le titre du programme ainsi que la partition requise. En bas vous avez deux séries de cases destinées à inscrire la fonction de chaque touche utilisateur (\boxed{A} à \boxed{E} pour les cases du bas, $\boxed{2nd} \boxed{A}$ à $\boxed{2nd} \boxed{E}$ pour les cases du haut).

Notez que lorsqu'un groupe est enregistré, le numéro du groupe et la partition correspondante sont également enregistrés sur la carte.



Affichage
lorsque

2nd **Write** est
utilisé, carte
introduite

	Programme normal	Programme protégé
1, 2, 3, 4	Enregistre sur un côté de carte le contenu du groupe (programme et/ou données), son numéro ainsi que l'indication de la partition.	Si le groupe contient uniquement un programme, la carte passe sans être enregistrée, l'affichage clignote. Si le groupe contient quelques données, celles-ci sont enregistrés mais non protégées par un signe négatif sur la carte.
-1, -2, -3, -4	Enregistre et protège le côté de carte avec le numéro signé du groupe. Enregistre également la partition	Si le groupe contient uniquement un programme, la carte passe sans être enregistrée, l'affichage clignote. Si le groupe contient quelques données, il est enregistré avec son numéro signé.
Tout autre nombre	La carte passe, elle n'est pas enregistrée. Deux caractères les plus à droite de l'affichage clignent.	Comme pour le programme normal.

Si l'affichage clignote lorsque vous essayez d'enregistrer une carte, celle-ci passe sans que l'enregistrement se produise et les deux caractères les plus à droite de l'affichage clignent.

La calculatrice ne doit pas être en virgule fixe lorsque vous enregistrez une carte.

Lorsque l'affichage clignote, le moteur d'entraînement continue à tourner jusqu'à ce que la carte soit retirée.

Attention : Ne jamais introduire dans le lecteur magnétique une carte magnétique à l'envers : Texas Instruments vers le sol. Ne jamais introduire une carte non magnétique dans le lecteur magnétique (cartes impression jaune, fond noir).



PROTECTION D'UN PROGRAMME

Si votre programme est confidentiel, vous pouvez le protéger en introduisant le numéro du groupe n à enregistrer avec le signe moins. En appuyant sur n **+/-** **2nd** **Write** vous enregistrez un programme qui pourra seulement être relu en mémoire programme, et exécuté mais sans plus. Les restrictions d'utilisation, une fois celui-ci relu par la calculatrice, sont les suivantes :

- Le programme ne pourra pas être listé sur l'imprimante
- Le mode Trace ne sera pas pris en considération
- Les étiquettes du programme ne pourront être listées
- La mise en mode programmation, le changement de partition et le re-enregistrement sont impossibles
- L'exécution en pas à pas grâce à la touche **SST** ou la touche **Pause** sont impossibles.

Il est impossible de forcer la calculatrice à lire un côté de carte protégé dans un groupe de mémoires différent de celui spécifié à l'enregistrement. Toutes ces impossibilités correspondent à la mise en position haute d'un drapeau interne qui ne peut être baissé qu'en appuyant sur **2nd** **CP** ou en éteignant la calculatrice. Vous pouvez lire ou transférer un autre programme après qu'un programme protégé ait été mis en mémoire programme. Néanmoins l'état du drapeau de protection restera inchangé. Le nouveau programme sera donc également protégé comme le précédent.

Il est impossible de protéger le contenu des mémoires de données. C'est pourquoi, si la partition se trouve au milieu d'un groupe, celui-ci ne pourra être protégé. Un groupe contenant partiellement des données sera considéré par la calculatrice comme un groupe formé uniquement de mémoires de données.

Un programme non protégé pourra être enregistré sur une carte contenant précédemment un programme protégé en utilisant la procédure mentionnée plus haut. Le nouveau programme sur la carte ne sera pas protégé.



LECTURE D'UNE CARTE.

Le moteur d'entraînement de votre calculatrice fait passer automatiquement la carte magnétique dans la calculatrice lorsque celle-ci est introduite dans la fente appropriée en mode calcul. La lecture ou la non-lecture de la carte dépend du contenu du registre d'affichage et de la partition.

1. Avec un zéro sur l'affichage, un groupe peut être lu pour autant que la partition de la calculatrice soit conforme à celle enregistrée sur la carte. Si non il n'y a pas de lecture et l'affichage clignote avec le numéro du groupe correspondant à la carte introduite.
2. Avec n sur l'affichage la calculatrice pourra lire uniquement le groupe n . Si la carte contient un autre numéro de groupe, celui-ci clignotera sur l'affichage; la lecture n'aura pas été effectuée. Comme précédemment si la partition est incorrecte, le numéro du groupe introduit clignotera sur l'affichage sans que la carte soit lue.
3. Avec $-n$ sur l'affichage tout groupe lu par la calculatrice est placé dans le groupe n . La partition est ignorée. Dès la lecture effectuée la calculatrice affiche le numéro du groupe qui vient d'être lu. Une carte protégée (numéro de carte négatif) ne peut pas être introduite dans un autre groupe que son groupe d'origine tel qu'il est enregistré sur la carte.

Un zéro clignotant sur l'affichage après la lecture de la carte indique qu'une erreur de lecture a été détectée, la carte doit être réintroduite.

Tout autre nombre que zéro ou $\pm n$ ($n = 1, 2, 3$ ou 4) sur l'affichage au moment de l'introduction d'une carte interdit toute lecture et entraîne le clignotement des deux caractères de droite de l'affichage.

LECTURE D'UNE CARTE AU COURS D'UN PROGRAMME.

Pendant l'exécution d'un programme, **INV** **2nd** **Write** demande à la calculatrice de lire une carte magnétique conformément aux règles précédentes. (Le numéro du groupe ou **CLR** doit être prévu dans le programme avant les instructions **INV** **2nd** **Write**). Une carte peut être introduite dans la fente de la calculatrice, mais ne sera pas lue tant que l'instruction **INV** **2nd** **Write** n'est pas rencontrée dans le programme. Ceci permet d'introduire des données au moment désiré.

Souvenez-vous qu'une carte introduite dans le lecteur sera automatiquement entraînée dès que le programme rencontrera une instruction du type **INV** **SBR** ou **R/S** .



LECTURE D'UNE CARTE MAGNETIQUE.

Affichage à l'introduction de la carte	Carte contenant un programme normal	Carte contenant un programme protégé
0	Lecture des informations dans le groupe dont le numéro est magnétisé sur la carte pour autant que la partition soit conforme à celle indiquée par la carte. Si la partition est incorrecte, la carte passe dans le lecteur sans être lue. Le n° de la carte clignote.	Même procédure que pour un programme normal.
1, 2, 3, 4	Lecture si le numéro de la carte est correct suivi de l'affichage de ce numéro. Si le numéro ou la partition sont incorrects, la carte passe sans être lue. Le numéro de la carte clignote.	Lecture si le numéro de la carte est correct suivi de l'affichage de ce numéro négatif. Même procédure que pour un programme normal.
-1, -2, -3, -4	Lecture forcée dans le groupe correspondant quel que soit le numéro de la carte ou la partition Un programme protégé ne peut être forcé dans aucun autre groupe que le sien propre.	Lecture si le numéro de la carte est correct suivi de l'affichage de ce numéro. Si numéro ou partition incorrects, la carte passe sans être lue. Le numéro de la carte clignote.
Tout autre nombre	La carte passe mais n'est pas lue. Les deux caractères de droite de l'affichage clignotent.	Même procédure que pour un programme normal.

Si l'affichage clignote avec une valeur quelconque lorsque vous essayez de lire une carte, celle-ci passe sans être lue. Les deux caractères de droite continuent à clignoter.

La calculatrice ne doit pas être en décimalisation fixe au moment de la lecture d'une carte.

Assurez-vous que la calculatrice soit correctement chargée ou branchée sur son chargeur avant d'entreprendre un long calcul. C'est particulièrement important pour assurer une bonne lecture ou écriture de la carte magnétique.

Un zéro clignotant sur l'affichage après la lecture d'une carte indique une mauvaise lecture. Une nouvelle lecture est donc nécessaire.



PRECAUTION A PRENDRE CONCERNANT LES CARTES MAGNETIQUES.

ATTENTION : Une carte magnétique enregistrée peut être altérée si elle est exposée aux poussières ou placée au voisinage de corps étrangers, d'un aimant permanent ou d'un champ magnétique (moteur électrique, transformateur, etc..).

Une carte magnétique peut conserver les informations enregistrées indéfiniment. Les informations ne risquent pas de s'estomper ni de s'affaiblir au fil du temps. Elles restent enregistrées tant qu'elles ne sont pas altérées par un champ magnétique externe. Si les signaux enregistrés ne se détériorent pas, par contre les caractéristiques physiques de la carte et le mécanisme de lecture de la calculatrice peuvent être endommagés.

Maniement des cartes magnétiques.

Il est primordial de prendre de bonnes habitudes pour manier les cartes. Une carte souillée, pliée ou entaillée devient inutilisable. Une dégradation physique résulte en général d'une accumulation de petits problèmes ou d'un mauvais maniement.

