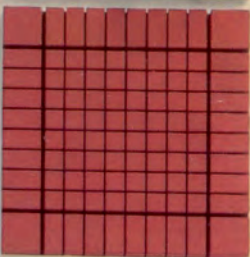
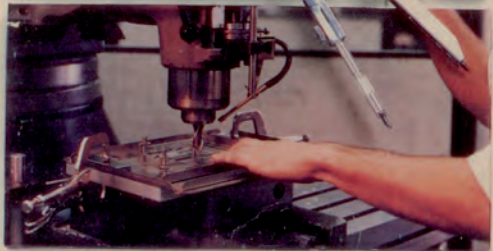
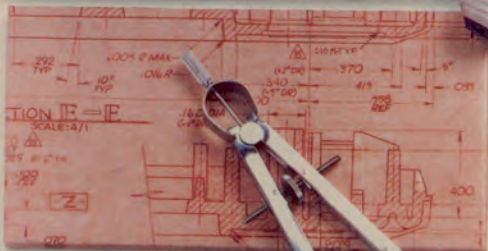


HEWLETT-PACKARD

HP-11C

BEDIENUNGS- UND
PROGRAMMIERHANDBUCH



Bemerkung

Hewlett-Packard übernimmt keinerlei Haftung für die Richtigkeit der in diesem Handbuch enthaltenen Programm-Materialien und Tastenfolgen. Die Programm-Materialien und Tastenfolgen dienen lediglich zur beispielhaften Illustration; der Benutzer trägt das volle Risiko hinsichtlich der Güte und der Durchführbarkeit. Sollten sich Programm-Materialien oder Tastenfolgen als fehlerhaft erweisen, so hat der Benutzer die gesamten Kosten für die entstandenen Schäden, die Folgeschäden und die notwendigen Korrekturen zu tragen. Hewlett-Packard haftet nicht für Schäden oder Folgeschäden, die sich im Zusammenhang mit der Verwendung von Tastenfolgen oder Programm-Materialien ergeben können.

VERKAUFS-NIEDERLASSUNGEN:

Hewlett-Packard GmbH:

- 6000 Frankfurt 56, Berner Straße 117, Postfach 560 140, Tel. (0611) 50 04-1
7030 Böblingen, Herrenberger Straße 110, Tel. (07031) 667-1
4000 Düsseldorf 11, Emanuel-Leutze-Straße 1 (Seestern), Tel. (0211) 59 71-1
2000 Hamburg 60, Kapstadtring 5, Tel. (040) 6 38 04-1
8028 Taufkirchen, Eschenstraße 5, Tel. (089) 61 17-1
3000 Hannover 91, Am Großmarkt 6, Tel. (0511) 46 60 01
8500 Nürnberg, Neumeyerstraße 90, Tel. (0911) 52 20 83/87
1000 Berlin 30, Keithstraße 2-4, Tel. (030) 24 90 86
6800 Mannheim, Roßlauer Weg 2-4, Tel. (0621) 7 00 50
7910 Neu-Ulm, Messerschmittstraße 7, Tel. (0731) 7 02 41

Hewlett-Packard (Schweiz) AG:

Allmend 2, CH-8967 Widen, Tel. (057) 5 01 11

Hewlett-Packard Ges. m. b. H., für Österreich/für sozialistische Staaten:

Wagramerstraße-Liebgassee, A-1220 Wien

Hewlett-Packard S.A., Europa-Zentrale:

7, rue du Bois-du-Lan, Postfach, CH-1217 Meyrin 2-Genf, Schweiz

SERVICE-NIEDERLASSUNGEN:

Hewlett-Packard GmbH:

6000 Frankfurt 56, Bernerstraße 117, Postfach 560 140, Tel. (0611) 50 04-1

Hewlett-Packard (Schweiz) AG:

Allmend 2, CH-8967 Widen, Tel. (057) 5 01 11

Hewlett-Packard Ges. m. b. H., für Österreich/für sozialistische Staaten:

Wagramerstraße-Liebgassee, A-1220



HEWLETT
PACKARD

HP-11 C

BEDIENUNGS- UND PROGRAMMIERHANDBUCH

Juni 1981

00011-90002

Printed in Singapore

© Hewlett-Packard Company 1981

Einleitung

Herzlichen Glückwunsch! Ihre Wahl des Taschenrechners HP-11C mit Permanent Speicher zeigt uns, daß Sie Wert auf Qualität, Vielseitigkeit und einfache Handhabung legen. Dieses Handbuch beschreibt die Möglichkeiten des HP-11C und soll Ihnen dabei helfen, die Ihnen unbekannt Details so schnell wie möglich zu erlernen.

Grundlagen, Programmierung, Anwendungsprogramme.

Das Handbuch HP-11C besteht aus drei Teilen. Die Teile I und II beschreiben die Bedienung des Tastenfelds und die Grundlagen der Programmierung; die Inhalte dieser Abschnitte dürften Ihnen vertraut sein, wenn Sie bereits mit anderen HP-Taschenrechnern gearbeitet haben. Der Teil III enthält eine Reihe von Anwendungsprogrammen sowie einige zusätzliche Informationen über die grundlegendsten Programmierpraktiken. Bevor Sie jedoch mit dem Lesen der einzelnen Abschnitte dieses Handbuchs beginnen, sollten Sie zunächst einige Erfahrungen im Umgang mit Ihrem HP-11C sammeln, indem Sie die im Abschnitt „Der HP-11C – Ein Problemlöser“ auf Seite 9 dargestellten, einführenden Materialien durcharbeiten.

Zusammenfassungen der Programmierungsmöglichkeiten.

Am Anfang jedes Abschnitts in Teil II (Abschnitt 5 bis 9) befindet sich eine kurze Diskussion der in dem jeweiligen Abschnitt beschriebenen Bedienungscharakteristika des Rechners. Wenn Sie bereits andere HP-Taschenrechner programmiert haben, dürfte es ausreichen, in jedem Abschnitt nur diese Zusammenfassung zu lesen. Wenn Sie dagegen das erste Mal mit einem HP-Taschenrechner in Berührung kommen, dürfte es für Sie von Interesse sein, auch die restlichen Materialien jedes Abschnitts durcharbeiten, in denen ausgewählte Schwerpunkte durch zusätzliche Erläuterungen und/oder Beispiele vertieft werden.

Weitere Anwendungsprogramme.

Bei vielen autorisierten Hewlett-Packard-Händlern ist zusätzlich das HP-11C Solutions Handbook erhältlich, das eine umfassende Sammlung von HP-11C Programmen aus den Bereichen Mathematik, Statistik, Ingenieurwissenschaften, Betriebswirtschaft und Spiele enthält.

Hinweis: Das HP-11C Solutions Handbook ist nur in *englischer* Sprache erhältlich.

Inhaltsverzeichnis

Einleitung	2
Tastatur und Permanentspeicher des HP-11 C	8
Der HP-11 C: Ein Problemlöser	9
Manuelle Lösungen	10
Programmierte Lösungen	12
TEIL I: Grundlagen des HP-11 C	15
Abschnitt 1: Erste Schritte	16
Ein- und Ausschalten des Geräts	16
Anzeige	16
Dezimal- und Gruppentrennzeichen	16
Statusanzeigen	16
Negative Zahlen	17
Löschen der Anzeige: CLx und ←	17
Running	18
Overflow und Underflow	18
Fehlermeldungen	19
Speicher	19
Permanentspeicher	19
Löschen des Speichers	20
Bedienung des Tastenfelds	20
Primär- und Alternativfunktionen	20
Löschen von Vorwahltasten	21
Funktionen einer Variablen	21
Funktionen von zwei Variablen	22
Abschnitt 2: Automatischer Speicherstack, LAST X, und Datenspeicherung.	26
Automatischer Speicherstack und Stackmanipulation	26
Funktionen zur Stackmanipulation	27
Rechnerfunktionen und Stack	29
Funktionen von zwei Variablen	29
Kettenrechnung	31
LAST X	33
Konstantenarithmetik	34
Speicherregister-Operationen	37
Speichern von Zahlen	37

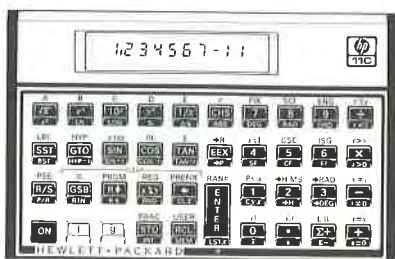
Rückruf von Zahlen	38
Übungen zum Speichern und Rückrufen von Zahlen	38
Löschen der Datenspeicher-Register	38
Speicherregister-Arithmetik	39
Übungen zur Speicherregister-Arithmetik	40
Übungsaufgaben	40
Abschnitt 3: Numerische Funktionen	42
Pi	42
Funktionen zur Zahlenmanipulation	42
Funktionen einer Variablen	43
Allgemeine Funktionen	43
Trigonometrische Operationen	44
Trigonometrische Funktionen	45
Zeit- und Winkelkonvertierung	46
Logarithmische Funktionen	48
Hyperbolische Funktionen	48
Funktionen von zwei Variablen	49
Potenzrechnung	49
Prozentrechnung	49
Transformation von Polar- und Rechteckskordinaten	51
Wahrscheinlichkeitsrechnung	52
Statistische Funktionen	54
Zufallszahlengenerator	54
Summationen	55
Korrektur fehlerhafter Eingaben	58
Mittelwert	60
Standardabweichung	61
Lineare Regression	63
Linearer Schätzwert und Korrelationskoeffizient	64
Abschnitt 4: Kontrolle der Anzeige	67
Wahl des Anzeigeformats	67
Festkommaformat	67
Wissenschaftliches Anzeigeformat	68
Technisches Anzeigeformat	70
Eingabe von Exponenten	71
Rundung auf der zehnten Stelle	72
TEIL II: Programmierung des HP-11 C	73
Abschnitt 5: Grundlagen der Programmierung	74
Was ist ein Programm?	74
Notwendigkeit von Programmen	74

Programmkontrolle	74
Automatische Speicherumwandlung	74
MEM	77
Tastencodes und Zeilennummern	77
Abgekürzte Tastenfolgen	78
Funktionen zur Programmkontrolle	78
User Mode	79
Programmspeicher	80
Interpretation von Tastencodes	81
Operationen zur Programmierung	83
Anfang und Ende eines Programms	84
Das fertige Programm	85
Laden eines Programms	85
Ausführen eines Programms	86
Verwendung des User Mode	88
Unterbrechung der Programmausführung	88
Geplantes Anhalten der Programmausführung	89
Pausen während der Programmausführung	91
Programmunterbrechung durch Fehler	93
Labels	94
Übungsaufgaben	94
Abschnitt 6: Programmkorrektur	96
Suchen von Programmfehlern	96
Korrekturfunktionen	97
Beispiel zur Programmkorrektur	98
Zeilenweise Ausführung eines Programms	99
Verwendung von SST und BST im	
Program Mode	100
Modifizieren eines Programms	101
Übungsaufgaben	107
Abschnitt 7: Programmverzweigungen	110
Vergleichsoperationen	110
Flags	111
Programmsteuerung	112
Go To	112
Sprünge und Programmschleifen	112
Verwendung von Flags	116
Abschnitt 8: Unterprogramme	119
Aufruf von Unterprogrammen	119
Einschränkungen bei der Verwendung von	
Unterprogrammen	120
Benutzung von Unterprogrammen	121

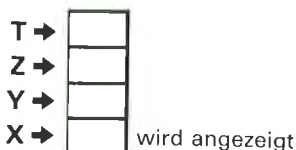
Abschnitt 9: Das Indexregister	127
Direkte Indexregister-Funktionen	127
Indirekte Indexregister-Funktionen	130
Beispiel für eine Schleifensteuerung mit ISG	132
Direkte R_1 Operationen	135
Indirekte R_1 Operationen	135
Indirekte Programmkontrolle	136
Indirekte Labeladressierung und indirekter Aufruf von Unterprogrammen	136
Indirekte Zeilennummer-Adressierung	137
TEIL III: Programmierte Problemlösungen	139
Abschnitt 10: Anwendungsprogramme	140
Matrizenrechnung	140
Lineare Gleichungssysteme mit 3 Unbekannten	149
Nullstellenbestimmung mit dem Newton-Verfahren	154
Numerische Integration	159
Kurvenanpassung	162
Dreiecksberechnungen	169
T-Test	176
Chiquadrat-Test	182
Finanzen: Jährliche Fälligkeiten und aufgelaufene Zinsen	185
U-Boot-Jagd	194
Abschnitt 11: Programmiertechniken	206
Struktur	206
Das Problemstatement	206
Der Algorithmus	206
Flußdiagramme	208
Unterprogramme	210
ISG mit RCL (i)	211
Dateneingabe	213
Programmschleifen	214
Flags	216
Zufallszahlen	217
Benutzer-definierbare Tasten	218
Datenspeicherung	218
Selektion von Routinen	219
Anhang A: Fehlerbedingungen	220
Anhang B: Stack Lift und LAST X	222
Abschluß der Zifferneingabe	222

Stack Lift	222
Disabling Operationen	222
Enabling Operationen	223
Neutrale Operationen	223
LAST X	224
Anhang C: Arbeitsweise der automatischen Speicherumwandlung	225
Umwandlung von Speicherregistern in Programmspeicher	225
Umwandlung von Programmspeicher in Speicherregister	228
Verwendung von MEM	228
Anhang D: Batterien, Gewährleistung und Serviceinformationen	230
Batterien	230
Anzeichen von Spannungsabfall	231
Einbau neuer Batterien	231
Funktionsprüfung	233
Gewährleistung	235
Änderungsverpflichtung	236
Gewährleistungsinformation	236
Service	237
Reparatur in den USA	237
Reparatur in Europa	237
Internationaler Service	238
Reparaturkosten	239
Garantie bei Reparaturen	239
Versandanweisungen	239
Sonstiges	240
Benutzerberatung	240
Bestellinformationen	240
Temperaturbereich	240
Index: Programmieretechniken	241
Index: Funktionstasten	243
Index: Programmtasten	247
Sachindex	249

HP-11C – Der programmierbare, wissenschaftliche Taschenrechner mit Permanentenspeicher



AUTOMATISCHER SPEICHERSTACK



LAST X



PROGRAMMSPEICHER

Permanent Konvertierbar Permanent

SPEICHERREGISTER

Konvertierbar

000-	064-
001-	065-
002-	066-
...	...
061-	201-
062-	202-
063-	203-

Permanent	Konvertierbar
R ₁ <input type="text"/>	R ₀ <input type="text"/> R ₀ <input type="text"/>
	R ₁ <input type="text"/> R ₁ <input type="text"/>
	R ₂ <input type="text"/> R ₂ <input type="text"/>
	R ₃ <input type="text"/> R ₃ <input type="text"/>
	R ₄ <input type="text"/> R ₄ <input type="text"/>
	R ₅ <input type="text"/> R ₅ <input type="text"/>
	R ₆ <input type="text"/> R ₆ <input type="text"/>
	R ₇ <input type="text"/> R ₇ <input type="text"/>
	R ₈ <input type="text"/> R ₈ <input type="text"/>
	R ₉ <input type="text"/> R ₉ <input type="text"/>

Die Standard-Programmspeicher/Speicherregister-Aufteilung besteht aus 63 Programmzeilen und 20 Datenspeicherregistern, sowie dem Indexregister (R₁). Der Rechner wandelt automatisch ein Datenspeicherregister in sieben Zeilen Programmspeicher um, wenn dies erforderlich ist. Es wird jeweils ein Register umgewandelt. Die Umwandlung beginnt mit R₉ und endet mit R₀.

Der HP-11C: Ein Problemlöser

Der programmierbare, wissenschaftliche Taschenrechner HP-11C ist ein äußerst leistungsfähiges Instrument, mit dem Sie jederzeit einfache oder komplexe Problemstellungen lösen oder wichtige Daten speichern können. Der HP-11C ist sehr einfach zu programmieren und zu bedienen, so daß keinerlei Programmiererfahrung und Kenntnisse von Programmiersprachen notwendig sind.

Der HP-11C hilft Ihnen Energie zu sparen, indem er seine Anzeige automatisch abschaltet, wenn länger als 8 bis 17 Minuten keine Eingabe erfolgt ist. Trotzdem gehen keinerlei Daten verloren – jede eingetastete Information bleibt durch den Permanentenspeicher erhalten.

Wir sind anders! Ihr Hewlett-Packard-Taschenrechner verwendet eine einzigartige Operationslogik, die durch die **ENTER** Taste verkörpert wird und sich wesentlich von der in anderen Taschenrechnern benutzten Logik unterscheidet. Die Vorteile der HP-Taschenrechnerlogik werden beim Gebrauch des Rechners deutlich. Später wird die **ENTER** Taste und die HP-11C-Logik im Detail dargestellt werden; doch zunächst wollen wir Sie mit **ENTER** vertraut machen, indem wir Ihnen zeigen, wie einfach Berechnungen auf dem HP-11C durchzuführen sind.

Ergebnisse erscheinen sofort nachdem Sie eine numerische Funktionstaste gedrückt haben. Dies sei, zum Beispiel, anhand der arithmetischen Funktionen erläutert. Die **+** Taste addiert die letzte Eingabe zu der Zahl, die sich bereits im Rechner befindet, die **-** Taste subtrahiert die letzte Eingabe; durch die **×** Taste wird die letzte Eingabe mit dem Rechnerinhalt multipliziert, und die **÷** Taste dividiert durch die letzte Eingabe. Bei all diesen Operationen müssen sich die beiden Zahlen zuvor im Rechner befinden. Dies geschieht wie folgt: Sie geben die erste Zahl, drücken **ENTER** um die erste Zahl von der zweiten zu trennen, und geben danach die zweite Zahl ein. Anschließend können Sie **+**, **-**, **×** oder **÷** drücken.

Um ein Gefühl für Ihren neuen Rechner zu bekommen, schalten Sie die Anzeige (das Display) durch Drücken der **ON** Taste ein. Wenn Ziffern ungleich Null erscheinen, drücken Sie **←** um die Anzeige auf **0,0000** zu setzen. Erscheinen keine vier Nullen auf der rechten Seite des Dezimalpunkts, drücken Sie **f** **FIX** 4. Nun sollte Ihre Anzeige mit der in den folgenden Beispielen übereinstimmen. Sämtliche in diesem Handbuch illustrierten Anzeigen haben das Format **FIX** 4, sofern anderes nicht explizit vermerkt ist.



0,0000

Manuelle Lösungen

Es ist nicht notwendig, den Rechner zwischen einzelnen Problemstellungen zu löschen. Sollten Sie bei der Eingabe der einzelnen Ziffern einer Zahl einen Fehler machen, drücken Sie **←** und geben Sie die korrekte Ziffer ein.

Problem	Eingaben	Anzeige
$9 + 6 = 15$	9 ENTER 6 +	15,0000
$9 - 6 = 3$	9 ENTER 6 -	3,0000
$9 \times 6 = 54$	9 ENTER 6 ×	54,0000
$9 \div 6 = 1,5$	9 ENTER 6 ÷	1,5000

Beachten Sie, daß in den vier Beispielen:

- Beide Zahlen im Rechner sind, bevor **+**, **-**, **×** oder **÷** gedrückt wird.
- **ENTER** nur zur Trennung der beiden nacheinander einzugebenden Zahlen benutzt wird.
- Das Auslösen einer numerischen Funktionstaste, in diesem Fall **+**, **-**, **×** oder **÷**, die sofortige Ausführung der Funktion und die Anzeige des Ergebnisses verursacht.

Um die enge Beziehung zwischen einer manuellen und einer programmierten Problemstellung zu verdeutlichen, soll zunächst die Lösung eines Problems manuell, d. h. über das Tastenfeld, berechnet werden. Danach werden wir ein Programm zur Lösung des gleichen und ähnlicher Probleme benutzen.

Die meisten konventionellen Warmwasserbereiter besitzen eine zylindrische Form. Der Wärmeverlust eines solchen Geräts lässt sich leicht über die Formel

$$q = h \times A \times T$$

berechnen, wobei

q der Wärmeverlust des Boilers (in Kalorien pro Stunde).

h der Wärmeübergangskoeffizient.

A die Gesamtoberfläche des Zylinders in m^2 .

T der Temperaturunterschied zwischen dem Zylinder und der umgebenden Luft in Grad Celsius.

Beispiel: Angenommen Sie besitzen einen zylinderförmigen, 200 Liter fassenden Warmwasserbereiter und Sie wollen bestimmen wieviel Energie wegen unzureichender Isolierung verloren geht. Bei ersten Messungen stellen Sie einen durchschnittlichen Temperaturunterschied zwischen Behälteroberfläche und umgebender Luft von 10 Grad Celsius fest. Die Oberfläche ist $3 m^2$ groß, und der Wärmeübergangskoeffizient beträgt ungefähr 1,78. Um den Wärmeverlust des Geräts zu berechnen, sind die folgenden Tasten in der angegebenen Reihenfolge zu drücken.



Tastenfolge

10
3

\times
1,78

\times

\leftarrow

Anzeige

10,0000
3

30,0000
1,78

53,4000

0,0000

Eingabe Temperaturunterschied (T) und Oberfläche (A).

Berechnung $A \times T$.

Eingabe Wärmeübergangskoeffizient (h).

Wärmeverlust in Kalorien pro Stunde ($h \times (AT)$).

Anzeige löschen.

Programmierte Lösungen

Der Wärmeverlust des Warmwasserbereiters im vorangegangenen Beispiel wurde für ein Temperaturunterschied von 10° Celsius berechnet. Angenommen, Sie wollen nun den Wärmeverlust für mehrere Temperaturunterschiede berechnen. Theoretisch könnten Sie jede Wärmeverlust-Berechnung manuell durchführen. Eine einfachere und schnellere Methode ist es jedoch, ein Programm zu schreiben, das den Wärmeverlust für beliebige Temperaturunterschiede berechnet.

Schreiben des Programms. Das Programm besteht aus der gleichen Tastenfolge wie bei der manuellen Lösung des Problems. Zwei zusätzliche Instruktionen, ein Label und ein Return (Rücksprung) werden benötigt, um den Anfang und das Ende des Programms zu definieren.

Laden des Programms. Um die Programmbefehle in den HP-11C zu laden, müssen Sie die folgenden Tasten in der angegebenen Reihenfolge drücken. Der Rechner speichert die einzelnen Befehle bei der Eingabe. (Im Display werden Informationen angezeigt, die für Sie erst später von Interesse sind und im Moment ignoriert werden können.)

Tastensequenz

[g] [P/R]

Anzeige

000-

Rechner auf Program Mode schalten.
(Program Statusanzeige erscheint.)

[f] CLEAR [PRGM]

000-

Löschen des Programmspeichers

[f] [LBL] [A]

001-42. 21. 11

Label „A“ definiert den Programmstart.

3

[X]

1

[.]

7

8

[X]

[g] [RTN]

002-

003-

004-

005-

006-

007-

008-

009-

3

20

1

48

7

8

20

43 32

Eingaben wie bei der manuellen Lösung des Problems.

„Return“ definiert das Ende des Programms.

g P/R

0,0000

Rechner wieder auf Run Mode schalten. (Programm Statusanzeige erlischt.)

Starten des Programms. Um das Programm zu starten, müssen Sie die folgenden Tasten drücken.

Tastenfolge

Anzeige

10

10

Erster Temperaturunterschied.

f A

53,4000

Der zuvor manuell berechnete Wärmeverlust.

12 f A

64,0800

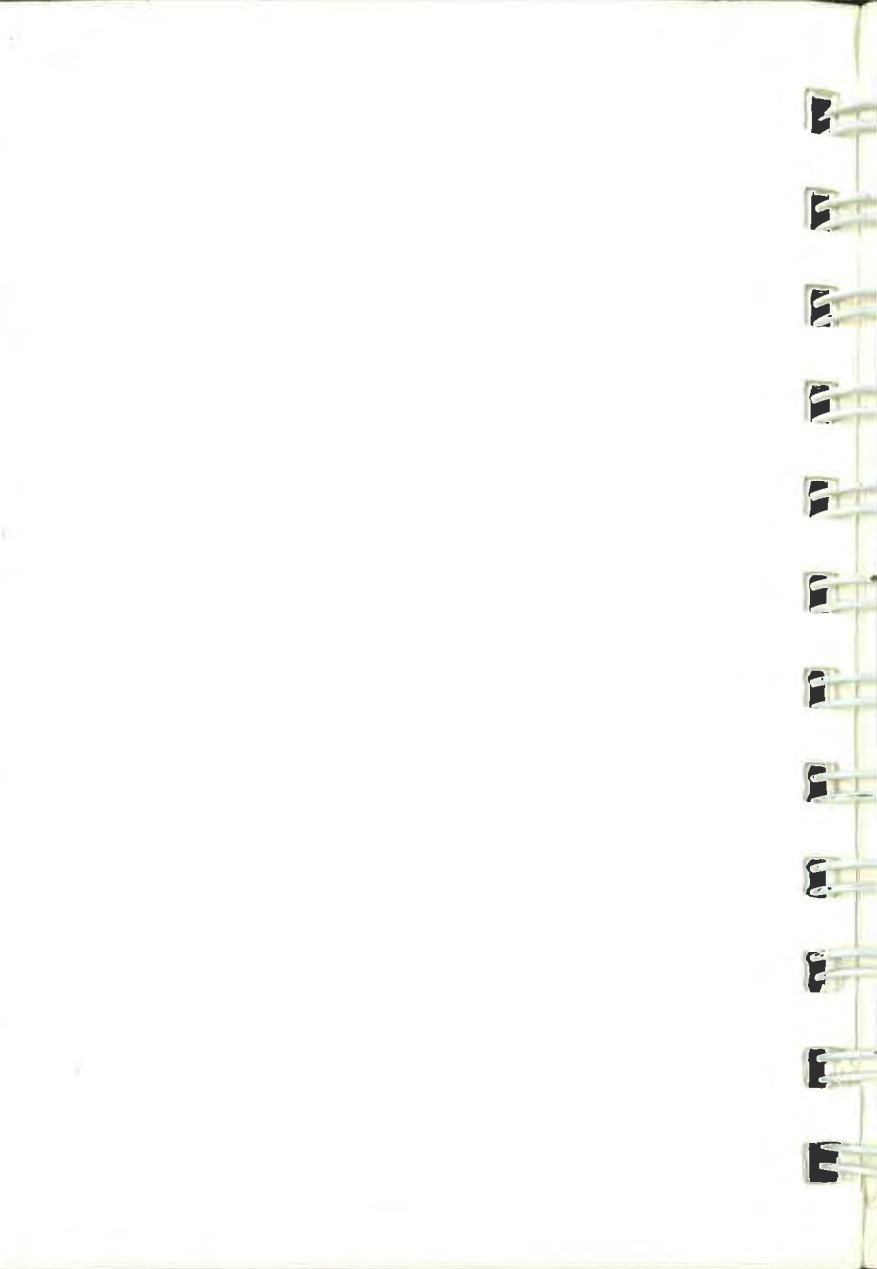
Wärmeverlust für einen Temperaturunterschied von 12° Celsius.

Mit dem geladenen Programm läßt sich nun sehr schnell der Wärmeverlust für beliebig viele Temperaturunterschiede berechnen. Sie brauchen lediglich den gewünschten Temperaturunterschied einzugeben und f A zu drücken. Vervollständigen Sie beispielsweise die rechtsstehende Tabelle.

Temperaturunterschied	Wärmeverlust
10	?
12	?
14	?
16	?
18	?
20	?

Als Ergebnisse sollten Sie erhalten 53,4000, 64,0800, 74,7600, 85,4400, 96,1200 und 106,8000.

Sie sehen, Programmieren ist sehr einfach! Der Rechner erinnert sich an eine Reihe von Tastenfolgen und führt sie aus, wenn Sie es wünschen. Nachdem Sie nun einige Erfahrung im Umgang mit Ihrem HP-11C gewonnen haben, lassen Sie uns einen Blick auf einige wichtige Details bei der Bedienung des Rechners werfen.



TEIL I

Grundlagen des HP-11C

Erste Schritte

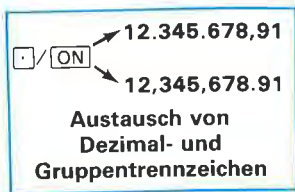
Ein- und Ausschalten des Geräts

Mit der **[ON]** Taste wird die Anzeige des HP-11C ein- und ausgeschaltet. Um Energie zu sparen, schaltet sich der HP-11C automatisch ab, wenn in einem Zeitraum von 8 bis 17 Minuten keine Eingabe erfolgt ist (Time-Out).

Anzeige

Dezimal- und Gruppentrennzeichen

Durch das Dezimaltrennzeichen wird der ganzzahlige und der dezimale Anteil einer Zahl unterschieden. Mit dem Gruppentrennzeichen können die Ziffern einer großen Zahl zu Gruppen zusammengefaßt werden. In manchen Ländern ist das Dezimaltrennzeichen der Dezimalpunkt und das Gruppentrennzeichen das Komma, während für andere Länder gerade die Umkehrung gilt. Um die Dezimal-/Gruppentrennzeichen-Konvention auf Ihrem HP-11C zu verändern, müssen Sie den Rechner ausschalten, die **[.]** Taste drücken, den Rechner wieder einschalten und die **[.]** Taste loslassen (**[.]**/**[ON]**).



In den Programmbeispielen und Illustrationen dieses Handbuchs wird der Punkt als Gruppentrennzeichen und das Komma als Dezimaltrennzeichen benutzt. Wir empfehlen Ihnen, diese Konvention beim Durcharbeiten des Handbuchs auch für Ihren Rechner zu benutzen. Bitte beachten Sie, daß zur Eingabe des Dezimalkommas ebenfalls die Taste **[.]** zu verwenden ist.

Statusanzeigen

Die Anzeige Ihres HP-11C enthält fünf Statusanzeigen, die Ihnen den Zustand des Rechners während bestimmter Operationen mitteilen. Die Statusanzeigen werden zusammen mit den zugehörigen Operationen in den jeweiligen Abschnitten dieses Handbuchs beschrieben.

USER f g GRAD PRGM

Statusanzeigen

Negative Zahlen

Um eine angezeigte Zahl, die entweder neu eingegeben wurde oder aus einer vorangegangenen Berechnung resultiert, mit einem negativen Vorzeichen zu versehen, ist die Taste $\boxed{\text{CHS}}$ (*change sign*) zu betätigen. Zeigt das Display bereits eine negative Zahl, d. h. vor der Zahl steht ein Minuszeichen, so wird durch Drücken von $\boxed{\text{CHS}}$ das Minuszeichen entfernt und die Zahl somit positiv.

Löschen der Anzeige: $\boxed{\text{CLx}}$ und $\boxed{\leftarrow}$

Die Anzeige des HP-11C kann mit zwei Befehlen gelöscht werden, mit $\boxed{\text{CLx}}$ (*Clear X*) und $\boxed{\leftarrow}$ (*Pfeil rückwärts*). Wenn $\boxed{\text{g}} \boxed{\text{CLx}}$ im Run Mode gedrückt wird, so wird jede angezeigte Zahl auf Null gesetzt. Das Betätigen von $\boxed{\text{g}} \boxed{\text{CLx}}$ im Program Mode beinhaltet die Speicherung dieser Anweisung als Programmbefehl. $\boxed{\leftarrow}$ ist eine nicht-programmierbare Funktion mit der Sie das Display sowohl im Run Mode als auch im Program Mode auf die folgende Weise löschen können:

1. Im Run Mode:

- A. Wird *nach* der Ausführung einer Funktion $\boxed{\leftarrow}$ gedrückt, so werden alle Ziffern der in der Anzeige angezeigten Zahl auf Null gesetzt.* (Die Ausführung einer HP-11C-Funktion beendet die Zifferneingabe – d. h. dem Rechner wird mitgeteilt, daß die Zahl in der Anzeige vollständig ist – und verursacht, daß $\boxed{\leftarrow}$ auf der gesamten Zahl wirkt.)

Tastenfolge

12345

$\boxed{\sqrt{x}}$

$\boxed{\leftarrow}$

Anzeige

12.345

111,1081

0,0000

Durch Drücken von $\boxed{\leftarrow}$ nach Ausführung einer Funktion werden alle Ziffern in der Anzeige auf Null gesetzt.

- B. Wenn Sie bei der Eingabe einer Zahl $\boxed{\leftarrow}$ *vor* der Ausführung einer Funktion drücken (d. h. vor der Beendigung der Zifferneingabe), wird die letzte eingegebene Ziffer gelöscht. Nachdem Sie eine oder mehrere Ziffern gelöscht haben, können Sie neue (Ersatz-)Ziffern eingeben.

* Eventuell vorhandene, nachlaufende Ziffern der im Display angezeigten Zahl können intern gespeichert sein und werden ebenfalls gelöscht.

Tastensequenz

12345



9


Anzeige


12.345

1.234



123

1.239

Wenn die Zifferneingabe nicht abgeschlossen wurde, wirkt  auf jede Ziffer einzeln.

2. Im Program Mode bewirkt das Drücken von  die Löschung des momentan angezeigten Programmbefehls.

Running

Während der Ausführung eines Programms oder der Ausführung von  (*Permutation*)  (*Kombination*) blinkt „running“ auf der Anzeige.



running

Overflow und Underflow

Overflow. Wenn das Ergebnis einer Berechnung im angezeigten X-Register eine Zahl mit einer Wertigkeit größer als $9,999999999 \times 10^{99}$ ist, werden alle Neunen und das korrekte Vorzeichen angezeigt. Tritt ein Overflow in einem ablaufenden Programm auf, so wird die Ausführung unterbrochen und die Overflow-Anzeige erscheint.



9,999999 99

Overflow-Anzeige

Underflow. Ist das Ergebnis einer Berechnung eine Zahl mit einer Wertigkeit kleiner als $1,000000000 \times 10^{-99}$, so wird diese Zahl durch Null ersetzt. Durch einen Underflow wird die Ausführung eines laufenden Programms nicht unterbrochen.

Fehlermeldungen

Wenn Sie eine Berechnung mit einem unzulässigen Parameter durchführen – z. B. die Quadratwurzel einer negativen Zahl ziehen – erscheint eine Fehlermeldung auf der Anzeige im Display.

Eine vollständige Auflistung aller Fehlermeldungen und ihrer Ursachen ist in Anhang A, Fehlerbedingungen, zu finden.

Tastenfolge

4 \square CHS

\square \sqrt{x}

\square

Anzeige

–4

Error 0

–4,0000

Zum Löschen einer Fehlermeldung ist \square oder eine sonstige Taste zu betätigen. Anschließend kann der normale Rechnerbetrieb fortgesetzt werden.

Speicher

Permanentspeicher

Durch den Permanentspeicher Ihres HP-11C bleiben die folgenden Informationen selbst nach Abschalten der Anzeige erhalten:


- Sämtliche im Rechner gespeicherten numerischen Daten.
- Sämtliche im Rechner gespeicherten Programme.
- Momentanes Anzeigeformat.
- Gesetzte Flags.
- Aktuelle Position des Rechners im Programmspeicher.
- Nicht abgearbeitete Unterprogramm-Rücksprünge.
- Momentaner trigonometrischer Mode (Altgrad, Bogenmaß oder Neugrad).

Nach dem Einschalten befindet sich der HP-11C immer im Run Mode (PRGM Statusanzeige gelöscht), selbst wenn beim Ausschalten der Rechner im Program Mode (PRGM Statusanzeige an) war.

Der Permanentenspeicher bleibt auch nach dem Entfernen der Batterien noch kurze Zeit erhalten. Dadurch können Sie erschöpfte Batterien austauschen, ohne die gespeicherten Daten und Programme zu zerstören.


Löschen des Speichers

Der Permanentenspeicher des HP-11C kann jederzeit wie folgt initialisiert (total gelöscht) werden:

1. Schalten Sie den HP-11C aus.
2. Drücken Sie die  Taste und lassen Sie sie gedrückt.
3. Schalten Sie den HP-11C ein.



**Permanentenspeicher
initialisieren**

Beim Initialisieren des Permanentenspeichers erscheint die rechts abgebildete Fehlermeldung. Durch Drücken von  (oder sonst einer Taste) wird die Meldung gelöscht.

Pr Error

Bemerkung: Der Permanentenspeicher kann unkontrolliert gelöscht werden, wenn der Rechner zu Boden fällt oder auf andere Art und Weise beschädigt wird, bzw. wenn die Stromzufuhr unterbrochen wird.

Bedienung des Tastenfelds

Primär- und Alternativfunktionen

Fast alle Tasten Ihres HP-11C beinhalten eine Primär- und zwei Alternativfunktionen. Die Primärfunktion wird durch den Text auf der Oberseite der Taste beschrieben. Die beiden Alternativfunktionen werden durch die Symbole auf der Vorderseite der Taste bzw. oberhalb der Taste angezeigt.



- Um die in goldfarbenen Buchstaben oberhalb der Taste angezeigte Alternativfunktion anzuwählen, ist zunächst die goldfarbene Vorwahltaste \boxed{f} und anschließend die eigentliche Funktionstaste zu drücken, d. h. hier $\boxed{f} \boxed{\pi}$.
- Um die Primärfunktion auf der Oberfläche der Taste anzuwählen, braucht nur die Taste selbst betätigt werden, d. h. hier \boxed{CHS} .
- Um die in blauen Buchstaben auf der Vorderseite der Taste angezeigte Alternativfunktion anzuwählen, ist zunächst die blaue Vorwahltaste \boxed{g} und anschließend die eigentliche Funktionstaste zu drücken, d. h. hier $\boxed{g} \boxed{ABS}$.

Beachten Sie, daß beim Drücken der Vorwahltasten \boxed{f} oder \boxed{g} die **f** oder **g** Statusanzeige erscheint, \boxed{f} . Die Statusanzeige erlischt, wenn eine Funktionstaste betätigt wird, um eine Eingabesequenz abzuschließen.

0,0000

Löschen von Vorwahltasten

Wenn Sie beim Tasten der Vorwahl für eine Funktion einen Fehler machen, drücken Sie \boxed{f} CLEAR \boxed{PREFIX} , um den Fehler zu annullieren. (CLEAR \boxed{PREFIX} löscht ebenfalls die \boxed{STO} , \boxed{RCL} , \boxed{GTO} , \boxed{GSB} , \boxed{HYP} und $\boxed{HYP^{-1}}$ Tasten.) Da die \boxed{PREFIX} Taste auch zur Anzeige der Gesamtmantisse einer Zahl in der Anzeige benutzt werden kann, erscheinen nach Drücken von \boxed{PREFIX} für einen Moment alle zehn Ziffern der Zahl.

Funktionen einer Variablen

Als Funktionen einer Variablen werden diejenigen numerischen Funktionen bezeichnet, die zur Ausführung der Operation nur eine Zahl (ein Argument) benötigen. Die Funktionen einer Variablen sind wie folgt zu benutzen:

1. Geben Sie die Zahl ein (falls nicht bereits in der Anzeige).
2. Drücken Sie die Funktionstaste(n).

Tastenfolge	Anzeige
45	45
$\boxed{9}$ $\boxed{\text{LOG}}$	1,6532

Funktionen von zwei Variablen

Bei einer Funktion zweier Variablen müssen zwei Zahlen im Rechner sein, bevor die Funktion ausgeführt werden kann. $\boxed{+}$, $\boxed{-}$, $\boxed{\times}$ und $\boxed{\div}$ sind Beispiele für Funktionen von zwei Variablen.

Die $\boxed{\text{ENTER}}$ Taste. Wenn sich eine der beiden Zahlen, die Sie für eine Funktion zweier Variablen benötigen, bereits als Ergebnis einer vorangegangenen Funktionsauswertung im Rechner befindet, brauchen Sie die $\boxed{\text{ENTER}}$ Taste nicht. Müssen Sie jedoch zwei Zahlen vor einer Funktionsausführung eingeben, so sind diese beiden Werte durch die $\boxed{\text{ENTER}}$ Taste zu trennen.

Geben Sie zum Beispiel zwei Zahlen in den Rechner ein und lösen Sie dann eine Funktion zweier Variablen aus, etwa $2 \div 3$:

1. Geben Sie die erste Zahl ein.
2. Drücken Sie $\boxed{\text{ENTER}}$, um die erste Zahl von der zweiten zu trennen.
3. Geben Sie die zweite Zahl ein.
4. Drücken Sie die Funktionstaste(n).

Tastenfolge	Anzeige
2	2
$\boxed{\text{ENTER}}$	2,0000
3	3
$\boxed{\div}$	0,6667

Reihenfolge der Eingabe. Wie aus den Grundregeln der Arithmetik bekannt ist, hat eine Vertauschung der Reihenfolge der Zahlen bei einer Addition oder Multiplikation keinen Einfluß auf das Ergebnis. Bei einer Subtraktion oder Division dagegen muß die Zahl, die Sie subtrahieren oder mit der Sie dividieren wollen, immer die als *zweite* eingetragene Zahl sein.

Zu lösen	Tastenfolge	Anzeige
$10 - 3$	10 $\boxed{\text{ENTER}}$ 3 $\boxed{-}$	7,0000
$3 - 10$	3 $\boxed{\text{ENTER}}$ 10 $\boxed{-}$	-7,0000
$10 \div 3$	10 $\boxed{\text{ENTER}}$ 3 $\boxed{\div}$	3,3333
$3 \div 10$	3 $\boxed{\text{ENTER}}$ 10 $\boxed{\div}$	0,3000

Wenn Sie mit den anderen Funktionen zweier Variabler (wie etwa \sqrt{x}) Ihres HP-11C arbeiten, beachten Sie, daß die auf der Taste mit x bezeichnete Zahl immer zuletzt einzugeben ist. Zur Berechnung des Werts von 2 hoch 3 ist beispielsweise zunächst die Zahl 2 einzugeben, dann ENTER zu drücken, dann der Exponent 3 einzugeben und schließlich $f \sqrt{x}$ zu drücken.

Tastenfolge	Anzeige	
2 ENTER	2,0000	
3	3	3 ist der x -Wert.
\sqrt{x}	8,0000	2 (y) hoch 3 (x).

Rechnen Sie das nächste und die darauf folgenden Beispiele. Beachten Sie, daß Sie nur dann ENTER zur Trennung von Zahlen zu drücken brauchen, wenn Sie die Zahlen direkt hintereinander eingeben. Ein zuvor berechnetes Ergebnis (Zwischenresultat) wird automatisch von der nächsten Zahl, die Sie eingeben, getrennt.

Lösen Sie $(2 + 4) \div 8$:

Tastenfolge	Anzeige	
2 ENTER	2,0000	
4	4	
$+$	6,0000	$(2 + 4)$
8	8	
\div	0,7500	$(2 + 4) \div 8$

Lösen Sie $(9 + 17 - 4 + 23) \div 4$:

Tastenfolge	Anzeige	
9 ENTER	9,0000	
17 $+$	26,0000	$(9 + 17)$
4 $-$	22,0000	$(9 + 17 - 4)$
23 $+$	45,0000	$(9 + 17 - 4 + 23)$
4 \div	11,2500	$(9 + 17 - 4 + 23) \div 4$

Sogar deutlich kompliziertere Probleme lassen sich auf diese einfache Weise lösen – nämlich durch Ausnutzung der automatischen Speicherung von Zwischenresultaten.

Beispiel: Lösen Sie $(6 + 7) \times (9 - 3)$.

Zunächst wird das Zwischenresultat $(6 + 7)$ berechnet:

Tastensequenz	Anzeige	
6	6	
<input type="text" value="ENTER"/>	6,0000	
7	7	
<input type="text" value="+"/>	13,0000	$(6 + 7)$

Als nächstes ist $(9 - 3)$ zu berechnen. Da jetzt wieder zwei Zahlen direkt hintereinander eingegeben werden müssen, muß mit der Taste die erste Zahl (9) von der zweiten (3) getrennt werden. (braucht nicht gedrückt zu werden, um die Zahl 9 von dem zuvor berechneten Zwischenresultat 13, das sich bereits im Rechner befindet, zu trennen – die Ergebnisse von vorangegangenen Berechnungen werden automatisch gespeichert.)

Zu lösen ist $(9 - 3)$:

Tastensequenz	Anzeige	
9	9	
<input type="text" value="ENTER"/>	9,0000	
3	3	
<input type="text" value="-"/>	6,0000	$(9 - 3)$

Nun sind die Zwischenresultate (13 und 9) zu multiplizieren, um das Endergebnis zu erhalten:

Tastensequenz	Anzeige	
<input type="text" value="X"/>	78,0000	$(6 + 7) \times (9 - 3) = 78$

Beachten Sie, daß der HP-11C die Zwischenresultate für Sie automatisch gespeichert und sie bei der letzten Multiplikation nach dem Prinzip last-in/first-out verwendet hat. Jedes noch so kompliziert aussehende Problem kann immer auf eine Folge von Operationen mit ein oder zwei Variablen reduziert werden.

Denken Sie daran:

- Bei jeder Operation, die die sequentielle Eingabe zweier Zahlen erfordert, wird die Taste zur Trennung der zweiten Zahl von der ersten benötigt.
- Nach einer Berechnung neu eingetastete Ziffern werden automatisch als neue Zahl behandelt.
- Zwischenresultate werden nach dem Prinzip last-in/first-out gespeichert.

Rechnen Sie jetzt die folgenden Beispiele durch. Gehen Sie dabei so vor, als wenn Sie mit Papier und Bleistift arbeiten würden. Ignorieren Sie die Zwischenresultate – sie werden von Ihrem HP-11C automatisch verarbeitet.

$$(16 \times 38) - (13 \times 11) = 465,0000$$

$$(27 + 63) \div (33 \times 9) = 0,3030$$

$$(\sqrt{16,38 \times 0,55}) \div 0,05 = 60,0300$$

$$4 \times (17 - 12) \div (10 - 5) = 4,0000$$

Automatischer Speicherstack, LAST X und Datenspeicherung

Automatischer Speicherstack und Stackmanipulation

Die automatische Speicherung und die Rückgabe von Zwischenresultaten sind Gründe für die einfache Durchführbarkeit von komplexen Berechnungen auf Ihrem HP-11C. Diese einfache Handhabung wird durch den automatischen Speicherstack und die **[ENTER]** Taste ermöglicht.

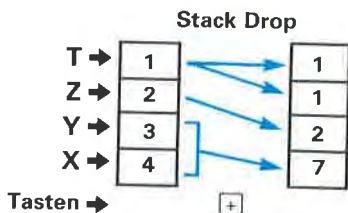
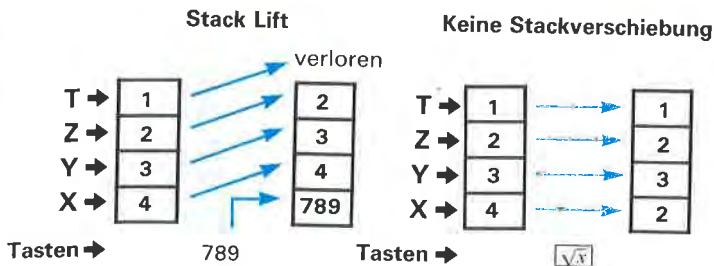
Automatische Speicherstack-Register

T →	0,0000	Wird immer angezeigt.
Z →	0,0000	
Y →	0,0000	
X →	0,0000	

Wenn sich Ihr HP-11C im Run Mode (die **PRGM** Statusanzeige ist aus) befindet, wird im Display immer der Wert des X-Registers angezeigt.

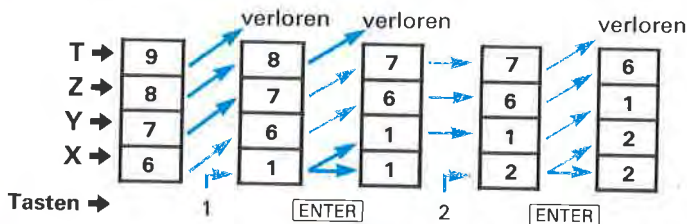
Jede neu eingegebene Zahl und jedes Ergebnis der Ausführung einer numerischen Funktion wird in dem angezeigten X-Register abgelegt. Durch die Ausführung einer Funktion oder die Eingabe einer Zahl werden die sich im Stack befindlichen Zahlen nach oben (Stack Lift) oder nach unten (Stack Drop) verschoben oder bleiben in den ursprünglichen Registern, je nach Typ der Operation. Die Zahlen im Stack sind auf der Basis last-in/first-out verfügbar. Wenn, zum Beispiel, der Stack durch vorangegangene Berechnungen wie auf der linken Seite der drei im folgenden gezeigten Abbildungen geladen ist, so bewirkt das Drücken der angedeuteten Tasten eine Stackverschiebung wie sie auf der rechten Seite der jeweiligen Abbildung zu sehen ist.*

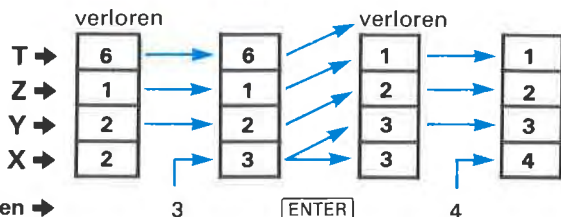
* Zur Vereinfachung der Illustration der in diesem Abschnitt beschriebenen Abläufe wird in den meisten Diagrammen eine einziffrige Zahlendarstellung an Stelle des sonst üblichen **[FIX] 4** Dezimalformats benutzt.



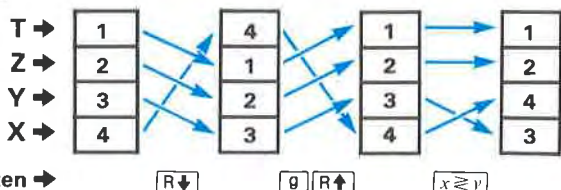
Funktionen zur Stackmanipulation

ENTER. Trennt zwei nacheinander eingegebene Zahlen. Wenn **ENTER** gedrückt wird, verschiebt der Rechner den Stackinhalt nach oben, indem er die Zahl im angezeigten X-Register in das Y-Register kopiert. Der Stack soll beispielsweise mit den Zahlen 1, 2, 3, 4 belegt werden (dabei wird vorausgesetzt, daß der Stack durch vorangegangene Berechnungen mit den gezeigten Zahlen geladen ist):

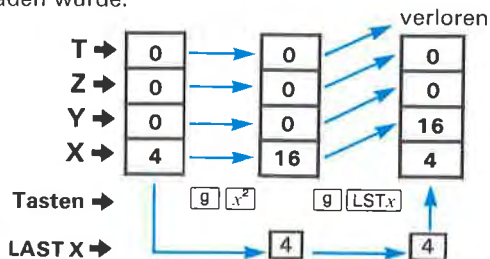




$\boxed{R\downarrow}$ (roll down), $\boxed{R\uparrow}$ (roll up) und $\boxed{x\geq y}$ (X exchange Y). $\boxed{R\downarrow}$ und $\boxed{R\uparrow}$ verschieben zyklisch den Inhalt der Stackregister um ein Register nach unten oder nach oben. Die Zahlen bleiben erhalten. $\boxed{x\geq y}$ vertauscht die Zahlen im X- und Y-Register. Wenn der Stack mit den Zahlen 1, 2, 3, 4 geladen ist, würde das Drücken von $\boxed{R\downarrow}$, $\boxed{9}$, $\boxed{R\uparrow}$ und $\boxed{x\geq y}$ die folgenden Verschiebungen bewirken:



$\boxed{LST_x}$ (LAST X). Bei der Ausführung einer numerischen Funktion wird der Wert, der das angezeigte X-Register vor der Funktionsausführung belegt hat, in das LAST X Register gerettet. Durch Drücken von $\boxed{9}$ $\boxed{LST_x}$ wird der aktuelle Inhalt des LAST X Registers wieder in das angezeigte X-Register zurück übertragen. In Anhang B, Stack Lift und LAST X, ist eine Aufstellung aller Funktionen, vor deren Ausführung der Inhalt des X-Registers in das LAST X Register gerettet wird, zu finden. Das folgende Diagramm illustriert den Ablauf einer LAST X Operation. Dabei wird unterstellt, daß der Stack zuvor mit den auf der linken Seite des Diagramms gezeigten Zahlen geladen wurde.



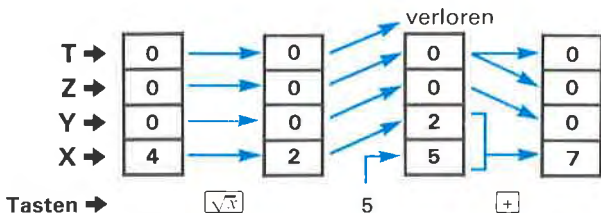
Rechnerfunktionen und Stack

Wenn Sie zwei Zahlen nacheinander eingeben wollen, müssen Sie zur Trennung der beiden Zahlen **ENTER** drücken. Wenn Sie jedoch nur eine Zahl eingeben wollen, weil sich die andere Zahl bereits im angezeigten X-Register als Ergebnis einer vorangegangenen Berechnung oder sonstigen Funktionsausführung (wie etwa $x \geq y$, **R**↑, usw.) befindet, brauchen Sie **ENTER** nicht betätigen.

Woran liegt dies? Die Ausführung einer HP-11C Funktion hat zwei Auswirkungen:

1. Die spezifizierte Funktion wird ausgeführt.
2. Der automatische Speicherstack wird *freigegeben*, d.h. bei Eingabe der *nächsten* Zahl verschiebt sich der Stackinhalt automatisch nach oben.

Das X-Register sei beispielsweise mit der Zahl 4 geladen:



Es gibt vier Funktionen – **ENTER**, **CL_x**, **Σ+** und **Σ-** – nach deren Ausführung der Stack *gesperrt* ist.* In diesen Fällen wird der Stack bei der Eingabe der *nächsten* Zahl nicht nach oben verschoben. Nach Ausführung einer der genannten Funktionen wird die momentan angezeigte Zahl bei der Eingabe eines neuen Werts überschrieben. (Es ist zu beachten, daß die Ausführung von **ENTER** einen direkten Stack Lift bewirkt, und die Eingabe der *nächsten* Zahl somit *keine* Verschiebung der Stackinhalte auslöst. Die auf den Seiten 27 und 28 illustrierte Arbeitsweise von **ENTER** zeigt, wie durch **ENTER** der Stack in diesem Fall gesperrt wird.) In den meisten Fällen sind die hier genannten Aspekte so natürlich in den Rechenprozeß eingliedert, daß sie Ihnen überhaupt nicht auffallen werden.

* Wenn durch Drücken von **☐** die gesamte Anzeige gelöscht wird, so ist dies identisch mit **CL_x** und der Sperrung des Stacks. Ansonsten ist **☐** neutral, d.h. der Stackinhalt bleibt unberührt. Eine weitergehende Diskussion des Stacks ist in Anhang B, Stack Lift und LAST X, zu finden.

Funktionen von zwei Variablen

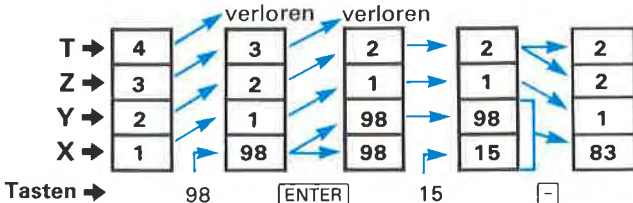
Ein wichtiger Aspekt bei den Funktionen zweier Variablen ist die Positionierung der Zahlen im Stack. Vor der Ausführung einer arithmetischen Funktion sollten die Zahlen so positioniert sein, wie man sie auch auf dem Papier anordnen würde. Wenn Sie zum Beispiel 15 von 98 subtrahieren wollen, schreiben Sie zunächst die Zahl 98 auf das Papier und anschließend die Zahl 15 darunter:

$$\begin{array}{r} 98 \\ -15 \\ \hline \end{array}$$

Danach würden Sie die Subtraktion ausführen:

$$\begin{array}{r} 98 \\ -15 \\ \hline 83 \end{array}$$

Die Zahlen werden im Rechner in der gleichen Weise positioniert; die erste Zahl, der Minuend, im Y-Register und die zweite Zahl, der Subtrahend, im angezeigten X-Register. Bei Ausführung der Subtraktionsfunktion wird die 15 im X-Register von der 98 im Y-Register abgezogen und der Stack nach unten verschoben, wodurch das Ergebnis im X-Register abgelegt wird. Das folgende Diagramm zeigt den Ablauf der gesamten Operation (dabei wird unterstellt, daß der Stack zuvor mit den auf der linken Seite gezeigten Zahlen geladen wurde):



Bei jeder arithmetischen Funktion werden die Zahlen immer zunächst in ihrer natürlichen Reihenfolge positioniert, danach wird die Funktion ausgeführt, und der Stack wird nach unten verschoben. In obigem Beispiel wurde 15 von 98 subtrahiert. Die gleiche Zahlenpositionierung würde benutzt werden, um 15 zu 98 zu addieren, 98 mit 15 zu multiplizieren und 98 durch 15 zu dividieren, d. h.:

$$\begin{array}{r} 98 \\ + 15 \\ \hline \end{array}$$

$$\begin{array}{r} 98 \\ \times 15 \\ \hline \end{array}$$

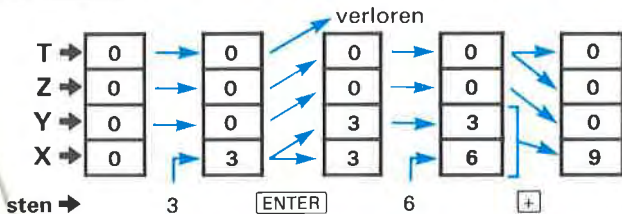
$$\begin{array}{r} 98 \\ \hline 15 \end{array}$$

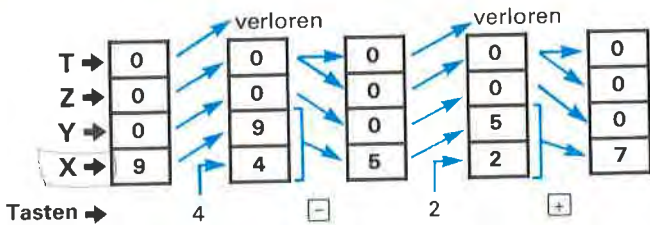
T →	
Z →	
Y →	98
X →	15

Kettenrechnung

Ob Sie nun Ihren HP-11C für manuelle oder programmierte Lösungen benutzen, in allen Fällen werden Sie häufig mit Kettenrechnungen zu tun haben. Auf diesem Gebiet wird die Einfachheit und Leistungsfähigkeit der HP-11C Logik besonders deutlich. Selbst bei extrem langen Berechnungen führen Sie zu jedem Zeitpunkt immer nur eine einzige Operation durch. Der automatische Speicherstack speichert bis zu vier Zwischenresultate solange für Sie, bis Sie sie benötigen, um sie dann in die Rechnung einzufügen. Dadurch wird die Abarbeitung eines Problems auf die gleiche natürliche Weise wie mit Papier und Bleistift möglich.

Sie haben bereits gelernt, wie ein Zahlenpaar mit Hilfe der **ENTER** Taste einzugeben ist, und wie danach eine Berechnung durchzuführen ist. Sie haben gesehen, wie sich der Stack durch die Ausführung bestimmter Funktionen nach unten verschiebt, und wie er sich durch die Eingabe einer neuen Zahl nach der Ausführung einer Funktion wieder nach oben verschiebt. Lassen Sie uns nun die Auswirkung einer Kettenrechnung auf den Stackinhalt betrachten. Dazu sei $3 + 6 - 4 + 2$ zu berechnen. (Es wird vorausgesetzt, daß der Stack vor Beginn der Rechnung durch Drücken von **←**, **ENTER**, **ENTER**, **ENTER** auf Null gesetzt wurde.)





Wie Sie sehen, wird zu jedem Zeitpunkt nur eine einzige Operation abgearbeitet. Nach jeder arithmetischen Operation wird der Stackinhalt automatisch nach unten verschoben und nach Abschluß der Berechnung bei Eingabe einer neuen Zahl wieder nach oben geschoben. Sogar weitaus kompliziertere Probleme lassen sich auf die gleiche, einfache Weise lösen.

Beispiel: An Stelle des in diesem Abschnitt bereits verwendeten Pfeildiagramms soll jetzt eine Tabelle benutzt werden, um die Stackoperationen bei der Lösung des Ausdrucks

$$\frac{(3 + 4) \times (6 - 4)}{2}$$

(96x2) + (713) - 56901
16 + 37

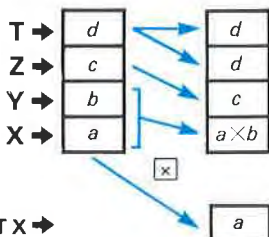
zu verfolgen.

T →	0	0	0	0	0	0
Z →	0	0	0	0	0	7
Y →	0	3	3	0	7	6
X →	3	3	4	7	6	6
Tasten →	3	ENTER	4	+	6	ENTER

T →	0	0	0	0	0
Z →	7	0	0	0	0
Y →	6	7	0	14	0
X →	4	2	14	2	7
Tasten →	4	-	x	2	÷

LAST X

Im LAST X Register des HP-11C, einem separaten Datenspeicher-Register, wird der vor der Ausführung einer numerischen Funktion zuletzt im Display angezeigte Wert abgelegt.*



Diese Funktion erspart Ihnen die Neueingabe von Zahlen, die Sie weiter benutzen wollen, und kann Ihnen bei der Fehlerkorrektur behilflich sein.

Beispiel: Es sind zwei Zahlen, etwa 45,575 Meter und 25,331 Meter, mit 0,175 zu multiplizieren:

T →	0,0000	0,0000	0,0000	0,0000
Z →	0,0000	0,0000	0,0000	0,0000
Y →	0,0000	45,5750	45,5750	0,0000
X →	45,575	45,5750	0,175	7,9756

Tasten → 45 575 175

LAST X →

T →	0,0000	0,0000	0,0000
Z →	0,0000	7,9756	0,0000
Y →	7,9756	25,3310	7,9756
X →	25,331	0,1750	4,4329

Tasten → 25.331

LAST X →

* Dies gilt für alle numerischen Funktionen mit Ausnahme der statistischen Funktionen , und

Durch $\boxed{\text{LST}_x}$ wird eine einfache Korrektur von Eingabefehlern, wie etwa das Auslösen der falschen Funktionstaste oder die Eingabe falscher Zahlenwerte, möglich. Dividieren Sie zum Beispiel 287 durch 13,9, nachdem sie zuvor versehentlich durch 12,9 geteilt haben.

Tastenfolge287 $\boxed{\text{ENTER}}$ 12,9 $\boxed{\div}$ $\boxed{\text{g}}$ $\boxed{\text{LST}_x}$ **Anzeige**

287,0000

22,2481

12,9000

Der falsche Divisor!!!

Aus dem LAST X Register wird der letzte Wert des X-Registers (der falsche Divisor) vor der Ausführung von $\boxed{\div}$ zurückgeholt.

 $\boxed{\times}$

287,0000

Ausführung der inversen Funktion zu der Funktion, die das falsche Ergebnis erzeugt hat.

13,9 $\boxed{\div}$

20,6475

Das korrekte Ergebnis.

Konstantenarithmetik

Da der Wert des T-Registers erhalten bleibt, wenn der Stack nach unten verschoben wird, kann diese Zahl als Konstante in arithmetischen Operationen benutzt werden.

T →	c	c
Z →	c	c
Y →	c	c
X →	x	cX
Tasten →		$\boxed{\times}$

Um eine Konstante in eine Berechnung einzufügen, muß der Stack mit dieser Konstante geladen werden, indem die Konstante in das X-Register eingegeben und anschließend dreimal $\boxed{\text{ENTER}}$ gedrückt wird. Die Konstante wird benutzt, indem Sie das eigentliche Argument eingeben und die gewünschten arithmetischen Operationen ausführen. Bei jeder Verschiebung des Stacks nach unten steht Ihnen eine Kopie der Konstanten für die nächste Berechnung zur Verfügung. Gleichzeitig wird die Konstante im T-Register reproduziert.

Beispiel: Ein Bakteriologe testet eine Kultur von Mikroorganismen, deren Population im Durchschnitt jeden Tag um 15 % wächst (d. h. der Wachstumsfaktor beträgt 1,15). Wie groß ist die Bakterienpopulation am Ende eines jeden von fünf aufeinanderfolgenden Tagen, wenn die Kultur mit 1000 angesetzt wird?



Verfahren: Laden Sie mit Hilfe von **ENTER** den konstanten Wachstumsfaktor (1,15) in das Y-, Z- und T-Register und die ursprüngliche Population (1000) in das angezeigte X-Register. Danach erhalten Sie durch jedes Auslösen von **X** die neue Tagespopulation. Drücken Sie **f** **FIX** 2, um das Anzeigeformat Ihres Rechners an das im folgenden benutzte Format anzugleichen.

T →	0,00	0,00	0,00	1,15	1,15
Z →	0,00	0,00	1,15	1,15	1,15
Y →	0,00	1,15	1,15	1,15	1,15
X →	1,15	1,15	1,15	1,15	1.000

Tasten → 1,15 **ENTER** **ENTER** **ENTER** 1000

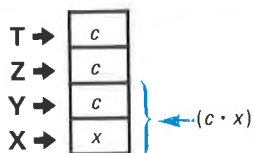
T →	1,15	1,15	1,15	1,15	1,15
Z →	1,15	1,15	1,15	1,15	1,15
Y →	1,15	1,15	1,15	1,15	1,15
X →	1.150,00	1.322,50	1.520,88	1.749,01	2.011,36

Tasten → **X** **X** **X** **X** **X**

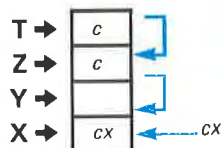
Beim ersten Drücken von **X** wird $1,15 \times 1000$ berechnet. Das Ergebnis (1.150,00) wird im X-Register angezeigt, der Stack wird

nach unten verschoben und die Konstante im T-Register reproduziert, d. h. jedes Drücken von $\boxed{\times}$ beinhaltet:

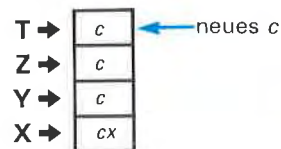
1. Der Beginn einer neuen Berechnung, die das X- und Y-Register benutzt.



2. Das Ergebnis der Berechnung wird im X-Register abgelegt, und der verbleibende Stackinhalt nach unten verschoben.



4. In T wird wieder eine neue Kopie der zuletzt in diesem Register gespeicherten Zahl (in diesem Fall unsere Konstante) erzeugt.



Da bei jedem Stack Drop eine neue Kopie des Wachstumsfaktors in das T-Register abgebildet wird, brauchen Sie diesen Wert niemals neu einzugeben.

Drücken Sie \boxed{f} \boxed{FIX} 4, um den HP-11C wieder auf das Anzeigeformat \boxed{FIX} 4 zu schalten.

Als Alternative zum Auffüllen des Stacks mit einer Konstanten ist Konstantenarithmetik auch mit Hilfe des LAST X Registers möglich. Bei dieser Verfahrensweise wäre das Ergebnis des letzten Beispiels wie folgt zu berechnen:

1. Tasten Sie die ursprüngliche Population ein und drücken Sie \boxed{ENTER} .
2. Geben Sie den konstanten Wachstumsfaktor (1,15) ein.
3. Drücken Sie $\boxed{\times}$, um die Population am Ende des ersten Tages zu berechnen.
4. Drücken Sie \boxed{g} \boxed{LSTx} $\boxed{\times}$, um die Population am Ende eines jeden Folgetages zu berechnen.

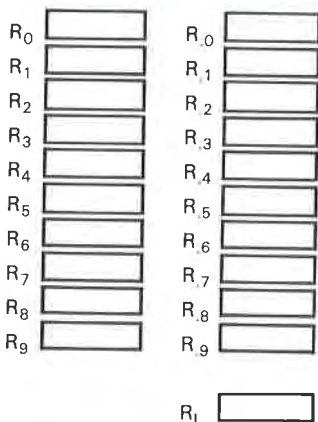
Speicherregister-Operationen

Das Speichern und der Rückruf von Zahlen sind Operationen, die das angezeigte X-Register und die 21 Datenspeicher-Register des HP-11C betreffen. Die Datenspeicher-Register sind völlig getrennt von den Stackregistern und dem LAST X Register.

Speichern von Zahlen

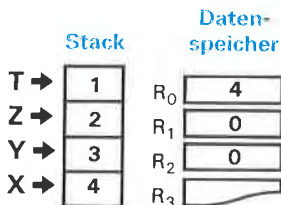
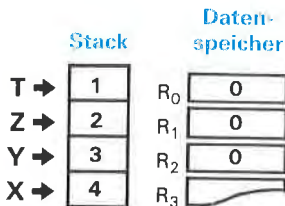
[STO] (*store*). Wenn hinter dem Befehl eine Registeradresse (0 bis 9, \square 0 bis \square 9 oder \square) folgt, wird die Zahl im angezeigten X-Register in das Datenspeicher-Register mit der spezifizierten Adresse kopiert.

Datenspeicher-Register



Wenn ...

... und Sie **[STO]** 0 drücken, dann ...



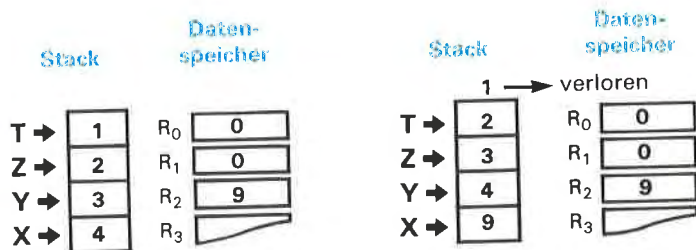
Die gespeicherte Zahl bleibt solange im Speicherregister bis eine neue Zahl dort gespeichert wird, oder die Speicherregister gelöscht werden, oder der gesamte Permanentenspeicher gelöscht wird.

Rückruf von Zahlen

RCL (*recall*). Wenn dem Befehl eine Speicherregister-Adresse (0 bis 9, \square 0 bis \square 9 oder \square) folgt, wird die Zahl aus dem spezifizierten Datenspeicher-Register in das angezeigte X-Register geladen. Der Inhalt des Datenspeicher-Registers bleibt unverändert. Wenn der Stack nicht gesperrt ist, wird durch eine **RCL** Operation der Stackinhalt nach oben verschoben.

Wenn ...

... und Sie **RCL** 2 drücken, dann ...



Übungen zum Speichern und Rückrufen von Zahlen

Führen Sie die folgenden Operationen aus:

Tastenfolge

123

STO 4

678

STO \square 7

RCL 4

RCL \square 7

Anzeige

123

123,0000

678

678,0000

123,0000

678,0000

Speichern von 123 in R₄.

Speichern von 678 in R₇.

Rückruf von 123 aus R₄.

Rückruf von 678 aus R₇.

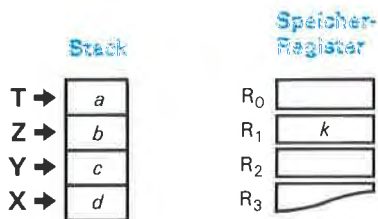
Löschen der Datenspeicher-Register

CLEAR **REG** (*clear registers*). Mit diesem Befehl wird der Inhalt aller Datenspeicher-Register auf Null gesetzt. **CLEAR** **REG** wirkt nicht auf

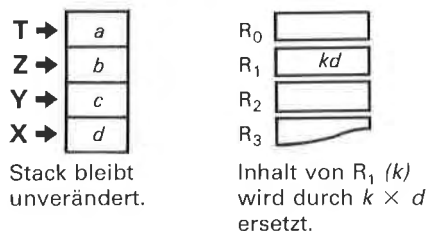
die Stackregister und das LAST X Register. (Um ein einzelnes Datenspeicher-Register zu löschen, laden Sie einfach Null in das betreffende Register.)

Speicherregister-Arithmetik

$\boxed{\text{STO}}$ ($\boxed{+}$, $\boxed{-}$, $\boxed{\times}$, $\boxed{\div}$) n (*storage register arithmetic*). Bei Befehlssequenzen dieser Form werden die Inhalte des angezeigten X-Registers und des durch n spezifizierten Datenspeicher-Registers als Argumente für die gewünschte arithmetische Operation benutzt. Diese Tastenfolgen haben den allgemeinen Aufbau: $\boxed{\text{STO}}$ gefolgt von einer arithmetischen Funktionstaste, wiederum gefolgt von einer Speicherregisteradresse (0 bis 9)*. Das Ergebnis einer Operation der Speicherregister-Arithmetik wird in dem spezifizierten Datenspeicher-Register abgelegt.



Wenn der Stack- und Speicher-Status durch die obige Abbildung dargestellt wird, so bewirkt die Ausführung von $\boxed{\text{STO}}$ $\boxed{\times}$ 1 die folgenden Veränderungen:



* Die Speicherregister-Arithmetik ist auch mit den Registern R₀ bis R₉ zulässig. In diesen Fällen muß eine indirekte Adressierung, wie sie in Abschnitt 9, das Indexregister, beschrieben wird, verwendet werden. Die Benutzung von R₁ innerhalb der Speicherregister-Arithmetik ist unzulässig.

Übungen zur Speicherregister-Arithmetik

18 [STO] 0	Anzeige 18,0000	Speichern von 18 in R ₀ .
3 [STO] [÷] 0	3,0000	Division der Zahl in R ₀ (18) durch 3.
[RCL] 0	6,0000	Rückruf des Ergebnisses aus R ₀ .
4 [STO] [×] 0	4,0000	Multiplikation des Ergebnisses in R ₀ (6,0000) mit 4.
[RCL] 0	24,0000	Rückruf des neuen Ergebnisses aus R ₀ .
[STO] [+] 0	24,0000	Additionen von 24 zu der Zahl in R ₀ .
[RCL] 0	48,0000	Rückruf des Ergebnisses aus R ₀ .
40 [STO] [−] 0	40,0000	Subtraktion von 40 von der Zahl in R ₀ .
[RCL] 0	8,0000	Rückruf des Ergebnisses aus R ₀ .

Übungsaufgaben

1. Berechnen Sie den Wert von x in der folgenden Gleichung.

$$x = \sqrt{\frac{8,33 \times (4 - 5,2) + ((8,33 - 7,46) \times 0,32)}{4,3 \times (3,15 - 2,75) - (1,71 \times 2,01)}}$$

Ergebnis: ~~4,6728~~

Einer der möglichen Lösungswege ist

4 **[ENTER]** 5,2 **[−]**
 8,33 **[×]** **[g]** **[LSTx]** 7,46
[−] ,32 **[×]** **[+]**
 3,15 **[ENTER]** 2,75 **[−]**
 4,3 **[×]** 1,71 **[ENTER]**
 2,01 **[×]** **[−]** **[÷]** **[√x]**

2,3789

2. Benutzen Sie die Konstantenarithmetik um den Restrückzahlungsbetrag eines Darlehens in Höhe von 1000 DM zu ermitteln, für das bereits 6 Rückzahlungsraten in Höhe von jeweils 100 DM geleistet worden sind. Der Zinssatz (i) pro Zahlungsperiode beträgt 1 %.

Vorgehensweise: Laden Sie den Stack mit $(1 + i)$ und geben Sie den ursprünglichen Restrückzahlungsbetrag ein. Benutzen Sie folgende Formel, um nach jeder Rückzahlungsrate den noch ausstehenden Restrückzahlungsbetrag zu ermitteln:

$$\text{Neuer Restrückzahlungsbetrag} = ((\text{Alter Restrückzahlungsbetrag}) \times (1 + i)) - \text{Rückzahlungsrate}$$

Ergebnis: 446,3186

3. Speichern Sie in Register R_5 die Zahl 100. Führen Sie die folgenden Operationen aus:
1. Dividieren Sie den Inhalt von R_5 durch 25.
 2. Subtrahieren Sie 2 vom Inhalt von R_5 .
 3. Multiplizieren Sie den Inhalt von R_5 mit 0,75.
 4. Addieren Sie 1,75 zum Inhalt von R_5 .
 5. Rufen Sie den Inhalt von R_5 auf.

Ergebnis: 3,2500

Numerische Funktionen

Die numerischen Funktionen Ihres HP-11C bieten Ihnen eine ganze Reihe von Operationen, die von der einfachen Zahlenmanipulation bis hin zu mathematischen und statistischen Anwendungen reichen. Die Verwendung einer Funktion innerhalb eines Programms oder über das Tastenfeld ist in beiden Fällen gleich.

Pi

Durch Drücken der Tasten $\boxed{f} \boxed{\pi}$ wird eine 10-stellige Approximation der Zahl Pi (3,141592654) im angezeigten X-Register abgelegt. Bei nicht gesperrtem Stack wird durch $\boxed{f} \boxed{\pi}$ der Stackinhalt nach oben verschoben.

Funktionen zur Zahlenmanipulation

Zusätzlich zu \boxed{CHS} (negative Zahlen; siehe Seite 17) enthält der HP-11C vier weitere Funktionen zur Zahlenmanipulation: \boxed{ABS} , \boxed{INT} , \boxed{FRAC} und \boxed{RND} .

Absolutwert. Durch Drücken von $\boxed{g} \boxed{ABS}$ wird die Zahl im angezeigten X-Register durch ihren Absolutwert ersetzt.

Rundung. Durch Drücken von $\boxed{g} \boxed{RND}$ wird die intern gespeicherte 10-stellige Mantisse der momentan angezeigten Zahl auf die Anzahl von Stellen gerundet, die durch das aktuelle \boxed{FIX} , \boxed{SCI} oder \boxed{ENG} Anzeigeformat festgelegt ist.

Ganzzahliger Anteil. Durch Drücken von $\boxed{g} \boxed{INT}$ wird die Zahl im angezeigten X-Register durch ihren ganzzahligen Anteil ersetzt, d. h. alle Ziffern rechts des Dezimalpunkts (-kommas) werden auf Null gesetzt.

Gebrochener Anteil. Durch Drücken von $\boxed{f} \boxed{FRAC}$ wird die Zahl im angezeigten X-Register durch ihren dezimalen Anteil ersetzt, d. h. alle Ziffern links des Dezimalpunkts (-kommas) werden auf Null gesetzt.

Gesucht	Tastensequenz	Anzeige
Absolutwert	12345 CHS g ABS	-12.345 12.345,0000
Ganzzahliger Anteil	123,4567 g INT	123,4567 123,0000
Gebrochener Anteil	123,4567 f FRAC	123,4567 0,4567
Rundung (bei Anzeigeformat FIX 4)	1,23456789 g RND	1,23456789
Kontrolle der Rundung	f FIX 8	1,23460000
Zurückschalten auf FIX 4	f FIX 4	1,2346

Funktionen einer Variablen

Die mathematischen Funktionen einer Variablen des HP-11C lassen sich wie folgt charakterisieren:

- Die Zahl im angezeigten X-Register wird als Argument für die Funktion benutzt.
- Die Zahl im angezeigten X-Register wird durch das Ergebnis der Funktionsausführung ersetzt.
- Der Inhalt des Y-, Z- und T-Registers bleibt unverändert.

Allgemeine Funktionen

Reziprokwert. Durch Drücken von $\boxed{1/x}$ wird der Reziprokwert der Zahl im angezeigten X-Register berechnet, d.h. die Zahl 1 wird durch den Wert des X-Registers dividiert.

Fakultät und Gammafunktion. Durch Drücken von $\boxed{f} \boxed{x!}$ wird wahlweise die Fakultät oder der Funktionswert der Gammafunktion berechnet:

1. Fakultät.

Wenn sich bei der Ausführung dieser Funktion im angezeigten X-Register eine nichtnegative ganze Zahl n ($1 \leq n \leq 69$) befindet, wird durch $\boxed{x!}$ die Fakultät von n , d.h. das Produkt aller ganzen Zahlen von 1 bis n , berechnet.

2. Gammafunktion.

Die Taste $\boxed{x!}$ kann auch zur Berechnung der Gammafunktion, die mit $\Gamma(x)$ bezeichnet wird, und bei gewissen Anwendungen in der höheren Mathematik und Statistik vorkommt, benutzt werden.* Durch Drücken von $\boxed{x!}$ wird $\Gamma(x+1)$ berechnet. Um den Funktionswert der Gammafunktion für ein bestimmtes Argument zu berechnen, ist von dem Argument die Zahl 1 zu subtrahieren. Wenn sich danach das Ergebnis der Subtraktion im angezeigten X-Register befindet, können Sie $\boxed{f} \boxed{x!}$ drücken.

Quadratwurzel. Durch Drücken von $\boxed{\sqrt{x}}$ wird die Quadratwurzel der Zahl im angezeigten X-Register berechnet.

Quadrat. Durch Drücken von $\boxed{g} \boxed{x^2}$ wird das Quadrat der Zahl im angezeigten X-Register berechnet.

Gesucht	Tastensequenz	Anzeige
Reziprokwert	25 $\boxed{1/x}$	25 0,0400
Fakultät	8 $\boxed{f} \boxed{x!}$	8 40.320,0000
Gammafunktion	2,7 $\boxed{\text{ENTER}} \boxed{1} \boxed{-}$ $\boxed{f} \boxed{x!}$	2,7 1,7000 1,5447
Quadratwurzel	3,9 $\boxed{\sqrt{x}}$	3,9 1,9748
Quadrat	12,3 $\boxed{g} \boxed{x^2}$	12,3 151,2900

Trigonometrische Operationen

Sie haben die Möglichkeit einen trigonometrischen Mode zu wählen, indem die sechs trigonometrischen Basisfunktionen arbeiten sollen.

* $\boxed{x!}$ kann für beide Funktionen (Fakultät und Gammafunktion) benutzt werden, da (wenn x eine nichtnegative ganze Zahl n ist) die Beziehung $\Gamma(x+1) = \Gamma(n+1) = n!$ gilt. Die Gammafunktion kann als Verallgemeinerung der Fakultät aufgefaßt werden, da der Wert im X-Register nicht notwendig eine nichtnegative ganze Zahl sein muß. Umgekehrt läßt sich die Fakultät als Spezialfall der Gammafunktion betrachten.

Trigonometrische Modi. Durch die Auswahl eines bestimmten trigonometrischen Modes werden bereits im Rechner befindliche Zahlen nicht in diesen Mode konvertiert. Die Spezifikation eines trigonometrischen Modes beinhaltet lediglich, daß dem Rechner mitgeteilt wird, welche Maßeinheit (Altgrad, Bogenmaß oder Neugrad) bei der Ausführung einer trigonometrischen Funktion benutzt werden soll.

DEG Mode. Durch Drücken von $\boxed{g} \boxed{DEG}$ wird der trigonometrische Mode „dezimale Altgrad“ gewählt. Im Display erscheint keine Statusanzeige.

RAD Mode. Durch Drücken von $\boxed{g} \boxed{RAD}$ wird der trigonometrische Mode „Radian“ (Bogenmaß) gewählt. Wenn der \boxed{RAD} Mode gewählt wird, erscheint im Display die Statusanzeige **RAD**.

$\boxed{g} \boxed{RAD}$ 0,0000
RAD

GRAD Mode. Durch Drücken von $\boxed{g} \boxed{GRD}$ wird der trigonometrische Mode „dezimale Neugrad“ (360 Altgrad = 400 Neugrad) gewählt. Wenn der \boxed{GRD} Mode gewählt wird, erscheint im Display Statusanzeige **GRAD**.

$\boxed{g} \boxed{GRD}$ 0,0000
GRAD

Der Rechner befindet sich stets in einem der drei trigonometrischen Modi. Der Permanentpeicher erhält den zuletzt gewählten trigonometrischen Mode, wenn der Rechner nach einem Abschalten wieder eingeschaltet wird. Nach einem Stromausfall oder nach dem Löschen des PermanentSpeichers befindet sich der Rechner automatisch im \boxed{DEG} Mode.

Trigonometrische Funktionen

Tastenfolge	Funktion
\boxed{SIN}	Sinus
$\boxed{g} \boxed{SIN^{-1}}$	Arcussinus
\boxed{COS}	Cosinus
$\boxed{g} \boxed{COS^{-1}}$	Arcuscosinus
\boxed{TAN}	Tangens
$\boxed{g} \boxed{TAN^{-1}}$	Arcustangens

Bei der Benutzung von trigonometrischen Funktionen ist zu beachten, daß der Rechner vor Ausführung der Funktion auf den gewünschten trigonometrischen Mode (**DEG**, **RAD** oder **GRD**) eingestellt ist.

Gesucht	Tastenfolge	Anzeige
(In allen Beispielen wird der trigonometrische Mode DEG unterstellt.) Alle trigonometrischen Funktionen, z. B. Sinus	33,5 SIN	33,5 0,5519
Arcussinus	,7982 g SIN⁻¹	0,7982 52,9586

Zeit- und Winkelkonvertierung

Zahlen, die eine Zeit oder einen Winkel darstellen, werden vom HP-11C in Abhängigkeit von der zuletzt ausgeführten Konvertierung in einem Dezimal- oder Minuten/Sekunden-Format interpretiert:

Stunden, Dezimalstunden (H,h)	Stunden, Minuten Sekunden Dezimalsekunden (H,MMSSs)
oder	oder
Grad, Dezimalgrad (D,d)	Grad, Minuten Sekunden Dezimalsekunden (D,MMSSs)

Stunden (oder Grad)/Minuten/Sekunden-Konvertierung. Durch Drücken von **f** **→ H.MS** wird die Zahl im angezeigten X-Register aus einem Dezimalstunden- (oder Dezimalgrad-)Format in ein Stunden (oder Grad)/Minuten/Sekunden-Format konvertiert.

H,h → H,MMSSs
oder
D,h → D,MMSSs

Dezimalstunden- (oder Dezimalgrad-)Konvertierung

Durch Drücken von \boxed{g} \rightarrow \boxed{H} wird die Zahl im angezeigten X-Register aus einem Stunden (oder Grad)/Minuten/Sekunden/Dezimalsekunden-Format in ein Dezimalstunden- (oder Dezimalgrad-)Format konvertiert.

H,MMSSs \rightarrow H,h
oder
D,MMSSs \rightarrow D,d

Altgrad/Bogenmaß (Radiant) Konvertierungen

Die $\boxed{\rightarrow DEG}$ und $\boxed{\rightarrow RAD}$ Funktionen dienen zur Konvertierung von Winkeln aus (dezimalen) Altgrad in Radiant und umgekehrt (D,d \rightarrow R,r und R,r \rightarrow D,d).

Altgrad/Radiant Konvertierung. Durch Drücken von \boxed{f} \rightarrow \boxed{RAD} wird die Zahl im angezeigten X-Register von einem dezimalen Altgradwert in den äquivalenten Radiant-(Bogenmaß-)Wert konvertiert.

Radiant/Altgrad Konvertierung. Durch Drücken von \boxed{g} \rightarrow \boxed{DEG} wird die Zahl im angezeigten X-Register von einem Radiant-Wert in den äquivalenten (dezimalen) Altgradwert konvertiert.

Zu konvertieren	Tastenfolge	Anzeige
Dezimalstunden (H,h) oder Grad (D,d) in ein H,MMSSs oder D,MMSSs Format.	17,553 \boxed{f} \rightarrow $\boxed{H.MS}$	17,553 17,3311
Anzeige der Dezimalsekunden im \boxed{FIX} 4 Format. H,MMSSs oder D,MMSSs in ein Dezimalstunden (H,h) oder Grad (D,d) Format.	\boxed{f} \boxed{PREFIX}	1733108000 17,3311
Altgrad in Radiant	12,3045 \boxed{g} \rightarrow \boxed{H}	12,3045 12,5125
Radiant in Altgrad	40,5 \boxed{f} \rightarrow \boxed{RAD}	40,5 0,7069
	1,1746 \boxed{g} \rightarrow \boxed{DEG}	1,1746 67,2996

Logarithmische Funktionen

Natürlicher Logarithmus. Durch Drücken von \boxed{g} \boxed{LN} wird der natürliche Logarithmus der Zahl im angezeigten X-Register berechnet, d.h. der Logarithmus zur Basis e (2,718281828) des Werts im X-Register.

Natürliche Exponentialfunktion. Durch Drücken von $\boxed{e^x}$ wird der Wert der e-Funktion mit der Zahl im angezeigten X-Register als Argument berechnet, d.h. es wird e (2,718281828) hoch dem Inhalt des X-Registers gebildet.

Dekadischer Logarithmus. Durch Drücken von \boxed{g} \boxed{LOG} wird der dekadische Logarithmus der Zahl im angezeigten X-Register berechnet, d.h. der Logarithmus zur Basis 10 des Werts im X-Register.

Dekadische Exponentialfunktion. Durch Drücken von $\boxed{10^x}$ wird der Wert der dekadischen Exponentialfunktion mit der Zahl im angezeigten X-Register als Argument berechnet, d.h. die Zahl 10 wird mit dem Inhalt des X-Registers potenziert.