Il y a plusieurs causes de détérioration à considérer. Les cendres de cigarettes, les particules d'aliment, les liquides, la poussière et les corps gras sont les contaminants les plus courants qu'il faut éviter. Une carte peut être endommagée par simple contact avec une surface souillée ou des doigts sales. Le contact des doigts, même propres, provoque un dépôt de gras qui, à la longue, s'accumule et retient les particules de poussière. Il est important de savoir qu'une carte souillée peut contaminer non seulement le mécanisme de lecture de la calculatrice, mais également les autres cartes qui sont passées par la suite. Dans certains cas, surtout lorsque la détérioration est provoquée par un corps huileux, le lecteur de cartes de la calculatrice peut être rendu inopérant, nécessitant le retour de la calculatrice auprès du service après-vente de Texas Instruments. Les instructions qui suivent sont primordiales pour assurer une durée de vie maximum aux cartes magnétiques.

1. Saisir la carte par la tranche.
2. Eloigner la carte des objets magnétiques ou tranchants qui pourraient endommager la surface magnétique.
3. Remettre la carte dans le porte cartes dès qu'elle n'est plus utilisée.
4. Si une carte est détériorée, la nettoyer immédiatement.
5. Ne pas tenter de lire ou d'enregistrer une carte visiblement sale.



NETTOYAGE D'UNE CARTE.

Une carte souillée peut être nettoyée sans utiliser de nettoyeurs ou de solvants spéciaux. Les produits à base d'essence ne doivent en aucun cas être utilisés pour nettoyer une carte. Les poussières et les particules étrangères peuvent être nettoyées avec un tissu doux et sec; les autres formes de détérioration peuvent être supprimées en passant la carte dans de l'eau tiède contenant une goutte d'un liquide détergent doux. Elle doit ensuite être rincée puis séchée avec un linge doux.

INSCRIPTIONS SUR UNE CARTE.

Les cartes magnétiques vierges fournies avec votre calculatrice ont une partie réservée aux différentes inscriptions, nombres, symboles et abréviations identifiant le programme. Ces inscriptions peuvent se faire avec un crayon à mine douce ou un marqueur à encre lavable; si vous désirez obtenir une identification permanente, utilisez des marqueurs à encre indélébile. Renseignez-vous auprès de votre revendeur; ces marqueurs existent en plusieurs couleurs.

UTILISATION DE LA CARTE DE NETTOYAGE DU GALET D'ENTRAÎNEMENT.

Cette carte peut être utilisée après 500 passages environ ou lorsque les cartes magnétiques commencent à glisser ou à avancer par saccades dans la calculatrice. Appuyez sur **1** **2nd** **Write** et introduisez la carte. Dès que le moteur tourne retenez la puis laissez la avancer, retenez la à nouveau et ceci pendant trois à quatre secondes, le temps que le galet soit correctement nettoyé. Retirez la carte et appuyez sur **R/S**. Si le moteur continue à tourner, il faut éteindre la calculatrice pour l'arrêter.

COMPATIBILITE DE LECTURE/ENREGISTREMENT ENTRE T1-59.

Le lecteur de carte de votre T1-59 a fait l'objet de test et de soin particuliers au montage. Toutefois, malgré l'emploi de composants de haute technologie, il se peut que vous ne puissiez pas lire une carte enregistrée sur une autre T1-59. Texas Instruments ne garantit pas la compatibilité des machines entre elles.



UTILISATION DE LA CARTE DE NETTOYAGE DE LA TÊTE MAGNETIQUE.

La carte de nettoyage fournie avec la calculatrice possède à la place de la surface magnétique une surface abrasive. Elle permet de chasser toutes particules qui auraient été déposées après les différentes lectures ou enregistrements effectués. Elle ne doit en aucun cas être utilisée inconsidérément. Un usage excessif pourrait modifier les caractéristiques de la tête de lecture/écriture. Dans l'appendice A, le paragraphe «EN CAS DE DIFFICULTES» vous fournira les instructions vous indiquant les cas pour lesquels cette carte peut remédier à un problème. Pour l'utiliser, il suffit de l'introduire dans la fente du bas située sur le côté de la calculatrice comme pour lire une carte ordinaire et de la laisser passer dans la calculatrice; appuyez ensuite sur **CLR** pour arrêter le clignotement de l'affichage. Cette carte doit être utilisée avec parcimonie et jamais plus d'une fois à la suite. Assurez-vous qu'elle soit propre avant de l'utiliser.

TEST DE LECTURE/ENREGISTREMENT.

La procédure suivante permet de tester la lecture et l'enregistrement de cartes magnétiques. Transférez le programme de diagnostic de la bibliothèque de base dans la calculatrice. Enregistrez ce programme sur une carte magnétique puis lisez la carte que vous venez d'enregistrer.

Vérifiez que l'enregistrement a bien fonctionné en exécutant le programme de diagnostic.

La séquence à exécuter est la suivante:

2nd **Prm** **01** **2nd** **Op** **09** **1** **2nd** **Write**

Introduisez une carte magnétique. **1** apparaît à l'affichage. Eteignez puis rallumez la calculatrice, introduisez la carte que vous venez d'enregistrer. **1** apparaît à l'affichage. Appuyez sur **SBR** **≡**. Au bout de quelques secondes **1** apparaît à l'affichage.

Si ce test ne s'effectue pas correctement, référez vous à l'Appendice A.

A

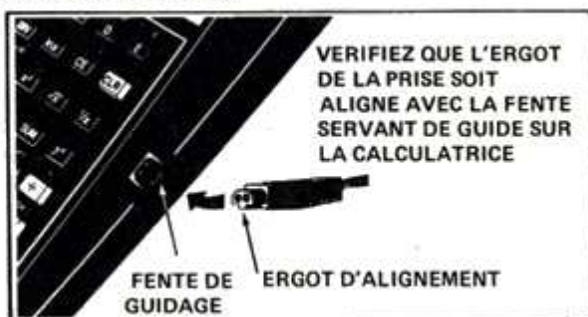
Entretien



ACCUMULATEUR ET CHARGEUR

Pour assurer une autonomie maximum, connectez l'adaptateur/chargeur [AC 9900 H] au secteur 220 V/50 Hz d'une part et à la calculatrice d'autre part et chargez le bloc accumulateur pendant au moins 4 heures quand la calculatrice est éteinte, 10 heures quand elle est en fonctionnement. L'accumulateur et le chargeur chauffent légèrement lorsqu'ils sont sous tension ce qui est normal et sans conséquence.

ATTENTION : La calculatrice peut être endommagée si l'adaptateur/chargeur est branché sans que le bloc accumulateur soit placé dans la calculatrice.



Lorsque le bloc accumulateur est complètement chargé la calculatrice a une autonomie approximative de 2 à 3 heures. Cependant n'hésitez pas à brancher la calculatrice sur son adaptateur/chargeur si vous pensez que le bloc accumulateur est presque déchargé. Un bloc accumulateur presque déchargé peut avoir un effet néfaste sur les calculs en cours. Un bloc accumulateur déchargé se repère par un affichage sombre, fantaisiste ou complètement aveugle ou encore par le démarrage du moteur d'entraînement des cartes. Si le bloc accumulateur se trouve complètement déchargé pendant la lecture d'une carte, le programme contenu sur la carte risque d'être endommagé. Si vous vous apercevez que les accumulateurs de votre TI-58C sont en bas de charge, éteignez-la immédiatement pour éviter la perte des informations en mémoire. Mettez en place un bloc d'accumulateur correctement chargé ou connectez votre TI-58C à son chargeur le plus rapidement possible. Veillez à toujours arrêter l'exécution d'un programme avant de mettre la calculatrice sur arrêt et d'enlever le bloc accumulateur.

La durée de vie du bloc accumulateur est difficile à prédire, cependant en usage normal il peut avoir une durée de vie de 2 à 3 ans, soit environ 500 à 1000 cycles de recharge:

RECHARGEMENT PERIODIQUE La calculatrice peut fonctionner branchée sur l'adaptateur/chargeur indéfiniment, toutefois le bloc accumulateur perdra de sa capacité s'il n'est pas régulièrement déchargé et rechargé. La durée de vie maximum du bloc accumulateur sera obtenue en utilisant au moins deux fois par mois la calculatrice sur son alimentation autonome.

DECHARGE EXCESSIVE DES BATTERIES Si la calculatrice reste accidentellement allumée pendant une période trop longue, (une nuit, par exemple) il est nécessaire de brancher la calculatrice éteinte sur son adaptateur/chargeur au moins 24 heures: Si cela ne restitue pas au bloc accumulateur sa charge normale il est alors nécessaire de le remplacer. Des déchargements excessifs et répétés peuvent, en effet, détériorer définitivement le bloc accumulateur. Vous pourrez trouver un bloc accumulateur BP1A de remplacement auprès de votre revendeur habituel.

STOCKAGE Si la calculatrice ne doit pas être utilisée pendant plusieurs semaines, il est préférable de recharger le bloc accumulateur auparavant. Les batteries ne coulant pas, il est conseillé de les laisser en place dans la calculatrice.