Gesucht	Tastenfolge	Anzeige
Natürlicher Logarithmus	45 \boxed{g} \boxed{LN}	45 3,8067
Natürliche Exponentialfunktion	3,4012 $\boxed{e^x}$	3,4012 30,0001
Dekadischer Logarithmus	12,4578 \boxed{g} \boxed{LOG}	12,4578 1,0954
Dekadische Exponentialfunktion	3,1354 $\boxed{10^x}$	3,1354 1.365,8405

Hyperbolische Funktionen

Tastenfolge	Funktion
\boxed{f} \boxed{HYP} \boxed{SIN}	Sinus hyperbolicus (<i>sinh</i>)
\boxed{g} $\boxed{HYP^{-1}}$ \boxed{SIN}	Inverser Sinus hyperbolicus (<i>sinh⁻¹</i>)
\boxed{f} \boxed{HYP} \boxed{COS}	Cosinus hyperbolicus (<i>cosh</i>)

Tastenfolge	Funktion
\boxed{g} $\boxed{HYP^{-1}}$ \boxed{COS}	Inverser Cosinus hyperbolicus ($cosh^{-1}$)
\boxed{f} \boxed{HYP} \boxed{TAN}	Tangens hyperbolicus ($tanh$)
\boxed{g} $\boxed{HYP^{-1}}$ \boxed{TAN}	Inverser Tangens hyperbolicus ($tanh^{-1}$)

Gesucht	Tastenfolge	Anzeige
Alle hyperbolische Funktionen, z. B.: Sinus hyperbolicus	2,53 \boxed{f} \boxed{HYP} \boxed{SIN}	2,53 6,2369
Arcussinus hyperbolicus	1,95 \boxed{g} $\boxed{HYP^{-1}}$ \boxed{SIN}	1,95 1,4210

Funktionen von zwei Variablen

Die mathematischen Funktionen zweier Variabler des HP-11C benutzen zur Berechnung eines Resultats die Werte des angezeigten X-Registers und des Y-Registers. Bei der Benutzung dieser Funktionen ist zunächst der Wert für das Y-Register einzugeben, danach ist die \boxed{ENTER} Taste zu drücken, um den Wert in das Y-Register zu laden und anschließend ist der Wert für das X-Register einzugeben. Dann kann die gewünschte Funktion ausgeführt werden.

Potenzrechnung

Durch Drücken von $\boxed{y^x}$ wird die Zahl im Y-Register mit dem Wert im X-Register potenziert.

Gesucht	Tastenfolge	Anzeige
Potenz y^x zu gegebenen Werten y und x	2 \boxed{ENTER} 3 $\boxed{y^x}$	2,0000 3 8,0000

Prozentrechnung

Prozentualer Anteil. Ein bestimmter prozentualer Anteil einer Zahl kann wie folgt berechnet werden:

1. Eingabe der Basiszahl.
2. \boxed{ENTER} drücken.
3. Prozentsatz eingeben.
4. \boxed{g} $\boxed{\%}$ drücken.

T →				
Z →				
Y →		150	150	150
X →	150	150	25	37,5
Tasten →	150	ENTER	25	g %
LAST X →				25

Der berechnete prozentuale Anteil erscheint im angezeigten X-Register, die Basiszahl bleibt im Y-Register, und der Prozentsatz wird im LAST X Register abgelegt. Der Stack wird nicht nach oben verschoben, so daß die Werte im T- und Z-Register auch nach dem Auslösen von $\boxed{g} \boxed{\%}$ erhalten bleiben. In der obigen Illustration wird die Wirkung der $\boxed{\%}$ Taste bei der Berechnung von 25 % von 150 gezeigt.

Prozentuale Differenz. Die $\boxed{\Delta\%}$ Funktion berechnet die prozentuale Differenz – den relativen Größenunterschied – zweier Zahlen. Die prozentuale Differenz wird wie folgt berechnet:

1. Eingabe der Basiszahl (in der Regel die Zahl, die zuerst auftritt).
2. \boxed{ENTER} drücken.
3. Eingabe der zweiten Zahl.
4. $\boxed{g} \boxed{\Delta\%}$ drücken.

T →				
Z →				
Y →		150	150	150
X →	150	150	225	50
Tasten →	150	ENTER	225	g $\Delta\%$
LAST X →				225

Bei der oben gezeigten Reihenfolge der Eingabe, beinhaltet ein positives Ergebnis einen Zuwachs der zweiten Zahl gegenüber der ersten; ein negatives Ergebnis deutet eine Abnahme der Wertigkeit der zweiten Zahl gegenüber der ersten an. Die obige Illustration

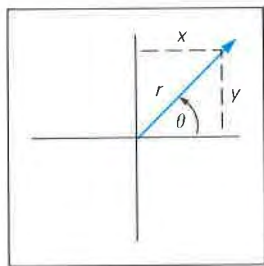
zeigt die Funktionsweise von $\Delta\%$ bei der Berechnung des prozentualen Zuwachs zwischen 150 (der Basiszahl) und 225 (der zweiten eingegebenen Zahl).

Gesucht	Tastenfolge	Anzeige
Prozentualer Anteil Basiszahl Prozentsatz Prozentualer Anteil	200 \square ENTER 75 \square g \square %	200,000 75 150,000
Prozentuale Differenz Basiszahl Zweite Zahl Prozentualer Zuwachs	40 \square ENTER 160 \square g \square $\Delta\%$	40,000 160 300,000

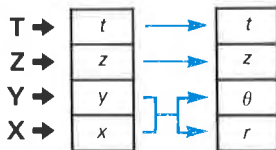
Transformation von Polar- und Rechteckskordinaten

Die beiden Funktionen \square →P und \square →R dienen zur Umwandlung von Polar- in Rechteckskordinaten und umgekehrt.

Der Winkel θ kann in Abhängigkeit von dem jeweiligen trigonometrischen Mode (\square DEG, \square RAD oder \square GRAD) in *dezimalen* Altgrad, Radian oder Neugrad angegeben werden. θ wird wie in der Abbildung rechts gemessen. Das Ergebnis einer Umwandlung (nach θ) liegt zwischen 180° und -180° .

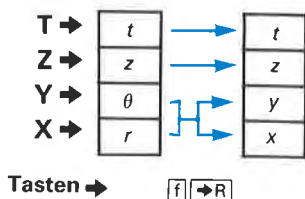


Konvertierung in Polarkoordinaten.
Durch Drücken von \square g \square →P (*polar*) werden die im X- und Y-Register gespeicherten Rechteckskordinaten (x , y) in Polarkoordinaten (Betrag r , Winkel θ) umgewandelt.



Tasten \square → \square g \square →P

Konvertierung in Rechteckskordinaten. Durch Drücken von $\boxed{f} \boxed{\rightarrow R}$ (*rectangular*) werden die im X- und Y-Register gespeicherten Polarkoordinaten (Betrag r , Winkel θ) in Rechteckskordinaten (x,y) umgewandelt.



Zu konvertieren	Tastensequenz	Anzeige
Umwandlung von Rechteckskordinaten in Polarkoordinaten:		
y	5 $\boxed{\text{ENTER}}$	5,0000
x	10	10
r	$\boxed{g} \boxed{\rightarrow P}$	11,1803
θ	$\boxed{x \approx y}$	26,5651
Umwandlung von Polarkoordinaten in Rechteckskordinaten:		
θ	30 $\boxed{\text{ENTER}}$	30,0000
r	12	12
x	$\boxed{f} \boxed{\rightarrow R}$	10,3923
y	$\boxed{x \approx y}$	6,0000

Wahrscheinlichkeitsrechnung

Permutation. Durch Drücken von $\boxed{f} \boxed{P_{y,x}}$ wird die Anzahl aller Möglichkeiten, y verschiedene Elemente zu Mengen mit jeweils x Elementen zusammenzufassen, berechnet. Mengen, die die gleichen Elemente in unterschiedlicher Reihenfolge enthalten, werden einzeln mitgezählt. (Jedes Element darf in einer Menge nur einmal vorkommen.) Zur Berechnung von $\boxed{P_{y,x}}$ wird die folgende Formel benutzt:

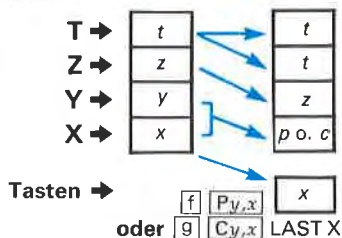
$$P_{y,x} = \frac{y!}{(y-x)!}$$

Kombination. Durch Drücken von $\boxed{g} \boxed{C_{y,x}}$ wird die Anzahl aller Möglichkeiten, y verschiedene Elemente zu Mengen mit jeweils x Elementen ohne Beachtung der Reihenfolge zusammenzufassen, berechnet. Jedes Element darf in einer Menge nur einmal vorkommen. $\boxed{C_{y,x}}$ berechnet die Anzahl der Kombinationen nach der folgenden Formel:

$$C_{y,x} = \frac{y!}{x! (y-x)!}$$

Zur Berechnung einer Permutation oder Kombination ist wie folgt vorzugehen:

1. Eingabe der Anzahl der Elemente (y).
2. \boxed{ENTER} drücken.
3. Eingabe der Anzahl der Elemente, die jeweils zu einer Menge (mit oder ohne Beachtung der Reihenfolge) zusammengefaßt werden sollen (x).
4. $\boxed{f} \boxed{P_{y,x}}$ oder $\boxed{g} \boxed{C_{y,x}}$ drücken.



Das Ergebnis, p oder c , erscheint im angezeigten X-Register, der Stack wird nach unten verschoben, und die Anzahl der Elemente pro Menge (x) wird im LAST X Register abgelegt.

Alle Eingaben für Kombinationen und Permutationen müssen nicht-negativ und ganzzahlig sein.

Bemerkung: Die Ausführung von $\boxed{P_{y,x}}$ und $\boxed{C_{y,x}}$ Berechnungen kann mehrere Sekunden oder länger in Anspruch nehmen, je nach Größe der eingegebenen x - und y -Werte. Der maximale Eingabewert für $\boxed{P_{y,x}}$ oder $\boxed{C_{y,x}}$ Berechnungen beträgt $10^{10} - 1$. Bei der Ausführung von längeren $\boxed{P_{y,x}}$ und $\boxed{C_{y,x}}$ Berechnungen erscheint im Display die Meldung „running“.

Gesucht	Tastensequenz	Anzeige
Permutation Beispiel: 10 Elemente sind unter Beachtung der Reihenfolge zu Mengen mit je 3 Elementen zusammenzufassen.	10 ENTER 3 f P_{y,x}	10,0000 720,0000
Kombination Beispiel: 10 Elemente sind ohne Beachtung der Reihenfolge zu Mengen mit je 3 Elementen zusammenzufassen.	10 ENTER 3 g C_{y,x}	10,0000 120,0000

Statistische Funktionen

Zufallszahlengenerator

Der Zufallszahlengenerator des HP-11C benutzt entweder einen automatischen Startwert (Null) oder einen von Ihnen vorgegebenen Startwert, um damit eine Folge von gleichverteilten Pseudo-Zufallszahlen im Bereich $0 \leq r \leq 1$ zu erzeugen.* Durch den Permanent-Speicher wird die Zufallszahlenfolge solange fortgesetzt, bis ein neuer (Anfangs-) Startwert gespeichert wird.

Startwert des Zufallszahlengenerators. Um zu einem beliebigen Zeitpunkt eine neue Zufallszahlenfolge zu erzeugen, müssen Sie einen neuen Startwert für den Zufallszahlengenerator definieren, indem Sie eine frei wählbare Zahl n ($0 \leq n \leq 1$) eingeben und anschließend **STO** **f** **RAN#** drücken. Um danach eine Zufallszahl zu erzeugen, brauchen Sie nur **f** **RAN#** betätigen. Die neue Zufallszahl erscheint dann im angezeigten X-Register. (**f** **RAN#** hat im Stack die gleiche Wirkung wie der Rückruf von Zahlen aus einem Speicherregister.) Die neu erzeugte Zufallszahl wird automatisch als Startwert für die nächste Zufallszahl der laufenden Sequenz benutzt. Bei einem Löschen des Permanent-Speichers oder einem Stromverlust wird der Startwert auf Null gesetzt. (Durch wiederholte Verwendung einer Zahl als Zufallszahlenstartwert wird jedesmal die gleiche Zufallszahlenfolge erzeugt.)

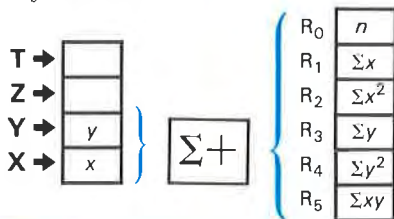
* Die erzeugte Sequenz erfüllt den Spektraltest (Knuth, V 2, Nr. 3.4).

Erzeugen von Zufallszahlen	Tastenfolge	Anzeige
Die Zahl 0,5764 ist als Zufallszahlenstartwert zu speichern: (Willkürlicher Startwert)	\square 5764	0,5764
Startwert gespeichert	STO \square f	0,5764
Erzeugen von Zufallszahlen, die auf obigem Startwert basieren:	RAN#	0,3422
	f RAN#	0,2809
	f RAN#	
	.	.
	.	.
	.	.

Eine weitergehende Diskussion des Zufallszahlengenerators ist auf Seite 217, Zufallszahlen, in Teil III dieses Handbuchs zu finden.

Summationen

$\Sigma+$ ist eine Funktion zweier Variabler, die zur Berechnung von mehreren statistischen Summen aus den Werten im X- und Y-Register dient. Die Ergebnisse werden automatisch in den Speicherregistern R_0 bis R_5 akkumuliert.

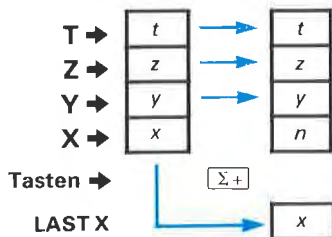


In R_0 bis R_5 zuvor gespeicherte, aus vorangegangenen Berechnungen oder Speicheroperationen resultierende Werte werden in nachfolgende Statistikakkumulationen mit einbezogen. Um sicherzustellen, daß vor einer neuen Summenbildung alle Akkumulationsregister auf Null gesetzt sind, ist \square CLEAR Σ zu drücken. (Dieser Befehl löscht den Stack und die Register R_0 bis R_5 .)

Durch Auslösen der $\Sigma +$ Taste werden die folgenden statistischen Summen in den angegebenen Datenspeicher-Registern abgelegt:

Register	Inhalt
R ₀	n : Anzahl der akkumulierten Datenelemente (Datenpaare). (n erscheint auch im angezeigten X-Register).
R ₁	Σx : Summe der x -Werte.
R ₂	Σx^2 : Summe der Quadrate der x -Werte.
R ₃	Σy : Summe der y -Werte.
R ₄	Σy^2 : Summe der Quadrate der y -Werte.
R ₅	Σxy : Summe der Produkte der x - und y -Werte.

Bei der Ausführung von $\Sigma +$ wird die zuvor im X-Register gespeicherte Zahl in das LAST X-Register gerettet, und im X-Register wird der neue n -Wert abgelegt. *Der Inhalt des Y-Registers bleibt unverändert.*



Bei der Eingabe einer neuen Zahl wird der n -Wert im angezeigten X-Register überschrieben; der Stack verschiebt sich nicht.

Wenn Sie ein statistisches Problem vorliegen haben, daß an Stelle von zwei Variablen (x und y) nur eine Variable (x) beinhaltet, müssen Sie sicherstellen, daß vor jeder Ausführung der $\Sigma +$ Funktion das Y-Register auf Null gesetzt ist. (Dies kann durch Auslösen von f CLEAR Σ vor Beginn der Summenbildung erreicht werden, da mit dieser Befehlsfolge sowohl der Stack (d. h. auch das Y-Register) als auch die Σ -Register (R₀ bis R₅) gelöscht werden.)

Datenmengen, die aus Folgen von annähernd gleichgroßen x -Werten (oder y -Werten) bestehen, sind besonders anfällig gegenüber Rundungsfehlern. Die Genauigkeit von statistischen Berechnungen

mit derartigen Daten kann gesteigert werden, wenn statt der ursprünglichen Daten die Differenzen der einzelnen Werte von einem ungefährem Durchschnittswert der Datenfolge eingegeben wird. Diese Zahl muß anschließend zum Resultat der jeweiligen Berechnung addiert werden. Wenn zum Beispiel die x -Werte aus der Folge 665999, 666000 und 666001 bestehen, sollten als Daten $-1,0$ und 1 eingegeben werden. Zu einem daraus berechneten \bar{x} muß anschließend 666000 addiert werden. In einigen Fällen kann der Rechner s , r , L.R. oder \hat{y} nicht mit zu dicht beieinander liegenden Daten berechnen; wenn dies versucht wird, erscheint im Display die Meldung „Error 3“. Dieser Fehler wird jedoch nicht auftreten, wenn die Daten wie oben beschrieben normalisiert werden.

Bemerkung: Im Gegensatz zur Speicherregister-Arithmetik lassen die $\Sigma +$ und $\Sigma -$ Funktionen einen Overflow in den Speicherregistern R_0 bis R_5 zu, ohne in der Anzeige „Error 1“ zu melden.

Jede der mit $\Sigma +$ oder $\Sigma -$ berechneten Statistiksummen kann in das angezeigte X-Register zurückgespeichert werden, in dem $\boxed{\text{RCL}}$ und die Nummer des Datenspeicher-Registers mit der gewünschten Summe gedrückt werden. Wenn Sie sowohl die Σx als auch die Σy Statistik zurückrufen wollen, können Sie $\boxed{\text{RCL}}$ $\Sigma +$ drücken. Dadurch wird simultan Σx aus R_1 in das X-Register und Σy aus R_3 in das Y-Register kopiert. (Durch Auslösen von $\boxed{\text{RCL}}$ $\Sigma +$ wird der Stack in der gleichen Weise nach oben verschoben, wie dies bei der aufeinanderfolgenden Eingabe zweier Zahlen der Fall wäre.)

Beispiel: Die in der Energieforschung tätige Wissenschaftlerin Helen I. Voltz vermutet einen möglichen Zusammenhang zwischen dem weltweiten Anstieg der Kohleproduktion in den Jahren 1972 bis 1976 und einem ähnlichen Zuwachs in der weltweiten Stromerzeugung im gleichen Zeitraum. Um ihre Studie mit Daten zu belegen, will Voltz ihren HP-11C zur Akkumulation der Produktionsmengen an Kohle und Strom benutzen. Gesucht sind Σx , Σx^2 , Σy und Σy^2 und Σxy für die paarweisen x - und y -Werte der Voltz'schen Daten.



Jahr	1972	1973	1974	1975	1976
Kohleproduktion (y) (in Milliarden Tonnen)	1,761	1,775	1,792	1,884	1,943
Stromerzeugung (x) (in Mrd. Megawattstd.)	5,552	5,963	6,135	6,313	6,713

Tastenfolge**Anzeige**f CLEAR Σ **0,0000**Löschen der statistischen Datenspeicher-Register (R_0 bis R_5 und Stack).

1,761 ENTER

1,76105,552 $\Sigma +$ **1,0000**

Daten für 1972.

1,775 ENTER

1,7750

Daten für 1973.

5,963 $\Sigma +$ **2,0000**

1,792 ENTER

1,7920

Daten für 1974.

6,135 $\Sigma +$ **3,0000**

1,884 ENTER

1,8840

Daten für 1975.

6,313 $\Sigma +$ **4,0000**

1,943 ENTER

1,9430

Daten für 1976.

6,713 $\Sigma +$ **5,0000**

RCL 1

30,6760Summe der x -Werte (Σx) aus Register R_1 .

RCL 2

188,9386Summe der Quadrate der x -Werte (Σx^2) aus Register R_2 .

RCL 3

9,1550Summe der y -Werte (Σy) aus Register R_3 .

RCL 4

16,7877Summe der Quadrate der y -Werte (Σy^2) aus Register R_4 .

RCL 5

56,2924Summe der Produkte der x - und y -Werte (Σxy) aus Register R_5 .**Korrektur fehlerhafter Eingaben**

Durch fehlerhafte Eingaben entstandene, inkorrekte Statistiksummen können sehr einfach korrigiert werden.

1. Geben Sie das fehlerhafte oder aus anderen Gründen zu entfernende Datenpaar in das X- und Y-Register ein.
2. Drücken Sie $\boxed{g} \boxed{\Sigma -}$; die Daten werden dann aus den verschiedenen Summen entfernt.
3. Geben Sie die korrekten Werte für x und y ein. (Auch wenn nur einer der Werte x und y zu korrigieren war, sind beide Werte zu entfernen und neu einzugeben.)
4. Drücken Sie $\boxed{\Sigma +}$.

Bemerkung: Beim Entfernen eines fehlerhaften (x, y) -Paares mit $\boxed{g} \boxed{\Sigma -}$ werden Rundungsfehler die durch die Summation der Werte auf die Inhalte der Akkumulationsregister R_1 bis R_5 entstehen können, nicht mitgelöscht. Folglich können die sich ergebende Resultate von den Ergebnissen unterscheiden, die entstehen würden, wenn das fehlerhafte Paar nicht mit $\boxed{\Sigma +}$ eingegeben und $\boxed{\Sigma -}$ wieder gelöscht worden wäre. Der Unterschied wird jedoch nicht sehr groß sein, solange sich das fehlerhafte Paar betragsmäßig nicht zu sehr vom korrekten Paar unterscheidet. In diesen Fällen ist es ratsam, die Berechnung neu zu beginnen, und die gesamte Dateneingabe (etwas sorgsamer) zu wiederholen.

Beispiel: Nach der Eingabe der letzten Daten erhält Voltz Informationen, nach denen die Kohleproduktion im Jahr 1976 1,946 Mrd. Tonnen statt 1,943 Mrd. Tonnen betragen hat. Benutzen Sie $\boxed{\Sigma -}$, um das fehlerhafte Wertepaar aus den akkumulierten Statistiksummen zu entfernen, und geben Sie das korrekte Datenpaar ein.

Tastenfolge

1,943 $\boxed{\text{ENTER}}$
6,713 $\boxed{g} \boxed{\Sigma -}$

Anzeige

1,9430
4,0000

Eingabe des zu ersetzenden Datenpaares. Die fehlerhaften Daten werden aus den Statistiksummen gelöscht, und der Zähler wird auf 4 gesetzt.

1,946 $\boxed{\text{ENTER}}$
6,713 $\boxed{\Sigma +}$

1,9460
5,0000

Eingabe und Akkumulation der korrekten Daten. Die Anzahl der akkumulierten Datenpaare ist wieder fünf.

Belassen Sie die bisher gespeicherten Statistikdaten in Ihrem Rechner; sie werden in den folgenden Beispielen benötigt.

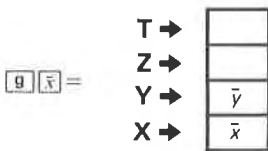
Mittelwert

Die \bar{x} -Funktion berechnet die Mittelwerte (die arithmetischen Mittel) der x - und y -Werte, die in den Registern R_1 und R_3 aufsummiert worden sind. Das Auslösen von $\boxed{g} \boxed{\bar{x}}$ beinhaltet die folgenden Operationen:

1. Der Inhalt der Stackregister wird in der gleichen Weise nach oben verschoben, wie dies bei $\boxed{RCL} \boxed{\Sigma +}$ (siehe Seite 57) der Fall ist.
2. Der Mittelwert der x -Werte (\bar{x}) wird mit Hilfe der Statistiksummen in den Registern R_1 (Σx) und R_0 (n) berechnet. Zur Berechnung des Mittelwerts der y -Werte (\bar{y}) wird der Inhalt der Register R_3 (Σy) und R_0 (n) verwendet. Die Berechnung erfolgt nach den folgenden Formeln:

$$\bar{x} = \frac{\Sigma x}{n} \qquad \bar{y} = \frac{\Sigma y}{n}$$

3. Die für \bar{x} und \bar{y} berechneten Werte werden im X- und Y-Register des Stacks abgelegt.



Beispiel: Die im vorangegangenen Beispiel akkumulierten statistischen 5-Jahres-Daten sollen zur Berechnung der durchschnittlichen Kohleproduktion und der durchschnittlichen Stromerzeugung im angegebenen Zeitraum benutzt werden.

Tastensequenz

$\boxed{g} \boxed{\bar{x}}$

Anzeige

6,1352

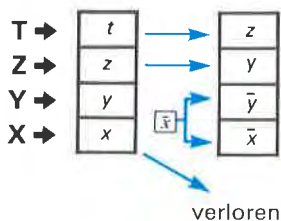
Durchschnittliche Stromerzeugung (Mittelwert der X-Register-Eingaben) im 5-Jahres Zeitraum.

$\boxed{x} \boxed{\geq} \boxed{y}$

1,8316

Durchschnittliche Kohleproduktion (Mittelwert der Y-Register-Eingaben) im 5-Jahres Zeitraum.

Die folgende Illustration verdeutlicht die Abläufe im Stack bei der Ausführung von \overline{x} (dabei wird unterstellt, daß der Stack wie etwa nach einer $\Sigma+$ Operation gesperrt ist):



Die jetzt in Ihrem HP-11C gespeicherten statistischen Daten werden auch für das nächste Beispiel benötigt.

Standardabweichung

Durch Drücken von \square \square wird die *Standardabweichung* (ein Maß für die Streuung um den Mittelwert) der akkumulierten statistischen Daten berechnet. Die zur Berechnung von s_x (Standardabweichung der akkumulierten x-Werte) und s_y (Standardabweichung der akkumulierten y-Werte) benutzten Formeln lauten:

$$s_x = \sqrt{\frac{n\sum x^2 - (\sum x)^2}{n(n-1)}} \quad s_y = \sqrt{\frac{n\sum y^2 - (\sum y)^2}{n(n-1)}}$$

Diese Formeln liefern aus den *Stichprobendaten* resultierende *beste Schätzungen* für die Standardabweichungen der Grundgesamtheit. Aus diesem Grund wird die aus diesen Formeln berechnete Standardabweichung gewöhnlich auch als Standardabweichung der *Stichprobe* bezeichnet. Das Auslösen von \square \square beinhaltet die folgenden Operationen:

1. Der Inhalt der Stackregister wird in der gleichen Weise nach oben verschoben, wie dies bei \square \square (siehe Seite 57) der Fall ist.
2. Die Standardabweichung der x-Werte (s_x) wird nach obiger Formel unter Benutzung der Statistiksummen in den Registern R_2 ($\sum x^2$), R_1 ($\sum x$) und R_0 (n) berechnet. Der für s_x berechnete Wert wird im X-Register abgelegt.

3. Die Standardabweichung der y -Werte (s_y) wird in analoger Weise aus dem Inhalt der Register R_4 (Σy^2), R_3 (Σy) und R_0 (n) ermittelt. Der für s_y berechnete Wert ist nach Abschluß der Rechnung im Y-Register verfügbar.

Beispiel: Die in den vorangegangenen Beispielen akkumulierten (und korrigierten) Statistiksummen sollen zur Berechnung der Standardabweichung bei der Kohleproduktion und der Stromerzeugung benutzt werden.

Tastenfolge

[g] [s]

Anzeige

0,4287

Standardabweichung bei der Stromerzeugung (X-Register-Eingaben) im betrachteten 5-Jahres-Zeitraum.

[x] [Σ] [y]

0,0800

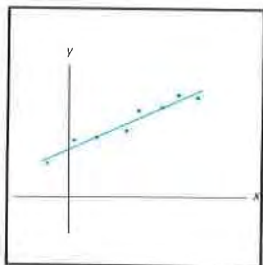
Standardabweichung bei der Kohleproduktion (Y-Register-Eingaben) im betrachteten 5-Jahres-Zeitraum.

Belassen Sie die Statistiksummen in ihrem HP-11C, da sie auch im nächsten Beispiel benötigt werden.

Wenn Ihre Daten nicht nur eine Stichprobe aus einer Grundgesamtheit, sondern die ganze Grundgesamtheit *selbst* darstellen, so ist die aus den Daten berechnete Standardabweichung die wahre Standardabweichung der Grundgesamtheit (und wird mit σ bezeichnet). Die Formel für die wahre Standardabweichung der Grundgesamtheit unterscheidet sich von der für die **[s]** Funktion benutzten um den Faktor $[(n - 1)/n]^{1/2}$. Die Differenz zwischen den beiden Werten ist gering und kann in den meisten Fällen ignoriert werden. Wenn Sie trotzdem den exakten Wert der Standardabweichung für eine Grundgesamtheit berechnen wollen, so erfordert dies nur geringen zusätzlichen Aufwand. Sie brauchen nur mittels der **[Σ+] [x]** Taste den Mittelwert (\bar{x}) der Daten zu den Daten addieren und anschließend **[g] [s]** zu drücken. Das Ergebnis ist dann die wahre Standardabweichung der Grundgesamtheit der ursprünglichen Daten.

Lineare Regression

Die lineare Regression ist ein statistisches Verfahren zum Auffinden derjenigen Geraden, die die Quadrate der Abstände von zwei oder mehreren Datenpaaren von der Geraden minimiert und damit einen Zusammenhang zwischen den beiden Variablen schafft. Nach der Akkumulation der Statistiksummen einer Folge von Datenpaaren in den Registern R_0 bis R_5 , können die Koeffizienten in der linearen Gleichung $y = Ax + B$ durch Drücken von $\boxed{f} \boxed{L.R.}$ berechnet werden. Die Bestimmung der Koeffizienten erfolgt nach der Methode der kleinsten Quadrate.



Um die lineare Regressionsfunktion Ihres HP-11C benutzen zu können, müssen Sie zuvor mit Hilfe der $\boxed{\Sigma +}$ Taste die Statistiksummen einer Folge von zwei oder mehr Datenpaaren akkumulieren. Danach beinhaltet die Ausführung von $\boxed{f} \boxed{L.R.}$ die folgenden Operationen:

1. Der Inhalt der Stackregister wird in der gleichen Weise nach oben verschoben, wie dies bei $\boxed{RCL} \boxed{\Sigma +}$ (siehe Seite 57) der Fall ist.
2. Die Steigung (A) und der y-Achsenabschnitt (B) der K.Q.-Geraden (K.Q. steht hier für Methode der kleinsten Quadrate) werden nach den folgenden Formeln berechnet:

$$A = \frac{n\sum xy - \sum x \sum y}{n\sum x^2 - (\sum x)^2}$$

$$B = \frac{\sum y \sum x^2 - \sum x \sum xy}{n\sum x^2 - (\sum x)^2}$$

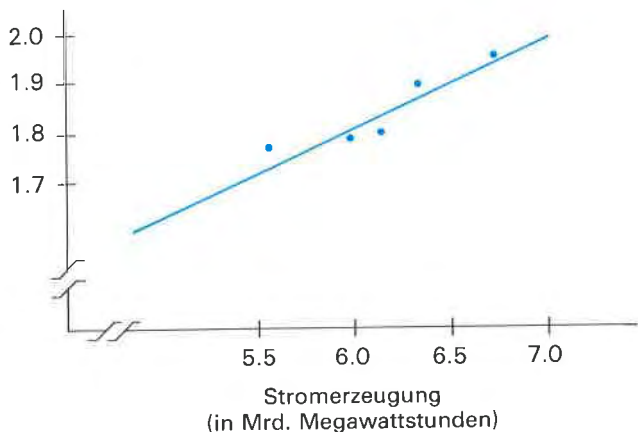
Die Steigung A wird im Y-Register, der y-Achsenabschnitt B im X-Register abgelegt.

$\boxed{f} \boxed{L.R.}$	T →	
	Z →	
	Y →	Steigung (A)
	X →	y-Achsenabschnitt (B)

Beispiel: Berechnen Sie die Steigung und den y-Achsenabschnitt der zu den Voltz'schen Daten gehörenden Regressionsgeraden.

Lösung: Eine mögliche Lösung bestünde darin, daß Voltz die Kohleproduktion, wie unten zu sehen, gegen die Stromerzeugung auftragen könnte. Erheblich einfacher ist es jedoch, mit dem HP-11C die Statistiksummen zu akkumulieren (was in unserem Fall bereits geschehen ist) und danach nur noch $\boxed{f} \boxed{L.R.}$ zu drücken.

Kohleproduktion
(Mrd. Tonnen)



Tastensequenz

\boxed{f} $\boxed{L.R.}$

Anzeige

0,7773

y -Achsenabschnitt der
K.Q.-Gerade.

$\boxed{x \approx y}$

0,1718

Steigung der K.Q.-Ge-
raden.

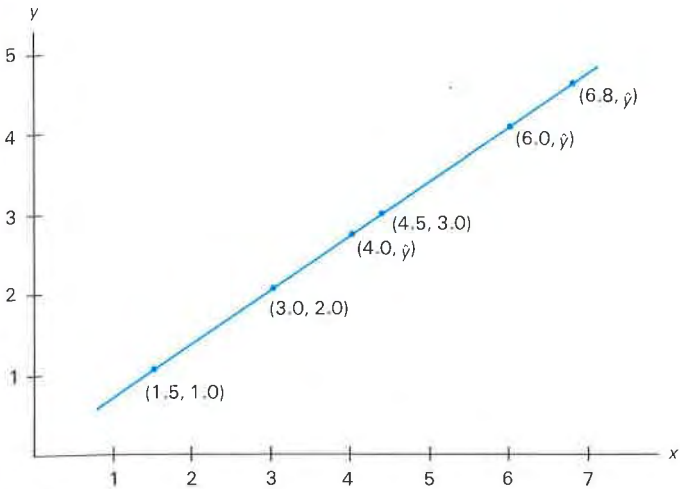
Belassen Sie die bisher akkumulierten Statistiksummen im Rechner, da sie im nächsten Beispiel noch benötigt werden.

Linearer Schätzwert und Korrelationskoeffizient

Bei der Ausführung der $\boxed{\hat{y}, r}$ Funktion wird der *lineare Schätzwert* (\hat{y}) im angezeigten X-Register und der *Korrelationskoeffizient* (r) im Y-Register abgelegt.

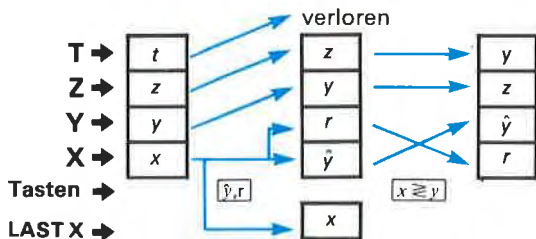
Linearer Schätzwert. Aus den in den Registern R_0 bis R_5 akkumulierten Statistiksummen wird nach Eingabe eines bekannten Werts für x durch Auslösen von $\boxed{f} \boxed{\hat{y}, r}$ ein geschätzter Wert für y (der mit \hat{y} bezeichnet wird) berechnet.

Korrelationskoeffizient. Bei der linearen Regression und der linearen Schätzung wird unterstellt, daß der Zusammenhang zwischen den x - und y -Werten bis zu einem gewissen Grad durch eine lineare



Funktion (d. h. durch eine Gerade) approximiert werden kann. Der Korrelationskoeffizient (r) ist ein Maß für die Güte dieser Approximation, d. h. ein Maß dafür, wie „eng“ gestreut die Datenpunkte um die Gerade liegen. Der Korrelationskoeffizient liegt zwischen -1 und $+1$. Bei $r = +1$ liegen alle Datenpunkte auf einer Geraden mit positiver Steigung, bei $r = -1$ auf einer Geraden mit negativer Steigung. Bei $r = 0$ ist eine Approximation der Datenpunkte durch eine Gerade nicht möglich. Der Korrelationskoeffizient wird aus den akkumulierten Statistiksummen in den Registern R_0 bis R_5 durch Drücken von $\boxed{f} \boxed{\hat{y}, r}$ berechnet. Der in der Anzeige dann angezeigte Wert des X-Registers ist ein \hat{y} -Wert (der ohne Bedeutung ist, sofern nicht zuvor ein x -Wert, wie oben beschrieben, vorgegeben wurde). Zur Anzeige des Werts des Korrelationskoeffizienten (r) müssen der Inhalt des X- und Y-Registers durch Drücken von $\boxed{x \cong y}$ vertauscht werden.

In der Beschreibung von LAST X in Abschnitt 2 wurde erwähnt, daß $\boxed{\bar{x}}$, \boxed{s} und $\boxed{L.R.}$ den letzten x -Wert nicht in das LAST X-Register kopieren. Da jedoch bei $\boxed{f} \boxed{\hat{y}, r}$ das Ergebnis \hat{y} aus dem Wert im X-Register berechnet wird, hat dies eine Verschiebung des Stacks nach oben und das Retten des x -Werts in das LAST X-Register zur Folge.



Beispiel: Auf der Grundlage der statistischen Daten der letzten Beispiele will Voltz die Kohleproduktion \hat{y} für 1977 voraussagen. Dazu gibt sie die für 1977 geschätzte Stromerzeugung (ein „bekannter“ x -Wert) ein und drückt $f \hat{y}, r$. Da der Korrelationskoeffizient für die Voltz'schen Daten automatisch mitberechnet wird, kann sie durch Drücken von $x \geq y$ eine Aussage darüber erhalten, wie gut ihre Daten durch eine Gerade approximiert werden können.

Tastenfolge

7,1417

Anzeige

7,1417

Die von Voltz für 1977 geschätzte Stromerzeugung.

 $f \hat{y}, r$

2,0046

Voraussichtliche Kohleproduktion im Jahr 1977.

 $x \geq y$

0,9211

Die Daten werden durch die Gerade relativ gut approximiert.

Kontrolle der Anzeige

Aufgrund des Permanentspeichers bleibt das von Ihnen gewählte Anzeigeformat auch nach einem Ausschalten des Rechners erhalten. Unabhängig von dem aktuellen Anzeigeformat werden alle Zahlen intern in Form einer Gleitkommazahl mit einer zehnstelligen Mantisse und einem zweistelligen Exponenten zur Basis 10 dargestellt. Beispielsweise wird bei einem Anzeigeformat mit vier Dezimalstellen die Kreisconstante π als 3,416 angezeigt. Intern wird π jedoch in der Form $3,141592654 \times 10^{00}$ gespeichert.

$$3,141592654 \times 10^{00}$$

Angezeigt werden nur diese Ziffern (auf die vierte Dezimalstelle gerundet) ...

... aber diese Ziffern sind intern ebenso vorhanden.

Wahl des Anzeigeformats

Der HP-11C verfügt über drei Anzeigemodi **[FIX]**, **[SCI]** und **[ENG]**, durch die mit Hilfe einer zusätzlichen Variablen (0 bis 9) das Anzeigeformat spezifiziert wird. Die unten zu sehende Illustration zeigt, in welchem Format die Zahl 123456 in jedem der drei Modes bei jeweils 4 Dezimalstellen angezeigt werden würde.

f	[FIX]	4 :	123.456,0000
f	[SCI]	4 :	1,2346 05
f	[ENG]	4 :	123,46 03

Festkommaformat

Bei Wahl von **[FIX]** (*fixed decimal*) werden alle Zahlen in einem Festkommaformat ohne Exponent angezeigt. Der Rechner verläßt automatisch den **[FIX]** Mode und springt in den **[SCI]** Mode, wenn eine Zahl zu groß oder zu klein ist, um in dem gewählten **[FIX]** Format angezeigt werden zu können. Der Rechner springt automatisch in den **[FIX]** Mode zurück, wenn eine Zahl wieder in dem gewählten Format angezeigt werden kann.

-1.234,567890

Vorzeichen 10-stellige Zahl

Das Festkommaformat wird mit den Tasten \boxed{f} \boxed{FIX} , gefolgt von der Anzahl der Dezimalstellen (0 bis 9), auf die gerundet werden soll, gewählt.

Tastenfolge
123,45678 \boxed{ENTER}

Anzeige
123,4568

Im Display wird die auf vier Dezimalstellen gerundete Zahl angezeigt. Intern wird jedoch der ursprüngliche Wert mit 10 Stellen gespeichert.

\boxed{f} \boxed{FIX} 6

123,456780

Die Anzeige wird aufgerundet, wenn die erste nicht angezeigte Zahl größer oder gleich 5 ist.

\boxed{f} \boxed{FIX} 0
 \boxed{f} \boxed{FIX} 4

123,
123,4568

Anzeige im bisher üblichen \boxed{FIX} 4 Format.

Wissenschaftliches Anzeigeformat

Bei Wahl von \boxed{SCI} (*scientific*) werden alle Zahlen in einem Gleitkommaformat mit einer Vorkomastelle angezeigt. Ein \boxed{SCI} Format wird mit den Tasten \boxed{f} \boxed{SCI} , gefolgt von der Anzahl der anzuzeigenden Dezimalstellen (0 bis 6) gewählt oder modifiziert. Zum Runden der angezeigten Zahl können auch 7, 8 oder 9 Dezimalstellen spezifiziert werden; im \boxed{SCI} Mode können jedoch nicht mehr als sechs Dezimalstellen angezeigt werden.*



Tastenfolge
123,4567895 \boxed{ENTER}

Anzeige
123,4568

Die Anzeige ist auf 4 Dezimalstellen gerundet.

\boxed{f} \boxed{SCI} 2

1,23 02

$1,23 \times 10^2$; abgerundete Anzeige.

* \boxed{SCI} 8 oder 9 und \boxed{ENG} 8 oder 9 werden bei der Eingabe als Programmbeefehle automatisch in \boxed{SCI} 7 bzw. \boxed{ENG} 7 umgewandelt.

Tastenfolge

f SCI 4

Anzeige

1,2346 02

1,2346 × 10²; aufgerundete Anzeige.

f SCI 6

1,234568 02

1,234568 × 10²; aufgerundete Anzeige.

Wie die obigen Beispiele bereits andeuten, wird die Rundung der Anzeige auf der letzten Dezimalstelle, die bei der Wahl des SCI Modes spezifiziert wurde, wirksam. Auch bei einer Angabe von mehr als sechs Stellen werden nur sechs Dezimalstellen angezeigt. Jedoch wird durch die Spezifikation von 7 oder mehr Dezimalstellen die Rundung in den auf das größte SCI Format folgenden (intern gespeicherten) Ziffernbereich verschoben. Wenn beispielsweise die Anzeige des letzten Beispiels benutzt wird, so erhöhen die folgenden Operationen nicht die Anzahl der Ziffern in der Anzeige, sondern verschieben nur die Rundung in den nicht mehr angezeigten Teil der Mantisse:*

Tastenfolge

f SCI 7

Anzeige

1,234567 02

Rundung auf der siebten Dezimalstelle; das Display kann im SCI Mode die siebte Dezimalstelle jedoch nicht anzeigen, daher bleibt die Rundung unsichtbar.

f SCI 8

1,234567 02

Rundung auf der achten Dezimalstelle; die Anzeige bleibt unverändert.

f SCI 9

1,234567 02

Rundung auf der neunten Dezimalstelle; die Anzeige bleibt unverändert.

* Wenn auf die letzte zulässige Ziffer des Anzeigeformats intern eine oder mehrere nachlaufende Neunen folgen, so kann die Rundung bei den Formaten SCI 7 und SCI 8 in den angezeigten Ziffernbereich fortgepflanzt werden. Beispielsweise wird im SCI 7 Format die Zahl 1,00000094 bei der Anzeige nicht gerundet, die Eingabe der Zahl 1,00000095 verursacht jedoch eine gerundete Anzeige (...95 auf ...99).

Technisches Anzeigeformat

Bei Wahl von **[ENG]** (*engineering*) werden alle Zahlen in einem Gleitkommaformat angezeigt, das sich in den folgenden Punkten vom **[SCI]** Format unterscheidet:

- Im technischen Anzeigeformat ist der Exponent zur Basis 10 immer ein Vielfaches von 3 (z. B. 10^3 , 10^6 , 10^{12}).
- Die bei der Formatwahl spezifizierte Ziffernanzahl bezieht sich auf die Anzahl der signifikanten Stellen, die nach der *führenden Ziffer* angezeigt werden sollen.



In der technischen Notation wird die erste signifikante Ziffer immer im Display angezeigt. Die nach dem Drücken von **[f] [ENG]** spezifizierte Zahl gibt die Anzahl der zusätzlichen signifikanten Stellen an, auf die die Anzeige gerundet werden soll. Zum Beispiel:

Tastenfolge	Anzeige	
[f] [ENG] 1	0,012345	Technisches Anzeigeformat. Die Anzeige wird auf eine signifikante Stelle nach der führenden Ziffer gerundet. Die Potenz von 10 ist ein Vielfaches von 3.
	12, -03	
[f] [ENG] 3	12,35	Die Anzeige ist auf drei signifikante Stellen nach der führenden Ziffer gerundet.
	12,35 -03	
[f] [ENG] 6	12,34500	
[f] [ENG] 0	10, -03	Die Anzeige wird auf die erste signifikante Ziffer gerundet.

Es ist zu beachten, daß im **[ENG]** Mode die Dezimalstellen automatisch verschoben werden, um den Exponenten als Vielfaches von 3 zu erhalten. Die folgenden Beispiele demonstrieren dieses Verhalten:

Tastenfolge

f [ENG] 2

Anzeige

12,3

-03 Anzeige des letzten Beispiels im [ENG] 2 Format.

10 [X]

123,

-03 Verschiebung der Dezimalstellen, um den Exponent als Vielfaches von 3 zu erhalten.

Eingabe von Exponenten

Die Taste [EEX] (*enter exponent*) ist zu benutzen, wenn ein Exponent als Teil einer Zahl eingegeben werden soll. Bei der Eingabe von Gleitkommazahlen mit Exponenten ist zunächst die Mantisse einzugeben, dann [EEX] zu drücken und danach der Exponent einzugeben. Tasten Sie zum Beispiel die Avogadro'sche Konstante ($6,0222 \times 10^{26} \text{ kmol}^{-1}$) ein:

Tastenfolge

f [FIX] 4

Anzeige

Rechner auf Anzeigeformat [FIX] 4 zurückschalten.