REPLACEMENT DU BLOC ACCUMULATEUR Le bloc accumulateur peut facilement être retiré; pour cela il suffit de retourner la calculatrice et de placer une languette ou une pièce de monnaie dans la fente centrale pour faire levier. Le bloc accumulateur se dégage et il n'y a plus qu'à le sortir de son logement. Veillez à toujours arrêter l'exécution de votre programme, mettre la calculatrice sur arrêt et débrancher le chargeur avant de procéder à l'enlèvement du bloc accumulateur.



Les deux languettes métalliques situées sur le bloc accumulateur sont ses deux contacts. Pour éviter toute détérioration, faites attention aux risques de court-circuits. Grattez de temps en temps ces deux contacts pour éviter tout mauvais contact dû à la corrosion.

Pour remettre le bloc accumulateur en place, introduire le côté rond vers le haut de la calculatrice puis faire pivoter le bloc accumulateur pour le mettre en place. Appuyez ensuite fermement, un dé clic vous avertira de son blocage.





EN CAS DE DIFFICULTES

En cas de difficultés avec votre calculatrice, les instructions qui suivent peuvent vous aider à déterminer le problème avec précision. Vous pouvez peut-être ainsi trouver vous-même la solution de votre problème. Dans le cas contraire, écrivez ou téléphonez à TEXAS INSTRUMENTS en donnant avec précision le diagnostic du problème rencontré.

Si des difficultés apparaissent alors que votre calculatrice est connectée à l'imprimante PC-100A et qu'elles disparaissent lorsque la calculatrice fonctionne de façon autonome, référez-vous au manuel d'utilisation de l'imprimante.

SYMPTOMES

1. L'affichage donne des résultats erronés, clignote avec des nombres fantaisistes, baisse d'intensité, s'éteint ou le moteur du lecteur de cartes tourne sans arrêt.
2. L'affichage reste éteint sans raison apparente.
3. L'affichage clignote pendant que vous exécutez des calculs au clavier.
4. L'affichage clignote chaque fois qu'un programme de la bibliothèque est appelé.

REMEDES

Le bloc accumulateur est sans doute déchargé reportez-vous au paragraphe "Accumulateur et chargeur".

Appuyez et maintenez appuyé un instant la touche **[R/S]**. Si l'affichage revient c'est que la calculatrice était en train de dérouler un programme ou était perdue dans une boucle ou encore qu'elle était en attente d'une introduction de carte.

Appuyez et maintenez appuyé un instant la touche **[RST]**. Si l'affichage revient c'est que la calculatrice était soit en train d'exécuter un programme de la bibliothèque, perdue dans une boucle (possible si les batteries sont peu chargées) soit d'exécuter un long programme.

Les batteries sont déchargées ou mal introduites dans leur logement.

Une opération ou une séquence invalide de touches a été composée ou la capacité limite a été dépassée. Reportez-vous à l'appendice B.

Le numéro ne correspond pas à un programme de la bibliothèque. Consultez le manuel de la bibliothèque.

Le module préenregistré est incorrectement placé. Voir section III de ce manuel.



SYMPTOMES

5. L'affichage clignote ou fourni des résultats incorrects pendant l'exécution d'un programme de la bibliothèque.

6. L'affichage clignote ou fourni des résultats incorrects pendant l'exécution d'un de vos programmes.

REMEDES

Un mauvais numéro de programme a été appelé.

Mode d'emploi suivi incorrectement. Référez-vous au manuel de la bibliothèque.

La partition ne laisse pas suffisamment de mémoire disponible pour que le programme s'exécute.

La calculatrice travaille en virgule fixe. Appuyez sur **INV** **2nd** **fix** avant de réexécuter le programme.

Appuyez sur **CLR** **2nd** **Pgm** **1** **SBR** **=** pour exécuter le diagnostic de la bibliothèque. Si le résultat clignote, vérifiez que le module est correctement placé (paragraphe III) et refaites cette séquence.

Une opération invalide ou un dépassement de capacité s'est produit durant l'exécution du programme. Voyez les appendices B, C et D.

Un des programmes de la bibliothèque a été appelé. Appuyez sur **RST** et essayez à nouveau.

Appuyez sur **CLR** **2nd** **Pgm** **1** **SBR** **=** pour exécuter le diagnostic de la bibliothèque. Si le résultat clignote, reportez-vous au symptôme numéro 5.

Si le programme a été lu à partir d'une carte magnétique, exécutez le programme de test de lecture/enregistrement (voir Section VII). Si le test est correct, vérifiez le programme enregistré avec celui que vous avez écrit pour déceler une erreur éventuelle. Voir le symptôme numéro 7.



SYMPTOMES

7. L'affichage clignote après lecture ou enregistrement d'un programme.

8. La calculatrice ne veut pas passer en mode programmation; l'exécution pas à pas, le listage et enregistrement sur carte sont impossibles.

Lorsque vous retournez la calculatrice pour réparation, renvoyez la calculatrice, l'adaptateur/chargeur, le module préenregistré et les cartes magnétiques ayant fait l'objet d'un problème. Par souci de sécurité, faites l'expédition en **EXPRESS RECOMMANDE** en joignant un justificatif de la date d'achat. Texas Instruments n'assume aucune responsabilité pour toute perte ou détérioration survenue au cours d'une expédition non assurée.

Si la calculatrice est hors garantie, une somme forfaitaire conforme au tarif en vigueur au moment du retour vous sera facturée. Joignez à votre envoi les informations décrivant les difficultés que vous avez rencontrées avec votre calculatrice ainsi que votre adresse : nom, adresse, ville et code postal. Pour l'expédition la calculatrice doit être correctement emballée dans un colis suffisamment résistant pour la protéger contre les chocs et les mauvais traitements éventuels et expédiée à l'adresse de Texas Instruments correspondant à votre pays: Voir la liste de ces adresses au dos de ce manuel.

REMEDES

Procédure incorrecte. Voir Section VII.

Partition incorrecte.

Une erreur de lecture a été détectée. Si les autres cartes sont lues normalement, regardez si la carte n'a pas été endommagée (Voyez page VII - 8). Si les autres cartes ne sont pas lues, utilisez une seule fois la carte de nettoyage de la tête de lecture à la page VII - 8.

Le programme placé en mémoire programme est protégé. Voyez le chapitre « Protection d'un programme » à la section VII.

Partition incorrecte.



SI VOUS AVEZ DES QUESTIONS A POSER OU BESOIN D'AIDE Si vous avez des questions à poser concernant la réparation de votre calculatrice, adressez-vous au service après-vente en vous référant aux adresses et numéros de téléphone situés au dos de ce manuel.

Si vous désirez obtenir un complément d'information sur votre calculatrice, adressez-vous au service d'assistance technique dont l'adresse se trouve également au dos de ce manuel.

En raison du grand nombre de suggestions qui parviennent à Texas Instruments contenant des idées nouvelles ou anciennes, Texas Instruments ne prendra en considération de telles suggestions que si celles-ci sont envoyées et données gratuitement à Texas Instruments. La politique de Texas Instruments est de refuser toute réception de suggestions confidentielles. Toutefois, si vous désirez développer vos suggestions avec Texas Instruments ou si vous désirez nous soumettre un programme que vous avez développé, veuillez inclure dans votre courrier le paragraphe qui suit :

"L'ensemble des informations ci-incluses est présenté à Texas Instruments à titre de suggestion et sans aucune obligation ni caractère confidentiel d'aucune sorte. Aucune relation confidentielle ou privilégiée n'est créée de ce fait avec Texas Instruments. Texas Instruments pourra donc utiliser, reproduire, dupliquer, publier, distribuer ou disposer de ces informations comme bon lui semblera sans qu'il me soit dû aucune compensation d'aucune sorte."