6,0222

6,0222

00 Die beiden Nullen sind die Aufforderung, den Exponenten einzugeben.

[EEX]

6,0222

00

26

6,0222

26

 $(6,0222 \times 10^{26})$

g [CLX]

0,0000

Anzeige löschen.

Um eine Zahl mit negativen Exponenten einzugeben, muß zuerst die Mantisse eingegeben und dann [EEX] gedrückt werden. Danach ist [CHS] (*change sign*) zu tasten, um das Vorzeichen des Exponenten umzukehren, und dann kann der Exponent eingegeben werden. Üben Sie diese Operation durch Eingabe des Planckschen Wirkungsquantum ($6,6262 \times 10^{-34}$ Joule/Sekunde) und multiplizieren Sie diesen Wert mit der Zahl 50:

Tastenfolge

6,6262 [EEX]

Anzeige

6,6262

00

[CHS]

6,6262

-00

3

-03

4

6,6262

-34

[ENTER]

6,6262

-34

50 [X]

3,3131

-32

Joule/Sekunde

Bemerkung: In den Exponentenbereich der Anzeige eingetastete Ziffern verschwinden beim Auslösen von **EEX** (bleiben jedoch intern gespeichert).

EEX verarbeitet keine Zahlen mit mehr als 7 Vorkommastellen und keine Dezimalzahlen mit mehr als fünf Nullen vor der ersten signifikanten Stelle. Um eine solche Zahl einzugeben, muß diese zuvor in eine Form mit einem geeigneten, höher- oder niederwertigeren Exponenten gebracht werden. Beispielsweise kann die Zahl $123456789,8 \times 10^{23}$ als $1234567,898 \times 10^{25}$, die Zahl $0,00000025 \times 10^{-15}$ als $2,5 \times 10^{-22}$ eingegeben werden.

Mantisse. Alle Zahlen im Stack und den Datenspeicher-Registern des Rechners werden intern als Gleitkommazahlen mit einer 10-stelligen Mantisse und einem zweistelligen Exponenten dargestellt. Zur Anzeige aller zehn Stellen der Mantisse einer Zahl im X-Register müssen Sie **f** CLEAR **PREFIX** drücken und dann die **PREFIX** Taste festhalten. Die Mantisse der momentan im X-Register gespeicherten Zahl wird solange im Display angezeigt, bis Sie die **PREFIX** Taste wieder loslassen.

Tastenfolge

f π

f CLEAR **PREFIX**

(**PREFIX** festhalten)

Anzeige

3,1416

3141592654

Rundung auf der zehnten Stelle

Wie Ihnen bereits bekannt ist, speichert der HP-11C jede Zahl, unabhängig von der Anzahl der im momentanen **FIX**, **SCI** oder **ENG** Format spezifizierten Dezimalstellen, intern mit 10 Stellen. Das Endresultat jeder Berechnung oder jeder Serie von Berechnungen wird auf 10 Stellen gerundet. Beispielsweise haben die Zahlen π und $2/3$ nichtabbrechende Dezimaldarstellungen (3,1415926535... und 0,6666666666...). Da der HP-11C nur endliche Approximationen (10 Stellen) derartiger Zahlen verarbeiten kann, tritt in der zehnten Stelle gegebenenfalls ein Rundungsfehler auf. Dieser Fehler kann im Verlauf von längeren Berechnungen anwachsen, bleibt jedoch in der Mehrheit aller Fälle außerhalb des Bereichs der signifikanten Stellen (einer gegebenen Anwendung). Die korrekte Abschätzung von Rundungsfehlerauswirkungen in einem Rechenprozeß erfordert Methoden der numerischen Analysis, die den Rahmen dieses Handbuchs sprengen würden.

Teil II

Programmierung des HP-11C

Grundlagen der Programmierung

Kurzdarstellung der Grundlagen

Was ist ein Programm?

Ein Programm besteht aus einer Folge von Tastenfolgen, die vom Rechner gespeichert wird. Sie können ein gespeichertes Programm durch ein oder zwei zusätzliche Befehle beliebig oft ausführen lassen. Der Stack reagiert auf die Instruktionen des ablaufenden Programms in der gleichen Weise, wie er auf diese Instruktionen reagieren würde, wenn Sie sie manuell über die Tastatur eingeben würden. Das am Ende einer Programmausführung angezeigte Resultat ist identisch mit dem Ergebnis bei manueller Instruktionausführung. Zum Erlernen der HP-11C Programmierung ist keinerlei Programmiererfahrung notwendig.

NACH EINEM BESTIMMTEN SCHEMA ABLAUF. RECHENVORGANG

Notwendigkeit von Programmen

Programme ersparen Ihnen Zeit bei sich häufig wiederholenden Berechnungen. Wenn Sie einmal eine Tastenfolge, aus der sich der Algorithmus zur Lösung eines bestimmten Problems zusammensetzt, niedergeschrieben und im Rechner aufgezeichnet haben, brauchen Sie keine Zeit mehr mit der manuellen Eingabe der einzelnen Befehle zu vergeuden. Der Rechner übernimmt für Sie die Lösung Ihrer Problemstellungen. Eine Kontrolle des Algorithmus Ihres Programms ist sehr einfach; folglich können Sie mehr Vertrauen zum berechneten Endresultat haben, weil die Gefahr, bei der Eingabe der Instruktionen eine falsche Taste gedrückt zu haben, nicht mehr gegeben ist.

Auf den folgenden Seiten werden die Programmierungsmöglichkeiten des HP-11C beschrieben. Weitere Hinweise, die Ihnen bei der Planung und Entwicklung von Programmen behilflich sein können, sind auf Seite 206, Struktur, in Teil III dieses Handbuchs zu finden.

Programmkontrolle

Automatische Speicherumwandlung

Die Aufteilung des verfügbaren Speicherplatzes in Programmspeicher und Datenspeicher wird automatisch vom Rechner kontrolliert.

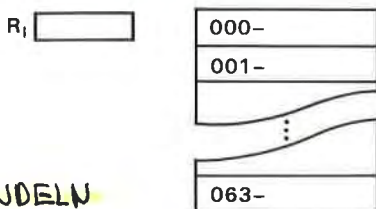
Aufgrund dieser internen Kontrolle werden das Display oder Tastaturoperationen nicht von der Speicherumwandlung berührt. Für Sie ist lediglich von Interesse:

1. Wodurch wird eine Umwandlung verursacht?
2. Was geschieht im Speicher während einer Umwandlung?

Nach dem Löschen des Programm- oder des Permanentspeichers besteht die Speicher-Konfiguration des Rechners aus 21 Datenspeicher-Registern und 63 Zeilen verfügbaren Programmspeicher.

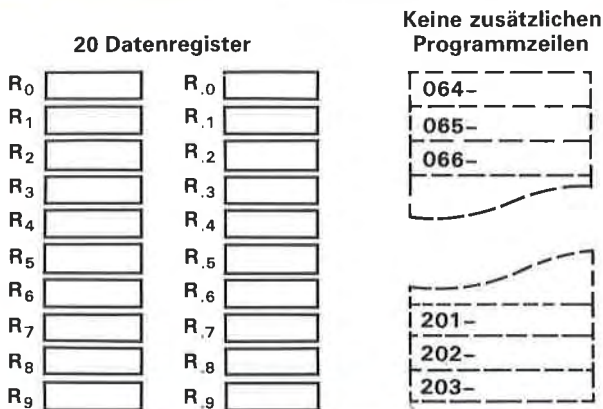
SPEICHER-KONFIGURATION DES HP-11C

Permanentspeicher



UMWANDELN

Konvertierbarer Speicher (Standardkonfiguration)



Bei der Eingabe von Instruktionen in den Programmspeicher werden diese sequentiell im verfügbaren Speicherbereich abgelegt. Wenn alle 63 Zeilen des ursprünglichen Programmbereichs belegt sind, und ein 64. Programmbefehl eingegeben wird, so wandelt der Rechner das Datenspeicher-Register R_9 in sieben Zeilen zusätzlichen Programmspeicher um. Dadurch wird Platz für den 64. Befehl (und sechs weitere) geschaffen. Die Speicher-Konfiguration des HP-11C besteht dann aus 70 Zeilen Programmspeicher und 20 Datenspeicher-Registern. Bei Eingabe eines 71. Programmbefehls würde automatisch das Datenspeicher-Register R_8 in sieben zusätzliche Zeilen Programmspeicher umgewandelt werden. Dieses Verfahren kann solange wiederholt werden, bis alle 20 konvertierbaren Datenspeicher-Register (R_9 bis R_0 und R_9 bis R_0) in Programmspeicher umgewandelt sind.

Nach Umwandlung aller konvertierbaren Datenspeicher-Register in Programmspeicher besteht die Speicher Konfiguration aus 203 Programmzeilen und einem Datenspeicher-Register (dem Indexregister R_1). Die folgende Tabelle zeigt die Zuordnung der Programmzeilen mit den entsprechenden Speicherregistern.

R_9	064—070	R_9	134—140
R_8	071—077	R_8	141—147
R_7	078—084	R_7	148—154
R_6	085—091	R_6	155—161
R_5	092—098	R_5	162—168
R_4	099—105	R_4	169—175
R_3	106—112	R_3	176—182
R_2	113—119	R_2	183—189
R_1	120—126	R_1	190—196
R_0	127—133	R_0	197—203

Zuordnung Speicherregister/Programmspeicher

Das Löschen von aufeinanderfolgenden Programmzeilen an beliebigen Stellen des Programmspeichers bewirkt die automatische Rückumwandlung von Programmspeicherbereichen in Datenspeicher-Register. Die Rückumwandlung verläuft in der umgekehrten Reihenfolge der Umwandlung.

Reihenfolge der Konvertierung Datenspeicher/Programmspeicher

$R_{9} R_{8} R_{7} \dots R_{1} R_{0} R_{9} R_{8} \dots R_{1} R_{0}$

Reihenfolge der Konvertierung Programmspeicher/Datenspeicher

Zusätzliche Beschreibungen und Illustrationen sind in Anhang C, Automatische Speicherumwandlung, zu finden.

MEM

Die aktuelle Speicheraufteilung können Sie sich jederzeit durch Drücken von **9** **MEM** anzeigen lassen. Solange Sie **MEM** gedrückt lassen, wird die Anzahl der vor der nächsten Speicherregister-Konvertierung verfügbaren Programmzeilen und der Name des nächsten, zu konvertierenden Datenspeicher-Registers angezeigt. (Eine graphische Darstellung der **MEM** Operation ist in Anhang C, Automatische Speicherumwandlung, zu finden.)



MEM Anzeige bei gelöschtem Programmspeicher

Tastencodes und Zeilennummern

Wenn Sie den Rechner auf Program Mode schalten und Programmbefehle eingeben, wird der Tastencode der gedrückten Tasten und die Zeilennummer des vollständigen Befehls im Display angezeigt. Der Tastencode eines Befehls besteht aus ein, zwei oder drei Elementen, je nach dem wieviele Tasten für diesen Befehl zu betätigen sind. Jedes Element besteht aus zwei Ziffern, die die Reihen-/Spalten-Position der durch das Element repräsentierten Taste beschreiben. (Eine Ausnahme bilden die numerischen Tasten, die durch ein einstelliges Element dargestellt werden.)

Zeilennummer } 

	017	42,	21,	11
Tastenreihe	4	2	1	
Tastenspalte	2	1	1	

Abgekürzte Tastenfolgen

Im Run oder Program Mode wird die Vorwahltaste [f], die in den Tastenfolgen für einige Befehle enthalten sein sollte, nicht immer benötigt. (Überflüssige [f] Tasten, die als Teil eines Programmbefehls betätigt wurden, erscheinen nicht im Tastencode dieses Befehls.) Beispielsweise liefert das Auslösen von [STO] [RAN #] die gleichen Ergebnisse wie [STO] [f] [RAN #]. Hinweise auf weitere Tasten, die in abgekürzten Tastenfolgen verwendet werden können, sind in den entsprechenden Abschnitten zu finden.

Funktionen zur Programmkontrolle

Program/Run. Durch Drücken von [g] [P/R] wird der Rechner wahlweise von Program auf Run Mode oder umgekehrt geschaltet. Wenn sich der Rechner im Program Mode befindet, erscheint im Display die PRGM Statusanzeige, und es können Programmbefehle eingegeben oder gelöscht werden. Im Run Mode können entweder im Programmspeicher gehaltene Programme oder einzelne Tastenfunktionen ausgeführt werden.

Clear Program Memory. Im *Program Mode* bewirkt die Tastenfolge [f] CLEAR [PRGM], daß sämtliche Programme im Permanentspeicher gelöscht werden, und daß der Permanentspeicher in 21 Datenspeicher-Register und 63 Zeilen verfügbaren Programmspeicher aufgeteilt wird. Im *Run Mode* wird der Rechner durch [f] CLEAR [PRGM] auf Zeile 000 gesetzt, der Programmspeicher wird jedoch nicht gelöscht.

Go to Line 000. Durch Drücken von [GTO] [.] 000 wird der Rechner auf Zeile 000 (Beginn des Programmspeichers) positioniert. Dies gilt sowohl im Run als auch im Program Mode.

Labels. Die Labels des HP-11C sind *Adressen* für Programme, Programmverzweigungen und Unterprogramme. Die alphanumerischen Labels ([A] bis [E]) und die numerischen Labels (0 bis 9)

werden in den Programmspeicher durch Auslösen der Tasten \boxed{f} \boxed{LBL} (*label*) und der gewünschten alphanumerischen oder numerischen Taste eingegeben. Im Run Mode wird ein mit einem alphanumerischen Label adressiertes Programm durch Drücken der Vorwahltaste \boxed{f} und der Label Taste gestartet. Die Labels 0 bis 9 können ebenfalls zur Programmadressierung benutzt werden, sollten jedoch für Programmunterteilungen (Sprünge, Unterprogramme) reserviert bleiben. Numerische Labels können durch Drücken von \boxed{GSB} und der gewünschten Zahlentaste angesprungen werden.

Return. Wenn der \boxed{RTN} (*return*) Befehl das Ende eines Programms markiert, wird die Programmausführung angehalten und zur Zeile 000 gesprungen.

Run/Stop. Der Befehl $\boxed{R/S}$ (*Run/Stop*) bewirkt in einem ablaufenden Programm die Unterbrechung der Programmausführung. Durch erneutes Auslösen von $\boxed{R/S}$ wird die Programmausführung mit der Zeile im Programmspeicher fortgesetzt, auf die der Rechner momentan positioniert ist.

Pause. In einem ablaufenden Programm bewirkt der Befehl \boxed{f} \boxed{PSE} (*pause*) ein Anhalten der Programmausführung für ungefähr eine Sekunde. Dadurch wird eine Anzeige des sich momentan im X-Register befindlichen Werts ermöglicht. Anschließend wird die Programmausführung fortgesetzt.

User Mode

Der User Mode ist eine Servicefunktion, durch die Sie sich bei der Ausführung von Programmen das Bedienen bestimmter Tasten ersparen können. Durch Drücken von \boxed{f} \boxed{USER} werden die primären mathematischen Zuordnungen und die durch die Vorwahltaste \boxed{f} anwählbaren Alpha-Zuordnungen der in der ersten Reihe angeordneten Tasten des Rechners vertauscht. Solange der User Mode eingeschaltet ist, erscheint im Display die **USER** Statusanzeige.

0,0000
USER

User Mode Statusanzeige

Vorwahl \boxed{f}	→	A	B	C	D	E
Primär	→	\sqrt{x}	e^x	10^x	1^x	$1/x$
Vorwahl \boxed{g}	→	x^2	LN	LOG	%	$\Delta\%$

Im Run Mode bewirkt dieser Austausch, daß Programme mit den Labels \boxed{A} bis \boxed{E} gestartet werden können, indem lediglich die

entsprechende Alpha-Funktionstaste betätigt wird, ohne vorher die Vorwahltaste **f** drücken zu müssen.

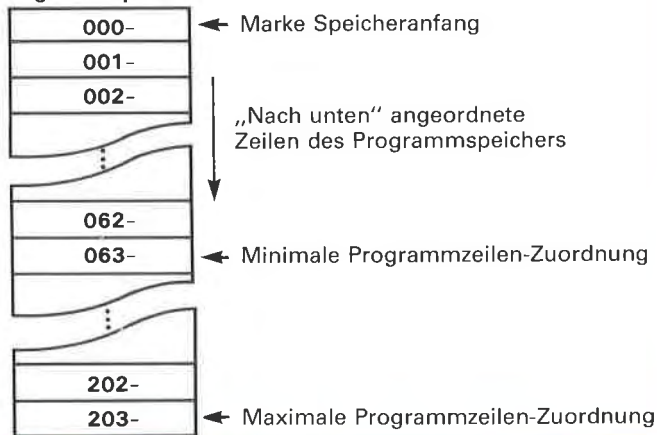
Bemerkung: Um ein versehentliches Starten von Programmen oder die Programmierung falscher Funktionen zu vermeiden, sollte der User Mode nur dann aktiviert werden, wenn dies für eine bestimmte Anwendung notwendig oder wünschenswert ist.

Zur Deaktivierung des User Mode sind wieder die Tasten **f** **USER** zu drücken.

Programmspeicher

Wie Sie bereits am Beispiel des Wärmeverlustprogramms am Anfang dieses Handbuchs gesehen haben, werden die zur manuellen Berechnung einer Lösung benötigten Tastenfolgen auch zum Schreiben eines Programms für die automatische Berechnung der Lösung verwendet. Diese Tastenfolgen werden im *Programmspeicher* des Rechners gespeichert. Drücken Sie nun **GTO** **.** **000**, um den Rechner an den Anfang des Programmspeichers zu setzen. Falls dies noch nicht bereits geschehen ist, schalten Sie den Rechner durch Drücken von **9** **P/R** auf Program Mode. (Beachten Sie, daß in der Anzeige die PRGM Statusanzeige erscheint, solange sich der Rechner im Program Mode befindet.) In der Anzeige sollte nun **000-**, die Marke für den Anfang des Programmspeichers, zu sehen sein.

Programmspeicher



Der Programmspeicher ist von den Stackregistern, dem LAST X und dem Indexregister, sowie von den nicht konvertierten Datenspeicher-Registern getrennt. Im Program Mode gibt die in der linken Seite des Display's angezeigte Zahl die Zeilennummer der Zeile im Programmspeicher, in der der Rechner momentan steht, an. Wenn Sie \boxed{f} CLEAR \boxed{PRGM} und danach \boxed{f} \boxed{LBL} \boxed{A} (die ersten Tastenfolgen für das Wärmeverlustprogramm, siehe Seite 12) drücken, so erscheint im Display:

001-42.21.11

Zeilennummer
↑
↑
Tastencode

Wie durch die Zahl 001 auf der linken Seite der Anzeige angedeutet wird, steht der Rechner nun in Zeile 001 des Programmspeichers. Die anderen Zahlen in der Anzeige sind die Tastencodes derjenigen Tastenfolgen, die in diese Programmzeile geladen worden sind. Drücken Sie nun 3. Im Display sollte jetzt angezeigt werden:

002- 3

Zeilennummer
↑
↑
Tastencode

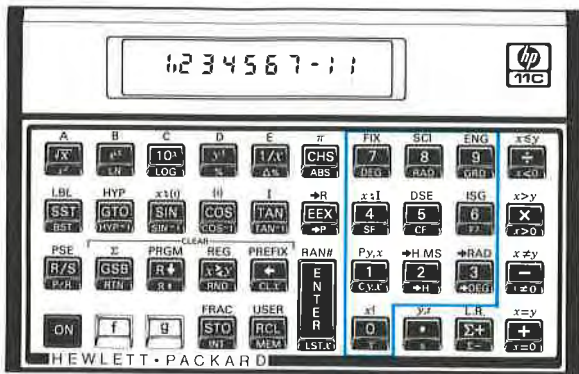
Die Zahl 002 auf der linken Seite der Anzeige besagt, daß Sie sich nun in Zeile 2 des Programms befinden.

Jede Zeile des Programmspeichers stellt einen einzigen Programm-befehl dar, egal ob dieser Befehl aus einem, zwei oder drei Tasten besteht. Daher kann eine Zeile des Programmspeichers einen aus einer einzigen Tastenfolge bestehenden Befehl wie \boxed{CHS} enthalten, während eine andere Zeile des Programmspeichers einen aus drei Tastenfolgen bestehenden Befehl wie \boxed{STO} $\boxed{+}$ 6 (Addition der Zahl im X-Register zu der Zahl in R_6) enthalten kann. Die Tastenfolgen für Programmbefehle werden im Rechner durch Tastencodes dargestellt.

Interpretation von Tastencodes

Die meisten Tastencodes für die Tastenpositionen des HP-11C werden durch eine einfache Reihen/Spalten-Matrix bestimmt. Die Tastenreihen tragen die Nummern 1 bis 4. Die Tastenspalten sind von 1 bis 10 durchnummeriert. (Die zehnte Spalte wird in den Tastencodes des HP-11C durch eine 0 dargestellt, d.h. „20“ bedeutet

beispielsweise Zeile 2, Spalte 10 und entspricht der \times Funktion.) Die einzigen Tastenpositionen, die nicht in dieses Matrixschema passen, sind die den Zahlentasten 0 bis 9 zugeordneten Funktionen. Der Code für die Funktionen dieser Tasten besteht lediglich aus der einzelnen Ziffer auf der Oberseite der entsprechenden Taste.



einstellige Tastencodes

Eine Programmzeile kann aus einem bis drei Tastencode-Elementen bestehen.

053- 23 054- 43 33 055-44. 40. 5

$\boxed{\text{SIN}}$ $\boxed{9}$ $\boxed{\text{R}\uparrow}$ $\boxed{\text{STO}}$ $\boxed{+}$ 5

Reihe 2 Reihe 4 Reihe 3 Reihe 4 Reihe 4 Zahlentaste „5“

Spalte 3 Spalte 3 Spalte 3 Spalte 4 Spalte 10

Lassen Sie uns nun noch einen Blick auf die eben eingegebenen Programmbefehle werfen (s. Seite 81). Drücken Sie $\boxed{9}$ $\boxed{\text{BST}}$. Das Display zeigt Ihnen nun die erste Zeile des Wärmeverlustprogramms:

001-42.21.11

Zeilennummer $\underbrace{\hspace{1.5cm}}$ $\underbrace{\hspace{1.5cm}}$ Tastencode

Der Zahlencode 001- bezeichnet die Nummer der Zeile im Programmspeicher. Das nächste Zahlenpaar, 42, steht für \boxed{f} (Reihe 4, Spalte 2); 21 repräsentiert $\boxed{\text{LBL}}$ (Reihe 2, Spalte 1), und 11 stellt $\boxed{\text{A}}$ (Reihe 1, Spalte 1) dar. Auf diese Weise werden alle programmier-

baren Tastenfolgen mit Ausnahme der den Zahlentasten zugeordneten Funktionen durch zweistellige Tastencodes dargestellt. Dies soll durch ein weiteres Beispiel verdeutlicht werden. Drücken Sie **[SST]**. Im Display des HP-11C wird nun der Tastencode für den zweiten Befehl des Wärmeverlustprogramms angezeigt:



002- ist wieder die Nummer der Programmzeile; 3 bedeutet in diesem Fall die Zahl 3. Im Schema rechts werden einige Tastencodes, die die Zahlentaste „3“ enthalten, aufgeführt.

Tasten	Tastencode
[f] [→RAD]	→ 42 3
3	→ 3
[g] [→DEG]	→ 43 3

Die verbleibenden Tastenfolgen für das Wärmeverlustprogramm werden im folgenden mit den zugehörigen Anzeigen aufgelistet. Drücken Sie nacheinander jede Taste noch einmal, und überprüfen Sie die im Display angezeigten Tastencodes.

Tastensequenz	Anzeige
[X]	003- 20 Taste in Zeile 2, Spalte 10.
1	004- 1 Zahlentaste „1“.
[.]	005- 48 Taste in Zeile 4, Spalte 8.
7	006- 7 Zahlentaste „7“.
8	007- 8 Zahlentaste „8“.
[X]	008- 20 Taste in Zeile 2, Spalte 10.
[g] [RTN]	009- 43 32 Ende des Programms.
[g] [P/R]	Rechner auf Run Mode schalten.

Operationen zur Programmierung

Im vorangegangenen Unterabschnitt wurden einzelne Programm-befehle beschrieben. Im folgenden soll nun der Ablauf eines vollständigen Programmierungsprozesses dargestellt werden. Die einzelnen Schritte der Programmierung werden anhand eines kleinen Programms illustriert.

Angenommen, Sie wollen über die Tastatur Ihres HP-11C manuell die Fläche eines Kreises nach der Formel $A = \pi r^2$ berechnen. Sie würden dabei als erstes den Radius r eintasten und diesen Wert anschließend mit **[g]** **[x²]** quadrieren. Als nächstes würden Sie mit **[f]**

π den Wert der Kreiskonstanten π in die Anzeige rufen. Abschließend würden Sie \times drücken, um den quadrierten Radius mit der Zahl π zu multiplizieren. Damit ergibt sich die Tastenfolge:

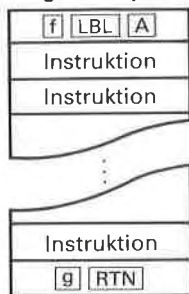


Anfang und Ende eines Programms

Der Anfang eines Programms wird durch den Befehl f LBL (*label*) gekennzeichnet, gefolgt von einer der numerischen oder Alpha-Tasten, durch die das gewünschte Label spezifiziert wird. Die Verwendung von Labels ermöglicht es, gleichzeitig mehrere Programme oder Teile von Programmen im Rechner zu speichern. Diese Programme können Sie dann jederzeit in beliebiger Reihenfolge ausführen lassen.

Das Ende eines Programms kann durch den Befehl g RTN (*return*) definiert werden. In einem ablaufenden Programm bewirkt eine in dieser Weise verwendete RTN Anweisung einen Sprung in Programmzeile 000 und damit die Terminierung der Programmausführung.

Programmspeicher



LBL kennzeichnet den Anfang des Programms.

RTN kennzeichnet das Ende des Programms.

Bemerkung: Wenn beim Ablauf eines Programms das Ende des *belegten* Programmspeichers erreicht wird, hat dies die gleiche Wirkung wie eine RTN Anweisung. Dies hat zur Folge, daß, wenn die *letzte* Instruktion im belegten Teil des Programmspeichers ein g RTN ist, dieser Befehl weggelassen werden kann, wodurch eine Programmzeile an Speicherplatz eingespart wird.

Das fertige Programm

Das fertige Programm zur Berechnung einer Kreisfläche bei vorgegebenem Radius sieht wie folgt aus:

- LBL A Weist den Namen „A“ zu und definiert den Programm-
anfang.
- x² Quadriert den Radius (Stack bleibt unverändert).
- π Lädt die Konstante π in das angezeigte X-Register (Stack
wird nach oben verschoben).
- Multipliziert den quadrierten Radius mit π (Stack wird
nach unten verschoben) und zeigt das Ergebnis an.
- RTN Definiert das Ende des Programms; der Rechner springt
in Zeile 000 und beendet die Programmausführung.

Laden eines Programms

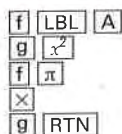
Ein Programm kann vor oder hinter bereits existierende Programme in den Speicher geladen werden. Wenn ein neues Programm vor ein bereits existierendes Programm (durch Sprung auf Zeile 000 und Eingabe der neuen Befehle) geladen wird, werden die Befehle des existierenden Programms zeilenweise im Speicher „nach unten“ verschoben.

Dor Rechner ist wie folgt auf das Laden des oben beschriebenen Programms zur Kreisflächenberechnung vorzubereiten:

1. Drücken Sie P/R, um den Rechner auf Program Mode zu schalten. In der Anzeige erscheint die PRGM Statusanzeige.
2. Drücken Sie CLEAR PRGM, um im Programmspeicher alle alten, unerwünschten Programme zu löschen. (Wenn Sie ein bereits gespeichertes Programm erhalten wollen, sind an Stelle von CLEAR PRGM die Tasten GTO . 000 zu drücken. Durch den Befehl GTO . 000 wird der Rechner auf Zeile 000 gesetzt, ohne daß der Inhalt des Programmspeichers verändert wird.)

Durch Anzeige der Ziffern 000 im linken Bereich des Display's teilt Ihnen der Rechner mit, daß er sich am Anfang des Programmspeichers befindet.

Das Programm zur Kreisflächenberechnung wird durch Drücken der folgenden Tasten geladen:



Drücken Sie die erste Taste, **f**, des Programms:

Tastenfolge	Anzeige
f	000—

Sie sehen, daß sich die Anzeige des Programmspeichers nicht verändert hat. Die Anzeige bleibt solange unverändert, bis alle zum vollständigen Befehl gehörenden Tasten gedrückt sind. Drücken Sie nun die restlichen Tasten des ersten Befehls:

Tastenfolge	Anzeige
LBL	000—
A	001—42.21.11

f **LBL** **A** wird in den Programmspeicher geladen.

Die Anzeige von Tastencodes und einer neuen Zeilennummer besagt, daß eine vollständige Operation in diese Zeile geladen wurde. Es ist zu beachten, daß, solange die Eingabe einer vollständigen Instruktion (egal, ob aus einer, zwei oder drei Tasten bestehend) nicht abgeschlossen ist, keinerlei Informationen in den Programmspeicher geladen werden.

Laden Sie jetzt den Rest des Programms:

Tastenfolge	Anzeige
g x²	002— 43 11
f π	003— 42 16
x	004— 20
g RTN	005— 43 32

Das Programm zur Berechnung des Flächeninhalts eines Kreises bei gegebenem Radius ist jetzt in den Programmspeicher Ihres HP-11C geladen.

Ausführen eines Programms

Programme können nur im Run Mode ausgeführt werden. Um das im obigen Beispiel geladene Programm zur Kreisflächenberechnung ausführen zu können, muß der HP-11C durch Drücken von **g** **[P/R]** auf Run Mode geschaltet werden.

Zur Ausführung eines Programms brauchen Sie jetzt nur noch die eventuell erforderlichen Daten einzugeben, und dann \boxed{f} und diejenige Alpha-Taste (\boxed{A} bis \boxed{E}), mit der Ihr Programm gelabelt ist, auszulösen. Im Fall des Kreisflächenprogramms ist der Radius einzugeben und dann \boxed{f} \boxed{A} zu drücken.

Beispiel: Gesucht sind die Flächeninhalte von Kreisen mit den Radien 7,5 cm, 9 mm und 15,3 m:

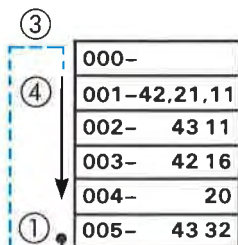
Tastenfolge	Anzeige
7,5 \boxed{f} \boxed{A}	176,7146 cm ²
9 \boxed{f} \boxed{A}	254,4690 mm ²
15,3 \boxed{f} \boxed{A}	735,4154 m ²

Wie sucht der Rechner nach einem Label? Beim Umschalten des HP-11C auf Run Mode befand sich der Rechner in Zeile 005 des Programmspeichers (der letzten Zeile, die beim Laden des Programms mit einer Instruktion belegt wurde). Durch Auslösen der Tasten \boxed{f} \boxed{A} beginnt der Rechner den Programmspeicher ab Zeile 005 sequentiell nach einem \boxed{LBL} \boxed{A} Befehl zu durchsuchen. Während der Suche werden keine Programmbehle ausgeführt.

Da in unserem Fall

1. die Zeile 005 den \boxed{LBL} \boxed{A} Befehl nicht enthält und
2. keine weiteren Programmzeilen belegt sind,

springt der HP-11C in Programmzeile 000 und setzt von dort aus die Suche fort. In Zeile 001 wird der gesuchte \boxed{LBL} \boxed{A} Befehl gefunden, und die Programmausführung beginnt.



② Ende des belegten Programmspeichers

Ausführung der Programmbehle. Der Rechner führt die Instruktionen in der Reihenfolge der Eingabe aus; d.h. zuerst die \boxed{g} $\boxed{x^2}$ Operation in Zeile 002, danach den \boxed{f} $\boxed{\pi}$ Befehl in Zeile 003, usw. Die Ausführung wird fortgesetzt, bis ein \boxed{g} \boxed{RTN} Befehl oder ein $\boxed{R/S}$ (run/stop) Befehl gefunden wird, oder das Ende des belegten Bereichs des Programmspeichers erreicht wird. Da sich in Zeile 005 eine \boxed{g} \boxed{RTN} Anweisung befindet, wird nach Zeile 000 gesprungen, und die Programmausführung wird angehalten. Anschließend wird das Resultat der Berechnung, der Wert im X-Register, angezeigt. Bei

Programmen mit längeren Ausführungszeiten blinkt während der Programmausführung im Display die Meldung „**running**“.

Nichtprogrammierbare Funktionen. Solange sich der Rechner im Program Mode befindet (PRGM Statusanzeige an), kann fast jede Funktion des Tastenfeldes als Programmbehehl im Programmspeicher abgelegt werden. Eine Ausnahme bilden die folgenden Anweisungen, die als nichtprogrammierbare Funktionen ausgelegt sind:

f	CLEAR	PRGM
f	CLEAR	PREFIX
GTO	.	nnn
f	USER	

g	P/R
g	MEM
.	/ ON

SST	
←	
g	BST

Verwendung des User Mode

Schalten Sie den Rechner nun auf User Mode und starten Sie das Kreisflächenprogramm erneut. Betätigen Sie anschließend die vom User Mode betroffenen Tastaturfunktionen.

Tastensequenz

f	USER
---	------

Anzeige
User

Aktivierung des User Mode; **USER** Statusanzeige erscheint.

7,5	A
-----	---

176,7146

Im User Mode werden **A**

9	A
---	---

254,4690

bis **E** zu den Primärfunktionen der entsprechenden Tasten.

15,3	A
------	---

735,4154

4	f	\sqrt{x}
---	---	------------

2,0000

Im User Mode werden die mathematischen Funktionen der obersten Zeile zu alternativen **f** Präfixfunktionen der entsprechenden Tasten.

1	f	e^x
---	---	-------

2,7183

1	f	10^x
---	---	--------

10,0000

2	ENTER
---	-------

2,0000

8	f	y^x
---	---	-------

256,0000

-	5	f	$1/x$
---	---	---	-------

2,0000

f	USER
---	------

2,0000

Deaktivierung des User Mode

Unterbrechung der Programmausführung

Sie werden während eines Programmablaufs häufig Anlaß haben, das Programm zu unterbrechen, um entweder Daten einzugeben

oder Ergebnisse in der Anzeige zu betrachten, bevor das Programm fortgesetzt wird. Zur Programmunterbrechung können zwei Tasten **R/S** (*run/stop*) und **PSE** (*pause*) benutzt werden.

Geplantes Anhalten der Programmausführung

Die **R/S** (*run/stop*) Funktion kann sowohl als Befehl in einem Programm wie auch als Tastaturoperation verwendet werden.

Die Verwendung als Tastaturoperation hat die folgende Wirkung:

1. Wenn sich ein Programm in der Ausführung befindet, wird durch **R/S** die Ausführung unterbrochen.
2. Wenn eine Programmausführung bereits unterbrochen ist oder sich ein Programm noch gar nicht in der Ausführung befindet, wird durch **R/S** die Programmausführung gestartet. Die Ausführung beginnt mit der auf den **R/S** Befehl folgenden Programmzeile. (Im Run Mode wird, solange **R/S** gedrückt ist, die Nummer und der Tastencode der momentanen Zeile angezeigt; wenn **R/S** losgelassen wird, beginnt die Programmausführung mit dieser Zeile.)

Sie können die **R/S** Instruktion auch dazu benutzen, um einen Programmablauf an Stellen, an denen Sie Daten eingeben wollen, zu unterbrechen. Nach der Dateneingabe kann das Programm mit der **R/S** Taste manuell wieder gestartet werden.

Beispiel: Dora's Dose, eine Konservenfabrik, will das Volumen von mehreren zylinderförmigen Konservendosen berechnen. Vor der Volumenberechnung soll die Grundfläche der Dosen angezeigt werden.



Das folgende Programm berechnet die Grundfläche jeder Dose und unterbricht dann die Ausführung. Nachdem Sie das Resultat notiert haben, kann das Programm für die Volumenberechnung fortgesetzt werden. Dazu wird die folgende Formel benutzt:

$$\text{Volumen} = \text{Grundfläche} \times \text{Höhe} = \pi r^2 \times h$$

Der Radius (r) und die Höhe (h) jeder Dose sind vor dem Start des Programms in das X- bzw. Y-Register zu laden.

Um das Programm zu speichern, schalten Sie den HP-11C auf Program Mode und geben die folgenden Befehle ein.

Tastenfolge

f CLEAR **PRGM**

Anzeige

000-

Löschen des Programmspeichers und Anzeige von Zeile 000.

f LBL **A**

001-42. 21. 11

g x^2

002- 43 11

Radius quadrieren.

f π

003- 42 16

X-Register mit π laden.

x

004- 20

Grundfläche berechnen.

R/S

005- 31

Unterbrechen der Programmausführung.

x

006- 20

Volumen berechnen.

g **RTN**

007- 43 32

Schalten Sie den HP-11C nun auf Run Mode und benutzen Sie das Programm, um die folgende Tabelle zu vervollständigen.

Höhe	Radius	Grundfläche	Volumen
25	10,0	?	?
8	4,5	?	?

Tastenfolge

25 **ENTER**

Anzeige

25,0000

Höhe in das Y-Register laden.

10 **f** **A**

314,1593

Radius in das X-Register laden und Berechnung der Grundfläche. Unterbrechung der Ausführung zur Anzeige der Fläche.

R/S

7.853,9816

Berechnung des Volumens der ersten Dose.

8 **ENTER**

8,0000

Höhe in das Y-Register laden.

Tastenfolge4,5 **f** **A****Anzeige****63,6173**

Radius in das X-Register laden und Berechnung der Grundfläche. Unterbrechung der Ausführung zur Anzeige der Fläche.

R/S**508,9380**

Berechnung des zweiten Volumens.

Mit der Höhe im Y-Register und dem Radius im X-Register wird durch Drücken von **f** **A** im Run Mode die Grundfläche der Dose berechnet; das Auffinden des **R/S** Befehls bewirkt die Unterbrechung des Programmablaufs. Durch Auslösen der Taste **R/S** wird das Volumen der Dose berechnet. Anschließend wird in Zeile 000 gesprungen und die Programmausführung beendet.

Eine Beschreibung der Dateneingabe ist auf den Seiten 213, Dateneingabe, und 218, Benutzer-definierbare Tasten, in Teil III dieses Handbuchs zu finden.

Pausen während der Programmausführung

Die **f** **PSE** Anweisung hält, als Bestandteil eines Programms, die Programmausführung für etwa eine Sekunde an. Während dieser Zeit wird der Inhalt des X-Registers angezeigt. Es können mehrere **PSE** Anweisungen in Folge eingegeben werden, um die Betrachtungszeit zu verlängern.

Um die Verwendung von **f** **PSE** innerhalb eines Programms zu illustrieren, wird das eben beschriebene Programm zur Berechnung der Dosenvolumina etwas modifiziert. Im neuen Programm soll die Grundfläche vor der Volumenberechnung kurz angezeigt werden. In diesem Beispiel soll ebenfalls demonstriert werden, wie verschiedene Programmierungsansätze zur Lösung des gleichen Problems verwendet werden können.

Zur Eingabe des Programms schalten Sie den HP-11C auf Program Mode. Löschen Sie zunächst den Programmspeicher mit **f** **CLEAR** **PRGM**. Im Display wird jetzt die Zeile 000 angezeigt. Geben Sie nun die folgenden Programmbefehle ein.

Tastenfolge**f** **CLEAR** **PRGM****f** **LBL** **A****g** **x²****Anzeige****000-****001-42. 21. 11****002- 43 11**

Quadrieren des Radius im X-Register.

Tastenfolgef π

X

f PSE

X

g RTN

Anzeige

003- 42 16

004- 20

005- 42 31

006- 20

007- 43 32

X-Register mit π laden.

Grundfläche berechnen.

Pause, um die Grundfläche für eine Sekunde anzuzeigen.

Berechnung des Volumens der Dose.

Bei diesem Programm wird wieder unterstellt, daß das Y-Register mit der Höhe und das X-Register mit dem Radius geladen ist. Wenn Sie die Befehle gespeichert haben, schalten Sie den HP-11C wieder auf Run Mode. Vervollständigen Sie die nachfolgende Tabelle mit Hilfe des neuen Programms.

Höhe	Radius	Grundfläche	Volumen
20	15	?	?
10	5	?	?

Tastenfolge

20 ENTER

15 f A

10 ENTER

5 f A

Anzeige

20,0000

705,8583

14.137,1669

10,0000

78,5398

785,3982

Y-Register mit der Höhe laden.

X-Register mit dem Radius laden und Berechnung der Grundfläche. Die Grundfläche wird für eine Sekunde angezeigt.

Ende Programm; Anzeige des Volumens.

Y-Register mit der zweiten Höhe laden.

X-Register mit dem Radius laden. Anzeige der Grundfläche für eine Sekunde.

Programmende; Anzeige des Volumens.

Programmunterbrechung durch Fehler

Manchmal kann durch einen Fehler in Ihrem Programm die Programmausführung unterbrochen werden. Als Hilfe bei der Fehlersuche sind im folgenden einige mögliche Fehlerursachen aufgeführt.

Fehlerhafte Verwendung von \boxed{g} \boxed{RTN} . Sofern \boxed{g} \boxed{RTN} außerhalb einer Subroutine gebraucht wird, bewirkt die Ausführung dieses Befehls den sofortigen Sprung nach Zeile 000 und die Terminierung der Programmausführung.

Erreichen des Endes des Programmspeichers. Wenn der letzte Befehl im Programmspeicher nicht eine der Instruktionen \boxed{GTO} , \boxed{GSB} , \boxed{RTN} oder $\boxed{R/S}$ ist, oder sich nicht in einem Unterprogramm befindet, wird die Ausführung eines Programms durch Erreichen des Endes des belegten Teils des Programmspeichers beendet und nach Zeile 000 gesprungen.

Auslösen beliebiger Tasten. Durch Drücken einer beliebigen Taste wird die Programmausführung angehalten. Der Rechner ist so ausgelegt, daß die Ausführung nicht während der Ausgabe der Ziffern einer Zahl unterbrochen werden kann. Wenn Sie eine Taste drücken, während das ablaufende Programm eine Zahl in das X-Register lädt, wird die Zahl „fertiggeschrieben“ und auch noch die nächste Zeile ausgeführt, bevor die Ausführung unterbrochen wird.

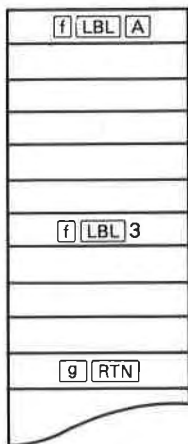
Im Run Mode kann die Ausführung eines angehaltenen Programms durch Drücken von $\boxed{R/S}$ wieder gestartet werden. Dieser Befehl bewirkt die Wiederaufnahme des Programmablaufs an der Stelle der Unterbrechung.

Unterbrechung durch Fehler. Wenn der Rechner versucht, eine fehlerverursachende Operation (siehe Anhang A, Fehlerbedingungen) auszuführen, wird die Programmausführung sofort abgebrochen und das Wort „Error“ zusammen mit einer Zahl angezeigt. Um die Zeilennummer und den Tastencode des fehlerverursachenden Befehls anzuzeigen, schalten Sie den Rechner auf Program Mode, indem Sie zunächst eine beliebige Taste (um die Fehlermeldung zu löschen) und anschließend \boxed{g} $\boxed{P/R}$ drücken. Wenn durch eine arithmetische Speicherregister-Operation ein Überlauf in einem Speicherregister verursacht würde, bricht der Rechner die Programmausführung ab und zeigt „Error 1“ an. Der Inhalt des betroffenen Speicherregisters wird nicht verändert. Nach dem Löschen der Fehlermeldung erscheint im Display wieder die zuvor angezeigte Zahl.

Wenn das Resultat einer Berechnung eine Zahl mit einer Wertigkeit kleiner als $1,000000000 \times 10^{-99}$ ist, wird diese Zahl durch Null ersetzt und die Programmausführung normal fortgesetzt. Diese Situation wird im allgemeinen als Underflow bezeichnet.