B

Conditions d'erreur



Un affichage clignotant indique que la limite de capacité de l'affichage de la calculatrice a été dépassée ou qu'une opération incomplète a été composée. En appuyant sur **CE** ou **CLR** vous arrêtez le clignotement. **CLR** efface également le contenu de l'affichage et les opérations en attente. **CE** arrête seulement le clignotement permettant ainsi de poursuivre les calculs avec la valeur affichée. L'affichage clignote pour les raisons suivantes:

1. Une introduction ou un résultat (sur l'affichage ou en mémoire) dépassent les limites de capacité de la calculatrice $\pm x \cdot 10^{-99}$ à $\pm 9.9999999 \times 10^{99}$. La limite en question clignote.
2. Fonction trigonométrique inverse ayant une valeur supérieure à ± 1 ; $\sin^{-1} x$ avec x supérieur à 1 par exemple, la valeur incorrecte de x clignote.
3. Puissance et logarithme d'un nombre négatif. La puissance ou le logarithme de la valeur absolue du nombre clignote.
4. Elever un nombre négatif à une puissance (ou extraction de racine). La puissance (ou la racine) de la valeur absolue clignote.
5. Appuyer sur deux touches d'opération à la suite, telles que $+$, $-$, \times , \div , γ^x et $\sqrt[\gamma]{\gamma}$. Le dernier nombre introduit clignote.
6. Appuyer sur **=** ou **)** après $+$, $-$, \times , \div , γ^x ou $\sqrt[\gamma]{\gamma}$ ou suivant un nombre précédé de **(**. Le dernier nombre introduit clignote.
7. Avoir plus de 9 ouvertures de parenthèses ou plus de 8 opérations en attente. La dixième parenthèse ou la neuvième opération n'est pas acceptée. Le dernier nombre introduit clignote.
8. Diviser un nombre par 0. "9.9999999 99" clignote.
9. Appeler une opération spéciale avec un code différent de 00 à 39.
10. Essayer de placer une astérisque (**09** 07) en dehors des limites 00 à 19. la valeur affichée clignote.
11. Essayer de faire une partition non admise par la capacité de la calculatrice.
12. Demander un transfert à une étiquette inexistante, ou à une adresse soit non définie par la partition soit en dehors des limites du programme.
13. Demander le transfert d'un programme de la bibliothèque dépassant la capacité laissée disponible par la partition.
14. Appeler un programme inexistant de la bibliothèque.
15. En calcul de régression linéaire et si la droite est parallèle à l'axe des y , essayer de calculer la pente, l'ordonnée à l'origine, le coefficient de corrélation, x' ou y' provoque le clignotement de l'affichage. Si la droite est parallèle à l'axe des x , l'affichage clignote si vous essayez de calculer x' ou le coefficient de corrélation.



- Essayer d'obtenir la pente, ordonnée à l'origine, coefficient de corrélation, x' ou y' pour moins de deux points introduits.
- Avoir plus de quatre opérations en attente pendant un calcul de régression linéaire, analyse de tendance ou calculs statistiques, ou encore pendant les conversions de coordonnées polaires ou cartésiennes ou de Degrés-Minutes-Secondes en Degrés décimaux.
- 0^{-x} et $^{-x}\sqrt{0}$ affichent "9.9999999 99" clignotant.
- Un angle dépassant la limite 10^{150} pour les conversions de coordonnées polaires ou cartésiennes.
- Arguments ne répondant pas aux limites ci-dessous.

Fonctions	Limites
$\sin^{-1}x, \cos^{-1}x$	$-1 \leq x \leq 1$
$\ln x, \log x$	$1 \times 10^{-99} \leq x < 1 \times 10^{100}$
e^x	$-227.9559242 \leq x \leq 230.2585092$
10^x	$-99 \leq x < 100$

- Les multiples impairs de $\pm 90^\circ$, $\pm \pi/2$ radians et ± 100 Grades sont des valeurs indéfinies pour la fonction tangente. Les petits multiples donnent une indication de dépassement de capacité ; les multiples plus importants fournissent des résultats incorrects sans pour autant provoquer une condition d'erreur. Voir Appendice C.
- Une lecture ou un enregistrement incorrect d'une carte magnétique provoque le clignotement de l'affichage.

ERREURS RENCONTREES AU COURS DE L'EXECUTION D'UN PROGRAMME

Lorsque l'une des erreurs citées plus haut est rencontrée dans un programme, ce qui se produit dépend de la façon dont a été rédigé le programme donc du programmeur. L'arrêt de l'exécution du programme n'est pas automatique après une mise en condition d'erreur, exception faite pour le cas de l'alinéa 12 ci-dessus. Le programme se poursuit en utilisant la valeur qui aurait clignoté si nous avions été en calcul manuel et la condition d'erreur n'apparaît qu'au moment où le programme s'arrête sur une «réponse» clignotante. Cette réponse peut être ou ne pas être correcte suivant le programme et le type de la condition d'erreur. Il y a toutefois une meilleure solution qui peut être utilisée sans pour autant augmenter trop le nombre d'instructions du programme. Si le programmeur le souhaite, il pourra demander à la calculatrice d'arrêter l'exécution du programme dès qu'une condition d'erreur est générée en levant le drapeau 8 ou en utilisant soit le test d'erreur soit les opérations spéciales OP 18 et 19.

C

Précision des résultats affichés



Une calculatrice, comme tout procédé électrique ou mécanique travaille avec un certain nombre de règles déterminées constituant ses limites.

Les limites mathématiques de la calculatrice sont liées au nombre de chiffres utilisables pour les calculs. La calculatrice semble utiliser 10 chiffres comme l'indique l'affichage ; en réalité 13 chiffres sont utilisés pour tous les calculs. Associés à l'arrondi interne 5/4, ces chiffres de garde sont conservés pour permettre d'accroître la précision des 10 chiffres affichés. Considérez l'exemple suivant en l'absence de chiffres de garde.

$$1/3 \times 3 = .9999999999 \text{ (inexact)}$$

Cet exemple montre que $1 \div 3 = .3333333333$, lorsqu'il est multiplié par 3 donne une réponse incorrecte, alors que la série des treize neufs sera arrondie à 1 lorsqu'elle sera placée dans les 10 chiffres de l'affichage.

Les fonctions mathématiques les plus complexes sont calculées par itérations. L'erreur cumulative de l'itération est généralement située au delà du dixième chiffre et ainsi son effet est invisible. La représentation d'un nombre sur 13 chiffres assure une précision du troisième ordre sur les chiffres affichés, c'est la raison pour laquelle l'affichage assure une restitution du résultat arrondi à 10 chiffres exacts.

Normalement il n'est pas nécessaire de prendre en considération ces chiffres de garde. Dans certains calculs cependant ces trois chiffres internes apparaissent comme une réponse alors qu'ils n'étaient pas attendus. Les limites mathématiques d'opérations finies telles que : longueur d'un mot, troncature au arrondi d'erreur ne permettent pas d'obtenir des chiffres de garde rigoureusement exacts. C'est pourquoi en soustrayant deux fonctions mathématiquement égales, la calculatrice n'affiche pas toujours zéro.

Exemple : $\sin 45^\circ - \cos 45^\circ \neq 0$.

Sélectionne le mode degrs

Appuyer

45 **2nd** **sin** **-**

45 **2nd** **cos**

=

Affichage

.7071067812

.7071067812

7.-13

Un résultat identique a été fourni pour $\sin 45^\circ$ et $\cos 45^\circ$ indiquant que les deux valeurs sont exactes tout au moins sur 10 chiffres. Le résultat final fait apparaître une différence sur le treizième chiffre. Ceci signifie que si un résultat est inférieur de 10^{-11} à 10^{-12} par rapport aux valeurs introduites ou aux résultats intermédiaires, ce résultat peut être considéré comme nul.

Ce principe est particulièrement important lorsque vous écrivez votre propre programme. Lorsque vous testez si un résultat de calcul est égal à une autre valeur en utilisant **≠**, des précautions doivent être prises pour éviter un branchement impropre dû à la présence des chiffres de garde. La séquence **EE** **INV** **EE** tronquera les chiffres de garde d'un résultat en ne conservant que la valeur arrondie affichée pour un usage ultérieur.



En affichage standard, les résultats sont exacts pour tous les calculs conformes aux restrictions données dans l'appendice B et aux limites indiquées ci-dessous.

FONCTIONS TRIGONOMETRIQUES – Tous les chiffres affichés en affichage standard sont exacts à $\pm 1 \times 10^{-9}$ pour des angles compris dans le domaine ± 36000 degrés, $\pm 200 \pi$ radians et ± 40.000 grades. Lorsqu'un argument atteint la limite $\pm 3,6 \times 10^{14}$ degrés ($\pm 2 \pi \times 10^{12}$ radians ou $\pm 4 \times 10^{14}$ grades) ou dépasse celle-ci, une rotation partielle n'est plus identifiable. En général, la précision décroît d'un caractère pour chaque décade supérieure à cette limite. Il y a une exception concernant la tangente : les angles multiples impairs de $\pm 90^\circ$, $\pm \pi/2$ radians ou ± 100 grades ont une tangente indéfinie; les petits multiples donnent une indication de dépassement de capacité, les multiples plus importants fournissent des résultats incorrects sans pour autant provoquer en condition d'erreur.

PUISSANCES ET RACINES – La précision peut diminuer si dans un calcul y est très voisin de 1 et la puissance x très grande. Par exemple $0.99999944^{-160000}$ est exact à 10^{-8} près; 0.99999944^{-400} est exact à 10^{-10} près soit sur la totalité de l'affichage standard.

D

Détection des anomalies d'un programme



DETECTION DES ANOMALIES D'UN PROGRAMME

Un programmeur même très vigilant se trouve de temps en temps face à face avec un programme qui ne se déroule pas correctement. Votre calculatrice dispose d'un certain nombre de touches permettant de faire des vérifications et des corrections. Le problème consiste à localiser l'erreur. Bien sûr, en premier lieu, vous pouvez vérifier que le programme ne comporte pas d'erreur d'introduction. Si vous n'avez rien trouvé, ce qui suit peut vous aider. Une liste des erreurs courantes de programmation vous est donnée, celle-ci peut vous permettre d'identifier votre erreur, sinon voyez le paragraphe **Diagnostic d'un programme** à la fin de ce chapitre.