Labels

Wie bereits erwähnt wurde, dienen die Labels in Ihren Programmen als Adressen – sie teilen dem Rechner mit, wo die Programmausführung beginnt oder wo sie neu aufgenommen werden soll. Dabei ist zu beachten, daß ein Label in einem Programm keine ausführbare Anweisung darstellt. Beispielsweise würde in dem unten gezeigten Programmsegment die Ausführung durch Drücken von **f** **A** bei **f** **LBL** **A** beginnen und „nach unten“ durch den Programmspeicher fortgesetzt werden, bis der **RTN** Befehl erreicht wird. Die **f** **LBL** **3** Anweisung wird dabei ohne jegliche Wirkung durchlaufen.



Durch Drücken von **f** **A** wird hier mit der Programmausführung begonnen.

Die Ausführung der **f** **LBL** **3** Anweisung bleibt ohne Wirkung und das Programm läuft weiter bis ...

... zum **RTN** Befehl. Hier wird die Ausführung beendet und nach Zeile 000 gesprungen.

Übungsaufgaben

1. Schreiben Sie ein Programm, das Temperaturwerte von Grad Celsius in Grad Fahrenheit konvertiert. Dazu ist die Formel $F = 1,8 \text{ } ^\circ\text{C} + 32$ zu benutzen. Definieren Sie den Programmumfang mit

f LBL A und das Programmende mit g RTN. Konvertieren Sie mit dem Programm die Celsius-Temperaturen -40° , 0° und 72° .

Lösungen: $-40,0000^\circ\text{F}$, $32,0000^\circ\text{F}$, $161,6000^\circ\text{F}$.

2. Schreiben Sie ein Programm, daß die Länge einer Sehne ℓ , die von einem Winkel θ in einem Kreis mit dem Radius r aufgespannt wird, berechnet. Dazu ist die Formel

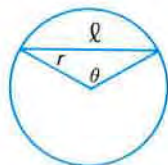
$$\ell = 2r \cdot \sin \frac{\theta}{2}$$

zu benutzen.

Das Programm soll so ausgelegt sein, daß die Daten in der Reihenfolge r , θ eingegeben werden.

Definieren Sie das neue Programm mit f LBL B und vervollständigen Sie die folgende Tabelle damit:

r (Meter)	θ	ℓ
25	30	?
50	45	?
100	90	?



(Die korrekten Lösungen müssen lauten: 12,9410 Meter, 38,2683 Meter und 141,4214 Meter.)

3. Nennen Sie die Tastencodes der folgenden Instruktionen.

g %, f \rightarrow RAD, STO + 1, SIN,
f LBL D, STO RAN #

Lösungen: 43,14; 42, 3; 44,40, 1; 23; 42,21,14; 44,36.*

4. Wieviel Programmzeilen sind zum Laden der folgenden Programmabschnitte erforderlich?

a. 2 ENTER 3 +

b. 10 STO 6 RCL 5 X

c. 100 STO 1 50 STO X 1 RCL 2 f π \div .

Lösungen: a. 4; b. 5; c. 10.

* Siehe Seite 78, Abgekürzte Tastenfolgen.

Programmkorrektur

Kurzdarstellung der Korrekturmöglichkeiten

Selbst der erfahrenste Programmierer wird Fehler in seinen Programmen entdecken. Diese reichen von Fehlern in der Logik des Programms bis zu Fehlern bei der Eingabe der einzelnen Programmbefehle. In jedem Fall besteht die Notwendigkeit diese Fehler zu finden und zu korrigieren. Der HP-11C ist darauf ausgelegt, den Prozeß der Fehlersuche und Fehlerkorrektur möglichst zu vereinfachen.

Bei der Ausführung eines Programms, das Speicherregister benutzt oder einen gewissen Rechnerstatus (wie gesetzte/nicht gesetzte Flags oder einen bestimmten trigonometrischen Mode) voraussetzt, können Fehler auftreten, wenn dieser Zustand beim Starten des Programms nicht gegeben ist. Die Wahrscheinlichkeit derartiger Fehler kann jedoch durch eine sogenannte *Initialisierungs-* (oder Lösch-)Prozedur eliminiert werden. Eine Möglichkeit zur Initialisierung eines Programms besteht darin, daß man die zum Löschen von Speicherregistern oder zum Zurücksetzen von Flags und Modi notwendigen Befehle vor Beginn der Programmausführung manuell über die Tastatur ausführt. Die zweite Möglichkeit, ein Programm selbstinitialisierend zu machen, besteht darin, die notwendigen Befehle am Anfang des Programms einzufügen.

Suchen von Programmfehlern

Wenn Sie herausfinden wollen, ob Ihr Programm korrekt arbeitet, besteht eine der einfachsten Möglichkeiten darin, das Programm an einem Beispiel zu testen, von dem Sie das Ergebnis kennen oder von dem das Ergebnis leicht berechnet werden kann. Eine andere Möglichkeit liegt darin, daß Sie die Reaktionen Ihres Programms mit Beispielen aus dem Grenzbereich der vorgegebenen Problemstellung oder der gewünschten Genauigkeit testen. In manchen Fällen kann es sogar von Interesse sein, die Reaktionen des Programms auf die Eingabe fehlerhafter Daten zu überprüfen. Potentielle Fehlerquellen in selbstinitialisierenden Programmen können unter Umständen durch wiederholte Ausführungen des Programms mit unterschiedlichen Daten in den Datenregistern, verschiedenen trigonometrischen Modi und Flagstellungen, und zur Kontrolle mit gelöschtem Speicher, lokalisiert werden.

Die Korrekturmöglichkeiten des HP-11C sollen Ihnen einen schnellen und einfachen Zugriff auf jeden beliebigen Teil eines Pro-

gramms, sei es zur Korrektur, Fehlersuche oder Dokumentation, ermöglichen. Wenn der Ablauf eines Programms durch einen Fehler oder Overflow unterbrochen wird, löschen Sie einfach die Fehlermeldung und schalten den Rechner auf Program Mode, um sich die Zeilennummer und den Tastencode der den Fehler oder Overflow verursachenden Operation anzusehen. Wenn Sie den Verdacht haben, daß der Fehler in einem bestimmten Bereich Ihres Programms liegt, können Sie die Ausführung dieses Bereichs Schritt für Schritt überprüfen und gegebenenfalls notwendige Änderungen vornehmen.

Korrekturfunktionen

Der HP-11C enthält vier nichtprogrammierbare Korrektur- und Manipulationsfunktionen, die Sie bei der Modifikation und Korrektur Ihrer Programme unterstützen:

[SST] [BST] [GTO] [·] nnn ←

[SST] (Single Step)

Im Program Mode:

Wenn Sie die Taste [SST] drücken und sofort wieder loslassen, springt der Rechner in die nächste Programmzeile und zeigt diese an. Bei gedrückter [SST] Taste zeigt Ihnen der Rechner solange nacheinander die Befehle im Programmspeicher an, bis Sie die Taste wieder loslassen. Bei Verwendung von [SST] im Program Mode werden die Programmbefehle nicht ausgeführt.

000-

[SST]

001-42.21.11

Im Run Mode:

Durch Drücken von [SST] springt der Rechner auf die nächste Zeile im Programmspeicher und zeigt diese an. Wenn Sie die Taste loslassen, wird die in dieser Zeile gespeicherte Anweisung ausgeführt.

[BST] (Back Step)

Durch Drücken von [g] [BST] läuft der Rechner schrittweise oder kontinuierlich „rückwärts“ durch den Programmspeicher; in analoger Weise, wie dies bei [SST] „vorwärts“ geschieht. (Die Programmbefehle werden nicht ausgeführt.)

002- 43 13

[g] [BST]

001-42.21.11

[GTO] [.] nnn (Go To Line nnn).

Durch Drücken von [GTO] [.] nnn springt der Rechner in die durch nnn spezifizierte Programmzeile. (Im Run Mode bewirkt das Auslösen von [GTO] gefolgt von dem Namen eines numerischen oder alphanumerischen Labels einen Sprung zu dem spezifizierten Label. Dies geschieht intern in der gleichen Weise, wie der durch [GTO] [.] nnn verursachte Sprung zu einer spezifizierten Zeilennummer.)

[←] Löschen einzelner Zeilen

Das Drücken von [←] im Program Mode bewirkt die Löschung der angezeigten Zeile im Programmspeicher. Alle auf die gelöschte Anweisung folgenden Programmbehele werden um eine Zeile nach oben verschoben und neu nummeriert. Im Run Mode läßt [←] den Inhalt des Programmspeichers unverändert; es wird lediglich der Inhalt des angezeigten X-Registers manipuliert (siehe Seite 17, Löschen der Anzeige [CLx] und [←]).

Programmspeicherinhalt

	Vor dem Drücken von [←]:		Nach dem Drücken von [←]:	
	001-42.21.11	→	001-42.21.11	Anzeige
Anzeige	002- 43 13	→	002- 23	
	003- 23	→	003- 45 0	
	004- 45 0	→	004- 45 1	
	005- 45 1	→		

Beispiel zur Programmkorrektur

Die Möglichkeiten zur Programm-Modifikation sollen an Hand des folgenden (auf dem Satz von Pythagoras basierenden) Programms zur Berechnung der Länge der Hypotenuse (Seite *c*) eines rechtwinkligen Dreiecks bei gegebener Länge der Seiten *a* und *b* demonstriert werden. Es sei unterstellt, daß zu Beginn der Berechnung das Y-Register mit der Seite *a* und das X-Register mit der Seite *b* geladen ist. Schalten Sie den HP-11C zur Eingabe des Programms auf Program Mode.



Tastensequenz

[f] CLEAR [PRGM]

Anzeige

000-

Löschen des Programmspeichers.

[f] [LBL] [E]

001-42. 21. 15

Programm wird gelabelt.

[g] [x²]

002- 43 11

Quadrieren von Seite *b*.

[x] [≥] [v]

003- 34

*b*² wird vom X- in das Y-Register geladen; *a* vom Y- in das angezeigte X-Register.[g] [x²]

004- 43 11

Quadrieren von Seite *a*.

[+]

005- 40

 $(a^2 + b^2)$

[√x]

006- 11

 $\sqrt{(a^2 + b^2)}$

[g] [RTN]

007- 43 32

Ende Programm

Schalten Sie den HP-11C auf Run Mode.

Zum Testen des Programms soll die Hypotenuse eines rechtwinkligen Dreiecks mit einer Seite *a* von 22 Metern und einer Seite *b* von 9 Metern berechnet werden.

Tastensequenz

22 [ENTER]

Anzeige

22,0000

Eingabe von *a*.

9

9

Eingabe von *b*.

[f] [E]

23,7697

Länge der Hypotenuse in Metern.

Zeilenweise Ausführung eines Programms

Bei längeren Programmen wird ein falsches Testergebnis in den seltensten Fällen direkt auf den Fehler deuten. Hier kann die Programmausführung durch Verwendung von [SST] im Run Mode verlangsamt werden.

Die zeilenweise Ausführung beginnt mit der Zeile auf die der Rechner momentan positioniert ist. Da durch die [RTN] Anweisung am Ende des Programms der Rechner in unserem Beispiel durch die Ausführung des Programms jetzt auf Zeile 000 positioniert ist, brauchen nur noch die Eingabedaten eingetastet werden, um mit der zeilenweisen Ausführung des Programms zu beginnen. Lassen Sie bei jedem Auslösen von [SST] die Taste für einen Moment gedrückt, um die Zeilennummer und den Tastencode der nächsten auszuführenden Anweisung anzuzeigen, und führen Sie anschließend die Anweisung durch Loslassen der [SST] Taste aus.

Tastensequenz22 **[ENTER]**

9

[SST]**[SST]****[SST]****[SST]****[SST]****[SST]****[SST]****Anzeige**

22,0000

9

001–42. 21. 15
9,0000002– 43 11
81,0000003– 34
22,0000004– 43 11
484,0000005– 40
565,0000006– 11
23,7697007– 43 32
23,7697Y-Register mit a laden.X-Register mit b laden.

Bei gedrückter **[SST]** wird der Tastencode in Zeile 001 (für **[f]** **[LBL]** **[E]**) angezeigt. Nach dem Loslassen wird Zeile 001 ausgeführt.

Tastencode für x^2 . **[SST]** losgelassen, x^2 wird ausgeführt.

Tastencode für $x \geq y$. Ausführung von $x \geq y$.

Ausführung von **[g]** x^2 .

Ausführung von **[+]**.

Ausführung von \sqrt{x} .

Ausführung von **[g]** **[RTN]**. Das Programm ist vollständig abgearbeitet, der Rechner springt in Zeile 000 und hält an.

Bemerkung: **[SST]** läuft nicht in unbelegte Zeilen des Programmspeichers. Wenn ein **[SST]** Befehl nach der Ausführung der letzten belegten Programmzeile (und diese Zeile keine **[GTO]** (*go to*) oder **[GSB]** (*go to subroutine*) Anweisung enthalten hat) gegeben wird, so springt der Rechner wieder auf Zeile 000. Wenn der letzte Befehl ein **[GTO]** oder **[GSB]** war, führt der Rechner diese Instruktion aus, in dem er auf die gewünschte Position im Programmspeicher springt.

Verwendung von **[SST] und **[BST]** im Program Mode**

Zeilenweises Durchlaufen des Programmspeichers. Im Program Mode bewirkt ein Drücken und wieder *Loslassen* der Taste **[SST]**, daß der Rechner in die nächste Zeile des belegten Bereichs des Programmspeichers springt, und diese anzeigt. Die Programmbeefehle werden nicht ausgeführt. (**[g]** **[BST]** arbeitet in der gleichen Weise mit der Ausnahme, daß die im Programmspeicher vor der momentanen Zeile stehende Zeile angezeigt wird.)

Tastenfolge

g P/R

SST

SST

SST

SST

SST

SST

SST

g P/R

Anzeige

000-

001-42. 21. 15

002- 43 11

003- 34

004- 43 11

005- 40

006- 11

007- 43 32

Der HP-11C wird auf
Program Mode geschaltet.

Der Rechner wird wieder
auf Run Mode geschaltet.

Kontinuierliches Durchlaufen des Programmspeichers. Im Program Mode bewirkt ein Drücken und *Festhalten* der Tasten **SST** oder **BST**, daß der Inhalt des Programmspeichers (vorwärts oder rückwärts) gelistet wird. (Der Rechner zeigt die momentane Programmzeile ungefähr zwei Sekunden lang an, anschließend wird jede Folgezeile für ungefähr eine halbe Sekunde angezeigt.)

Modifizieren eines Programms

Um die Möglichkeiten des HP-11C zur Modifizierung von Programmen zu illustrieren, soll das Pythagoras-Programm von Seite 99 so abgeändert werden, daß an bestimmten Stellen des Programms der Inhalt des X-Registers automatisch angezeigt wird. Dies soll durch Einfügen von **f PSE** (*pause*) Anweisungen in die entsprechenden Positionen im Programmspeicher geschehen.

Tastenfolge

f LBL E

g x^2 $x \geq y$ g x^2

+

 \sqrt{x}

g RTN

Anzeige

001-42. 21. 15

002- 43 11

003- 34

004- 43 11

005- 40

006- 11

007- 43 32

Hinter diese drei Befehle
soll jeweils eine **f PSE**
Anweisung eingefügt
werden.

Einfügen von Anweisungen. Ein neuer Programmbehl kann an jeder beliebigen Stelle zwischen Zeile 000 und dem Ende des belegten Teils des Programmspeichers eingefügt werden. Eine neue Anweisung wird wie folgt eingefügt:

1. Schalten Sie den Rechner auf Program Mode.
2. Durchlaufen Sie den Programmspeicher bis zu derjenigen Programmzeile, die der Position, an der eine neue Anweisung eingefügt werden soll, *unmittelbar vorangeht*.
3. Geben Sie den neuen Befehl ein. Dieser Befehl wird in die *nächste* Programmzeile geladen, und alle nachfolgenden Befehle werden um eine Zeile nach „unten“ verschoben (und neu nummeriert).

Da jedes Einfügen eines neuen Befehls eine Umnummerierung der Folgebefehle bedingt, wird das Einfügen von Änderungen in ein Programm vereinfacht, wenn die am weitesten von Zeile 000 entfernte Änderung zuerst eingefügt wird, und danach die Änderungen in der Reihenfolge zum Programmanfang hin abgearbeitet werden.

Beispiel: Hinter dem $\boxed{+}$ Befehl soll eine $\boxed{\text{PSE}}$ Anweisung eingefügt werden.

Tastenfolge
 $\boxed{\text{GTO}}$ $\boxed{\cdot}$ 000

$\boxed{\text{g}}$ $\boxed{\text{P/R}}$
 $\boxed{\text{SST}}$
 $\boxed{\text{SST}}$
 $\boxed{\text{SST}}$
 $\boxed{\text{SST}}$
 $\boxed{\text{SST}}$
 $\boxed{\text{f}}$ $\boxed{\text{PSE}}$

Anzeige

000–
 001–42. 21. 11
 002– 43 11
 003– 34
 004– 43 11
 005– 40
 006– 42 31

Der Rechner wird auf Zeile 000 positioniert.

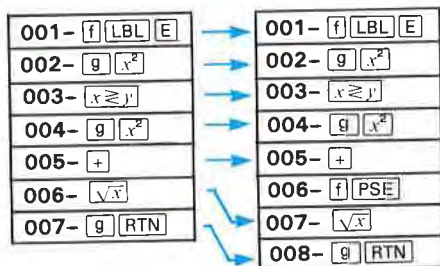
Zeile 000.

Durchlaufen des Speichers bis Zeile 005.

Zeile 005.

$\boxed{\text{PSE}}$ wird vor Zeile 006 eingefügt.

Das Einfügen von $\boxed{\text{f}}$ $\boxed{\text{PSE}}$ bewirkt die folgende Veränderung im Inhalt des Programmspeichers.



Die $\boxed{\text{f}}$ $\boxed{\text{PSE}}$ Anweisung ist hier eingefügt. Alle Folgebefehle sind um eins nach unten geschoben.

Sprung zu einer Zeilennummer. Es ist offensichtlich, daß das Durchlaufen des Programmspeichers bis zu einer bestimmten Zeile einen gewissen Zeitaufwand in Anspruch nehmen kann. Diese unnötige Wartezeit läßt sich vermeiden, wenn Sie die Anweisung **GTO** \cdot *nnn*, die Sie schon benutzt haben, um Zeile 000 anzuspringen, verwenden. Durch Drücken von **GTO** \cdot und einer dreistelligen Zeilennummer im belegten Teil des Programmspeichers springt der Rechner sofort in die durch die drei Ziffern spezifizierte Zeile. (Sie können **GTO** \cdot *nnn* im Program und im Run Mode verwenden – es werden in beiden Fällen keine Programmbefehle ausgeführt.) Springen Sie beispielsweise in Zeile 004, um eine **PSE** Anweisung hinter die zweite **g** x^2 Instruktion einzufügen.

Tastenfolge

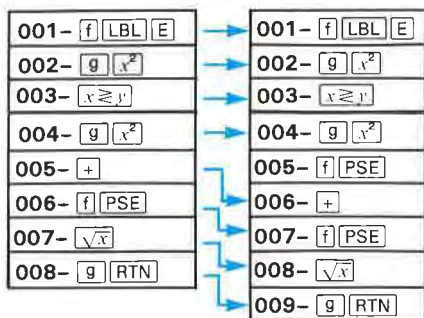
GTO \cdot 004
f **PSE**

Anzeige

004– 43 11
 005– 42 31

Sprung in Zeile 004.
 Einfügen von **PSE** nach
 Zeile 004.

Das Einfügen von **f** **PSE** bewirkt die folgende Änderung im Inhalt des Programmspeichers.



Die **f** **PSE** Anweisung ist hier eingefügt.

Alle Folgebefehle sind um eins nach unten geschoben.

Der letzte **f** **PSE** Befehl in unserem Beispiel soll nun mit **BST** eingefügt werden.

Tastensequenz

g **BST**
g **BST**
g **BST**

f **PSE**

Anzeige

005- 42 31

004- 43 11

003- 34

002- 43 11

003- 42 31

Momentane Position im
 Programmspeicher.

Die letzte **PSE** Anweisung
 ist hinter dieser Zeile einzu-
 fügen.

Der **PSE** Befehl ist jetzt
 eingefügt. Alle Folgebefeh-
 le werden um eins nach un-
 ten verschoben.

Unser modifiziertes Beispielprogramm hat nun die unten gezeigte Form. Sie können sich durch Verwendung von **SST** im Program Mode davon überzeugen, daß das Programm im Rechner mit dieser modifizierten Version übereinstimmt.

Tastensequenz

f **LBL** **E**
g x^2
f **PSE**
 $x \geq y$
g x^2
f **PSE**
 $+$
f **PSE**
 \sqrt{x}
g **RTN**

Anzeige

001-42.21.15

002- 43 11

003- 42 31

004- 34

005- 43 11

006- 42 31

007- 40

008- 42 31

009- 11

010- 43 32

Lassen Sie das Programm nun noch einmal ablaufen. Benutzen Sie die Werte 22 und 9 aus dem letztem Beispiel für die Seiten a und b .

Tastenfolge

[g] [P/R]

22 [ENTER]

9

[f] [E]

Anzeige

22,0000

9

81,0000

484,0000

565,0000

23,7697

Der HP-11C wird Run Mode geschaltet.

Y-Register mit Seite *a* laden.

X-Register mit Seite *b* laden.

Pause um das Resultat der $[x^2]$ (b^2) Anweisung in Zeile 003 anzuzeigen.

Pause um das Resultat der $[x^2]$ (a^2) Anweisung in Zeile 006 anzuzeigen.

Pause um das Ergebnis der $[+]$ ($a^2 + b^2$) Anweisung in Zeile 007 anzuzeigen.

Endergebnis Seite *c*
($c = \sqrt{a^2 + b^2}$)

Einfügungen in größeren Programmen. Wenn *alle* 203 Programmzeilen *belegt* sind, werden zusätzliche Programmbeefehle vom Rechner nicht mehr akzeptiert. Bei dem Versuch, eine neue Anweisung in die 203 Programmzeilen einzufügen, erscheint im Display die Meldung „**Error 4**“. Der Inhalt des Programmspeichers wird nicht verändert. (Siehe Anhang C, Automatische Speicherumwandlung).

Löschen von Programmbeefehlen. Bei der Korrektur oder Modifizierung von Programmen kommt es häufig vor, daß Instruktionen aus dem Programmspeicher gelöscht werden müssen. Dazu ist der Rechner zunächst auf Program Mode zu schalten. Dann ist der Rechner mit $[GTO]$, $[SST]$ oder $[BST]$ auf die zu löschende Anweisung zu positionieren. Der eigentliche Löschvorgang wird danach durch Drücken der nichtprogrammierbaren Funktion $[↵]$ ausgelöst. (Nach dem Löschen eines Programmbeefehls mit $[↵]$ werden die nachfolgenden Programmbeefehle um eine Zeile nach *oben* verschoben und neu numeriert. Der Rechner zeigt dann die *vor* dem gelöschten Befehl stehende Anweisung an.)

Beispiel: Das Pythagoras-Programm aus dem letzten Beispiel ist so zu modifizieren, daß nur noch eine Pause zum Anzeigen der

Summe der Quadrate übrig bleibt. Dazu sind die **f** **PSE** Anweisungen in den Zeilen 003 und 006 zu löschen.

Tastenfolge**g** **P/R****GTO** **.** 006**←****GTO** **.** 003**←****Anzeige**

000-

006- 42 31

005- 43 11

003- 42 31

002- 43 11

Der HP-11C wird auf Program Mode geschaltet.

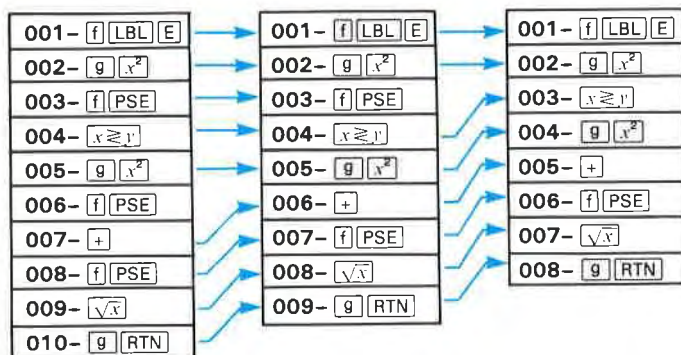
Positionieren des Rechners auf Zeile 006.

Löschen der **f** **PSE** Anweisung; der Rechner zeigt die Zeile davor (002) an. Die Folgebefehle werden um eine Zeile nach oben verschoben (und neu nummeriert).

Positionieren des Rechners auf Zeile 003.

Löschen der **f** **PSE** Anweisung; der Rechner zeigt die Zeile davor (002) an. Die Folgebefehle werden um eine Zeile nach oben verschoben (und neu nummeriert).

Das Löschen der **PSE** Anweisungen in den Zeilen 006 und 003 bewirkt die folgende Änderung im Inhalt des Programmspeichers.



Das so modifizierte Pythagoras-Programm hält nun nur noch an, um die Summe der Quadrate anzuzeigen. Danach wird die Länge der Hypotenuse berechnet, und die Ausführung des Programms beendet.

Lassen Sie das Programm jetzt noch einmal für ein rechtwinkliges Dreieck mit einer Seite a von 17 Metern und einer Seite b von 34 Metern ablaufen.

Tastensequenz

g P/R

17 ENTER 34

f E

Anzeige

34

1.445,0000

38,0132

Der HP-11C wird auf Run Mode geschaltet.

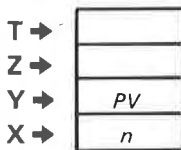
Pause um die Summe der Quadrate der Seiten a und b anzuzeigen.

Länge der Hypotenuse.

Beim Löschen von Programmbefehlen aus Programmen mit mehr als 63 Zeilen läuft der Prozeß der automatischen Umwandlung von Datenspeicher-Registern in Programmzeilen in umgekehrter Reihenfolge ab. Zum Beispiel werden durch die Löschung irgendeines Befehls in einem 71-Zeilen Programm die Zeilen 71 bis 77 in das Datenspeicher-Register R_2 zurückgewandelt. (Siehe Anhang C, Automatische Speicherumwandlung.)

Übungsaufgaben

- Das folgende Programm wird vom Manager einer Bank verwendet, der damit die zukünftigen Erträge aus den Sparguthaben errechnet. Dazu wird die Formel $FV = PV(1 + i)^n$ benutzt. FV ist der zukünftige, PV der gegenwärtige Wert der Sparguthaben; i der Zinssatz (dezimal) pro Periode und n die Anzahl der Perioden. Es wird unterstellt, daß PV und n vor Beginn der Ausführung, wie nebenstehend gezeigt, geladen sind. Des Weiteren wird ein Zinssatz von 7,5% angenommen.



Tastenfolge

f LBL A

f FIX 2

1

.

0

7

5

 $x \geq y$ v^x \times

g RTN

Anzeige

001-42. 21. 11

002-42. 7. 2

003- 1

004- 48

005- 0 Zinssatz

006- 7

007- 5

008- 34

009- 14 $(1 + i)^n$ 010- 20 $PV(1 + i)^n$

011- 43 32

- a. Laden Sie das Programm in den Rechner.
- b. Berechnen Sie mit dem Programm den Wert in fünf Jahren eines Sparguthabens von 1000 DM. (Lösung: 1435,63 DM)
Berechnen Sie das gleiche für 2300 DM in 4 Jahren. (Lösung: 3071,58 DM)
- c. Ändern Sie das Programm, um eine Änderung des Jahreszinssatzes von 7,5 % auf 8 % zu berücksichtigen.
- d. Berechnen Sie mit dem geänderten Programm den Wert eines Sparguthabens von 500 DM in 4 Jahren, bzw. von 2000 DM in 10 Jahren. (Lösungen: 680,24 DM und 4317,85 DM)
2. Das folgende Programm berechnet die Zeit, die ein aus einer Höhe h abgeworfener Gegenstand braucht, bis er die Erdoberfläche erreicht. (Der Einfluß des Luftwiderstandes wird dabei außer Betracht gelassen.) Wenn Sie die Höhe h (in Meter) in das angezeigte X-Register laden und dann \boxed{B} drücken, wird die Fallzeit

$$t = \sqrt{\frac{2h}{9.8 \text{ m/s}^2}}$$

berechnet und angezeigt.

- a. Löschen Sie alle bisher gespeicherten Programme, setzen Sie das Format auf \boxed{FIX} 4 und laden Sie die folgenden Programm-befehle:

Tastenfolge

f CLEAR PRGM

f LBL B

2

x

9

.

8

÷

√x

g RTN

Anzeige

000-

001-42.21.12

002- 2

003- 20

004- 9

005- 48

006- 8

007- 10

008- 11

009- 43 32

- b. Berechnen Sie mit Hilfe des Programms die Fallzeit für einen Stein, der vom 300,51 Meter hohen Eiffelturm geworfen wird. Führen Sie die gleiche Rechnung für einen Gegenstand durch, der aus einem in 1050 Meter Höhe fliegenden Luftschiff abgeworfen wird.

(Lösungen: 7,8313 Sek. und 14,6385 Sek.)

- c. Ändern Sie das vorstehende Programm jetzt so ab, daß die Höhe in *Fuß* eingegeben werden kann. Es gilt:

$$t = \sqrt{\frac{2h}{32,1740 \text{ ft/s}^2}}$$

- d. Berechnen Sie mit Hilfe des abgeänderten Programms die Fallzeit für einen Stein, der vom 550 Fuß hohen Grand Coulee Dam geworfen wird. Wie lange fällt eine Münze, die vom 1350 Fuß hohen World Trade Center Gebäude in New York City geworfen wird?

(Lösungen: 5,8471 Sek. und 9,1607 Sek.)

Programmverzweigungen

Kurzdarstellung Programmverzweigungen

Vergleichsoperationen

Die acht Vergleichsoperationen des HP-11C bestehen aus gleich/ungleich Abfragen, durch deren Verwendung in Programmen der Rechner Entscheidungen fällen kann. Wenn in einem ablaufenden Programm das Ergebnis einer Vergleichsoperation „gleich“ („TRUE“) lautet, wird die Programmausführung mit dem auf die Vergleichsoperation folgenden Befehl fortgesetzt. Ist das Ergebnis einer Vergleichsoperation „ungleich“ („FALSE“), wird der auf die Abfrage folgende Befehl *übersprungen* und die Ausführung mit dem zweiten, auf die Abfrage folgenden Befehl fortgesetzt.

Sprungverhalten nach einer Vergleichsoperation



Bei den Vergleichsoperationen des HP-11C wird der Wert im X-Register mit der Zahl im Y-Register bzw. mit Null in der folgenden Weise verglichen:

- $x \leq y$ Abfrage, ob der Wert im X-Register kleiner oder gleich dem Wert im Y-Register ist.
- $x > y$ Abfrage, ob der Wert im X-Register größer als der Wert im Y-Register ist.
- $x \neq y$ Abfrage, ob der Wert im X-Register ungleich dem Wert im Y-Register ist.

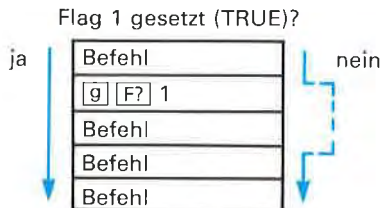
f	$x = y$	Abfrage, ob der Wert im X-Register gleich dem Wert im Y-Register ist.
g	$x > 0$	Abfrage, ob der Wert im X-Register größer als Null ist.
g	$x < 0$	Abfrage, ob der Wert im X-Register kleiner als Null ist.
g	$x \neq 0$	Abfrage, ob der Wert im X-Register ungleich Null ist.
g	$x = 0$	Abfrage, ob der Wert im X-Register gleich Null ist.

Flags

Eine andere Art der Abfrage besteht in den sogenannten Flag-Tests. Ein Flag ist eigentlich ein Statusindikator, der entweder gesetzt („TRUE“) oder nicht gesetzt („FALSE“) ist. Im Ablauf eines Programms kann ein Flag abgefragt und eine Entscheidung davon abhängig gemacht werden, ob der Flag gesetzt oder nicht gesetzt war. Die Auswirkungen von Flagabfragen auf die Programmausführung sind identisch mit denen bei Vergleichsoperationen.

Die beiden Flags Ihres HP-11C tragen die Nummern 0 und 1. Zum Setzen eines Flags müssen Sie **g** **SF** (*set flag*) gefolgt von der Zahlentaste mit der Nummer 0 oder 1 des gewünschten Flags drücken. Durch Auslösen von **g** **CF** (*clear flag*) gefolgt von der entsprechenden Zifferntaste wird ein Flag zurückgesetzt. Das Abfragen eines Flags erfolgt durch den Befehl **g** **F?** gefolgt von der Nummer des abzufragenden Flags.

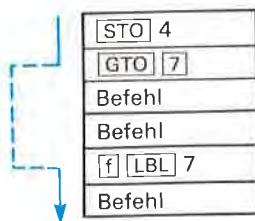
Ein mit einem **g** **SF** *n* Befehl gesetzter Flag bleibt solange gesetzt, bis er mit einem **g** **CF** *n* Befehl explizit zurückgesetzt oder der Permanentenspeicher gelöscht wird.



Programmsteuerung

Go To.

Die `GTO` Anweisung, die Sie bereits benutzt haben, um den Rechner zu Korrekturzwecken auf eine bestimmte Programmzeile zu positionieren, kann auch zum Sprung zu einem Label irgendwo im Programmspeicher dienen. Wenn `GTO` als Programmbefehl verwendet wird, muß die Adresse eines numerischen oder Alpha-Labels spezifiziert werden.



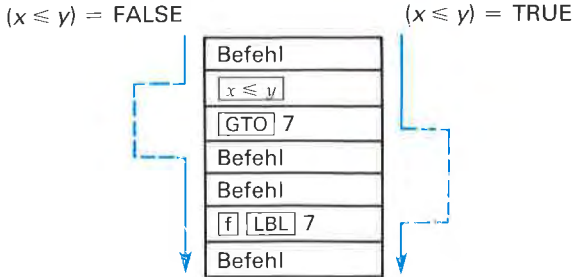
Wenn beim Ablauf eines Programms eine `GTO label` Anweisung auftritt, wird die Programmausführung für eine gewisse Zeitspanne unterbrochen, in der der Rechner den Programmspeicher nach dem spezifizierten Label durchsucht. Die Programmausführung wird mit der ersten Zeile, die das gesuchte Label enthält, fortgesetzt.

Sprünge und Programmschleifen

Unbedingte Sprünge. Ein unbedingter Sprung besteht aus einer `GTO` Anweisung, die, unabhängig vom jeweiligen Datenstatus, immer im Ablauf eines Programms ausgeführt wird, um die Programmkontrolle an eine beliebige Stelle im Programm zu übergeben. Die oben illustrierte `GTO` Operation ist ein Beispiel für einen unbedingten Sprung.

Bedingte Sprünge. Wenn, wie im folgenden illustriert, `GTO label` zusammen mit einer Abfrage benutzt wird, so wird diese Anweisung zu einem *bedingten Sprung*, d.h. die Ausführung der `GTO` Anweisung hängt von dem Ergebnis der Abfrage ab.

Bedingter Sprung

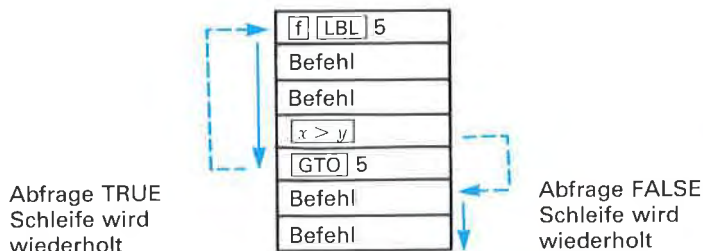


Im Run Mode bewirkt die Anweisung `GTO label`, daß der Rechner zu dem spezifizierten Label springt und dort anhält. Dieser Befehl ermöglicht Ihnen auf einfache Weise auf ein bestimmtes Label folgende Zeilen im Programmspeicher anzuzeigen oder zu korrigieren, ohne diese Zeilen als Teil des Programms ausführen zu lassen.

Programmschleifen. Programmschleifen sind spezielle Anwendungen vom Sprungbefehlen, in denen eine `GTO` Anweisung zur wiederholten Ausführung einer Reihe von Programmbefehlen dient. Eine häufige Anwendung von Programmschleifen besteht in der sequentiellen Berechnung einer Reihe von Resultaten unter Verwendung der gleichen Programmbefehle. Die Entscheidung, ob eine Programmschleife noch einmal durchlaufen oder verlassen werden soll, wird durch einen bedingten Sprungbefehl kontrolliert.

Die folgende Illustration zeigt, daß, solange das Ergebnis der Vergleichsoperation vor der `GTO` Anweisung TRUE ist, die Programmschleife erneut durchlaufen wird. Liefert die Vergleichsoperation das Ergebnis FALSE, wird die `GTO` Anweisung übersprungen und der Rest des Programms ausgeführt. Die Verwendung von unbedingten Sprungbefehlen zur Steuerung von Programmschleifen liefert sogenannte Endlosschleifen, d. h. die Programmschleife wird unendlich oft durchlaufen.

Steuerung einer Laufschleife durch einen bedingten Sprungbefehl



Beispiel für die Steuerung einer Programmschleife. Das folgende Programm dient zur Berechnung und Anzeige der Quadratwurzeln der Zahlen von 1 bis 10. Nach der Berechnung der Quadratwurzel der Zahl n wird durch $[x > y]$ abgefragt, ob n kleiner als 10 ist. Solange dies gilt, wird ein bedingter Sprungbefehl $[GTO] 0$ ausgeführt und die Schleife (d. h. die Quadratwurzelberechnung) für das Wurzelargument $n = n + 1$ wiederholt. Wenn die Abfrage ergibt, daß n größer oder gleich 10 ist, wird der Befehl $[GTO] 0$ übersprungen und die Laufschleife verlassen. (Der Anteil der Laufschleife im folgenden Programm besteht aus den Zeilen 004 bis 015.)

Tastensequenz

$[g]$ $[P/R]$

$[f]$ $[CLEAR]$ $[PRGM]$

$[f]$ $[LBL]$ $[C]$

0

$[STO]$ 1

$[f]$ $[LBL]$ 0

1

$[STO]$ $[+]$ 1

$[RCL]$ 1

$[f]$ $[PSE]$

$[\sqrt{x}]$

$[f]$ $[PSE]$

$[g]$ $[LSTX]$

Anzeige

000-

Rechner auf Program
Mode schalten.

000-

001-42. 21. 13

002- 0

003- 44 1

004-42. 21. 0

005- 1

006-44. 40. 1

007- 45 1

008- 42 31

009- 11

010- 42 31

011- 43 36

Löschen von alten Daten in
Register R_1
Schleifenanfang.

Zähler (Wurzelargument)
erhöhen

Nächste Zahl anzeigen.

Quadratwurzel ziehen.

Quadratwurzel anzeigen.

Wurzelargument aus dem
LAST X Register zurück-
laden.

Tastenfolge	Anzeige	
1	012-	1 Zählerstand mit dem
0	013-	0 maximal zulässigen
$\boxed{f} \boxed{x > y}$	014- 42 20	Wurzelargument vergleichen.
$\boxed{GTO} 0$	015- 22 0	Bedingter Sprung. Wenn das letzte Wurzelargument kleiner als 10 ist, Sprung nach $\boxed{LBL} 0 \dots$
$\boxed{g} \boxed{CLX}$	016- 43 35	... andernfalls Löschen des X-Registers und Beenden der Programm- ausführung.
$\boxed{g} \boxed{RTN}$	017- 43 32	
$\boxed{g} \boxed{P/R}$		Rechner auf Run Mode schalten.

Das Programm kann jetzt mit $\boxed{f} \boxed{C}$ gestartet werden. Der Rechner zeigt dann eine Tabelle der ganzen Zahlen von 1 bis 10 mit den zugehörigen Quadratwurzeln an. Die Programmausführung ist beendet, wenn 0,0000 angezeigt wird.

Arbeitsweise. Das Drücken der Tasten $\boxed{f} \boxed{C}$ bewirkt, daß der Rechner den Programmspeicher nach einer $\boxed{f} \boxed{LBL} \boxed{C}$ Anweisung durchsucht. Das Label und die folgenden Befehle werden in sequentieller Folge abgearbeitet, bis Zeile 014, die $\boxed{x > y}$ Abfrage, erreicht wird. Wenn diese Vergleichsoperation das Ergebnis TRUE liefert – d. h. das zuletzt benutzte Wurzelargument war kleiner als 10 – wird die nachfolgende $\boxed{GTO} 0$ Anweisung ausgeführt; der Rechner springt zu dem $\boxed{LBL} 0$ Befehl in die Zeile 004 und die Schleife wird erneut durchlaufen. Falls jedoch die Vergleichsoperation das Ergebnis FALSE liefert – dadurch 10 als letztes Wurzelargument implizierend – wird die $\boxed{GTO} 0$ Anweisung übersprungen und die Laufschleife verlassen.

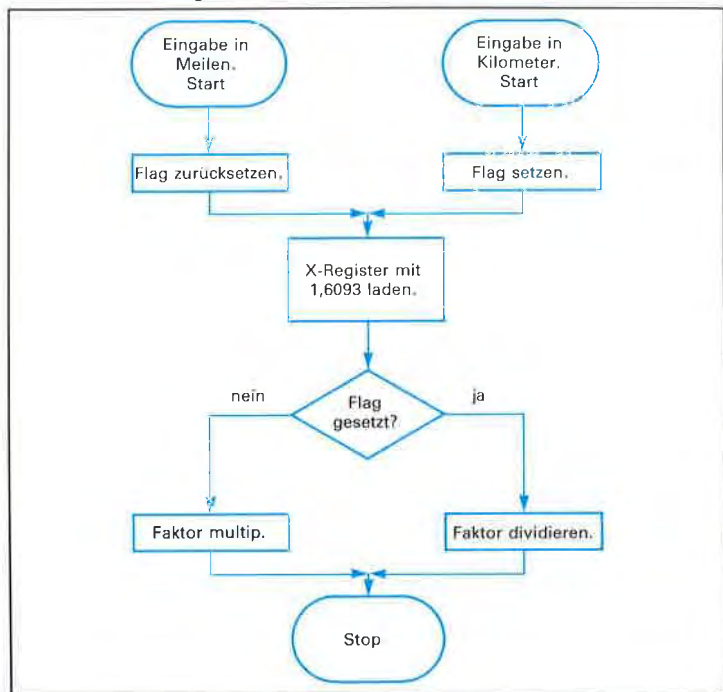
Eine weitergehende Diskussion der Laufschleifen und der Techniken zu ihrer Steuerung ist auf Seite 214, Programmschleifen, in Teil III dieses Handbuchs zu finden.

Übungsaufgabe: Es ist ein Programm zu schreiben, das Verkaufsprovisionen berechnet und anzeigt. Dabei soll bei Verkäufen von unter 100,- DM ein Provisionssatz von 10 %, bei Verkäufen im Wert von 100,- DM und mehr ein Provisionssatz von 15 % gewährt werden. Verwenden Sie zur Lösung einen bedingten Sprungbefehl.

Verwendung von Flags

Analog zu den Register-Vergleichsoperationen ($x < y$, $x < 0$ usw.) bieten Ihnen die Flagabfragen die Möglichkeit, einzelne Zeilen im Programmspeicher wahlweise zu überspringen oder ausführen zu lassen. Während bei Register-Vergleichsoperationen Werte verglichen werden, stellen Flagabfragen einen Zugriff auf einen Rechnerzustand dar. Beispielsweise kann ein in einer Programmverzweigung mit `g SF n` gesetzter Flag durch den Befehl `g F? n` an einer beliebigen Programmstelle abgefragt werden, um dadurch festzustellen, ob diese Programmverzweigung durchlaufen wurde oder nicht.

Beispiel: Das folgende Programm dient zur Konvertierung einer Eingabe von Meilen in Kilometer (`f A`) oder umgekehrt (`f B`). Der Rechner prüft den Status des Flags 0, um festzustellen, ob eine Benutzereingabe in Kilometer (Multiplikation der Eingabe mit dem Konvertierungsfaktor) oder in Meilen (Division der Eingabe durch den Konvertierungsfaktor) zu konvertieren ist.



Tastensequenz

[g] [P/R]

[f] CLEAR [PRGM]

[f] [LBL] [A]

[g] [CF] 0

[GTO] 1

[f] [LBL] [B]

[g] [SF] 0

[f] [LBL] 1

1
[.]