L'imprimante optionnelle vous aidera très utilement dans ce cas pour lister les instructions ou pour effectuer une exécution pas à pas de votre programme.

CONSIDERATIONS DE BASE

Notation algébrique directe

La plupart des problèmes peuvent être introduits dans la calculatrice dans l'ordre de l'écriture, mais cela ne signifie pas qu'ils sont interprétés dans ce même ordre. La hiérarchie algébrique de la calculatrice donne à une expression du type $2 + 3 \times 6$ l'interprétation suivante $2 + (3 \times 6) = 20$ et non pas $(2 + 3) \times 6 = 30$.

Il faut se souvenir également que les fonctions d'une seule variable doivent suivre le nombre sur lequel elles opèrent. Ainsi $\sin \pi$ est calculé par la séquence [2nd] [PI] [2nd] [sin].

Touche d'égalité – [=]

L'instruction d'égalité déclenche l'exécution de toutes les opérations en attente, c'est pourquoi il faut l'utiliser avec prudence surtout dans un sous-programme.

L'oubli de cette instruction peut également créer des problèmes. Considérons l'expression $(2 \times 3)^2$: La séquence [2] [X] [3] [x²] effectue le calcul de 2×3^2 , la séquence correcte est dans ce cas [2] [X] [3] [=] [x²] ou encore [(] [2] [X] [3] [)] [x²].

Opérations en attente

Votre calculatrice peut emmagasiner jusqu'à neuf niveaux de parenthèses et huit opérations en attente. Toutefois certaines fonctions particulières emploient quatre opérations en attente. Les conversions de coordonnées polaires en cartésiennes ou de degrés, minutes, secondes en degrés décimaux font partie de ces fonctions au même titre que les fonctions statistiques. Une introduction qui dépasse cette limite est simplement ignorée provoquant le clignotement de l'affichage pour vous prévenir de ce qui s'est produit.

Étiquettes multiples

Une étiquette ne peut être utilisée qu'une seule fois dans un programme. Le mécanisme de recherche d'une étiquette s'effectue à partir du pas 000 et non pas à partir de l'endroit où elle est appelée. C'est pourquoi le transfert ne se ferait qu'à la première étiquette rencontrée, la seconde étiquette restant ignorée.



Sous-Programme

Vous n'utiliserez peut-être jamais les six niveaux de sous-programme ; néanmoins si vous appelez un sous programme depuis le sixième niveau, vous ne stockerez pas la nouvelle adresse de retour dans la mémoire des adresses de retour des sous programmes. Lorsque l'instruction **INV** **SBR** se présente au septième niveau de sous programme, le processus de retour s'effectue sur la sixième adresse de retour qui, bien sur, ne correspond pas à ce qui était attendu. Ce cas ne déclenche pas le clignotement de l'affichage et par conséquent vous ne serez pas prévenu du mauvais transfert effectué.

Ne perdez pas de vue qu'une touche utilisateur est interprétée comme un appel de sous programme sauf si elle est précédée d'une instruction de transfert telle que **GTO** ou **2nd** **SC=1**

Les conversions de coordonnées polaires en cartésiennes ou de degrés, minutes, secondes en degrés décimaux ainsi que les fonctions statistiques utilisent un niveau de sous programme:

Instruction **RST**

C'est une instruction très souvent employée , toutefois il ne faut pas oublier ses différentes fonctions pour éviter d'avoir des conséquences inattendues. **RST** a quatre fonctions : placer le pointeur au pas 000, baisser tous les drapeaux, effacer les registres des adresses de retour des sous programmes et ramener le pointeur en mémoire programme après avoir arrêté un programme de bibliothèque.

Fonctions Statistiques

Lorsque vous utilisez les fonctions statistiques préprogrammées les données sont cumulées dans les mémoires $M_1 - M_6$. C'est pourquoi un programme utilisant ces fonctions doit non seulement les laisser disponibles mais également prévoir de les effacer avant de démarrer le programme. Il convient également de préserver la valeur contenue dans le registre T si celui-ci doit être utilisé. Comme vous l'avons dit précédemment, ces fonctions utilisent un niveau de sous-programme et quatre niveaux d'opérations en attente.

Conversion de Coordonnées Polaires en Cartésiennes

Il faut se souvenir de sélectionner le mode angulaire correct et ne pas oublier que cette conversion utilise un niveau de sous-programme et quatre niveaux d'opérations en attente.

Sélection du mode Angulaire

A la mise en marche, la calculatrice est en mode degrés. Pour sélectionner les radians ou les grades, il convient de l'indiquer à la calculatrice avec les touches prévues à cet effet. La calculatrice garde un mode angulaire sélectionné tant qu'un autre mode n'a pas été sélectionné. Il n'y a pas de témoin indiquant le mode angulaire sélectionné.

Fonctions opérant uniquement sur l'affichage

EE et **2nd** **MS** opèrent uniquement sur le contenu de l'affichage et non pas sur le registre d'affichage. Ceci signifie que les chiffres internes de garde et ceux supprimés par un mode d'affichage en virgule fixe sont perdus lorsque vous utilisez ces fonctions.



Comparaison par rapport au registre T

2^{nd} **Calc** et 2^{nd} **Calc** comparent la totalité du contenu du registre d'affichage avec celui du registre T avant de décider si le branchement aura lieu ou non. Pour illustrer ceci prenez l'exemple suivant :

```

 $2^{nd}$  Deg
45  $2^{nd}$  sin
Ans
45  $2^{nd}$  Ans
 $2^{nd}$  Calc 114
  
```

Si vous passez en mode programmation, vous remarquez que le transfert ne se fait pas, bien que mathématiquement les deux résultats soient égaux. Ceci est dû à l'arrondi effectué sur les chiffres internes lorsque la calculatrice calcule le résultat (voir l'**appendice C**). Calculez vous-même $\sin 45^\circ$ moins $\cos 45^\circ$, vous vérifierez que le résultat est différent de zéro, faisant apparaître la différence des chiffres non affichés. Normalement vous ne pouvez identifier la conséquence de ceci, mais vous devez être vigilant dans l'emploi de ce type de transfert conditionnel. La séquence **EE INV EE** qui tronque les chiffres de garde d'un résultat, permet de conserver uniquement la valeur arrondie contenue dans l'affichage pour la suite des calculs.

Vous devez être prudent également lorsque vous utilisez les instructions **EE** ou 2^{nd} **DMS** pour lesquelles les chiffres de garde se trouvent éliminés.

Modifications dans un programme

Vous devez être prudent lorsque vous modifiez un programme ; même un simple changement qui semble sans importance peut perturber l'ensemble du programme. Il faut considérer toutes les conséquences possibles de façon à ce que tout reste dans l'ordre. Il faut être vigilant en ce qui concerne adresses contractées, les doubles étiquettes, les instructions regroupées (comme **INV SBR**) et les adresses qui peuvent être interprétées comme des codes de touches.

Souvenez-vous qu'ajouter ou supprimer une instruction entraîne le décalage d'un pas vers le haut ou vers le bas du reste du programme. Les instructions de transfert à une adresse numérique doivent être modifiées en conséquence.

Partition

Assurez-vous que les mémoires de données et la mémoire programme dont vous avez besoin correspondent à la partition actuelle de la machine. Une mémoire peut contenir 8 instructions, chaque couple de chiffres dans une mémoire peut être interprété comme une instruction. Aussi soyez prudent quand vous modifiez une partition ; veillez à ce que le contenu d'une mémoire ne soit pas interprété comme 8 instructions de programme ou inversement.



DIAGNOSTIC D'UN PROGRAMME

Étudiez ce qui suit avec attention, vous y trouverez des conseils utiles pour diagnostiquer vos programmes. Le but est de vous fournir des principes de recherche vous permettant de trouver la raison pour laquelle un programme ne fonctionne pas correctement.

Un programme ne s'arrête plus

Si un programme ne s'arrête pas comme prévu, c'est souvent parce qu'il reste bloqué dans une boucle. La meilleure solution consiste alors à vérifier les instructions une à une et particulièrement celles qui régissent les transferts. Souvent les instructions de transfert sont en cause. Il est bon de vérifier en premier lieu les instructions de transfert inconditionnel. Faites attention aux séquences du type `2nd` `IM` `D` ... `GTO` `D` pour lesquelles une instruction de transfert conditionnel est indispensable si vous voulez sortir de la boucle ainsi créée. Examinez le transfert conditionnel le plus proche surtout lorsqu'il commande le début ou la fin d'une boucle.

L'instruction `2nd` `0%` ne doit pas entraîner le programme dans une boucle infinie à moins que celui-ci ne soit venu modifier le contenu de la mémoire décrétementée ou que cette valeur soit $\geq 10^{10}$. Assurez-vous que le contenu de cette mémoire puisse atteindre zéro. Si le nombre décrétementé est trop important, le programme peut prendre un temps exceptionnel pour s'achever et c'est seulement à ce moment que vous vous apercevrez que le programme bouclait. Si le nombre de boucles désiré est calculé par programme, vérifiez celui-ci.