6

0

9

3

[g] [F?] 0

[GTO] 2

[X]

[g] [RTN]

[f] [LBL] 2

[÷]

[g] [RTN]

[g] [P/R]

Anzeige

000-

000-

001-42. 21. 11

002-43. 5. 0

003- 22 1

004-42. 21. 12

005-43. 4. 0

006-42. 21. 1

007- 1

008 48

009- 6

010- 0

011- 9

012- 3

013-43. 6. 0

014- 22 2

015- 20

016- 43 32

017-42. 21. 2

018- 10

019- 43 32

Rechner auf Program
Mode schalten.Programmspeicher
löschen.Programm beginnt mit der
Eingabe in Meilen.Flag 0 zurücksetzen (Kenn-
zeichnung der Meilenein-
gabe zur Sprungsteuer-
ung).Unbedingter Sprung zu
[LBL] 1.Programm beginnt mit der
Eingabe in Kilometern.Flag 0 setzen (Kennzeich-
nung der Kilometereingabe
zur Sprungsteuerung).Beginn des Rechenteils des
Programms.Konvertierungsfaktor in
das X-Register laden.Flagabfrage. Falls gesetzt
(Kilometereingabe) Sprung
zu [LBL] 2.Falls Flag 0 nicht gesetzt
(Meileneingabe) Konvertie-
rung in Kilometer und Ende
Programm.Konvertierung in Meilen
und Ende Programm.Rechner wieder auf Run
Mode schalten.

Konvertieren Sie mit dem Programm 26 Meilen in Kilometer und 1,5 Kilometer in Meilen.

Tastensequenz26
f A**Anzeige**26
41,8418Eingabe der Meilen.
Meilen/Kilometer-Konvertierung;
Anzeige des Resultats.1,5
f B1,5
0,9321Eingabe der Kilometer.
Kilometer/Meilen-Konvertierung;
Anzeige des Resultats.

Arbeitsweise. Nach Eingabe eines Werts und der Wahl der gewünschten Konvertierungsart (durch Drücken von f A für Meilen/Kilometer- und f B für Kilometer/Meilen-Konvertierung) wird der Flag 0 vom Rechner gesetzt oder zurückgesetzt. Anschließend wird mit der Ausführung des Rechenteils (ab Label 1) des Programms begonnen. Nach dem Laden des Konvertierungsfaktors in das X-Register fragt das Programm den Flag 0 ab. Wenn der Flag gesetzt ist (TRUE), wird zu Label 2 gesprungen; die Konvertierung in Meilen ausgeführt und die Programmausführung beendet. Bei nicht gesetztem Flag (FALSE) wird der bedingte Sprung zu LBL 2 übergangen, die Konvertierung in Kilometer durchgeführt und die Programmausführung beendet. Der Flag hat die Aufgabe dem Rechner mitzuteilen, welches der beiden Labels (A oder B) angesprungen wurde, und damit welche Konvertierung gewünscht wird.

Die Verwendung von Flags wird auf Seite 216, Flags, in Teil III dieses Handbuchs weiter beschrieben.

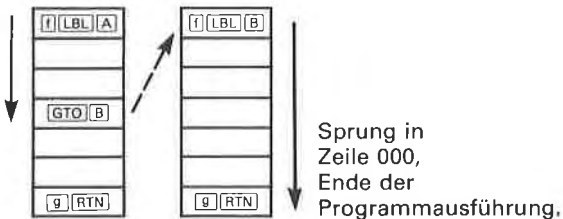
Unterprogramme

Kurzdarstellung der Unterprogrammtechniken

Es kommt häufig vor, daß ein Programm eine Folge von Befehlen enthält, die während eines Programmablaufs mehrmals ausgeführt werden. Wenn eine Befehlsfolge mehrmals innerhalb eines Programms auftritt, kann durch die Ausführung dieser Befehlsfolge als Unterprogramm Speicherplatz gespart werden.

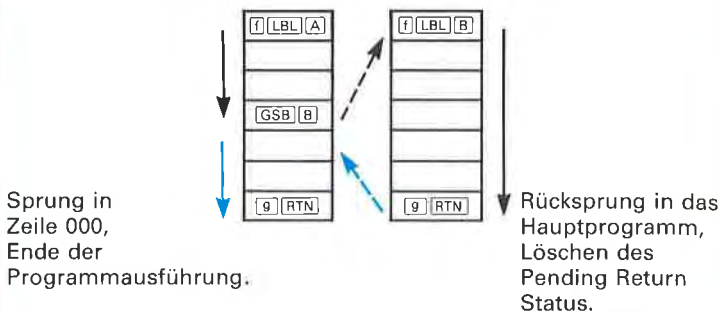
Aufruf von Unterprogrammen

Ein Unterprogramm wird durch den Befehl **[GSB]** (*go to subroutine*), gefolgt von dem gewünschten numerischen oder Alpha-Label, ausgeführt.* Bei einer **[GSB]** Anweisung wird die Kontrolle in der gleichen Weise wie bei einem **[GTO]** Befehl an die spezifizierte Labeladresse übergeben. Zusätzlich wird jedoch bei der Ausführung einer **[GSB]** Anweisung im Rechner eine „Pending Return“ (anstehende Rücksprung-) Bedingung gesetzt. Solange im Rechner eine Pending Return Bedingung existiert, bewirkt die als nächste im Programm gefundene **[RTN]** (*return*) Anweisung eine Fortsetzung der Programmausführung mit der ersten auf den letzten **[GSB]** Befehl folgenden Anweisung. Anschließend werden die Befehle im Programmspeicher wieder sequentiell „nach unten“ abgearbeitet. (Bei der Ausführung des **[RTN]** Befehls wird die Pending Return Bedingung gelöscht.) Vergleichen Sie die folgenden Illustrationen der Auswirkungen eines unbedingten Sprungs und des Sprungs in ein Unterprogramm.



* Ein **[GSB]** gefolgt von einem Alpha Label stellt eine abgekürzte Tastenfolge dar (siehe Seite 78/79, abgekürzte Tastenfolgen).

Sprung in ein Unterprogramm



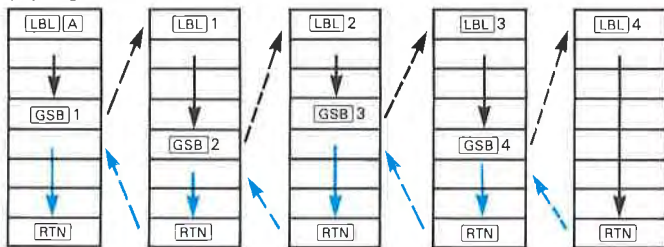
Wie Sie sehen, liegt der einzige Unterschied zwischen **GSB** (*go to subroutine*) und **GTO** (*go to branch*) in der Adresse, die durch die Ausführung der **RTN** Anweisung angesprungen wird.

Nach einem **GTO** bewirkt der nächste **RTN** Befehl einen Sprung in Zeile 000 und die Beendigung der Programmausführung; nach einem **GSB** bewirkt der nächste **RTN** Befehl einen Rücksprung in das Hauptprogramm und die Fortsetzung der Programmausführung dort.

Einschränkungen bei der Verwendung von Unterprogrammen

Ein Unterprogramm kann ein zweites Unterprogramm aufrufen, dieses wiederum ein drittes. Die Tiefe der Unterprogrammverschachtelung – der Aufruf von Unterprogrammen innerhalb eines

Hauptprogramm



Ende

Unterprogramms – wird nur durch die Anzahl der $\overline{\text{RTN}}$ Anweisungen beschränkt, die sich der HP-11C intern „merken“ kann. Das vorstehende Diagramm verdeutlicht die Abarbeitung von verschachtelten Unterprogrammen.

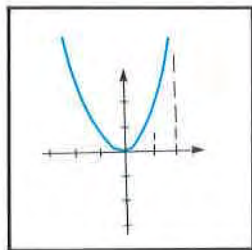
Der HP-11C kann jederzeit die Rücksprungadressen für vier Unterprogramme speichern. Wenn Sie versuchen in der vierten Unterprogrammebene ein weiteres Unterprogramm aufzurufen, so führt dies zu einem Abbruch der Programmausführung und zur Anzeige der Fehlermeldung „Error 5“.

Bemerkung: Während eine Unterprogrammverschachtelung nur aus bis zu vier Ebenen bestehen kann, unterliegt die Anzahl der Unterprogrammverschachtelungen oder die Anzahl der nicht verschachtelten Unterprogramme keinerlei Einschränkungen.

Benutzung von Unterprogrammen

Beispiel: Es ist ein Programm zu schreiben, das die durchschnittliche Steigung zwischen den Punkten x_1 und x_2 (siehe Zeichnung) berechnet. Die gezeichnete Kurve ist der Graph der Funktion.

$$y = x^2 - \sin(x).$$



Lösung: Die durchschnittliche Steigung wird durch die Formel

$$\frac{y_2 - y_1}{x_2 - x_1} = \frac{(x_2^2 - \sin(x_2)) - (x_1^2 - \sin(x_1))}{x_2 - x_1}$$

beschrieben.

Beachten Sie, daß bei der Lösung zwei Auswertungen der Funktion $x^2 - \sin(x)$ (eine an der Stelle $x = x_1$ und die zweite an der Stelle $x = x_2$) benötigt werden.

Da die Lösung einen Ausdruck enthält, der für beide Werte von x wiederholt werden muß, ist es sinnvoll, diesen Ausdruck in einem Unterprogramm zusammenzufassen, um dadurch Platz im Programmspeicher zu sparen.

Hauptprogramm

001- f LBL A
 002- STO 0
 003- $x \approx y$
 004- STO - 0
 005- GSB 0
 006- CHS
 007- $x \approx y$
 008- GSB 0
 009- +
 010- RCL 0
 011- \div
 012- g RTN

Unterprogramm

013- f LBL 0
 014- g x^2
 015- g LST x
 016- SIN
 017- -
 018- g RTN

(In diesem Programm wird der trigonometrische Mode [DEG] unterstellt.)

Wenn Sie [f] [A] auslösen, und zuvor das Y-Register mit x_1 und das X-Register mit x_2 geladen haben, beginnt die Ausführung mit der [LBL] [A] Anweisung in Zeile 001. Beim Erreichen des [GSB] 0 Befehls in Zeile 005 wird zu der [LBL] 0 Anweisung in Zeile 013 gesprungen und der Funktionswert y_1 berechnet. Für $x_1 = 2$ und $x_2 = 3$ würde beispielsweise y_1 wie folgt berechnet werden.

	001	002	003	004
T \rightarrow				
Z \rightarrow				
Y \rightarrow	2	2	3	3
X \rightarrow	3	3	2	2

Tasten \rightarrow [f] [LBL] [A] [STO] 0 $\text{[x} \approx \text{y]}$ [STO] [-] 0
 (x_1 im Y-Reg. (x_2 in R_0) (ver ($x_2 - x_1$ in R_0)
 x_2 im X-Reg.) tauschen)

	005	013	014	015
T →				
Z →				3
Y →	3	3	3	4
X →	2	2	4	2

Tasten → $\boxed{\text{GSB}} 0$ $\boxed{f} \boxed{\text{LBL}} 0$ $\boxed{g} \boxed{x^2}$ $\boxed{g} \boxed{\text{LST}} x$
 (Sprung zu (Anfang
 Label 0) Unter-
 programm)

	016	017	018	006
T →				
Z →	3			
Y →	4	3	3	3
X →	0,0349	3,9651	3,9651	-3,9651

Tasten → $\boxed{\text{SIN}}$ $\boxed{-}$ $\boxed{g} \boxed{\text{RTN}}$ $\boxed{\text{CHS}}$
 $\sin(x_1)$ $(x_1^2 - \sin x_1 = y_1)$ (Rücksprung (-
 Haupt $- \sin(x_1)$
 programm)

Der $\boxed{\text{RTN}}$ Befehl in Zeile 013 bewirkt einen Rücksprung in das Hauptprogramm und die Fortsetzung der Ausführung mit der auf die letzte $\boxed{\text{GSB}}$ Anweisung folgenden Zeile. Durch die $\boxed{\text{GSB}} 0$ Anweisung in Zeile 008 wird danach wieder zu dem $\boxed{\text{LBL}} 0$ Befehl in Zeile 013 gesprungen.

	007	008	013	014
T →				
Z →				
Y →	-3,9651	-3,9651	-3,9651	-3,9651
X →	3	3	3	9

Tasten → $\boxed{x \rightleftharpoons y}$ $\boxed{\text{GSB}} 0$ $\boxed{f} \boxed{\text{LBL}} 0$ $\boxed{g} \boxed{x^2}$
 (vertauschen) (Sprung zu (Anfang
 Label 0) Unter-
 programm)

	015	016	017	018
T →				
Z →	-3,9651	-3,9651		
Y →	9	9	-3,9651	-3,9651
X →	3	0,0523	8,9477	8,9477

Tasten → $\boxed{9} \boxed{\text{LST}_x}$ $\boxed{\text{SIN}}$ $\boxed{-}$ $\boxed{9} \boxed{\text{RTN}}$
 (x_2) $(\sin(x_2))$ $(x_2^2 - \sin(x_2) = y_2)$ (Rücksprung Hauptprogramm)

Wenn der Rechner zur Berechnung von y_2 ein zweites Mal das Unterprogramm mit dem $\boxed{\text{LBL}} 0$ durchlaufen hat, bewirkt der $\boxed{\text{f}} \boxed{\text{RTN}}$ Befehl in Zeile 018 den Rücksprung nach Zeile 009, dem ersten, auf die letzte $\boxed{\text{GSB}} 0$ Anweisung folgenden Befehl. Zu diesem Zeitpunkt befindet sich y_2 im X-Register, $-y_1$ im Y-Register, und die restlichen Programmbefehle vervollständigen die Berechnung der Steigung.

	009	010	011	012
T →				
Z →				
Y →		4,9826		
X →	4,9826	1	4,9826	4,9826

Tasten → $\boxed{+}$ $\boxed{\text{RCL}} 0$ $\boxed{+}$ $\boxed{9} \boxed{\text{RTN}}$
 $(y_2 - y_1)$ $(x_2 - x_1)$ $\frac{y_2 - y_1}{x_2 - x_1}$ (Ende Programm)

Nach beendiger Berechnung wird im Display die Steigung der Geraden, die durch die Punkte (x_1, y_1) und (x_2, y_2) geht, angezeigt.

Geben Sie nun die auf Seite 122 gelisteten Programmbefehle in den Rechner ein und berechnen Sie die durchschnittlichen Steigungen für die folgenden Wertepaare von x_1 und x_2 : 0,5, 1,25; 2,52, 3,72; 5,7.

Lösungen: 1,7325, 6,2226, 11,9826

Weitere Bemerkungen betreffend den Umgang mit Unterprogrammen, sind auf Seite 210 Unterprogramme, in Teil III dieses Handbuchs zu finden.

Beachten Sie, daß bei **[GSB]** und **[GTO]** Anweisungen der Rechner den Programmspeicher immer *nach unten* nach dem spezifizierten Label durchsucht. Dadurch ist es möglich, in Programmen ein bestimmtes Label mehrmals zu verwenden.

Beispiel: Das folgende Programm berechnet den Wert des Ausdrucks $\sqrt{x^2 + y^2 + z^2 + t^2}$

Dabei wird **[LBL] [A]** sowohl zur Kennzeichnung des Programmansfangs wie auch zur Definition eines Unterprogramms innerhalb des Programms verwendet. Das Programm wird gestartet, indem der Stack mit den Variablen x , y , z und t geladen und dann **[f] [A]** gedrückt wird.*

Tastensequenz	Anzeige	Bedeutung
[g] [P/R]	000-	
[f] CLEAR [PRGM]	000-	
[f] [LBL] [A]	001-42. 21. 11	
[g] x^2	002- 43 11	Berechnung von x^2
[GSB] [A]	003- 32 11	Berechnung von y^2 und $x^2 + y^2$
[GSB] [A]	004- 32 11	Berechnung von z^2 und $x^2 + y^2 + z^2$
[GSB] [A]	005- 32 11	Berechnung von t^2 und $x^2 + y^2 + z^2 + t^2$
\sqrt{x}	006- 11	Berechnung der Wurzel.
[g] [RTN]	007- 43 32	
[f] [LBL] [A]	008-43. 21. 11	
$x \approx y$	009- 34	
[g] x^2	010- 43 11	
+	011- 40	
[g] [RTN]	012- 43 32	
[g] [P/R]		

* In der Regel sollte die mehrfache Verwendung eines Labels vermieden werden, da dadurch die Gefahr von Programmierfehlern gesteigert und die Lesbarkeit eines Programms verringert wird. Das folgende Programm illustriert, wie ein Label (falls notwendig) mehrmals verwendet werden kann.

Testen Sie das Programm mit den folgenden Werten:

$$x = 4,3, \quad y = 7,9, \quad z = 1,3, \quad t = 8,0$$

Tastensequenz8 1,3 7,9 4,3 **Anzeige**

8,0000

1,3000

12,1074

Das Indexregister

Kurzdarstellung der Indexregister-Funktionen

Das Indexregister (R_1) ist eines der effektivsten, der auf Ihrem HP-11C verfügbaren Programmierhilfsmittel. Zusätzlich zum einfachen Speichern und Rückrufen von Daten kann das Indexregister auch zu

- Kontrollfunktionen und als Programmschleifenzähler,
- sowie zur indirekten Adressierung von Datenspeicher-Registern, Sprüngen und Unterprogrammen

verwendet werden.

Die Steuerung von Programmschleifen und die indirekte Adressierung über das Indexregister geschieht mittels eines Kontrollwerts, den Sie im Indexregister ablegen müssen. Der *ganzzahlige* Anteil bestimmt das Resultat jedes Schleifendurchlaufs oder der indirekten Adressierungsoperation. Der *dezimale* Anteil des Kontrollwerts enthält Parameter zur Änderung und Begrenzung des ganzzahligen Anteils.

Abgekürzte Tastenfolgen mit \boxed{I} und $\boxed{(i)}$. Der Rechner ist so ausgelegt, daß die Vorwahltaste \boxed{f} in Tastenfolgen mit \boxed{I} oder $\boxed{(i)}$ weggelassen werden kann. Selbst wenn Sie die Vorwahltaste \boxed{f} innerhalb einer \boxed{I} oder $\boxed{(i)}$ Tastenfolge drücken, wird der Code für \boxed{f} vom Rechner automatisch eliminiert.

Direkte Indexregister-Funktionen

Die direkten Indexregister-Funktionen betreffen den Inhalt des Indexregisters selbst.

Speicherung und Rückruf mit dem Indexregister. Die Befehle $\boxed{STO} \boxed{I}$ (*store in R_1*) und $\boxed{RCL} \boxed{I}$ (*recall from R_1*) arbeiten in der gleichen Weise wie \boxed{STO} und \boxed{RCL} mit den Datenspeicher-Registern R_0 bis R_9 und R_{10} bis R_{99} .

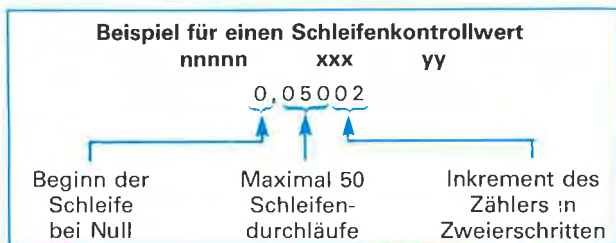
Vertauschen des Inhalts der Register X und I. Durch den Befehl $\boxed{X} \boxed{\rightleftharpoons} \boxed{I}$ werden die Inhalte des angezeigten X-Registers und des Indexregisters in der gleichen Weise vertauscht, wie dies durch Drücken von $\boxed{X} \boxed{\rightleftharpoons} \boxed{Y}$ mit den Inhalten des X- und des Y-Registers geschieht.

Steuerung von Programmschleifen mit dem Indexregister. Die Befehle **ISG** (*Inkrement und Sprung wenn größer*) und **DSE** (*Dekrement und Sprung wenn kleiner oder gleich*) benutzen einen von Ihnen im Indexregister abzulegenden Schleifenkontrollwert. Der Rechner verwendet den ganzzahligen Anteil des Schleifenkontrollwerts, um die Anzahl der aufeinanderfolgenden Schleifendurchläufe hochzuzählen und um zu bestimmen, wann die Schleife verlassen werden soll. Der ganzzahlige Anteil des Schleifenkontrollwerts wird ebenfalls zur indirekten Adressierung von Speicherregistern, Sprüngen und Unterprogrammen benutzt.

ISG und **DSE** interpretieren und vergleichen die Komponenten des Schleifenkontrollwerts, der nach dem folgenden Schema aufgebaut sein muß:

nnnnn.xxxyy , wobei	± nnnnn momentaner Zählerwert
	xxx Zählertestwert
	yy Wert des Inkrements oder Dekrements

nnnnn, der ganzzahlige Anteil des Kontrollwerts, wird als Zähler für die Programmschleifendurchläufe sowie zur indirekten Adressierung von Sprüngen, Unterprogrammen und Speicherregistern benutzt. **nnnnn** kann aus bis zu fünf Ziffern bestehen; falls nicht spezifiziert gilt der Defaultwert Null. **nnnnn** wird durch die Ausführung von **ISG** oder **DSE** erhöht oder herabgesetzt.



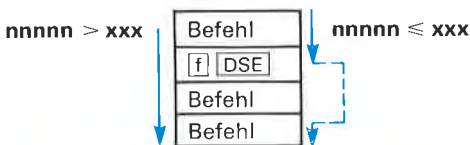
xxx besteht aus den ersten drei Stellen des dezimalen Anteils des Schleifenkontrollwerts. Bei jedem Inkrement oder Dekrement von **nnnnn** durch **ISG** oder **DSE** wird **xxx** vom Rechner intern mit **nnnnn** verglichen, um festzustellen, ob die gewünschte Anzahl an Inkrementen oder Dekrementen ausgeführt worden sind. **xxx** muß als dreistellige Zahl eingegeben werden, d. h. ein **xxx** Wert von 10 muß als 010 spezifiziert werden. (**xxx** ist ein Referenzwert, der durch die Ausführung von **ISG** oder **DSE** nicht verändert wird.)

yy bildet die vierte und fünfte Stelle des dezimalen Anteils des Kontrollwerts. **yy** bestimmt die Schrittweite, d.h. **nnnnn** wird um den Wert von **yy** erhöht oder herabgesetzt. Ein **yy** Wert muß zweistellig (z. B. 02, 03, 55) spezifiziert sein. Bei fehlender Angabe von **yy** gilt der Defaultwert 01. (**yy** ist ein Referenzwert, der durch die Ausführung von **[ISG]** oder **[DSE]** nicht verändert wird.)

Ablauf von [ISG] und [DSE]. Die Inkrement- und die Dekrementfunktion benutzen zur Steuerung des Programmablaufs **nnnnn/xxx** Vergleiche, die in ähnlicher Weise wie die Vergleichsoperationen in Abschnitt 7 arbeiten. Bei jeder Ausführung von **[f ISG]** wird **nnnnn** um **yy** erhöht. Danach wird abgefragt, ob **nnnnn** größer als **xxx** ist. Wenn ja, wird die nächste Zeile im Programmspeicher bei der Ausführung übersprungen.



Bei jeder Ausführung von **[f DSE]** wird **nnnnn** um **yy** herabgesetzt. Danach wird abgefragt, ob **nnnnn** kleiner oder gleich **xxx** ist. Wenn ja, wird die nächste Zeile im Programmspeicher bei der Ausführung übersprungen.



	Urspr. Kontrollwert in R_1	Veränderungen von nnnnn.xxyy während sukzessiver Durchläufe einer Programmschleife.			
Durchlauf →	0	1	2	3	4
[ISG] →	0,00602	2,00602	4,00602	6,00602	8,00602 entf.
[DSE] →	6,00002	4,00002	2,00002	0,00002 entf.	-2,00002 entf.

Indirekte Indexregister-Funktionen

Die indirekten Indexregister-Funktionen verändern den Inhalt von R_i nicht. Statt dessen wird **nnnnn**, der ganzzahlige Anteil des in R_i gespeicherten Kontrollwerts, als Adresse benutzt, die festlegt, wo im Programmspeicher die Funktion auszuführen ist. Indirekte Funktionen werden häufig zusammen mit **ISG** oder **DSE** Anweisungen in Programmen verwendet, wo die gleiche Programmstruktur wiederholt zur sequentiellen Adressierung von Speicherregistern, Sprüngen oder Unterprogrammen benutzt werden soll.

Indirekte Speicherung und Rückruf. Durch **STO** (i) und **RCL** $(i)^*$ werden Daten in dasjenige Datenspeicher-Register gespeichert bzw. von demjenigen Datenspeicher-Register zurückgerufen, das durch den Absolutwert von **nnnnn** adressiert ist. Die Register R_0 bis R_9 werden durch **nnnnn** Werte von 0 bis 9 adressiert; R_{10} bis R_{19} durch Werte von 10 bis 19 und R_i durch **nnnnn** = 20. (Siehe Indirekte Adressierungstabelle auf Seite 131.)

Indirektes Vertauschen des Inhalts des X-Registers. Durch den Befehl **f** $x \approx (i)$ werden die Inhalte des X-Registers und des durch den Absolutwert von **nnnnn** adressierten Speicherregisters vertauscht.

Indirekte Speicherregister-Arithmetik. Durch **STO** $(+)$, **STO** $(-)$, **STO** (\times) oder **STO** (\div) (i) werden die jeweiligen arithmetischen Operationen mit dem Inhalt des durch den Absolutwert von **nnnnn** adressierten Speicherregisters durchgeführt.

Beispiele:

Im Indexregister sei der Wert 5,01202 gespeichert, d. h.

nnnnn = 5, **xxx** = 012, **yy** = 02

STO (i) = **STO** 5

f $x \approx (i)$ = Inhalt der Register X und R_5 wird vertauscht

STO $(+)$ (i) = **STO** $(+)$ 5

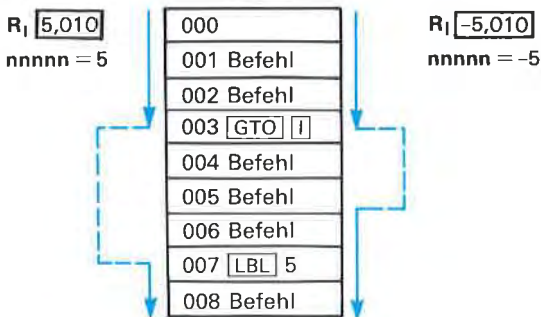
Indirekter Sprung zu einem Label oder einer Zeilennummer. Innerhalb eines ablaufenden Programms gelten die folgenden Regeln:

1. Wenn **nnnnn** ≥ 0 ist, wird durch **GTO** (i) zum nächsten durch **nnnnn** spezifizierten Label gesprungen. Der Programmspeicher wird „nach unten“ nach dem Label durchsucht.

* Siehe Seite 127, Abgekürzte Tastenfolgen mit (i) und (i) .

2. Wenn $nnnn < 0$ ist, wird durch $\boxed{\text{GTO}} \boxed{I}$ direkt zu der durch den Absolutwert von $nnnn$ spezifizierten *Zeilennummer* gesprungen. Die anzuspringende Zeile muß im belegten Bereich des Programmspeichers liegen.

Zum Beispiel:



Indirekter Unterprogrammaufruf. In einem ablaufenden Programm übergibt $\boxed{\text{GSB}} \boxed{I}$ in der gleichen Weise indirekt die Kontrolle an ein Unterprogramm, wie bei $\boxed{\text{GTO}} \boxed{I}$ die Kontrolle über einen indirekten Sprung übergeben wird.

nnnn Wert	Indirekte Adressierungstabelle Resultierende Adresse bei $\boxed{\text{GTO}} \boxed{I}$ oder $\boxed{\text{GSB}} \boxed{I}$:	Durch $\boxed{\text{STO}} \boxed{(i)}$ oder $\boxed{\text{RCL}} \boxed{(i)}$ adressiertes Register
0	$\boxed{f} \boxed{\text{LBL}} \boxed{0}$	R_0
⋮	⋮	⋮
9	$\boxed{f} \boxed{\text{LBL}} \boxed{9}$	R_9
10	" " \boxed{A}	$R_{,0}$
11	" " \boxed{B}	$R_{,1}$
12	" " \boxed{C}	$R_{,2}$
13	" " \boxed{D}	$R_{,3}$
14	" " \boxed{E}	$R_{,4}$
15	—	$R_{,5}$
16	—	$R_{,6}$
17	—	$R_{,7}$
18	—	$R_{,8}$
19	—	$R_{,9}$
20	—	R_I

Beispiel für eine Schleifensteuerung mit **ISG**

Beispiel: Das folgende Programm illustriert die Arbeitsweise von **ISG**. Es enthält eine Programmschleife mit einer Pause zur Anzeige des aktuellen Inhalts von R_1 und benutzt **ISG** zur Steuerung der Anzahl der Schleifendurchläufe und des Werts der Quadratzahl. Das Programm erzeugt eine Tabelle mit den Quadraten der geraden Zahlen von 2 bis 50.

Tastenfolge

g P/R

f CLEAR **PRGM**

f LBL **A**

2

.

0

5

0

0

2

STO **I**

f LBL 1

RCL **I**

g INT

f PSE

g x^2

f PSE

f ISG

GTO 1

g RTN

g P/R

Anzeige

000-

Program Mode

000-

001-42. 21. 11

Programmlabel

002-

2

Vorgabe des Zählerwerts
(nnnnn)

003-

48

004-

0

005-

5

006-

0

007-

0

008-

2

Wert des Inkrements (yy)

009-

44 25

Schleifenkontrollwert in R_1
speichern

010-42. 21. 1

Schleifenanfang

011-

45 25

Rückruf des Werts in R_1

012-

43 44

Ganzzahliger Anteil des
Kontrollwerts.

013-

42 31

Pause zur Anzeige von
nnnnn.

014-

43 11

nnnnn quadrieren.

015-

42 31

Anzeige des Quadrats von
nnnnn.

016-

42 6

Erhöhen des Zählers
nnnnn in R_1 um 2 und Ab-
frage ob Zählerwert größer
als Endwert (50). Wenn ja,
wird die nächste Pro-
grammzeile übersprungen.

017-

22 1

Rücksprung zum Schleifen-
anfang

018-

43 32

Ende Programm
Run Mode

Starten Sie nun das Programm mit $\boxed{f} \boxed{A}$.

Tastensequenz

$\boxed{f} \boxed{A}$

Anzeige

2,0000
4,0000
4,0000
16,0000
.
.
.
50,0000
2.500,0000

Zu Beginn der Ausführung wird zunächst die zu quadrierende Zahl und anschließend das Quadrat der Zahl angezeigt. Wenn der Zählerwert auf über 50 wächst, wird das Programm beendet

Der Programmablauf soll im folgenden etwas genauer betrachtet werden:

1. In dem auf das Label \boxed{A} folgenden Programmteil wird die Zahl 2,05002 als Schleifenkontrollwert im Indexregister gespeichert.

nnnnn	xxx	yy
(0000)2 Aktueller Zählerwert	050 Zähler- testwert	02 Inkrement- wert

2. Unter Label 1 wird die folgende Sequenz ausgeführt:

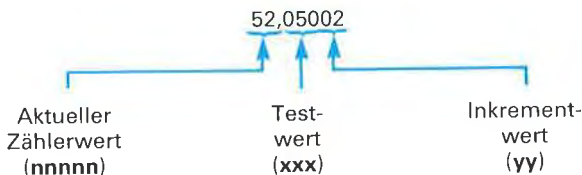
Nach der Anzeige von 2 und 4 (dem Quadrat von 2) wird der aktuelle Zählerwert in R_1 , 00002 (nnnnn) um den Inkrementwert 02 (yy) erhöht. Das Indexregister enthält jetzt den neuen Wert 4,05002, der vom Rechner wie folgt interpretiert wird:

nnnnn	xxx	yy
(0000)4 Aktueller Zählerwert	050 Zähler- testwert	02 Inkrement- wert

Anschließend wird der neue Zählerwert mit dem Testwert 050 (xxx) verglichen. Solange der Zählerwert den Testwert nicht überschreitet, wird die Ausführung mit der nächsten Zeile fortgesetzt. Da die nächste Zeile den Sprung an den Schleifenanfang, $\boxed{GTO} \boxed{1}$, enthält, wird der Prozeß mit der neuen Zahl wiederholt.

3. Nach der Anzeige der 25 geraden Zahlen (2–50) und ihrer Quadrate wird der aktuelle Zählerwert auf über 50 erhöht. Dies hat zur Folge, daß der Rechner die Zeile nach der **[ISG]** Anweisung in Zeile 16 überspringt. Daraus ergibt sich wiederum, daß der **[GTO]** 1 Befehl in Zeile 17 umgangen und stattdessen der **[RTN]** Befehl in Zeile 18 ausgeführt wird, was einen Sprung in Zeile 000 und die Beendigung der Programmausführung bedingt.

Drücken Sie nach dem Ablauf des Programms **[RCL] [I]**. Der zurückgerufene Indexregisterwert im Display sollte jetzt sein:



Einschränkungen bei der Verwendung von **[ISG] und **[DSE]**.** Beachten Sie, daß mit **[ISG]** und **[DSE]** jede beliebige Zahl, die der Rechner anzeigen kann, erhöht bzw. herabgesetzt werden kann. Dabei wird jedoch der dezimale Anteil des Schleifenkontrollwerts verändert, wenn der aktuelle Zählerwert den 5-stelligen **nnnnn** Wert überschreitet.

Beispielsweise würde die Zahl 99.950,50055 bei einer Erhöhung mit **[ISG]** zu 100.005,5006 werden. Die ursprüngliche Zahl ist dabei um den Wert 55 erhöht worden. Da aber die neue Zahl 100.005,50055 nicht vollständig angezeigt werden kann, wird der gebrochene Anteil dieser Zahl gerundet. Nun unterstellt der Rechner eine zweistellige Zahl als Inkrementwert (**yy**), das nächste Inkrement wäre also 60 und nicht 55. Der gerundete Wert (100.005,5006) werde jetzt um 60 reduziert. Wenn die herabgesetzte Zahl 999.945,5006 mit **[ISG]** wieder erhöht würde, wäre wegen der erneuten Rundung des gebrochenen Anteils das Ergebnis 1.000.005,501 zu erwarten. Da in diesem Fall kein Inkrementwert **yy** mehr vorhanden ist, würde als nächstes Inkrement der Defaultwert 01 an Stelle von 60 verwendet werden.

Direkte R_1 Operationen

Operation	Tastenfolge	Anzeige
Speichern in R_1 . (Anzeige löschen) Rückruf aus R_1 (der zuvor mit $\boxed{\text{STO}} \boxed{\text{I}}$ gespeicherten Zahl). (Anzeige löschen)	12345 $\boxed{\text{STO}} \boxed{\text{I}}$ \leftarrow	12345 12345,0000 0,0000
(Anzeige löschen) Vertauschen der Inhalte der Register X und I (unter Verwendung des oben im Indexregister gespeicherten Werts).	$\boxed{\text{RCL}} \boxed{\text{I}}$ \leftarrow	12345,0000 0,0000
	$\boxed{\text{f}}$ $\boxed{x \rightleftharpoons \text{I}}$	12345,0000

Indirekte R_1 Operationen

Operation	Tastenfolge	Anzeige
Speichern der Zahl 3 in R_1 als Adresse, d. h. nnnnn = 3.	3 $\boxed{\text{STO}} \boxed{\text{I}}$	3,0000
Indirektes Speichern und Rückruf von $\sqrt{7}$ unter Verwendung von R_3 (die indirekte Adressierung erfolgt über das obige Speichern von 3 in R_1).	7 $\boxed{\sqrt{x}}$ $\boxed{\text{STO}} \boxed{(i)}$ \leftarrow $\boxed{\text{RCL}} \boxed{(i)}$ \leftarrow	2,6458 2,6458 0,0000 2,6458 0,0000

Operation	Tastenfolge	Anzeige
Indirektes Vertauschen der Zahl 0.0000 im X-Register und der Zahl 2.6458 in R_3 (die nnnnn Adresse 3 bleibt in R_1).		
	$f \ x \geq (i)$	2,6458
	$f \ x \geq (i)$	0,0000
Indirekte Speicherregisterarithmetik; der Inhalt von R_3 ist durch π zu teilen.	$f \ \pi$	3,1416
	STO \div (i)	3,1416
	RCL (i)	0,8422

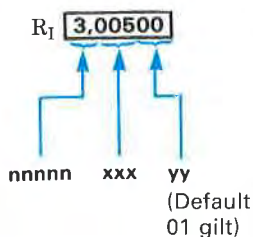
Eine weitergehende Diskussion der Techniken zur indirekten Speicherregisteradressierung ist auf Seite 211, **ISG** mit **RCL (i)**, in Teil III dieses Handbuchs zu finden.

Indirekte Programmkontrolle

Indirekte Label-Adressierung und indirekter Aufruf von Unterprogrammen

Analog zur indirekten Adressierung von Datenspeicher-Registern können auch Zeilennummern und Labels indirekt angesprungen werden. Die Tabelle auf Seite 132 enthält die jedem möglichen Label zugeordnete numerische Adresse. Jede numerische Adresse besteht aus dem ganzzahligen Anteil **nnnnn** eines im Indexregister gespeicherten, *positiven* Schleifenkontrollwerts.

Der indirekte *Sprung* zu einem Label erfolgt durch den Befehl **GTO** **I**. Wenn bei der Ausführung eines Programms eine **GTO** **I** Anweisung gefunden wird, so bewirkt dies die Übergabe der Kontrolle an dasjenige Label, das durch den aktuellen **nnnnn** Wert indirekt adressiert wird. Die folgende Illustration zeigt das Sprungverhalten für einen Kontrollwert von 3,005.



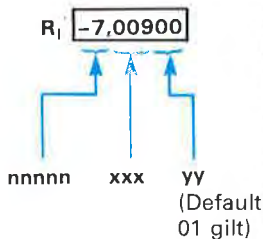
005- Befehl
006- Befehl
007- [GTO] [I]
008- Befehl
009- Befehl
010- Befehl
011- [f] [LBL] 3
012- Befehl
013- Befehl

Sprung zu Label 3 – dort Fortsetzung der Programmausführung.

Das gleiche Verfahren wird zur indirekten Ausführung von Unterprogrammen verwendet, mit der Ausnahme, daß an Stelle von [GTO] [I] der Befehl [GSB] [I] benutzt werden muß.

Indirekte Zeilennummer-Adressierung

Die Nummern von Programmzeilen können durch Sprung- oder Unterprogrammanweisungen in fast der gleichen Weise wie Labels indirekt adressiert werden. Wie bereits erwähnt wird bei einem *positiven* nnnnn Wert im Indexregister durch die Ausführung einer indirekten Sprung- oder Unterprogrammanweisung die Kontrolle an das durch nnnnn adressierte *Label* übergeben. Im Unterschied dazu wird bei einem *negativen* nnnnn Wert im Indexregister die Kontrolle an die durch den *Absolutwert* von nnnnn adressierte *Zeile* übergeben, d. h. an die Zeile mit der Nummer - nnnnn. Dieses Sprungsverhalten soll für den Kontrollwert -7,009 illustriert werden:



001- Befehl
002- Befehl
003- [GTO] [I]
004- Befehl
005- Befehl
006- Befehl
007- [f] [LBL] 3
008- Befehl
009- Befehl

Sprung zu Zeile 007 – dort Fortsetzung der Programmausführung.



Teil III

Programmierte Problemlösungen

Anwendungsprogramme

Matrizenrechnung

Diese Programm berechnet die Determinante und die Inverse einer 3×3 Matrix (sofern diese existiert, d.h. wenn die Determinante ungleich Null ist). Mittels einer geringfügigen Programmänderung kann zusätzlich die Lösung eines durch die Matrix dargestellten, linearen Gleichungssystems berechnet werden.

Gleichungen:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{33} & a_{33} \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$C = \frac{1}{\det A} \begin{bmatrix} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix}$$

C ist die Inverse von A und

$$\det A = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

und $\begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc.$

Die Gleichung $AX = B$ wird gelöst durch $X = CB$, d.h.

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \frac{1}{\det A} \begin{bmatrix} b_1 \begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix} & -b_2 \begin{bmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{bmatrix} & +b_3 \begin{bmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{bmatrix} \\ -b_1 \begin{bmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{bmatrix} & +b_2 \begin{bmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{bmatrix} & -b_3 \begin{bmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{bmatrix} \\ b_1 \begin{bmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{bmatrix} & -b_2 \begin{bmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{bmatrix} & +b_3 \begin{bmatrix} a_{12} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \end{bmatrix}$$

Bemerkung:

- Matrizenoperationen sind sehr anfällig gegenüber Rundungsfehlern, so daß in bestimmten Fällen verfälschte Ergebnisse auftreten können.
- Wenn sie mehrere lineare Gleichungssysteme zu lösen haben, berechnen Sie zuerst mit der Determinantenroutine die Determinante jeder der 3×3 Matrizen. Ersetzen Sie danach die Determinantenroutine durch die Gleichungssystemroutine und speichern Sie die jeweilige Determinante vor jeder Berechnung in R_0 .
- Eine 2×2 Matrix sollte wie folgt gespeichert werden:

$$\begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Wenn die Matrix A singularär ist, wird die Meldung **Error 0** angezeigt.

Hauptprogramm

Tastenfolge	Anzeige	Tastenfolge	Anzeige
f CLEAR PRGM	000-	RCL 1	027- 45 ,1
f LBL 9	001-42.21. 9	RCL 0	028- 45 ,0
f ISG	002- 42 6	RCL 8	029- 45 8
f PSE	003- 42 31	RCL 3	030- 45 ,3
RCL (i)	004- 45 24	GSB 8	031- 32 8
x	005- 20	RCL 5	032- 45 5
x	006- 20	RCL 3	033- 45 ,3
+	007- 40	RCL 1	034- 45 ,1
STO 0	008- 44 0	RCL 7	035- 45 7
g RTN	009- 43 32	GSB 8	036- 32 8
f LBL B	010-42.21.12	RCL 8	037- 45 8
RCL 9	011- 45 9	RCL 7	038- 45 7
RCL 3	012- 45 ,3	RCL 5	039- 45 5
RCL 2	013- 45 ,2	RCL 0	040- 45 ,0
RCL 0	014- 45 ,0	GSB 8	041- 32 8
GSB 8	015- 32 8	GSB 2	042- 32 2
RCL 2	016- 45 ,2	RCL 8	043- 45 8
RCL 7	017- 45 7	RCL 2	044- 45 ,2
RCL 6	018- 45 6	RCL 1	045- 45 ,1
RCL 3	019- 45 ,3	RCL 9	046- 45 9
GSB 8	020- 32 8	GSB 8	047- 32 8
RCL 6	021- 45 6	RCL 1	048- 45 ,1
RCL 0	022- 45 ,0	RCL 6	049- 45 6
RCL 9	023- 45 9	RCL 5	050- 45 5
RCL 7	024- 45 7	RCL 2	051- 45 ,2
GSB 8	025- 32 8	GSB 8	052- 32 8
GSB 2	026- 32 2	RCL 5	053- 45 5

Tastenfolge	Anzeige	Tastenfolge	Anzeige
$\boxed{\text{RCL}} \ 9$	054- 45 9	$\boxed{\text{R/S}}$	067- 31
$\boxed{\text{RCL}} \ 8$	055- 45 8	$\boxed{\text{g}} \ \boxed{\text{RTN}}$	068- 43 32
$\boxed{\text{RCL}} \ 6$	056- 45 6	$\boxed{\text{f}} \ \boxed{\text{LBL}} \ 2$	069-42.21. 2
$\boxed{\text{f}} \ \boxed{\text{LBL}} \ 8$	057-42.21. 8	$\boxed{\text{RCL}} \ 4$	070- 45 4
$\boxed{\times}$	058- 20	$\boxed{\text{g}} \ \boxed{\text{F?}} \ 0$	071-43. 6. 0
$\boxed{\text{R}} \ \boxed{\downarrow}$	059- 33	$\boxed{\text{R/S}}$	072- 31
$\boxed{\times}$	060- 20	$\boxed{\text{f}} \ \boxed{\text{LBL}} \ 3$	073-42.21. 3
$\boxed{\text{g}} \ \boxed{\text{R}} \ \boxed{\uparrow}$	061- 43 33	1	074- 1
$\boxed{-}$	062- 30	$\boxed{\text{STO}} \ \boxed{\text{I}}$	075- 44 25
$\boxed{\text{RCL}} \ 0$	063- 45 0	0	076- 0
$\boxed{\div}$	064- 10	$\boxed{\text{STO}} \ 4$	077- 44 4
$\boxed{\text{g}} \ \boxed{\text{F?}} \ 0$	065-43. 6. 0	$\boxed{\text{g}} \ \boxed{\text{RTN}}$	078- 43 32
$\boxed{\text{GTO}} \ 1$	066- 22 1		

Register			R_i : Index
R_0 : Det A	R_1 : b_1	R_2 : b_2	R_3 : b_3
R_4 : x_1, x_2, x_3	R_5 : a_{11}	R_6 : a_{12}	R_7 : a_{13}
R_8 : a_{21}	R_9 : a_{22}	R_{10} : a_{23}	R_{11} : a_{31}
R_{12} : a_{32}	R_{13} : a_{33}		

Determinantenroutine

Tastenfolge	Anzeige	Tastenfolge	Anzeige
$\boxed{\text{f}} \ \boxed{\text{LBL}} \ \boxed{\text{A}}$	079-42.21.11	0	083- 0
$\boxed{\text{g}} \ \boxed{\text{CF}} \ 0$	080-43. 5. 0	$\boxed{\text{RCL}} \ \boxed{\cdot} \ 0$	084- 45 ,0
4	081- 4	$\boxed{\text{RCL}} \ \boxed{\cdot} \ 2$	085- 45 ,2
$\boxed{\text{STO}} \ \boxed{\text{I}}$	082- 44 25	$\boxed{\text{GSB}} \ 9$	086- 32 9

Tastenfolge	Anzeige	Tastenfolge	Anzeige
RCL 8	087- 45 8	RCL 2	095- 45 ,2
RCL 3	088- 45 ,3	GSB 9	096- 32 9
GSB 9	089- 32 9	RCL 5	097- 45 5
RCL 9	090- 45 9	RCL 3	098- 45 ,3
RCL 1	091- 45 ,1	GSB 9	099- 32 9
GSB 9	092- 32 9	RCL 6	100- 45 6
CHS	093- 16	RCL 1	101- 45 ,1
RCL 7	094- 45 7	GSB 9	102- 32 9

Gleichungssystemroutine

Tastenfolge	Anzeige	Tastenfolge	Anzeige
f LBL 1	079-42.21. 1	STO 3	088- 44 3
RCL (i)	080- 45 24	R ↓	089- 33
x	081- 20	STO 2	090- 44 2
STO + 4	082-44.40. 4	R ↓	091- 33
f ISG	083- 42 6	STO 1	092- 44 1
f PSE	084- 42 31	GSB 3	093- 32 3
g RTN	085- 43 32	GSB B	094- 32 12
f LBL C	086-42.21.13	g CF 0	095-43. 5. 0
g SF 0	087-43. 4. 0	RCL 4	096- 45 4

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
1	Eingabe des Hauptprogramms (Zeile 000-078)			
2	Eingabe der Determinantenroutine (Zeile 079-102)			
			GTO 078	
			g P/R...g P/R	

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
3	User Mode wählen			
4	Matrizelemente speichern			
	a_{11}	a_{11}	$\boxed{\text{STO}} \ 5$	
	a_{12}	a_{12}	$\boxed{\text{STO}} \ 6$	
	a_{13}	a_{13}	$\boxed{\text{STO}} \ 7$	
	a_{21}	a_{21}	$\boxed{\text{STO}} \ 8$	
	a_{22}	a_{22}	$\boxed{\text{STO}} \ 9$	
	a_{23}	a_{23}	$\boxed{\text{STO}} \ \boxed{\cdot} \ 0$	
	a_{31}	a_{31}	$\boxed{\text{STO}} \ \boxed{\cdot} \ 1$	
	a_{32}	a_{32}	$\boxed{\text{STO}} \ \boxed{\cdot} \ 2$	
	a_{33}	a_{33}	$\boxed{\text{STO}} \ \boxed{\cdot} \ 3$	
5	Determinantenberechnung		$\boxed{\text{A}}$	Det A
6	Berechnung der			
	Inversen (optional)		$\boxed{\text{B}}$	c_{11}
			$\boxed{\text{R/S}}$	c_{12}
			$\boxed{\text{R/S}}$	c_{13}
			$\boxed{\text{R/S}}$	c_{21}
			$\boxed{\text{R/S}}$	c_{22}
			$\boxed{\text{R/S}}$	c_{23}
			$\boxed{\text{R/S}}$	c_{31}
			$\boxed{\text{R/S}}$	c_{32}
			$\boxed{\text{R/S}}$	c_{33}
7	Gleichungssystemroutine (optional)			
	(Löschen der Zeilen 079 bis 102)		$\boxed{\text{g}} \ \boxed{\text{RTN}} \ \boxed{\text{g}} \ \boxed{\text{BST}}$	
			$\boxed{\text{g}} \ \boxed{\text{P/R}}$	102- 32 9
			$\boxed{\leftarrow} \ \dots \ \boxed{\leftarrow}$	078- 43 32
8	Eingabe Unterprogramm $\boxed{\text{LBL}} \ 1$		$\boxed{\text{f}} \ \boxed{\text{LBL}} \ 1$	079-42 21. 1
	(Zeile 079 bis 096)		$\dots \ \boxed{\text{g}} \ \boxed{\text{P/R}}$	

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
9	Eingabe der rechten Seite	b_1	ENTER	
	und Berechnung von x	b_2	ENTER	
		b_3	C	x_1
			R/S	x_2
			R/S	x_3
10	Bei neuer rechter Seite			
	mit Schritt 9 fortfahren			
11	Für neue Determinante			
	oder Inverse Zeile 079		g RTN g BST	
	bis 096 löschen		g P/R	096- 45 4
			← ... ←	078- 43 32
	und Fortsetzung mit Schritt 2			

Beispiel 1: Berechnen Sie die Inverse von $\begin{bmatrix} 14 & -8 \\ -8 & 12 \end{bmatrix}$ und lösen

Sie dann

$$\begin{bmatrix} 14 & -8 \\ -8 & 12 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 20 \\ 5 \end{bmatrix}$$

Tastensequenz

Anzeige

User Mode wählen

f **FIX** 4

Eingabe der Determinantenroutine

14 **STO** 5

8 **CHS** **STO** 6

0 **STO** 7

8 **CHS** **STO** 8

12 **STO** 9

0 **STO** 0

STO 1 **STO** 2

1 **STO** 3

A

104,0000

Det A

B

0,1154

c_{11}

Tastensequenz

R/S

Anzeige

0,0769

 C_{12}

R/S

0,0000

 C_{13}

R/S

0,0769

 C_{21}

R/S

0,1346

 C_{22}

R/S

0,0000

 C_{23}

R/S

0,0000

 C_{31}

R/S

0,0000

 C_{32}

R/S

1,0000

 C_{33}

g RTN g BST

g P/R

102- 32 9

← ← ... ←

078- 43 32

Eingabe der Gleichungssystemroutine.

Tastensequenz

g P/R

Anzeige

20 ENTER 5 ENTER

0 C

2,6923

 X_1

R/S

2,2115

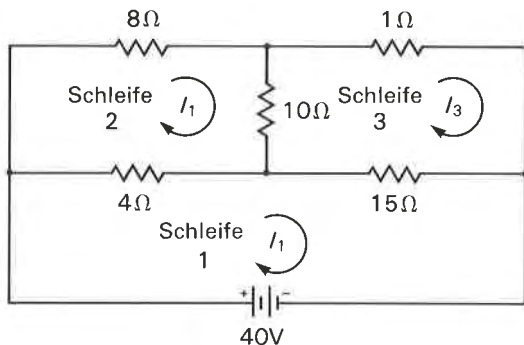
 X_2

R/S

0,0000

 X_3

Beispiel 2: Berechnen Sie die Schleifenspannungen im folgenden Schaltkreis.



Die drei Gleichungen lauten

$$\begin{array}{l} \text{Schleife 1} \quad 4I_1 - 4I_2 - 15I_3 - 40 = 0 \\ \text{Schleife 2} \quad 4I_2 - 4I_1 + 8I_2 + 10I_2 - 10I_3 = 0 \\ \text{Schleife 3} \quad 10I_3 - 10I_2 - 1I_3 + 15I_3 - 15I_1 = 0 \end{array}$$

oder

$$\begin{array}{r} 19I_1 - 4I_2 - 15I_3 = 40 \\ -4I_1 + 22I_2 - 10I_3 = 0 \\ -15I_1 - 10I_2 + 26I_3 = 0 \end{array}$$

bzw. in Matrixform

$$\begin{bmatrix} 19 & -4 & -15 \\ -4 & 22 & -10 \\ -15 & -10 & 26 \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} = \begin{bmatrix} 40 \\ 0 \\ 0 \end{bmatrix}$$

Tastenfolge

19 **[STO]** 5
 4 **[CHS]** **[STO]** 6
 15 **[CHS]** **[STO]** 7
 4 **[CHS]** **[STO]** 8
 22 **[STO]** 9
 10 **[CHS]** **[STO]** **[.]** 0
 15 **[CHS]** **[STO]** **[.]** 1
 10 **[CHS]** **[STO]** **[.]** 2
 26 **[STO]** **[.]** 3

Anzeige

Löschen Sie die Gleichungssystemroutine (falls noch nicht geschehen) und fügen Sie die Determinantenroutine ein.

[A] **2.402,0000**

Löschen Sie die Determinantenroutine und fügen Sie die der Gleichungssystemroutine ein.

40 **[ENTER]**
 0 **[ENTER]** **[C]** **7,8601**
[R/S] **4,2298**
[R/S] **6,1615**

Lineare Gleichungssysteme mit drei Unbekannten

Dieses Programm benutzt die Cramersche Regel zur Lösung von linearen Gleichungssystemen mit drei Unbekannten.

Das Programm wurde an dieser Stelle eingefügt, da es im Vergleich mit dem Programm zur Matrizenrechnung auf relativ einfache Weise zur Lösung linearer Gleichungssysteme verwendet werden kann.

Gleichungen: Ein lineares Gleichungssystem mit drei Unbekannten kann dargestellt werden als

$$AX = B$$

wobei

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad B = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

Für die Determinante des Systems gilt

$$\det A = a_{11}(a_{22}a_{33} - a_{23}a_{32}) - a_{12}(a_{21}a_{33} - a_{23}a_{31}) + a_{13}(a_{21}a_{32} - a_{22}a_{31})$$

Die Komponenten x_i der Lösung X werden bestimmt durch

$$x_i = \frac{\det(i)}{\det A}$$

wobei $\det(i)$ die Determinante der Matrix A ist, wenn man zuvor die i -te Spalte durch B ersetzt. (Dies gilt nur für $\det A \neq 0$.)

Bemerkungen:

1. Wenn die Determinante von A gleich 0 ist, sind mindestens zwei der Gleichungen linear abhängig. In diesem Fall ist dieses Programm nicht anwendbar, und die Ausführung wird mit der Fehlermeldung **Error 0** beendet.
2. Ist der Absolutwert der Determinante sehr klein, so enthält die Rechnerdarstellung dieser Zahl einen signifikanten Rundungsfehleranteil und der Quotient $\det(i)/\det A$ wird stark fehlerbehaftet sein.

Tastenfolge	Anzeige	Tastenfolge	Anzeige
f CLEAR PRGM	000-	RCL 5	029- 45 5
f LBL A	001-42.21.11	RCL 7	030- 45 7
1	002- 1	GSB 9	031- 32 9
.	003- 48	CHS	032- 16
0	004- 0	RCL 3	033- 45 3
0	005- 0	RCL 8	034- 45 8
9	006- 9	GSB 9	035- 32 9
STO I	007- 44 25	RCL 1	036- 45 1
f LBL 2	008-42.21. 2	RCL 9	037- 45 9
RCL I	009- 45 25	GSB 9	038- 32 9
g INT	010- 43 44	RCL 2	039- 45 2
f PSE	011- 42 31	RCL 7	040- 45 7
RCL (i)	012- 45 24	f LBL 9	041-42.21. 9
R/S	013- 31	f DSE	042- 42 5
STO (i)	014- 44 24	f PSE	043- 42 31
f ISG	015- 42 6	RCL (i)	044- 45 24
GTO 2	016- 22 2	x	045- 20
GSB 0	017- 32 0	x	046- 20
STO 0	018- 44 0	+	047- 40
R/S	019- 31	g RTN	048- 43 32
f LBL 0	020-42.21. 0	f LBL B	049-42.21.12
0	021- 0	STO . 3	050- 44 ,3
STO I	022- 44 25	R↓	051- 33
RCL 6	023- 45 6	STO . 2	052- 44 ,2
RCL 8	024- 45 8	R↓	053- 33
GSB 9	025- 32 9	STO . 1	054- 44 ,1
RCL 4	026- 45 4	1	055- 1
RCL 9	027- 45 9	GSB 7	056- 32 7
GSB 9	028- 32 9	R/S	057- 31

Tastenfolge	Anzeige	Tastenfolge	Anzeige
4	058- 4	$\boxed{\text{RCL}} \boxed{\cdot} 0$	075- 45 ,0
$\boxed{\text{GSB}} 7$	059- 32 7	$\boxed{\text{g}} \boxed{\text{RTN}}$	076- 43 32
$\boxed{\text{R/S}}$	060- 31	$\boxed{\text{f}} \boxed{\text{LBL}} 8$	077-42.21. 8
7	061- 7	$\boxed{\text{RCL}} \boxed{\cdot} 1$	078- 45 ,1
$\boxed{\text{GSB}} 7$	062- 32 7	$\boxed{\text{f}} \boxed{\text{x} \approx (i)}$	079- 42 23
$\boxed{\text{R/S}}$	063- 31	$\boxed{\text{STO}} \boxed{\cdot} 1$	080- 44 ,1
$\boxed{\text{f}} \boxed{\text{LBL}} 7$	064-42.21. 7	$\boxed{\text{f}} \boxed{\text{ISG}}$	081- 42 6
$\boxed{\text{STO}} \boxed{\cdot} 4$	065- 44 ,4	$\boxed{\text{f}} \boxed{\text{PSE}}$	082- 42 31
$\boxed{\text{STO}} \boxed{\text{I}}$	066- 44 25	$\boxed{\text{RCL}} \boxed{\cdot} 2$	083- 45 ,2
$\boxed{\text{GSB}} 8$	067- 32 8	$\boxed{\text{f}} \boxed{\text{x} \approx (i)}$	084- 42 23
$\boxed{\text{GSB}} 0$	068- 32 0	$\boxed{\text{STO}} \boxed{\cdot} 2$	085- 44 ,2
$\boxed{\text{RCL}} 0$	069- 45 0	$\boxed{\text{f}} \boxed{\text{ISG}}$	086- 42 6
$\boxed{\div}$	070- 10	$\boxed{\text{f}} \boxed{\text{PSE}}$	087- 42 31
$\boxed{\text{STO}} \boxed{\cdot} 0$	071- 44 ,0	$\boxed{\text{RCL}} \boxed{\cdot} 3$	088- 45 ,3
$\boxed{\text{RCL}} \boxed{\cdot} 4$	072- 45 ,4	$\boxed{\text{f}} \boxed{\text{x} \approx (i)}$	089- 42 23
$\boxed{\text{STO}} \boxed{\text{I}}$	073- 44 25	$\boxed{\text{STO}} \boxed{\cdot} 3$	090- 44 ,3
$\boxed{\text{GSB}} 8$	074- 32 8	$\boxed{\text{g}} \boxed{\text{RTN}}$	091- 43 32

Register			R_i : Index
R_0 : Det	R_1 : a_{11}	R_2 : a_{21}	R_3 : a_{31}
R_4 : a_{12}	R_5 : a_{22}	R_6 : a_{32}	R_7 : a_{13}
R_8 : a_{23}	R_9 : a_{33}	R_0 : det (i)	R_1 : b_1
R_2 : b_2	R_3 : b_3	R_4 : Index	

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
1	Eingabe des Programms			
2	User Mode wählen			

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
3	Eingabe der 3×3 Matrix		A	1, R_1
	(Bemerkung: Durch Drücken	a_{11}	R/S	2, R_2
	von R/S bleibt der	a_{21}	R/S	3, R_3
	angezeigte Wert im Register)	a_{31}	R/S	4, R_4
		a_{12}	R/S	5, R_5
		a_{22}	R/S	6, R_6
		a_{32}	R/S	7, R_7
		a_{13}	R/S	8, R_8
		a_{23}	R/S	9, R_9
		a_{33}	R/S	Det.
	Anzeige der Matrix durch A			
4	Eingabe der rechten Seite	b_1	ENTER	b_1
		b_2	ENTER	b_2
		b_3	B	x_1
			R/S	x_2
			R/S	x_3
5	Bei neuer rechter Seite Fortsetzung mit Schritt 4			
6	Bei neuer 3×3 Matrix Fortsetzung mit Schritt 3			

Beispiel 1: Gesucht ist die Lösung des Gleichungssystems

$$\begin{bmatrix} 19 & -4 & -15 \\ -4 & 22 & -10 \\ -15 & -10 & 26 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 40 \\ 0 \\ 0 \end{bmatrix}$$

Tastenfolge

f CLEAR **REG**

User Mode wählen

f **FIX** 4

A

Anzeige

1,0000

0,0000

Tastensequenz	Anzeige		
19 R/S	2,0000	0,0000	
4 CHS R/S	3,0000	0,0000	
15 CHS R/S	4,0000	0,0000	
4 CHS R/S	5,0000	0,0000	
22 R/S	6,0000	0,0000	
10 CHS R/S	7,0000	0,0000	
15 CHS R/S	8,0000	0,0000	
10 CHS R/S	9,0000	0,0000	
26 R/S	2.402,0000		(Determinante)
40 ENTER			
0 ENTER B	7,8601		(x ₁)
R/S	4,2298		(x ₂)
R/S	6,1615		(x ₃)

Beispiel 2: Gesucht ist die Lösung des Gleichungssystems

$$\begin{bmatrix} 19 & -4 & 4 \\ 5 & -12 & -10 \\ -15 & 8 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -3 \\ 4 \end{bmatrix}$$

Tastensequenz	Anzeige		
A	1,0000	19,0000	
R/S	2,0000	-4,0000	
5 R/S	3,0000	-15,0000	
R/S	4,0000	-4,0000	
R/S	5,0000	22,0000	
12 CHS R/S	6,0000	-10,0000	
8 R/S	7,0000	-15,0000	
4 R/S	8,0000	-10,0000	
R/S	9,0000	26,0000	
3 R/S	-264,0000		(Determinante)
5 ENTER			
3 CHS ENTER			
4 B	-1,6667		(x ₁)
R/S	-4,4091		(x ₂)
R/S	4,7576		(x ₃)

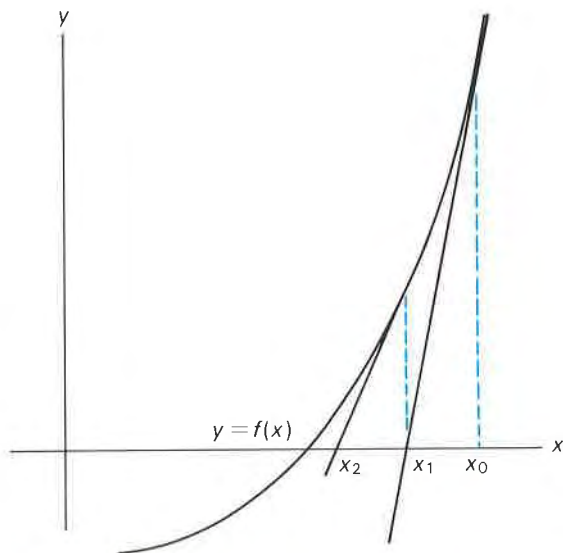
Nullstellenbestimmung mit dem Newton-Verfahren

Eines der häufigsten und schwierigsten Probleme der Algebra ist die Lösung einer Gleichung wie

$$\ln x + 3x = 10,8074$$

in der sich die Unbekannte x nicht isolieren läßt. Dies besagt, daß in diesem Fall keine einfache, algebraische Lösung existiert. Das folgende Programm benutzt das Newton-Verfahren zur Berechnung einer Lösung der Gleichung $f(x) = 0$, wobei die Funktion $f(x)$ vom Benutzer vorzugeben ist.

Der Benutzer muß die Funktion $f(x)$ definieren, in dem er die zur Berechnung des Funktionswerts notwendige Tastenfolge in den Programmspeicher eingibt. Dabei wird vorausgesetzt, daß sich das Argument x im X-Register befindet. Das Hauptprogramm selbst besteht aus 68 Programmzeilen und benötigt die Register 0 bis 4. Der restliche Speicherbereich und die anderen Register können vom Benutzer zur Definition von $f(x)$ verwendet werden. Zusätzlich muß der Benutzer dem Programm eine Startnäherung x_0 für die unbekannte Nullstelle zur Verfügung stellen. Je näher die Startnäherung an der gesuchten Nullstelle liegt, desto schneller konvergiert das Verfahren gegen eine Lösung.



Die Programmausführung wird abgebrochen, wenn die folgenden Bedingungen erfüllt sind:

1. Iterationszähler \leq Iterationsmaximum

2. $|f(x)| \leq \varepsilon$ 3. $|x_{i+1} - x_i| \leq \Delta x$

Dabei ist ε eine Toleranz, die festlegt, ab wann ein Funktionswert numerisch als Nullstelle betrachtet werden soll. Δx ist ein Grenzwert, der festlegt, daß die Iteration abgebrochen werden soll, wenn sich das Ergebnis einer Iteration vom Ergebnis der vorangegangenen Iteration betragsmäßig um weniger als Δx unterscheidet. Die beiden Werte können vom Benutzer vorgegeben werden; bei fehlender Angabe wählt das Programm interne Defaultwerte.

Der Newton'sche Algorithmus lautet:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)} \quad i = 0, 1, 2, \dots$$

Im Programm wird die folgende numerische Approximation der Ableitung $f'(x)$ benutzt:

$$x_{i+1} = x_i - \delta_i \frac{f'(x_i)}{f(x_i + \delta_i) - f(x_i)} \quad i = 0, 1, 2, \dots$$

Dabei ist $\delta_i = 10^{-5} x_0$ für $i = 0$ und $\delta_i = x_{i+1} - x_i$ sonst

Bemerkungen:

1. Nach Beendigung des Programms wird die berechnete Nullstelle angezeigt. Dieser Wert befindet sich zusätzlich auch in Register 2.
2. Die Fehlermeldung **Error 0** erscheint, wenn die Gleichung zur Berechnung von x_{i+1} durch Null geteilt wird. In diesem Fall ist das Programm mit einer neuen Startnäherung, die sich von der alten Startnäherung nur geringfügig unterscheiden sollte, erneut zu starten.
3. Das Programm kontrolliert die Anzahl der zur Berechnung einer Nullstelle benötigten Iterationen. Das Indexregister wird in den Zeilen 020, 021 und 022 mit dem Iterationsmaximum 50 initialisiert. Bei Überschreiten dieses Werts wird die Meldung **Error 4** ausgegeben. Der Benutzer kann das Iterationsmaximum durch Löschen der Zeilen 020 und 021 und Eingabe der gewünschten Zahl beliebig verändern.
4. Unter Umständen ist der Benutzer an der Folge der berechneten Zahlen x_{i+1} interessiert. Diese Werte werden angezeigt, wenn hinter dem **[STO] 2** Befehl in Zeile 047 (sofern keine Änderungen gemacht wurden) eine **[f] [PSE]** Anweisung eingefügt wird.
5. Wenn die Toleranz ε zu klein ist, wächst der Rundungsfehlereinfluß und die berechneten Nullstellen können fehlerbehaftet sein.

Tastenfolge	Anzeige	Tastenfolge	Anzeige
f CLEAR PRGM	000-	GSB C	029- 32 13
f LBL A	001-42.21.11	STO 2	030- 44 2
EEX	002- 26	RCL 0	031- 45 0
CHS	003- 16	RCL 1	032- 45 1
5	004- 5	+	033- 40
STO 0	005- 44 0	GSB C	034- 32 13
EEX	006- 26	RCL 2	035- 45 2
CHS	007- 16	-	036- 30
8	008- 8	g x=0	037- 43 40
STO 3	009- 44 3	STO . 9	038- 44 ,9
STO 4	010- 44 4	RCL 2	039- 45 2
R/S	011- 31	x≅y	040- 34
STO 3	012- 44 3	÷	041- 10
R/S	013- 31	RCL 0	042- 45 0
STO 4	014- 44 4	x	043- 20
g CLx	015- 43 35	CHS	044- 16
g RTN	016- 43 32	RCL 1	045- 45 1
f LBL B	017-42.21.12	+	046- 40
STO 2	018- 44 2	STO 2	047- 44 2
STO x 0	019-44.20. 0	g LSTx	048- 43 36
5	020- 5	-	049- 30
0	021- 0	STO 0	050- 44 0
STO 1	022- 44 25	f DSE	051- 42 5
GSB 1	023- 32 1	GTO 2	052- 22 2
RCL 2	024- 45 2	GSB 9	053- 32 9
g RTN	025- 43 32	f LBL 2	054-42.21. 2
f LBL 1	026-42.21. 1	RCL 2	055- 45 2
RCL 2	027- 45 2	GSB C	056- 32 13
STO 1	028- 44 1	g ABS	057- 43 16

Tastenfolge	Anzeige	Tastenfolge	Anzeige
$\boxed{\text{RCL}} \ 3$	058- 45 3	$\boxed{\text{g}} \ \boxed{\text{ABS}}$	064- 43 16
$\boxed{x \approx y}$	059- 34	$\boxed{\text{f}} \ \boxed{x \leq y}$	065- 42 10
$\boxed{\text{f}} \ \boxed{x > y}$	060- 42 20	$\boxed{\text{g}} \ \boxed{\text{RTN}}$	066- 43 32
$\boxed{\text{GTO}} \ 1$	061- 22 1	$\boxed{\text{GTO}} \ 1$	067- 22 1
$\boxed{\text{RCL}} \ 4$	062- 45 4	$\boxed{\text{f}} \ \boxed{\text{LBL}} \ \boxed{\text{C}}$	068-42.21.13
$\boxed{\text{RCL}} \ 0$	063- 45 0		

Register			R_i : Schleifenzähler
$R_0: \delta_i$	R_1 : temporär	R_2 : temporär	$R_3: \epsilon$
$R_4: \Delta x$	R_5 - R_8 : unbelegt		

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
1	Eingabe des Programms			
2	Sprung zu $\boxed{\text{LBL}} \ \boxed{\text{C}}$		$\boxed{\text{GTO}} \ \boxed{\text{C}}$	
3	Program Mode wählen		$\boxed{\text{g}} \ \boxed{\text{P/R}}$	068-42.21.13
4	Eingabe der Funktion	$f(x)$		
5	Run Mode wählen		$\boxed{\text{g}} \ \boxed{\text{P/R}}$	
6	User Mode wählen			
7	Toleranzen initialisieren		$\boxed{\text{A}}$	10^{-8}
8	(Optionale) Eingabe ϵ .	ϵ	$\boxed{\text{R/S}}$	ϵ
	Wenn der Defaultwert für ϵ (10^{-8}) nicht geändert werden soll, Schritt 9		$\boxed{\text{R/S}}$	10^{-8}
9	(Optionale) Eingabe Δx .	Δx	$\boxed{\text{R/S}}$	Δx
10	Eingabe Startnäherung u. Berechn. d. Nullst.	x_0	$\boxed{\text{B}}$	x_i

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
11	Bei neuer Startnäherung			
	Fortsetzung m., Schritt 10			
12	Bei neuen Toleranzen Fortsetzung m., Schritt 8			
13	Bei neuer Funktion $f(x)$			
	Programm beenden;		\boxed{g} \boxed{RTN} \boxed{g} \boxed{BST}	
	Program Mode wählen und		\boxed{g} $\boxed{P/R}$	
	alte Funktion (ab \boxed{LBL} \boxed{C})		$\boxed{\leftarrow}$ $\boxed{+}$ $\boxed{\leftarrow}$	068-42.21.13
	löschen. Forts. mit Schritt 4.			

Beispiel: Gesucht ist die größte Nullstelle des Polynoms

$$f(x) = x^6 - x - 1$$

d. h. die größte Lösung der Gleichung

$$x^6 - x - 1 = 0$$

Die Startnäherung sei $x_0 = 2$.

Tastenfolge

\boxed{GTO} \boxed{C}
 \boxed{g} $\boxed{P/R}$
 \boxed{STO} $\boxed{5}$
 $\boxed{6}$ $\boxed{V^x}$
 \boxed{RCL} $\boxed{5}$ $\boxed{=}$ $\boxed{1}$ $\boxed{=}$
 \boxed{g} $\boxed{P/R}$
 \boxed{f} \boxed{FIX} $\boxed{9}$
 \boxed{f} \boxed{USER} ,
 \boxed{A}
 $\boxed{2}$ \boxed{B}
 \boxed{C}

Anzeige

068-42. 21. 13
 069- 44 5
 071- 14
 075- 30

0,00000010
 1,134724138
 -0,00000004

Voreinstellung ε und Δx
 x_i
 $f(x_i)$

Numerische Integration

Dieses Programm dient zur numerischen Integration einer Funktion, deren Funktionswerte an einer endlichen Anzahl äquidistanter Punkte bekannt sind. Die Integrale werden entweder mit der Simpson- oder der Trapezregel approximiert.

Verfahren:

Durch x_0, x_1, \dots, x_n seien $n + 1$ äquidistante Punkte ($x_j = x_0 + j \cdot h$, $j = 0, 1, \dots, n$) gegeben, an denen die zugehörigen Funktionswerte $f(x_0), f(x_1), \dots, f(x_1), \dots, f(x_n)$ der Funktion $f(x)$ bekannt sind.

Das Integral $\int_{x_0}^{x_n} f(x) dx$

kann dann approximiert werden durch:

1. Trapezregel:

$$\int_{x_0}^{x_n} f(x) dx \sim \frac{h}{2} \left[f(x_0) + 2 \left(\sum_{j=1}^{n-1} f(x_j) \right) + f(x_n) \right]$$

2. Simpson-Regel:

$$\int_{x_0}^{x_n} f(x) dx \sim \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + \dots + 4f(x_{n-3}) + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]$$

Zur Anwendung der Simpson-Regel muß n gerade sein. Ist dies nicht der Fall, erscheint die Meldung **Error 0**.

Tastenfolge	Anzeige	Tastenfolge	Anzeige
\boxed{f} CLEAR $\boxed{\text{PRGM}}$	000-	$\boxed{\text{STO}}$ 0	005- 44 0
\boxed{f} $\boxed{\text{LBL}}$ $\boxed{\text{A}}$	001-42.21.11	$\boxed{\text{STO}}$ 5	006- 44 5
$\boxed{\text{STO}}$ 4	002- 44 4	0	007- 0
$\boxed{\text{R/S}}$	003- 31	$\boxed{\text{STO}}$ 3	008- 44 3
\boxed{f} $\boxed{\text{LBL}}$ $\boxed{\text{B}}$	004-42.21.12	\boxed{f} $\boxed{\text{LBL}}$ 1	009-42.21. 1

Tastenfolge	Anzeige	Tastenfolge	Anzeige
R/S	010- 31	GSB 3	035- 32 3
f LBL B	011-42.21.12	3	036- 3
STO 1	012- 44 1	RCL 5	037- 45 5
GSB 2	013- 32 2	f LBL 0	038-42.21. 0
ENTER	014- 36	RCL 1	039- 45 1
+	015- 40	-	040- 30
STO + 5	016-44.40. 5	RCL 4	041- 45 4
1	017- 1	x	042- 20
STO + 3	018-44.40. 3	$x \approx y$	043- 34
RCL 3	019- 45 3	÷	044- 10
R/S	020- 31	R/S	045- 31
f LBL B	021-42.21.12	f LBL 2	046-42.21. 2
STO 1	022- 44 1	ENTER	047- 36
GSB 2	023- 32 2	+	048- 40
STO + 5	024-44.40. 5	STO + 0	049-44.40. 0
1	025- 1	g RTN	050- 43 32
STO + 3	026-44.40. 3	f LBL 3	051-42.21. 3
RCL 3	027- 45 3	2	052- 2
GTO 1	028- 22 1	÷	053- 10
f LBL C	029-42.21.13	f FRAC	054- 42 44
2	030- 2	g $x=0$	055- 43 40
RCL 0	031- 45 0	g RTN	056- 43 32
GTO 0	032- 22 0	0	057- 0
f LBL D	033-42.21.14	+	058- 10
RCL 3	034- 45 3	g RTN	059- 43 32

Register		R _i : unbelegt	
R ₀ : belegt	R ₁ : $f(x)$, a	R ₂ : unbelegt	R ₃ : j
R ₄ : h	R ₅ : belegt	R ₆ -R ₉ : unbelegt	

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
1	Eingabe des Programms			
2	User Mode wählen			
3	Eingabe d. Schrittweite zwischen d. x-Werten	h	\boxed{A}	h
4	Eingabe d. Funktionswerte an d. Punkten $x_j, j = 0, 1, \dots, n$	$f(x_j)$	\boxed{B}	j
5	Berechnung des Integrals mit der Trapezregel oder der Simpson-Regel (hier muß n gerade sein)		\boxed{C} \boxed{D}	Trap \int Simp \int
6	Zur Berechnung des nächsten Integrals Forts. mit Schritt 3			

Beispiel: Mit den in der Tabelle gelisteten Werten für $f(x_i), i = 0, 1, \dots, 8$, sind Approximationen des Integrals

$$\int_0^2 f(x) dx$$

mittels der Trapezregel und der Simpson-Regel zu berechnen. Die Schrittweite h beträgt 0,25.

i	0	1	2	3	4	5	6	7	8
x_i	0	0,25	0,5	0,75	1	1,25	1,5	1,75	2
$f(x_i)$	2	2,8	3,8	5,2	7	9,2	12,1	15,6	20

Tastenfolge

\boxed{f} \boxed{FIX} 2

User Mode wählen

,25 \boxed{A}

2 \boxed{B}

2,8 \boxed{B}

3,8 \boxed{B}

5,2 \boxed{B}

Anzeige

0,25

0,00

1,00

2,00

3,00

Tastensequenz

7 **B**
 9,2 **B**
 12,1 **B**
 15,6 **B**
 20 **B**
C
D

Anzeige

4,00
 5,00
 6,00
 7,00
 8,00
 16,68
 16,56

(Trapezregel)
 (Simpson-Regel)

Kurvenanpassung

Der HP-11C enthält die Funktion „Lineare Regression“ (**L.R.**), die es auf einfache und schnelle Weise ermöglicht, eine Gerade durch eine vorgegebene Datenmenge zu legen (s. Seite 63).

Diese Funktion wird hier in einem Programm dazu benutzt, auch andere Kurven durch vorgegebenen Datenpunkte zu legen. Dies sind:

1. Exponentialkurven $y = a \cdot e^{bx}$ ($a > 0$)
2. logarithmische Kurven $y = a + b \cdot \ln x$
3. Potenzkurven $y = a \cdot x^b$ ($a > 0$)

Alle diese Kurven lassen sich in eine allgemeine, lineare Funktion der Form

$$Y = A + bX$$

transformieren.

Die Regressionskoeffizienten A und b werden durch das folgende, lineare Gleichungssystem bestimmt.

$$\begin{bmatrix} n & \sum X_i \\ \sum X_i^2 & \sum X_i^2 \end{bmatrix} \begin{bmatrix} A \\ b \end{bmatrix} = \begin{bmatrix} \sum Y_i \\ \sum (Y_i X_i) \end{bmatrix}$$

wobei n die Anzahl der Datenpaare ist.

Zwischen den einzelnen Variablen gelten die folgenden Beziehungen:

Regression	A	x_i	y_i	Code
Exponentiell	$\ln a$	x_i	$\ln y_i$	1
Logarithmisch	a	$\ln x_i$	y_i	2
Potenz	$\ln a$	$\ln x_i$	$\ln y_i$	3

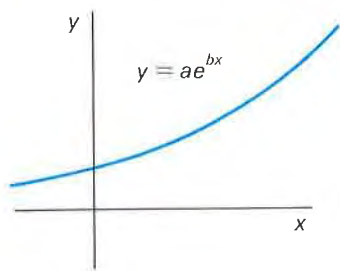
Die Güte einer Regression wird durch den Korrelationskoeffizienten

$$r^2 = \frac{A \sum Y_i + b \sum X_i Y_i - \frac{1}{n} (\sum Y_i)^2}{\sum (Y_i^2) - \frac{1}{n} (\sum Y_i)^2}$$

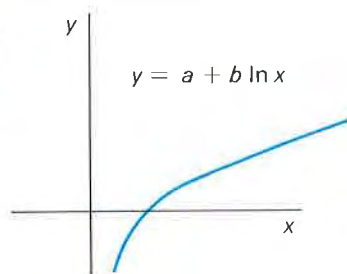
gemessen. Die Regression ist um so schlechter, je näher r^2 an Null und um so besser, je näher r^2 an eins liegt. Die Regressionskoeffizienten a und b definieren die erzeugte Kurve über die am Anfang dieses Abschnitts angegebenen Beziehungen.

Die gewünschte Regressionsart wird durch die Eingabe der Codenummer vor der eigentlichen Dateneingabe bestimmt.

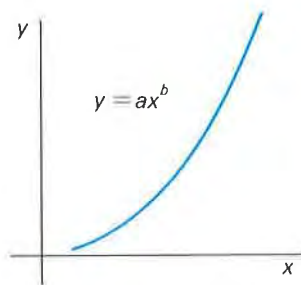
Exponentielle Regression
Code = 1



Logarithmische Regression
Code = 2



Potenzregression
Code = 3

**Bemerkungen:**

- Das Programm benutzt die Methode der kleinsten Quadrate zur Bestimmung der ursprünglichen Gleichung (logarithmische Kurve) bzw. der transformierten Gleichungen (Exponential- und Potenzkurve).
- Bei der logarithmischen Regression wird durch x_i -Werte kleiner oder gleich Null ein Internfehler verursacht. Das gleiche gilt bei der exponentiellen Regression für y_i -Werte kleiner oder gleich Null. Bei der Potenzregression dürfen sowohl die x_i - als auch die y_i -Werte nicht kleiner oder gleich Null sein.
- Bei sehr kleinen Differenzen zwischen den x - und/oder y -Werten kann ein Genauigkeitsverlust in den Regressionskoeffizienten auftreten.
- Während des Programmablaufs werden sämtliche Speicherregister gelöscht, d. h. alle zuvor gespeicherten Daten werden zerstört.

Tastensequenz	Anzeige	Tastensequenz	Anzeige
\boxed{f} CLEAR \boxed{PRGM}	000-	6	006- 6
\boxed{f} LBL \boxed{A}	001-42.21.11	$\boxed{+}$	007- 40
\boxed{f} CLEAR \boxed{REG}	002- 42 34	\boxed{STO} \boxed{I}	008- 44 25
\boxed{f} FIX 2	003-42. 7. 2	\boxed{R} $\boxed{\downarrow}$	009- 33
\boxed{ENTER}	004- 36	\boxed{g} \boxed{SF} 0	010-43. 4. 0
\boxed{ENTER}	005- 36	\boxed{g} \boxed{SF} 1	011-43. 4. 1

Tastensequenz	Anzeige	Tastensequenz	Anzeige
$\boxed{\text{GTO}} \boxed{1}$	012- 22 25	$\boxed{\text{g}} \boxed{\Sigma^-}$	036- 43 49
$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{7}$	013-42.21. 7	$\boxed{\text{GTO}} \boxed{9}$	037- 22 9
$\boxed{\text{g}} \boxed{\text{CF}} \boxed{1}$	014-43. 5. 1	$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{B}}$	038-42.21.12
$\boxed{\text{GTO}} \boxed{9}$	015- 22 9	$\boxed{\text{f}} \boxed{\text{L.R.}}$	039- 42 49
$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{8}$	016-42.21. 8	$\boxed{\text{g}} \boxed{\text{F?}} \boxed{0}$	040-43. 6. 0
$\boxed{\text{g}} \boxed{\text{CF}} \boxed{0}$	017-43. 5. 0	$\boxed{e^1}$	041- 12
$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{9}$	018-42.21. 9	$\boxed{\text{R/S}}$	042- 31
$\boxed{\text{R/S}}$	019- 31	$\boxed{x \approx y}$	043- 34
$\boxed{\text{g}} \boxed{\text{F?}} \boxed{0}$	020-43. 6. 0	$\boxed{\text{R/S}}$	044- 31
$\boxed{\text{g}} \boxed{\text{LN}}$	021- 43 12	$\boxed{\text{f}} \boxed{\text{I.r}}$	045- 42 48
$\boxed{x \approx y}$	022- 34	$\boxed{x \approx y}$	046- 34
$\boxed{\text{g}} \boxed{\text{F?}} \boxed{1}$	023-43. 6. 1	$\boxed{\text{g}} \boxed{x^2}$	047- 43 11
$\boxed{\text{g}} \boxed{\text{LN}}$	024- 43 12	$\boxed{\text{g}} \boxed{\text{RTN}}$	048- 43 32
$\boxed{\text{RCL}} \boxed{6}$	025- 45 6	$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{D}}$	049-42.21.14
$\boxed{\text{g}} \boxed{x \neq 0}$	026- 43 30	1	050- 1
$\boxed{\text{GTO}} \boxed{6}$	027- 22 6	$\boxed{\text{STO}} \boxed{6}$	051- 44 6
$\boxed{\text{R}} \boxed{\downarrow}$	028- 33	$\boxed{\text{GTO}} \boxed{9}$	052- 22 9
$\boxed{\Sigma+}$	029- 49	$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{\text{C}}$	053-42.21.13
$\boxed{\text{GTO}} \boxed{9}$	030- 22 9	$\boxed{\text{g}} \boxed{\text{F?}} \boxed{1}$	054-43. 6. 1
$\boxed{\text{f}} \boxed{\text{LBL}} \boxed{6}$	031-42.21. 6	$\boxed{\text{g}} \boxed{\text{LN}}$	055- 43 12
$\boxed{\text{R}} \boxed{\downarrow}$	032- 33	$\boxed{\text{f}} \boxed{\text{I.r}}$	056- 42 48
0	033- 0	$\boxed{\text{g}} \boxed{\text{F?}} \boxed{0}$	057-43. 6. 0
$\boxed{\text{STO}} \boxed{6}$	034- 44 6	$\boxed{e^1}$	058- 12
$\boxed{\text{R}} \boxed{\downarrow}$	035- 33		

Register			R_i : Code
$R_0: i$	$R_1: \Sigma x_i$	$R_2: \Sigma x_i^2$	$R_3: \Sigma y_i$
$R_4: \Sigma y_i^2$	$R_5: \Sigma x_i y_i$	$R_6: 1 \text{ oder } 0$	$R_7 - R_9$: unbelegt

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
1	Eingabe des Programms			
2	User Mode wählen			
3	Regressionsart wählen			
	Exponentiell	1	[A]	1,00
	Logarithmisch	2	[A]	2,00
	Potenz	3	[A]	3,00
4	Eingabe der x - und y -Werte	x_i	[ENTER]	
	(Schritt 4 ist für alle Datenpunkte zu wiederholen)	y_i	[R/S]	i
5	Berechnung der Regressionskoeffizienten und des Korrelationskoeffizienten		[B] [R/S] [R/S]	a b r^2
6	Schätzung eines y -Werts für einen bekannten x -Wert (Schritt 6 ist evtl für weitere x -Werte zu wiederholen Fehlerkorrektur.)	x	[C]	\hat{y}
7	Fehlerhafte Eingaben in Schritt 4 können durch Drücken von [D] und Neuingabe der fehlerhaften Daten korrigiert werden. Danach ist mit Schritt 4 und Eingabe der korrekten Daten fortzufahren.	fehlerh. x_i und y_i	[D] [ENTER] [R/S]	$i - 1$

Beispiel 1:

(Exponentielle Regression, Code = 1)

x_i	0,72	1,31	1,95	2,58	3,14
y_i	2,16	1,61	1,16	0,85	0,5

Lösung:

$$a = 3,45, b = -0,58$$

$$y = 3,45 e^{-0,58x}$$

$$r^2 = 0,98$$

Tastensequenz**Anzeige**

User Mode wählen

1	A		1,00	
,	72	ENTER	2,16	R/S
1,	31	ENTER	1,61	R/S
1,	95	ENTER	1,16	R/S
2,	58	ENTER	,85	R/S
3,	14	ENTER	,5	R/S
	B		3,45	a
	R/S		-0,58	b
	R/S		0,98	r^2
1.5	C		1,44	\hat{y}

Beispiel 2:

(Logarithmische Regression, Code = 2)

x_i	3	4	6	10	12
y_i	1,5	9,3	23,4	45,8	60,1

Lösung

$$a = -47,02, b = 41,39$$

$$y = 47,02 + 41,39 \ln x$$

$$r^2 = 0,98$$

Die Schätzung des Funktionswert y an der Stelle $x = 8$ ergibt $\hat{y} = 39,06$. Für $x = 14,5$ erhält man $\hat{y} = 63,67$ **Tastensequenz****Anzeige**

2	A		2,00
3	ENTER	1,5	R/S
4	ENTER	9,3	R/S
6	ENTER	23,4	R/S

Tastensequenz**Anzeige**

10	ENTER	45,8	R/S	
12	ENTER	60,1	R/S	
B				-47,02
R/S				41,39
R/S				0,98
8	C			39,06
14,5	C			63,67

a
b
*r*²
y
y

Beispiel 3:

(Regression mit Potenzkurven, Code = 3)

x_i	1,3	2,6	4	5,4	7,1
y_i	3,74	6,15	7,21	8,03	8,84

Lösung:

$$a = 3,49, \quad b = 0,50$$

$$y = 3,49 x^{0,50}$$

$$r^2 = 0,97$$

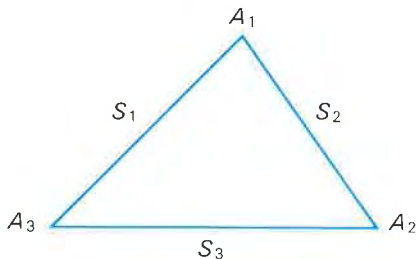
Tastensequenz**Anzeige**

3	A			3,00
1,3	ENTER	3,74	R/S	
2,6	ENTER	6,15	R/S	
4	ENTER	7,21	R/S	
5,4	ENTER	8,03	R/S	
7,1	ENTER	8,84	R/S	
B				3,49
R/S				0,50
R/S				0,97

a
b
*r*²

Dreiecksberechnungen

Mit dem folgenden Programm lassen sich die Seiten, die Winkel und die Fläche eines ebenen Dreiecks berechnen.