Les transferts conditionnels réalisés par comparaison avec le contenu du registre T doivent être examinés avec attention. Généralement, lorsque vous utilisez ce type de transfert pour sortir d'une boucle c'est que vous pensez que vos calculs convergent vers une valeur déterminée. Naturellement, si ces calculs ne convergent pas, la boucle ne s'arrête jamais. Regardez aussi bien votre expression mathématique que la séquence utilisée pour la calculer. Assurez-vous également que le registre T contient la bonne valeur, si celle-ci est calculée dans le programme, vérifiez la séquence correspondante. N'oubliez pas en plus que cette instruction compare la totalité du contenu du registre T à la totalité du contenu de l'affichage avant de décider du transfert.

Si vous utilisez un programme d'une bibliothèque à titre de sous-programme et que l'exécution semble perdue dans une boucle, appuyez sur `RST` pour arrêter son déroulement. Toutefois l'emploi de `RST` ne doit être qu'une mesure d'urgence utilisée en dernier recours du fait que l'endroit où le programme s'arrête est imprévisible et en conséquence la valeur affichée est non significative.

Si vous n'avez pas trouvé de solution à votre problème dans ce qui vient d'être énoncé, voyez le chapitre Utilisation de la calculatrice en diagnostic à la fin de ce chapitre.



Mêmes données et résultats différents

Ce genre de situation est souvent provoqué par un transfert conditionnel qui, utilisé de façon incorrecte, donne une fois un résultat correct et la fois suivante un résultat faux. A titre d'exemple considérez ce qui suit :

```

2nd [L1]
A
2nd [x=1]
B
2nd [St I/q]
3
2nd [L1]
B
.
.
.
2nd [I/q]
3
C
.
.
.
2nd [L1]
C
.
.
.

```

Ici, le programme doit sauter une portion du programme si la donnée introduite est plus petite que zéro. Regardons ce qui se passe lorsque l'on introduit successivement dans le programme 12, -16, 12. Le programme se déroule normalement pour les deux premières introductions; à la troisième le résultat est incorrect. La raison en est que l'introduction de -16 a levé le drapeau 3 et comme rien n'était prévu pour baisser à nouveau ce drapeau avant l'introduction suivante, celle-ci est considérée par le programme comme étant négative. Cette situation peut être corrigée en intercalant la séquence `INV 2nd [St I/q] 3` après l'instruction `2nd [L1] A`.

Un problème analogue peut se produire avec une opération de transfert. Malheureusement il n'est pas possible de donner un exemple pour chaque cas. En règle générale cependant, la première chose à faire lorsque vous diagnostiquez un problème de ce type est d'identifier la nature de la réponse. Dans l'exemple précédent seules les introductions positives fournissent des réponses erronées et ceci seulement après avoir introduit une valeur négative. Ce genre de constatations peut vous aider à identifier le problème.



D

Bien sûr toutes les erreurs de ce type ne proviennent pas forcément d'une opération de transfert. Prenons l'exemple suivant qui restitue des résultats différents pour des introductions successives analogues.

```
2nd |bl|
A
.
.
.
SBR
SUM
.
.
.
2nd |bl|
SUM
SUM
1 | 2
.
.
.
INV | SBR
```

Ici le problème est dû à une mauvaise utilisation d'une mémoire. Si M_{12} n'est jamais effacée, ou si la première exécution n'utilise pas l'instruction `STO|`, un cumul systématique modifiera les résultats obtenus.

Comme indiqué précédemment, la nature des résultats vous permet d'identifier ce type de problème. Si vous n'avez pas trouvé la solution à votre problème, voyez le chapitre *Utilisation de la calculatrice en diagnostic* à la fin de ce chapitre.



Dans la plupart des cas de telles situations peuvent être évitées en plaçant dans chaque programme une séquence d'initialisation du type indiqué ci-dessous. Son exécution se fait en appuyant sur la touche **E**.

Séquence de touches	Commentaires
R/S	Termine une séquence (pas 000)
2nd 1/x E	
2nd CM	Efface les mémoires de données
CLR	Efface les registres de calcul
2nd CP	Efface le contenu du registre T
INV 2nd Fix	Retourne en virgule flottante
RST	Abaisse tous les drapeaux
	Efface les registres d'adresses de retour des sous-programmes
	Place le pointeur au pas 000

Résultats régulièrement faux

Il est possible qu'un programme fournisse régulièrement des réponses erronées; regardez alors si la résolution du problème n'est pas traitée de façon incorrecte. Si, ayant vérifié manuellement l'équation vous êtes certain de sa validité dans tous les cas et que vous ne découvrez aucune erreur d'instruction alors reportez-vous à ce qui suit.

Utilisation de la calculatrice en diagnostic

Une fois déterminé les valeurs qui doivent être calculées et affichées et l'endroit exact où elles doivent être stockées aux différents stades du programme, votre calculatrice est alors le meilleur outil pour examiner le mauvais fonctionnement d'un programme.

Précaution importante : lorsque vous rappelez le contenu d'une mémoire pour l'examiner à un endroit déterminé du programme, assurez-vous de ne pas perdre pour autant la valeur contenue dans le registre d'affichage. Sinon la suite de l'examen serait totalement inutile. Une solution consiste à utiliser l'instruction **2nd** **Exc** pour étudier le contenu d'une mémoire de façon à pouvoir restituer les conditions initiales en réutilisant cette même fonction, la valeur contenue dans le registre d'affichage se trouve ainsi correctement remise en place.

Il y a plusieurs instructions que vous pouvez utiliser pour analyser un programme pendant son déroulement. En intercalant l'instruction **R/S** à plusieurs endroits dans un programme vous pouvez déterminer à partir de quel moment une erreur se produit en examinant, séquence après séquence, si les valeurs contenues tant dans le registre d'affichage que dans les mémoires sont correctes.

Pour insérer l'instruction **R/S**, il est plus pratique de commencer par la fin du programme et de revenir de proche en proche jusqu'au début. L'insertion en redescendant dans la mémoire programme permet d'éviter de chercher une instruction qui aurait été décalée par une insertion précédente.



Dès qu'une erreur est découverte, faites une nouvelle exécution du programme; arrêtez celle-ci sur l'instruction **R/S** précédent celle où l'erreur a été détectée puis utilisez la touche **SST** pour continuer l'exécution du programme pas à pas jusqu'à ce que vous ayez identifié l'endroit exact où se trouve l'erreur. Une fois le problème identifié et l'erreur corrigée, supprimez les instructions **R/S** en commençant par le début du programme. Utilisez cette méthode tant que tous les problèmes n'auront pas été identifiés et corrigés.

L'utilisation de **SST** peut être remplacée par le maintien enfoncé de la touche **GTO** qui a pour effet de ménager un temps d'arrêt après chaque exécution, vous permettant d'avoir un aperçu bref mais automatique sur le déroulement du programme. Pour utiliser cette particularité vous devez commencer par démarrer le programme en suivant les consignes indiquées ci-dessous :

1. Appuyez et maintenez appuyée la touche **R/S** .
2. Tout en maintenant **R/S** , appuyez et maintenez appuyée la touche **GTO** .
3. Relâchez la touche **R/S** .

Les problèmes les plus simples à identifier sont ceux qui provoquent une condition d'erreur. Il suffit simplement de lever le drapeau 8 et de réexécuter le programme qui automatiquement s'arrêtera dès qu'une condition d'erreur sera rencontrée. (L'exécution s'arrête sur le pas suivant celui qui provoque la condition d'erreur). Il reste à identifier le type de l'erreur rencontrée et faire la correction nécessaire.

Note : il n'est pas possible de demander une exécution pas à pas dans un programme de bibliothèque. Lorsque vous explorez pas à pas la mémoire programme en utilisant **SST** ou **GTO** et qu'un programme d'une bibliothèque se trouve appelé, l'affichage disparaît pendant son exécution et réapparaît dès que le retour au programme principal est effectué. (Si vous utilisez **GTO** , ne relâchez pas cette touche pendant qu'une exécution est en cours dans la bibliothèque, elle pourrait vous faire sauter l'inspection d'une séquence du programme principal).

Utilisation de l'imprimante pour effectuer un diagnostic.

L'imprimante optionnelle constitue un support efficace pour diagnostiquer un programme. Une fois votre calculatrice connectée au berceau imprimant, vous pouvez obtenir :

- (1) En appuyant sur **RST** **2nd** **List** , une liste complète des instructions de votre programme comportant le numéro du pas de programme, le code de l'instruction et le symbole de l'instruction, instruction par instruction. Ceci vous évite la tâche fastidieuse d'interprétation des différents codes fournis par l'affichage de la calculatrice lorsque vous contrôlez l'exactitude des instructions introduites.
- (2) L'exécution instruction par instruction de votre programme en mode "Trace". Ainsi vous pouvez suivre pas à pas une séquence de calculs et découvrir exactement l'endroit où se trouve le problème cherché.
- (3) La liste du contenu des mémoires à partir de la mémoire xx en appuyant sur **XX** **INV** **2nd** **List** . Exécutée à différents endroits, cette séquence permet de vérifier les valeurs contenues dans les différentes mémoires.
- (4) La liste des étiquettes utilisées dans un programme avec le numéro du pas où elles sont utilisées en appuyant sur **RST** **2nd** **09** **08**.



A	
Adaptateur/Chargeur	A-1
Adressage	
Absolu	IV-44, 86, V-57
Forme réduite	IV-15, 44, V-22, 58
Indirect	IV-84, V-68
Mémoire	II-6
Programme	IV-44, V-57
Adressage contracté	IV-15, 44, V-22, 58
Adressage indirect	
des Mémoires	IV-84, V-68
du Programme	IV-86, V-68
Affichage	
Caractéristiques	II-8, V-1, 5, 44
Clignotement	V-9, B-1
Contrôle	II-8, V-8
Dépassement de capacité	B-1
Mode Programmation	V-17
Registre d’Affichage	II-8, V-5
Standard	II-8, V-1
Analyse de tendance	V-39
Antilogarithme décimal	II-11, V-16
Arcosinus	II-12, V-18
Arithmétique directe en mémoire	II-7, V-24
Arctangente	II-12, V-18
Arcsinus	II-12, V-18
Arrondi	C-1
Assistance	A-4
B	
Batteries	A-1
Bibliothèque de base	I-3, III-3
Boucles programmées	IV-68, V-64
Conditionnelles	IV-70
DSZ	IV-71
Inconditionnelles	IV-68
Branchement	V-62
C	
Calculs angulaires	II-12, V-16
Calculs au clavier	I-4, II-2
Caractères numériques non affichés	V-5, C-1
Caractéristiques de la Calculatrice	I-1
Carte de nettoyage	
du galet d’entraînement	VII-8
de la tête magnétique	VII-9
Cartes magnétiques	VII-1
Charge du bloc accumulateur	A-1
Clavier codé	V-49
Codes	
contractés	IV-17, V-51
C	
des touches	IV-17, V-44, 48
des opérations spéciales	V-27
Conditions d’erreurs	V-67, B-1, D-1
Conversions	
Coordonnées Polaires/Coordonnées cartésiennes	II-14, V-30
Degrés, Minutes, secondes / Degrés décimaux	II-13, V-30
Degrés / Radians / Grades	V-19
De températures	I-4
Heures, Minutes, Secondes / Heures décimales	II-13, V-30
Correction d’un programme	IV-21, V-48, 51
D	
Décharge des batteries	A-1
Décision dans un programme	IV-57
Degrés décimaux / Degrés Minutes Secondes	II-13, V-30
Diagnostic	
de la Bibliothèque	A-4
de la calculatrice	A-3
Drapeaux	IV-61, V-65
Adressage indirect	IV-81
Fonctions spéciales	IV-65, V-67
E	
Écart type	V-33
Effacement d’une condition d’erreur	B-1
Enregistrer une carte magnétique	VII-2
Entretien et Service	A-1
Étiquettes	
Ordinaires	IV-43, V-56
Touche utilisateur	IV-11, V-55
Exoés de capacité, Affichage	B-1
Exécuter un programme	V-46
Exécution du programme pas à pas	D-8
Exemple de racine carrée	IV-59
Exposant	V-5
Puissance de 10	II-8
F	
Fonctions algébriques	II-10, V-15
Fonctions arithmétiques	II-2, V-10
Fonctions trigonométriques	II-12, V-17
Limites	C-2
Format de l’affichage	II-8, V-5
H	
Hiérarchie Algébrique	II-3, V-11



I	
Impression sélective	VI-2
Imprimante	VI-1
Caractéristiques	VI-1
Listage	VI-4
Opérations	VI-1
Précautions	VI-12
Tracé d'une courbe par points	VI-10
Incrémentation/Décrémentation des mémoires	V-29
Index des touches	Couverture
Informations de service	A-1
Instruction DSZ	IV-57, 71, V-63
Instructions d'expédition	A-4
Interprétation des symboles imprimés	VI-5
Introduction de la puissance de 10	II-8, V-5
Introduction d'un programme	IV-16
Inverse des Fonctions Trigonométriques	II-12, V-18
Isolement de la partie entière	V-20
J	
Jeu du Code Secret	VI-101
Jeu du nombre mystérieux	I-3
L	
Lecture d'une carte magnétique	VII-5
Listage	
du contenu des mémoires	VI-4
du contenu du programme	VI-4
des étiquettes	VI-11
Logarithme	
décimal	II-11, V-16
népérien	II-11, V-16
Logiciel	III-1
M	
Mantisse	II-8, V-5
Marche/Arrêt	I-2
Mécanisme de la programmation	IV-10
Mémoires de données	
Adressage	II-6, V-22
Adressage indirect	IV-84, V-68
Incrémentat. / décrémentation	V-29
Listage du contenu des mémoires	VI-4
nombre de mémoires	II-6, V-22, 42
Mémoire programme	V-41
Module préprogrammé enfichable	I-3, III-1
Moyenne	V-33
Multiplication en mémoire	II-7, V-24
Multiplication implicite	II-4, V-13
N	
Nettoyage de la tête d'impression	VI-12
Nombres négatifs	V-1, 2
Notation	
algébrique directe	II-3, V-11
ingénieur	II-9, V-8
scientifique	II-8, V-5
O	
Opérations alphanumériques	VI-7
Opérations comportant un nombre répété	V-15
Opérations de base	II-2
Opérations de contrôle spéciales	V-27, VI-7
Opérations de test	V-29, 62
Opérations en attente	II-3, V-11
Optimisation d'un programme	IV-89
Organigramme	IV-4
P	
Partition	V-22, 29, 42
Pas de programme, pointeur	V-44
Passage d'une carte magnétique	VII-2
Point d'interrogation imprimé	VI-2, 5
Pointeur	IV-7, V-44, 47
Pondération statistique	V-34
Précision	C-1
Première fonction d'une touche	II-5, IV-17, V-3
Programmation	IV-1
avancée	IV-43
élémentaire	IV-2
langage	IV-1
mécanismes	IV-10
opérations de base	IV-1, V-41
pas de programme	IV-10
Programme	IV-1
affichage	IV-17, V-44
avancement du pointeur	V-47
bibliothèque de programmes	I-3, II-1
conditions d'erreur	B-2, D-1
correction	IV-21, V-48, 51
drapeaux dans un programme	IV-61
écriture d'un programme	I-4
effacement	IV-16, V-3, 43
exécution	VI-46
introduction	IV-16, V-45
listage	IV-4
mémoire de programme	V-42
module de programme	III-1
optimisation	IV-89



P (suite)

organigramme	IV-4
pas	IV-17, V-44
personnalisation	IV-27
protection	VII-4
sous-programme	IV-46
trace	D-8
transfert (instructions)	IV-44
transfert	III-4, V-28

Programme

de calcul de temps écoulé	IV-18
de calcul du coût d'exploitation	IV-93
de calcul d'une factorielle	IV-72, V-65
de calcul d'un investissement	IV-27
de chiffrage de contrôle de prix	IV-32
de conversion de coordonnées sphériques	IV-38
de conversions métriques	IV-65
de l'équation du second degré	IV-79
d'intérêts composés	IV-75
du biorythme	IV-53

Programme principal

Programme pré-enregistré (bibliothèque)	I-3, III-1
Puissance	II-10, V-21
limites	C-2

R

Racines	II-10, V-21
limites	C-2

Recherche des problèmes (En cas de difficultés)

sur les opérations	A-3
dans un programme	D-1

Registre

d'affichage	II-8, V-5
arithmétiques	II-7, V-24
d'adresses de retour des sous-progr.	IV-46, V-58
de la calculatrice	I-1, V-22
d'impression	VI-8
mémoire de données	II-6, V-22
partition	V-22
T	II-14, IV-57, V-62

Registre T

comparaison avec	II-15, IV-57
	V-62

Régression linéaire	II-17, V-36
Remplacement du bloc accumulateur	A-2

S

Secante	V-18
Signe moins flottant	II-8, V-1
Sommation en mémoire	II-7, V-24
Sous-programmes	
accès	IV-48

S (suite)

programme de la bibliothèque utilisé en	
sous-programmes	IV-52, V-60
registres d'adresses de retour	IV-46, V-58
remarques	IV-49
touche de sous-programme	IV-46, V-58
Soustraction en mémoire	II-7, V-24
Statistiques	V-32
Symboles imprimés	VI-5, 7

T

Tableau des codes de touches	V-29
------------------------------	------

Touche

addition	II-2, V-10
adressage indirect	IV-86, V-68
antilogarithme décimal	II-11, V-16
d'avance papier	VI-3
d'avance pas à pas	IV-21, V-48
d'avancement de papier	VI-3
carré	II-10, V-10
de changement de signe	II-2, 8, V-2
Cosecante	V-18
cosinus	II-12, V-17
cotangente	V-18
décimale fixe	II-9, V-8
décrément et saut sur zéro	IV-71, V-63
D.MS	II-13
division	II-2, V-10
division en mémoire	II-7, V-24
10 puissance x	II-11, V-16
à double fonction	II-5, V-3
e puissance x	II-11, V-16
échange	II-6, V-26
d'effacement	V-3
de la dernière introduction	II-2, V-3, 15
des mémoires de données	II-6, V-23
de la mémoire programme	IV-16, V-3, 41, 43
d'effacement d'une instruction	IV-21, V-51, 52
égalité	II-2, V-10
d'enregistrement	VII-2, 5
«Go To» (aller A)	IV-44, V-56
grades	II-12, V-16
insertion	IV-21, V-52
Inverse	II-5, V-3, 18
inverse d'une	II-10, V-15
lever d'un drapeau	IV-61, V-65
listage	VI-4
logarithme décimal	II-11, V-16
mémoires	II-6, V-23
mode degrés	II-12, V-16
mode programmation	I-4, IV-8, V-43, 44



T (suite)

moyenne	V-33
multiplication	II-2, V-10
numériques	II-2, V-2
opération blanche	IV-99, V-51
parenthèses	II-4, V-12
partie entière	II-8, V-10
pause	V-44
Pi (π)	II-2, V-2
+/-	II-2, V-2
point décimal	II-2, V-2
produit en mémoire	II-7, V-24
racine xième de y	II-10, V-21
radian	II-12, 16
rappel mémoire	II-6, V-23
«Reset»	V-44
retour pas à pas	IV-21, V-48
seconde	II-5, V-3
seconde fonction	II-5, IV-17, V-3
sigma plus	V-32
sinus	II-12, V-17
de sommation en mémoire	II-7, V-24
de soustraction	II-2, V-10
de soustraction en mémoire	II-7, V-24
stockage en mémoire	II-6, V-23
tangente	II-12, V-17
test de l'état d'un drapeau	IV-61, V-65
test $x = t$	IV-57, 60, V-62
test $x \geq t$	IV-57, 60, V-62
trace	V1-5
utilisateur	IV-11, V-57
valeur absolue	V-20
x au carré	II-10, V-20
x échange t.	II-14, IV-57, V-30, 32, 62
$x = t$	IV-57, 60, V-62
$x \geq t$	IV-57, 60, V-62
y puissance x	II-10, V-21
Tracé d'une courbe point par point	VI-10
Transfert	IV-43, V-56
conditionnel	IV-57, V-62
inconditionnel	IV-44, V-56
indirect	IV-86, V-58
Transfert d'un programme de la bibliothèque	III-4, V-28
Type d'opérations	I-2

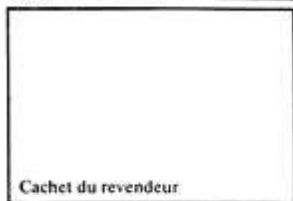
V

Variables dans un programme	IV-2
Variance	V-33
Vérification d'un programme	IV-21, V-48, 51
Virgule flottante	II-8, V-2

TEXAS INSTRUMENTS

JUSTIFICATIF DE LA DATE D'ACHAT

Date _____



Modèle **TI-58** _____ N° de série _____

Modèle **TI-58C** _____ N° de série _____

Modèle **TI-59** _____ N° de série _____ 1 Monsieur 2 Mme, Melle 3 Société

Nom _____ Prénom _____

Société _____ Adresse _____

Ville _____ Code postal _____ Pays _____

Description de la panne _____



CARTE DE GARANTIE
CALCULATRICES

TEXAS INSTRUMENTS



MODELE : _____ DATE : _____

NOM : _____ PRENOM : _____ 1 Mr.

2 Mme.
Melle

ADRESSE : _____

PAYS : _____ AGE: 1 2 3 4 5

6-12 13-18 19-26 27-45 45+

A CETTE CALCULATRICE :

- 1 Est un cadeau 2 A été achetée par votre entreprise 3 A été achetée par vous-même

B VOTRE DECISION D'ACHAT EST MOTIVEE PAR : (Cochez 2 cases)

- 1 Le prix 2 La performance 3 La présentation
4 La marque 5 Une recommandation 6 La publicité. Spécifier

C OU A-T-ELLE ETE ACHETEE ?

- 1 Supermarché 2 Grand magasin 3 Par correspondance
4 Bijoutier 5 Magasin equip. bureau 6 Papeterie
7 Magasin Radio/TV 8 Groupement d'achat ou association 9 Autres. Spécifier

D QUELLE EST VOTRE PROFESSION ?

- 1 Ecolier, lycéen, étudiant 2 Enseignant 3 Femme au foyer
4 Ouvrier spécialisé 5 Employé 6 Vendeur, marketing
7 Tech., engin., agent tech. 8 Directeur 9 Patron d'entreprise profession libérale
10 Autres. Spécifier

E POSSEDEZ-VOUS UNE AUTRE CALCULATRICE ?

- 1 Non Modèle : _____
2 Oui Marque : _____

F INDIQUER LES DEUX APPLICATIONS PRINCIPALES DE VOTRE CALCULATRICE :

- 1 Utilisation personnelle 2 Loisirs, sport, distractions 3 Ventes, marketing
4 Comptabilité, usage au bureau 5 Banque, finance assurance 6 Calculs tech. et scientifiques
7 Math, statistiques 8 Autres

Garantie contractuelle limitée à un an

La garantie légale des vices cachés ou défauts des marchandises vendues, s'applique en tout état de cause.

La TI-58/58C/59 de Texas Instruments est garantie pièces et main-d'œuvre au premier acheteur pour une durée d'un an à partir de la date d'achat pour des conditions d'utilisation normales. Sont exclus de cette garantie tous dommages causés par une fuite éventuelle des piles.

La garantie est nulle si :

1. La calculatrice a été endommagée par accident ou utilisation abusive, par négligence, par réparation impropre, ou tout autre cause ne trouvant pas son origine dans les pièces détachées ou leur assemblage;
2. Le numéro de série a été modifié ou effacé.

TEXAS INSTRUMENTS NE SAURAIT ETRE TENUE POUR RESPONSABLE DES PERTES DE JOUISSANCE CONSECUTIVES A UNE PANNE DE LA CALCULATRICE ET/OU TOUT AUTRE DOMMAGE INDIRECT SUBI PAR L'ACHETEUR.

Pendant la période de garantie, la calculatrice ou ses pièces défectueuses seront réparées, ajustées et/ou remplacées par une calculatrice refabriquée d'une qualité équivalente, marquée "RECONDITIONNEE", au choix du fabricant, gratuitement lorsque la calculatrice aura été renvoyée à Texas Instruments franco de port et assurée, accompagnée d'une justification de la date d'achat.

TOUTE CALCULATRICE RENVOYEE SANS JUSTIFICATION DE LA DATE D'ACHAT SERA REPARÉE AU COÛT DE REPARATION EN VIGUEUR AU MOMENT DU RETOUR.

En cas de remplacement par une calculatrice refabriquée, cette dernière bénéficiera de la poursuite de la garantie contractuelle initialement accordée au modèle acheté. Cette garantie contractuelle ne sera en aucun cas inférieure à 90 jours.

IMPORTANT : Avant d'effectuer toute expédition pour réparation, il est recommandé de relire avec soin dans ce manuel les instructions relatives au service et à l'expédition.

SERVICE CENTERS

BELGIË-BELGIQUE

Mercure Centre
Rue de la Fusée/Raketstraat 100
(Parallèle av. Leopold III)
1130 Brussel/Bruxelles
Tel. (2) 7208000

CANADA

41 Shelley Road
Richmond Hill, Ontario
Tel. (416) 8897373

DANMARK

Marielundvej 46E
2730 Herlev
Tel. (02) 917400

DEUTSCHLAND

Haggerty Straße 1
8050 - Freising
Tel. (08161) 801

ENGLAND

Manton Lane
Bedford, MK41 7PU
Tel. (0234) 67466

ESPAÑA

Carretera Antigua a Barcelona
K. M. 23. 100
Apartado de Correos 98
Torrejón de Ardoz - Madrid
Tel. 6755300 - 6755350

FRANCE

B. P. 28
06021 - Nice Cedex
Tel. (93) 200101

ITALIA

Cassella Postale 1
02015 - Cittaducale
Tel. (746) 69034/35/36

NEDERLAND

Postbus 43
Kolthofsingel 8
7600 AA Almelo
Tel. (05490) 63967

NORGE

Kr. Augusts gt. 13
Oslo 1
Tel. (02) 206040

ÖSTERREICH

Rennweg 17
1030 - Wien
Tel. (0222) 724186

PORTUGAL

Dept ECD
R. Eng. Frederico Ulrich 2650
Moreira da Maia
4470 - Maia
Tel. (02) 9481003

SCHWEIZ-SUISSE

Aargauerstrasse 250
CH 8048 - Zürich
Tel. (01) 643455/56

SUOMI FINLAND

Elimäenkatu 14-16
P.L. 53
00511 Helsinki 51
Tel. (80) 7013133

SVERIGE

Norra Hamnvägen 3
Fack
10054 Stockholm 39
Tel. (08) 235480

TEXAS INSTRUMENTS