In der Regel ist die Spezifikation von drei der sechs Parameter eines Dreiecks (3 Seiten, 3 Winkel) zur Definition des Dreiecks ausreichend. (Die Ausnahme besteht bekanntermaßen in dem Fall, daß ein Dreieck durch drei Winkel nicht eindeutig festgelegt ist.) Das Programm kann die verbleibenden fünf möglichen Fälle behandeln: zwei Seiten und der eingeschlossene Winkel (SAS), zwei Winkel und die eingeschlossene Seite (ASA), zwei Seiten und ein benachbarter Winkel (SSA – ein zweideutiger Fall), zwei Winkel und eine benachbarte Seite (AAS) und drei Seiten (SSS).

Wenn die drei Vorgabegrößen im Uhrzeigersinn um das Dreieck geordnet eingegeben werden, erfolgt die Ausgabe der berechneten Größen ebenfalls im Uhrzeigersinn.

Bemerkungen:

- Die Winkeleingabe kann in jedem der drei trigonometrischen Modes ($\boxed{\text{DEG}}$, $\boxed{\text{RAD}}$, $\boxed{\text{GRAD}}$) erfolgen. Dabei ist sicherzustellen, daß der Rechner auf diejenige Maßeinheit geschaltet ist, in der die Winkel vorliegen.
- Beachten Sie, daß die vom Programm verwendete Dreiecksbeschreibung nicht mit der üblichen Notation übereinstimmt, d. h. A_1 liegt nicht gegenüber von S_1 .
- Die Winkel müssen dezimal eingegeben werden. Winkel, die in Grad, Minuten und Sekunden vorliegen, können mit $\boxed{\text{g}} \rightarrow \boxed{\text{H}}$ in Dezimalgrad konvertiert werden.
- Bei Dreiecken mit extrem kleinen Winkeln kann die berechnete Lösung von geringer Genauigkeit sein.

Tastenfolge	Anzeige	Tastenfolge	Anzeige
\boxed{f} CLEAR \boxed{PRGM}	000-	\boxed{RCL} 2	029- 45 2
\boxed{f} \boxed{LBL} \boxed{A}	001-42.21.11	\boxed{RCL} 6	030- 45 6
\boxed{STO} 5	002- 44 5	$\boxed{+}$	031- 40
$\boxed{R\downarrow}$	003- 33	\boxed{SIN}	032- 23
\boxed{STO} 3	004- 44 3	$\boxed{\div}$	033- 10
$\boxed{R\downarrow}$	005- 33	\boxed{RCL} 1	034- 45 1
\boxed{STO} 1	006- 44 1	$\boxed{\times}$	035- 20
\boxed{RCL} 3	007- 45 3	\boxed{STO} 3	036- 44 3
\boxed{g} $\boxed{\rightarrow P}$	008- 43 26	\boxed{GTO} 0	037- 22 0
\boxed{g} $\boxed{x^2}$	009- 43 11	\boxed{f} \boxed{LBL} \boxed{C}	038-42.21.13
\boxed{RCL} 5	010- 45 5	\boxed{STO} 4	039- 44 4
\boxed{g} $\boxed{x^2}$	011- 43 11	$\boxed{R\downarrow}$	040- 33
$\boxed{-}$	012- 30	\boxed{STO} 2	041- 44 2
\boxed{RCL} 1	013- 45 1	$\boxed{R\downarrow}$	042- 33
\boxed{RCL} 3	014- 45 3	\boxed{STO} 1	043- 44 1
$\boxed{\times}$	015- 20	\boxed{RCL} 4	044- 45 4
2	016- 2	\boxed{RCL} 2	045- 45 2
$\boxed{\times}$	017- 20	$\boxed{+}$	046- 40
$\boxed{\div}$	018- 10	\boxed{SIN}	047- 23
\boxed{g} $\boxed{\cos^{-1}}$	019- 43 24	\boxed{RCL} 4	048- 45 4
\boxed{STO} 2	020- 44 2	\boxed{SIN}	049- 23
\boxed{GTO} 0	021- 22 0	$\boxed{\div}$	050- 10
\boxed{f} \boxed{LBL} \boxed{B}	022-42.21.12	\boxed{RCL} 1	051- 45 1
\boxed{STO} 2	023- 44 2	$\boxed{\times}$	052- 20
$\boxed{R\downarrow}$	024- 33	\boxed{STO} 3	053- 44 3
\boxed{STO} 1	025- 44 1	\boxed{GTO} 0	054- 22 0
$\boxed{R\downarrow}$	026- 33	\boxed{f} \boxed{LBL} \boxed{E}	055-42.21.15
\boxed{STO} 6	027- 44 6	\boxed{STO} 4	056- 44 4
\boxed{SIN}	028- 23	$\boxed{R\downarrow}$	057- 33

Tastenfolge	Anzeige	Tastenfolge	Anzeige
[STO] 3	058- 44 3	f [LBL] [D]	087-42.21.14
[R↓]	059- 33	[STO] 3	088- 44 3
[STO] 1	060- 44 1	[R↓]	089- 33
[RCL] 3	061- 45 3	[STO] 2	090- 44 2
[RCL] 4	062- 45 4	[R↓]	091- 33
[SIN]	063- 23	[STO] 1	092- 44 1
[RCL] 1	064- 45 1	f [LBL] 0	093-42.21. 0
[÷]	065- 10	[RCL] 2	094- 45 2
[×]	066- 20	[RCL] 1	095- 45 1
[g] [SIN] ¹	067- 43 23	f [→R]	096- 42 26
[RCL] 4	068- 45 4	[RCL] 3	097- 45 3
[+]	069- 40	[y ≥ x]	098- 34
[GSB] 9	070- 32 9	[−]	099- 30
[STO] 2	071- 44 2	[g] [→P]	100- 43 26
[GSB] 0	072- 32 0	[STO] 5	101- 44 5
[RCL] 1	073- 45 1	[y ≥ x]	102- 34
[RCL] 3	074- 45 3	[STO] 4	103- 44 4
f [y ≤ x]	075- 42 10	[RCL] 2	104- 45 2
[GTO] 8	076- 22 8	[+]	105- 40
2	077- 2	[GSB] 9	106- 32 9
[R/S]	078- 31	[STO] 6	107- 44 6
[RCL] 6	079- 45 6	[SIN]	108- 23
[GSB] 9	080- 32 9	[×]	109- 20
[STO] 6	081- 44 6	[RCL] 1	110- 45 1
[RCL] 4	082- 45 4	[×]	111- 20
[+]	083- 40	2	112- 2
[GSB] 9	084- 32 9	[÷]	113- 10
[STO] 2	085- 44 2	[STO] 0	114- 44 0
[GTO] 0	086- 22 0	[GTO] 1	115- 22 1

Tastenfolge	Anzeige	Tastenfolge	Anzeige
\boxed{g} \boxed{RTN}	116- 43 32	6	126- 6
\boxed{f} \boxed{LBL} 9	117-42.21. 9	\boxed{STO} \boxed{I}	127- 44 25
\boxed{COS}	118- 24	\boxed{f} \boxed{LBL} 2	128-42.21. 2
\boxed{CHS}	119- 16	\boxed{RCL} $\boxed{(i)}$	129- 45 24
\boxed{g} \boxed{COS}°	120- 43 24	$\boxed{R/S}$	130- 31
\boxed{g} \boxed{RTN}	121- 43 32	\boxed{f} \boxed{ISG}	131- 42 6
\boxed{f} \boxed{LBL} 1	122-42.21. 1	\boxed{GTO} 2	132- 22 2
$\boxed{\cdot}$	123- 48	\boxed{f} \boxed{LBL} 8	133-42.21. 8
0	124- 0	\boxed{g} \boxed{CLR}	134- 43 35
0	125- 0		

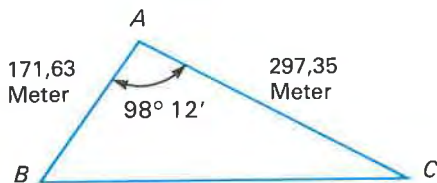
Register			R_1 : Index
R_0 : Fläche	R_1 : S_1	R_2 : A_1	R_3 : S_2
R_4 : A_2	R_5 : S_3	R_6 : A_3	R_7 - R_8 : unbelegt

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
1	Eingabe des Programms			
2	User Mode wählen			
3	Gew. trig. Mode schalten			
	(DEG, RAD, GRAD)			

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
4	SSS (3 Seiten)			
	Eingabe:			
	Seite 1	S_1	<input type="button" value="ENTER"/>	
	Seite 2	S_2	<input type="button" value="ENTER"/>	
	Seite 3	S_3	<input type="button" value="A"/>	Fläche
	Forts. mit Schritt 9			
5	ASA (2 Winkel u. eing. Seite)			
	Eingabe:			
	Winkel 3	A_3	<input type="button" value="ENTER"/>	
	Seite 1	S_1	<input type="button" value="ENTER"/>	
	Winkel 1	A_1	<input type="button" value="B"/>	Fläche
	Forts. mit Schritt 9			
6	SAA (2 Winkel u. ben. Seite)			
	Eingabe:			
	Seite 1	S_1	<input type="button" value="ENTER"/>	
	Winkel 1	A_1	<input type="button" value="ENTER"/>	
	Winkel 2	A_2	<input type="button" value="C"/>	Fläche
	Forts. mit Schritt 9			
7	SAS (2 Seiten u. eing. Winkel)			
	Eingabe:			
	Seite 1	S_1	<input type="button" value="ENTER"/>	
	Winkel 1	A_1	<input type="button" value="ENTER"/>	
	Seite 2	S_2	<input type="button" value="D"/>	Fläche
	Forts. mit Schritt 9			
8	SSA (2 Seiten u. ben. Winkel)			
	Eingabe:			
	Seite 1	S_1	<input type="button" value="ENTER"/>	
	Seite 2	S_2	<input type="button" value="ENTER"/>	

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
	Winkel 2	A_2	\boxed{E}	Fläche
9	Berechnung der verbleibenden Seiten und Winkel		$\boxed{R/S}$	S_1
			$\boxed{R/S}$	A_1
			$\boxed{R/S}$	S_2
			$\boxed{R/S}$	A_2
			$\boxed{R/S}$	S_3
			$\boxed{R/S}$	A_3
9a	Wenn Lösung eindeutig:		$\boxed{R/S}$	0
9b	Wenn zweite Lösung existiert:		$\boxed{R/S}$	2
			$\boxed{R/S}$	Fläche
			$\boxed{R/S}$	S_1
			$\boxed{R/S}$	A_1
			$\boxed{R/S}$	S_2
			$\boxed{R/S}$	A_2
			$\boxed{R/S}$	S_3
			$\boxed{R/S}$	A_3

Beispiel 1: Ein Landvermesser hat eine dreieckige Landparzelle zu vermessen. Vom Punkt A werden die Entfernungen zu den Punkten B und C elektronisch gemessen. Der Winkel zwischen \overline{AB} und \overline{AC} wird gleichfalls gemessen. Gesucht sind die Fläche und die verbleibenden Seiten und Winkel des Dreiecks.



Dies entspricht einem Seite-Winkel-Seite Problem mit

$$S_1 = 171,63, \quad A_1 = 98^\circ 12', \quad S_2 = 297,35$$

Tastensequenz

g | DEG

f | FIX | 2

User Mode wählen

171,63 | ENTER

98,12 | g | → | H

297,35 | D

R/S

R/S

R/S

R/S

R/S

R/S

Anzeige

171,63

98,20

25.256,21

171,63

98,20

297,35

27,83

363,91

53,97

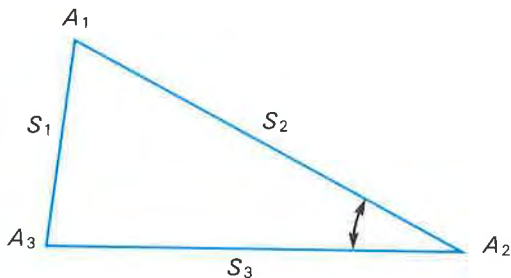
Fläche

 S_1 A_1 S_2 A_2 S_3 A_3

Beispiel 2: Gegeben sind zwei Seiten und ein benachbarter Winkel eines Dreiecks.

$$\text{Seite 1} = 25,6 \quad \text{Seite 2} = 32,8 \quad \text{Winkel 2} = 42,3$$

Berechnen Sie die fehlenden Größen.



Tastenfolge

25,6	ENTER
32,8	ENTER
42,3	E
R/S	
R/S	
R/S	
R/S	
R/S	
R/S	
R/S	
R/S	
R/S	
R/S	
R/S	
R/S	
R/S	
R/S	

Anzeige

25,60	
32,80	
410,85	Fläche
25,60	S ₁
78,12	A ₁
32,80	S ₂
42,30	A ₂
37,22	S ₂
59,58	A ₃
2,00	2. Lösung
124,68	Fläche
25,60	S ₁
17,28	A ₁
32,80	S ₂
42,30	A ₂
11,30	S ₃
120,42	A ₃

T-Test**T-Test bei Stichproben mit gleichem Umfang**

Gegeben seien paarweise Beobachtungen aus zwei normalverteilten Grundgesamtheiten mit den unbekanntem Mittelwerten μ_1 und μ_2 .

x_i	x_1	$x_2 \dots x_n$
y_i	y_1	$y_2 \dots y_n$

Sei nun

$$D_i = x_i - y_i$$

$$\bar{D} = \frac{1}{n} \sum_{i=1}^n D_i$$

$$s_D = \sqrt{\frac{\sum D_i^2 - \frac{1}{n} (\sum D_i)^2}{n - 1}}$$

Die Testgröße

$$t = \frac{\bar{D}}{s_D} \sqrt{n}$$

hat $n - 1$ Freiheitsgrade (df) und kann zum Testen der Nullhypothese

$$H_0: \mu_1 = \mu_2$$

verwendet werden.

T-Test bei Stichproben mit unterschiedlichem Umfang

Die Mengen $\{x_1, x_2, \dots, x_{n_1}\}$ und $\{y_1, y_2, \dots, y_{n_2}\}$ seien unabhängige, zufällige Stichproben aus zwei normalverteilten Grundgesamtheiten mit den unbekanntem Mittelwerten μ_1 und μ_2 und derselben (unbekanntem) Varianz σ^2 .

Zu testen ist die Nullhypothese:

$$H_0: \mu_1 - \mu_2 = d$$

Die Testgröße t wird erklärt über die Gleichungen

$$\bar{x} = \frac{1}{n_1} \sum_{i=1}^{n_1} x_i$$

$$\bar{y} = \frac{1}{n_2} \sum_{i=1}^{n_2} y_i$$

$$t = \frac{\bar{x} - \bar{y} - d}{\sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \sqrt{\frac{\sum x_i^2 - n_1 \bar{x}^2 + \sum y_i^2 - n_2 \bar{y}^2}{n_1 + n_2 - 2}}}$$

Diese Testgröße, die eine T-Verteilung mit $n_1 + n_2 - 2$ Freiheitsgraden (df) besitzt, kann zum Testen der Nullhypothese H_0 verwendet werden.

Literatur:

1. Statistics in Research, B. Ostle, Iowa State University Press, 1963
2. Statistical Theory and Methodology in Science and Engineering, K. A. Brownlee, John Wiley and Sons, 1965.

Tastensequenz	Anzeige	Tastensequenz	Anzeige
\boxed{f} CLEAR \boxed{PRGM}	000-	\boxed{STO} $\boxed{\cdot}$ 2	029- 44 ,2
\boxed{f} LBL \boxed{A}	001-42.21.11	\boxed{RCL} 0	030- 45 0
$\boxed{-}$	002- 30	\boxed{STO} $\boxed{\cdot}$ 0	031- 44 ,0
$\boxed{\Sigma+}$	003- 49	0	032- 0
$\boxed{R/S}$	004- 31	\boxed{f} CLEAR $\boxed{\Sigma}$	033- 42 32
\boxed{f} LBL \boxed{C}	005-42.21.13	$\boxed{R/S}$	034- 31
\boxed{g} $\boxed{\bar{x}}$	006- 43 0	\boxed{f} LBL \boxed{D}	035-42.21.14
\boxed{STO} 6	007- 44 6	\boxed{STO} 6	036- 44 6
$\boxed{R/S}$	008- 31	\boxed{RCL} $\boxed{\cdot}$ 1	037- 45 ,1
\boxed{g} \boxed{s}	009- 43 48	\boxed{RCL} $\boxed{\cdot}$ 2	038- 45 ,2
$\boxed{R/S}$	010- 31	\boxed{STO} 4	039- 44 4
\boxed{RCL} 6	011- 45 6	$\boxed{x \geq y}$	040- 34
\boxed{RCL} 0	012- 45 0	\boxed{STO} 3	041- 44 3
$\boxed{\sqrt{x}}$	013- 11	\boxed{g} $\boxed{\bar{x}}$	042- 43 0
\boxed{x}	014- 20	\boxed{STO} 8	043- 44 8
$\boxed{x \geq y}$	015- 34	$\boxed{x \geq y}$	044- 34
$\boxed{\div}$	016- 10	\boxed{RCL} 0	045- 45 0
$\boxed{R/S}$	017- 31	\boxed{x}	046- 20
\boxed{RCL} 0	018- 45 0	\boxed{RCL} $\boxed{\cdot}$ 0	047- 45 ,0
1	019- 1	$\boxed{\div}$	048- 10
$\boxed{-}$	020- 30	\boxed{STO} 7	049- 44 7
$\boxed{R/S}$	021- 31	$\boxed{x \geq y}$	050- 34
\boxed{GTO} \boxed{C}	022- 22 13	$\boxed{-}$	051- 30
\boxed{f} LBL \boxed{B}	023-42.21.12	\boxed{RCL} 6	052- 45 6
$\boxed{\Sigma+}$	024- 49	$\boxed{-}$	053- 30
$\boxed{R/S}$	025- 31	\boxed{RCL} 4	054- 45 4
\boxed{RCL} 1	026- 45 1	\boxed{RCL} 3	055- 45 3
\boxed{STO} $\boxed{\cdot}$ 1	027- 44 ,1	\boxed{RCL} 7	056- 45 7
\boxed{RCL} 2	028- 45 2	\boxed{x}	057- 20

Tastenfolge	Anzeige	Tastenfolge	Anzeige
\square	058- 30	\sqrt{x}	072- 11
\square 2	059- 45 2	\square 0	073- 45 0
\square	060- 40	$1/x$	074- 15
\square 1	061- 45 1	\square 0	075- 45 ,0
\square 8	062- 45 8	$1/x$	076- 15
\square	063- 20	\square	077- 40
\square	064- 30	\sqrt{x}	078- 11
\square 0	065- 45 0	\square	079- 20
\square 0	066- 45 ,0	\square	080- 10
\square	067- 40	R/S	081- 31
2	068- 2	\square 9	082- 45 9
\square	069- 30	R/S	083- 31
\square 9	070- 44 9	\square 6	084- 45 6
\square	071- 10	GTO \square	085- 22 14

Register			R ₁ : unbelegt
R ₀ : $n \mid n_1, n_2$	R ₁ : $\Sigma D \mid \Sigma x, \Sigma y$	R ₂ : $\Sigma D^2 \mid \Sigma x^2, \Sigma y^2$	R ₃ : $df \mid \Sigma x$
R ₄ : $\mid \Sigma x^2$	R ₅ : belegt	R ₆ : $\bar{D} \mid d$	R ₇ : $\mid \bar{x}$
R ₈ : $\mid \bar{y}$	R ₉ : $\mid df$	R ₀ : $\mid n_1$	R ₁ : $\mid \Sigma x$
R ₂ : $\mid \Sigma x^2$	R ₃ —R ₆ : unbelegt		

Die Werte vor den senkrechten Strichen beziehen sich auf den T-Test bei Stichproben mit gleichem Umfang; die Werte hinter den Strichen auf den T-Test bei Stichproben mit unterschiedlichem Umfang.

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
	T-Test bei gleich großen Stichproben			
1	Eingabe des Programms			
2	Initialisieren des Programms		\square CLEAR \square REG	
3	User Mode wählen			
	Schritt 4–5 sind für $i=1, \dots, n$ zu wiederholen.			

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
4	Eingabe			
	x_i	x_i	ENTER	
	y_i	y_i	A	i
5	(Optional) Korrektur			
	fehlerhafter x_k	x_k	ENTER	
	oder y_k	y_k	- 0 Σ-	$k-1$
6	Berechnung der Testgrößen:			
	\bar{D}		C	\bar{D}
	s_D		R/S	s_D
	t		R/S	t
	df		R/S	df
7	Schritt 6 zur Anzeige der Resultate wiederholen			
	T-Test für unterschiedlich große Stichproben			
1	Eingabe des Programms			
2	Initialisieren des Programms		f CLEAR REG	
3	User Mode wählen			
	Schritt 3–4 sind für $i = 1, \dots, n_1$ zu wiederholen			
4	Eingabe x_i	x_i	B	i
5	Korrektur fehlerhafter Eingaben x_k	x_k	0 Σ-	$k-1$
6	Initialisieren für den			
	2. Datensatz		R/S	
	Schritt 7–8 sind für $i = 1, \dots, n_2$ zu wiederholen			
7	Eingabe y_i	y_i	B	i
8	Korrektur fehlerhafter Ein- gaben y_k	y_k	0 Σ-	$k-1$
9	Eingabe von d zur Berechnung d. Testgrößen			
	t	d	D	t
	df		R/S	df
10	Schritt 9 zur Anzeige der Resultate wiederholen			

Beispiel 1:

x_i	14	17,5	17	17,5	15,4
y_i	17	20,7	21,6	20,9	17,2

Gesucht sind \bar{D} , s_D , t und df für den obigen Datensatz.

Tastensequenz

Anzeige

f FIX 4		
User Mode wählen		
f CLEAR REG		
14 ENTER 17 A	1,0000	
17 ENTER 15 A	2,0000	(Fehler)
17 ENTER		
15 - g Σ-	1,0000	(Korrektur)
17,5 ENTER 20,7 A	2,0000	
17 ENTER 21,6 A	3,0000	
17,5 ENTER 20,9 A	4,0000	
15,4 ENTER 17,2 A	5,0000	
C	-3,2000	(\bar{D})
R/S	1,0000	(s_D)
R/S	-7,1554	(t)
R/S	4,0000	(df)

Beispiel 2:

x	79	84	108	114	120	103	122	120		
y	91	103	90	113	108	87	100	80	99	54

Mit den obigen Daten sind t und df für $d = 0$ (d. h. $H_0: \mu_1 = \mu_2$) zu berechnen.

Tastensequenz

Anzeige

f CLEAR REG	
79 B 84 B 99	
B 99 g Σ-	2,0000
108 B 114 B 120 B	
103 B 122 B 120 B	8,0000
R/S	0,0000

Tastenfolge

91 [B] 103 [B]
 90 [B] 113 [B]
 108 [B] 87 [B]
 100 [B] 80 [B]
 99 [B] 54 [B]
 0 [D]
 [R/S]

Anzeige

10,0000
1,7316 (t)
16,0000 (df)

Chiquadrat-Test

Dieses Programm berechnet den χ^2 -Wert für die Güte des Zusammenhangs zwischen beobachteten und erwarteten Häufigkeiten. Es gilt

$$\chi_1^2 = \sum_{i=1}^n \frac{(O_i - E_i)^2}{E_i} \quad (df = n - 1)$$

wobei O_i = beobachtete Häufigkeit
 E_i = erwartete Häufigkeit

Wenn die erwarteten Werte gleich sind, d. h.

$$E = E_i = \frac{\sum O_i}{n} \text{ für alle } i$$

gilt statt der obigen Formel

$$\chi_2^2 = \frac{n \sum O_i^2}{\sum O_i} - \sum O_i$$

Bemerkung: Bei der Anwendung des χ^2 -Tests auf eine gegebene Datenmenge kann es notwendig sein, die Daten in mehreren Klassen zusammenzufassen, damit die erwarteten Häufigkeiten nicht zu klein (in der Regel nicht kleiner als 5) sind.

Literatur:

Mathematical Statistics, J. E. Freund, Prentice Hall, 1962.

Tastenfolge	Anzeige	Tastenfolge	Anzeige
f CLEAR PRGM	000-	f LBL B	022-42.21.12
f LBL A	001-42.21.11	$\Sigma+$	023- 49
ENTER	002- 36	R/S	024- 31
R↓	003- 33	f LBL D	025-42.21.14
-	004- 30	RCL 0	026- 45 0
g x^2	005- 43 11	RCL 0	027- 45 0
g R↑	006- 43 33	RCL 2	028- 45 2
+	007- 10	x	029- 20
g F? 0	008-43. 6. 0	RCL 1	030- 45 1
CHS	009- 16	÷	031- 10
STO + 1	010-44.40. 1	RCL 1	032- 45 1
1	011- 1	-	033- 30
g F? 0	012-43. 6. 0	R/S	034- 31
CHS	013- 16	RCL 1	035- 45 1
STO + 0	014-44.40. 0	RCL 0	036- 45 0
RCL 0	015- 45 0	±	037- 10
g CF 0	016-43. 5. 0	R/S	038- 31
R/S	017- 31	GTO D	039- 22 14
f LBL C	018-42.21.13	f LBL E	040-42.21.15
RCL 1	019- 45 1	g SF 0	041-43. 4. 0
R/S	020- 31	GTO A	042- 22 11
GTO C	021- 22 13		

Register			R ₁ : unbelegt
R ₀ : Index	R ₁ : $\gamma_2^2, \Sigma O_i$	R ₂ : ΣO_i^2	R ₃ : belegt
R ₄ : belegt	R ₅ : belegt	R ₆ -R ₉ : unbelegt	

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
	Ungleiche erwartete Häufigkeiten			
1	Eingabe des Programms			
2	Initialisieren des Programms		\boxed{f} CLEAR \boxed{REG}	
			\boxed{g} \boxed{CF} 0	
3	User Mode wählen			
	Schritt 4–5 ist für $i = 1, \dots, n$ zu wiederholen			
4	Eingabe			
	O_i	O_i	\boxed{ENTER}	
	E_i	E_i	\boxed{A}	i
5	Fehlerhafte Eingabe	O_k	\boxed{ENTER}	
	O_k oder E_k korrigieren	E_k	\boxed{E}	$k-1$
6	χ_1^2 berechnen		\boxed{C}	χ_1^2
7	Schritt 6 zur Anzeige des Resultats wiederholen			
	Gleiche erwartete Häufigkeiten			
1	Programm initialisieren		\boxed{f} CLEAR \boxed{REG}	
2	User Mode wählen			
	Schritt 3–4 für $i = 1, \dots, n$ wiederholen			
3	Eingabe O_i	O_i	\boxed{B}	i
4	Fehlerhafte Eingabe O_n korrigieren	O_h	$\boxed{0}$ $\boxed{\Sigma-}$	$h-1$
5	Berechnung von			
	χ_2^2 und		\boxed{D}	χ_2^2
	E		$\boxed{R/S}$	E
6	Schritt 5 zur Anzeige der Resultate wiederholen			

Beispiel 1: Führen Sie den χ^2 -Test mit den folgenden Daten durch:

O_i	8	50	47	56	5	14
E_i	9,6	46,75	51,85	54,4	8,25	9,15

Tastensequenz

\boxed{f} CLEAR \boxed{REG}
 \boxed{f} \boxed{FIX} 4

Anzeige

Tastensequenz**Anzeige**

User Mode wählen

[g] [CF] 0

8 [ENTER] 9,6 [A] **1,000**50 [ENTER] 46,75 [A] **2,000**47 [ENTER] 51,85 [A] **3,000**56 [ENTER] 54,4 [A] **4,000**100 [ENTER] [A] **5,000** (Fehler)100 [ENTER] [E] **4,000** Korrektur5 [ENTER] 8,25 [A] **5,000**14 [ENTER] 9,15 [A] **6,000**[C] **4,8444** (χ_1^2)

Beispiel 2: Die folgende Tabelle zeigt die beim 120maligen Werfen eines Würfels beobachteten Häufigkeiten. χ^2 kann zum Testen der Güte des Würfels (man testet, ob der Würfel „fair“ ist) benutzt werden. df ist in diesem Fall 5.

Bemerkung: Es wird unterstellt, daß die erwarteten Häufigkeiten gleich sind.

Zahl	1	2	3	4	5	6
Häufigkeit O_i	25	17	15	23	24	16

Tastensequenz**Anzeige**

[f] CLEAR [REG]

25 [B] 17 [B] 19 [B] **3,000** (Fehler)19 [g] [Σ -] **2,000** (Korrektur)

15 [B] 23 [B]

24 [B] 16 [B] **6,000**[D] **5,000** (χ_2^2)[R/S] **20,000** (E)

Der χ^2 -Wert für $df = 5$ und 5 % Signifikanz beträgt 11,070 (siehe J. E. Freund, Mathematical Statistics, Seite 438). Da 5,00 kleiner als 11,070 ist, bestehen in unserem Fall keine Unterschiede zwischen beobachteter und erwarteter Häufigkeit.

Finanzen: Jährliche Fälligkeiten und aufgelaufene Zinsen

Mit diesem Programm kann eine Vielfalt von Problemstellungen, die die Variablen Geld und Zeit beinhalten, behandelt werden. Die folgenden Parameter (mit Ausnahme des Zinssatzes, der immer

vorgegeben werden muß) können sowohl in der Eingabe als auch in der Ausgabe auftreten:

n: Laufzeit in Perioden (für ein Darlehen mit 30jähriger Laufzeit und monatlichen Raten gilt: $n = 12 \times 30 = 360$).

i: Zinssatz pro Periode in Prozent. (Bei nichtjährlicher Verzinsung ist der Jahreszinssatz durch die Anzahl der Abrechnungsperioden pro Jahr zu dividieren, d. h. ein Jahreszins von 10 % ergibt ein Monatszins von 10/12 % oder 0,83 %.)

PV: Gegenwärtiger Wert eines Darlehens oder Hypothek.

PMT: Zahlungsbetrag pro Periode.

FV: Zukünftiger Wert; Restrückzahlungsbetrag oder aufgelaufener Wert einer Reihe von Einzahlungen.

Das Programm bietet des weiteren die Möglichkeit zur Berechnung von Zinseszinsen und Restrückzahlungsbeträgen.

Das Programm läßt Zahlungen sowohl am Anfang als auch am Ende des Abrechnungszeitraums zu. Zahlungen am Ende des Abrechnungszeitraums (gewöhnliche Fälligkeiten) treten bei Darlehen und Hypotheken auf; Zahlungen am Anfang des Abrechnungszeitraums (Vorausfälligkeiten) kommen häufig bei Leasing-Geschäften vor. Drücken Sie bei gewöhnlicher Fälligkeit **A** und dann **R/S** bis eine 1 angezeigt wird. Bei Vorausfälligkeiten ist **A** und dann **R/S** zu drücken, bis eine 0 angezeigt wird.

Im Programm wird die Konvention benutzt, daß Auszahlungen negativ und Einzahlungen positiv einzugeben sind.

Das Drücken der Tastenfolge **f** CLEAR **REG** stellt eine sichere, bequeme und leicht zu merkende Methode zur Vorbereitung des Rechners auf ein neues Problem dar. Es ist jedoch nicht notwendig, **f** CLEAR **REG** bei Problemen mit der gleichen Variablenkombination zu benutzen. Beispielsweise kann eine beliebige Anzahl von *n*, *i*, *PV*, *PMT*, *FV* Problemen mit verschiedenen Daten und unterschiedlichen Vorgabekombinationen hintereinander ohne Löschen der Register gerechnet werden. Dabei müssen nur diejenigen Werte, die sich von Problem zu Problem ändern, neu eingegeben werden. Zur Änderung von Vorgabekombinationen ohne Verwendung von **f** CLEAR **REG** brauchen Sie nur für die nicht länger verfügbaren Variablen Null einzugeben. Um beispielsweise von einem *n*, *i*, *PMT*, *PV* Problem zu einem *n*, *i*, *PV*, *FV* Problem zu gelangen, muß an Stelle von *PMT* nur Null gespeichert werden. Die folgende Tabelle enthält eine Zusammenstellung der zulässigen Variablenkombinationen.

**Mögliche Lösungen bei der
Berechnung von Fälligkeiten und Zinseszinsen
(bei vorgegebenem Zins)**

Zulässige Variablenkombinationen	Anwendungen		Initialisierungsprozedur
	Gewöhnliche Fälligkeit	Vorausfälligkeit	
n, PV, PMT (Vorgabe von 2 Variablen, Berechnung der dritten).	Darlehen mit direkter Rückzahlung Diskontierter Schuldverschreibungen, Hypotheken	Leasing-Verträge	<input type="checkbox"/> f CLEAR <input type="checkbox"/> REG benutzen oder <i>FV</i> auf Null setzen.
n, PV, PMT, FV (Vorgabe von 3 Variablen, Berechnungen der vierten).	Darlehen mit kumulierter Rückzahlung Schuldverschreibung mit kumulierter Rückzahlung	Leasing-Verträge mit Restwerten	Keine
n, PMT, FV (Vorgabe von 2 Variablen, Berechnung der dritten).	Rentenpapiere	Ratensparverträge	<input type="checkbox"/> f CLEAR <input type="checkbox"/> REG benutzen oder <i>PV</i> auf Null setzen.
n, PV, FV (Vorgabe von 2 Variablen, Berechnung der dritten).	Sparguthaben (Fälligkeit ist nicht anwendbar und bleibt ohne Wirkung).		<input type="checkbox"/> f CLEAR <input type="checkbox"/> REG benutzen oder <i>PMT</i> auf Null setzen.

Allen Berechnungen liegt die folgende Gleichung zugrunde:

$$-PV = \frac{PMT}{i} A [1 - (1+i)^{-n}] + FV(1+i)^{-n}$$

wobei

$$A = \begin{cases} 1 & \text{bei gewöhnlicher Fälligkeit} \\ (1+i) & \text{bei Vorausfälligkeit} \end{cases}$$

Bemerkungen:

1. Die Angabe einer Zinsrate 0 induziert eine „Error 0“-Meldung.
2. Die Angabe von extrem hohen (10^6) oder extrem niedrigen (10^{-6}) Werten für n oder i kann unsinnige Ergebnisse verursachen.

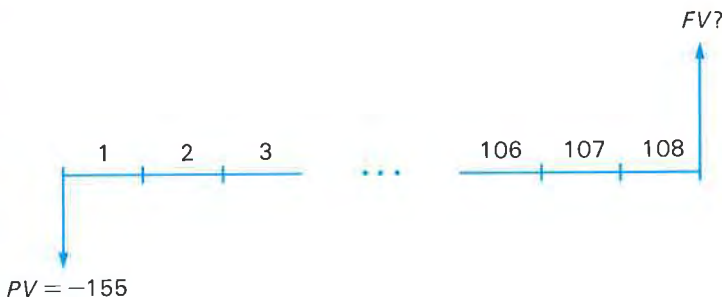
Tastensequenz	Anzeige	Tastensequenz	Anzeige
\boxed{f} CLEAR \boxed{PRGM}	000-	\boxed{g} LST \boxed{x}	018- 43 36
\boxed{f} LBL \boxed{A}	001-42.21.11	$\boxed{-}$	019- 30
\boxed{f} FIX 2	002-42. 7. 2	\boxed{RCL} 3	020- 45 3
\boxed{STO} 2	003- 44 2	\boxed{g} LST \boxed{x}	021- 43 36
\boxed{f} LBL 4	004-42.21. 4	$\boxed{+}$	022- 40
$\boxed{R/S}$	005- 31	$\boxed{\div}$	023- 10
0	006- 0	\boxed{CHS}	024- 16
\boxed{g} $\boxed{F?}$ 0	007-43. 6. 0	\boxed{g} LN	025- 43 12
1	008- 1	\boxed{RCL} 6	026- 45 6
\boxed{g} CF 0	009-43. 5. 0	\boxed{g} LN	027- 43 12
\boxed{g} $\boxed{x=0}$	010- 43 40	$\boxed{\div}$	028- 10
\boxed{g} SF 0	011-43. 4. 0	\boxed{STO} 1	029- 44 1
\boxed{GTO} 4	012- 22 4	\boxed{g} RTN	030- 43 32
\boxed{f} LBL \boxed{B}	013-42.21.12	\boxed{f} LBL \boxed{C}	031-42.21.13
\boxed{STO} 1	014- 44 1	\boxed{STO} 3	032- 44 3
$\boxed{R/S}$	015- 31	$\boxed{R/S}$	033- 31
\boxed{GSB} 1	016- 32 1	\boxed{GSB} 1	034- 32 1
\boxed{RCL} 5	017- 45 5	\boxed{GSB} 2	035- 32 2

Tastensequenz	Anzeige	Tastensequenz	Anzeige
[CHS]	036- 16	[RCL] 2	065- 45 2
[STO] 3	037- 44 3	[g] [%]	066- 43 14
[g] [RTN]	038- 43 32	[STO] 8	067- 44 8
[f] [LBL] [D]	039-42.21.14	1	068- 1
[STO] 4	040- 44 4	[STO] 0	069- 44 0
[R/S]	041- 31	[+]	070- 40
1	042- 1	[STO] 6	071- 44 6
[STO] 4	043- 44 4	[g] [F?] 0	072-43. 6. 0
[GSB] 1	044- 32 1	[STO] 0	073- 44 0
[1/x]	045- 15	[RCL] 1	074- 45 1
[RCL] 3	046- 45 3	[CHS]	075- 16
[GSB] 2	047- 32 2	[r ⁿ]	076- 14
[x]	048- 20	[STO] 7	077- 44 7
[CHS]	049- 16	1	078- 1
[STO] 4	050- 44 4	[\sqrt{x}]	079- 34
[g] [RTN]	051- 43 32	-	080- 30
[f] [LBL] [E]	052-42.21.15	[RCL] 4	081- 45 4
[STO] 5	053- 44 5	[RCL] 8	082- 45 8
[R/S]	054- 31	[+]	083- 10
[GSB] 1	055- 32 1	[RCL] 0	084- 45 0
[RCL] 3	056- 45 3	[x]	085- 20
[+]	057- 40	[x]	086- 20
[RCL] 7	058- 45 7	[g] [RTN]	087- 43 32
[÷]	059- 10	[f] [LBL] 2	088-42.21. 2
[CHS]	060- 16	[RCL] 5	089- 45 5
[STO] 5	061- 44 5	[RCL] 7	090- 45 7
[g] [RTN]	062- 43 32	[x]	091- 20
[f] [LBL] 1	063-42.21. 1	[+]	092- 40
1	064- 1		

Register			R ₁ : unbelegt
R ₀ : 1 oder 1 + i	R ₁ : n	R ₂ : i(%)	R ₃ : PV
R ₄ : PMT	R ₅ : FV	R ₆ : 1 + i	R ₇ : (1 - i) ⁻ⁿ
R ₈ : i/100	R ₉ —R ₄ : unbelegt		

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
1	Eingabe des Programms			
2	User Mode wählen			
3	Initialisieren		f CLEAR REG	
4	Eingabe des Zinssatzes pro Periode	i(%)	A	i(%)
5	Fälligkeitsmode wählen		R/S	1,00
	(Wiederholung dieses		R/S	0,00
	Schritts bewirkt ein Springen zwischen beiden Modi)			
6	Eingabe Vorgabegrößen			
	• Anzahl der Perioden	n	B	n
	• gegenwärtiger Wert	PV	C	PV
	• Zahlungsrate	PMT	D	PMT
	• zukünftiger Wert	FV	E	FV
7	Berechnung der unbekanntenen Werte:			
	• Anzahl der Perioden		B R/S	n
	• gegenwärtiger Wert		C R/S	PV
	• Zahlungsrate		D R/S	PMT
	• zukünftiger Wert		E R/S	FV
8	Zur Modifikation d. Problems			
	Forts. m. Schritt 4 u. Änderung d. entspr. Werte.			
	Eingabe von Null für nicht weiter gebrauchte Werte.			
9	Bei neuem Problem Fortsetzung mit Schritt 3.			

Beispiel 1: Sie zahlen 155 DM auf ein Sparkonto mit einer monatlichen Verzinsung von $5\frac{3}{4}\%$ ein. Welchen Betrag können Sie nach neun Jahren abheben?



Tastenfolge

f CLEAR **REG**

User Mode wählen

5,75 **ENTER** 12 **÷** **A**

R/S

9 **ENTER** 12 **×** **B**

155 **CHS** **C**

E **R/S**

Anzeige

0,48

0,00

108,00

-155,00

259,74

% monatlicher Zinssatz

Vorausfälligkeit

Anzahl der Monate

Einzahlungsbetrag

FV

Beispiel 2: Sie nehmen bei der Bank eine Hypothek in Höhe von 30000 DM mit einer Laufzeit von 30 Jahren zu einem Zinssatz von 13% auf. In welchen Monatsraten müssen Sie die Hypothek bei vollständiger Tilgung zurückzahlen?

$PV = 30,000$



(MONATSRATEN)

Tastensequenz

f CLEAR REG
 13 ENTER 12 ÷ A
 R/S
 30 ENTER 12 × B
 30000 C
 D R/S

Anzeige

1,08	% monatl. Zinssatz
1,00	gewöhnliche Fälligkeit
360,00	<i>n</i>
30.000,00	<i>PV</i>
-331,86	<i>PMT</i>

Beispiel 3: Zwei Geschäftsleute schließen einen Darlehensvertrag mit kumulativer Rückzahlung. Das Darlehen hat eine Höhe von 3.600 Dollar, und es wird eine jährliche Verzinsung von 10 % sowie die Rückzahlung in 36 Monatsraten zu je 100 Dollar vereinbart. Welcher Zinsbetrag ist zusammen mit der 36. Monatsrate zur Erfüllung des Darlehensvertrags fällig?

(Das folgende Diagramm bezieht sich auf den Darlehensgeber. Das Diagramm für den Darlehensnehmer wäre genau umgekehrt.)

**Tastensequenz**

f CLEAR REG
 10 ENTER 12 ÷ A
 R/S
 36 B
 3600 CHS C
 100 D E R/S

Anzeige

0,83	(% monatl. Zinssatz)
1,00	gewöhnliche Fälligkeit
36,00	(<i>n</i>)
-3.600,00	(<i>PV</i>)
675,27	(<i>FV</i>)

(Beachten Sie, daß die letzte Zahlung $675,27 + 100,00 = 775,27$ Dollar beträgt, da sie mit dem Ende der letzten Periode zusammenfällt.)

Beispiel 4: Das Programm kann auch zur Berechnung der aufgelaufenen Zinsen bzw. des Restrückzahlungsbetrags eines Darlehens benutzt werden. Die zwischen zwei Zeitpunkten n_1 und n_2 aufgelaufenen Zinsen ergeben sich aus den Gesamtzahlungen minus der Tilgung für diesen Zeitraum. Die Tilgung entspricht der Differenz zwischen den Restrückzahlungsbeträgen zu den Zeitpunkten n_1 und n_2 . Das folgende Beispiel soll zur Begriffsklärung dienen:

Für ein Darlehen in Höhe von 50.000 Dollar mit einer Laufzeit von 360 Monaten und einer jährlichen Verzinsung von 14% ist der Restrückzahlungsbetrag nach der 24. Rate und der anteilige Zinsbetrag für die Raten 13 bis 24 (d.h. die zwischen der 12. und 24. Rate aufgelaufenen Zinsen) zu berechnen.

Zunächst sind die Raten zu berechnen:

Tastenfolge	Anzeige	
\boxed{f} CLEAR \boxed{REG}		
360 \boxed{B}	360,00	(<i>n</i>)
14 \boxed{ENTER} 12 $\boxed{\div}$ \boxed{A}	1,17	(<i>i</i>)
$\boxed{R/S}$	1,00	gewöhnliche Fälligkeit
50 000 \boxed{CHS} \boxed{C}	-50.000,00	(<i>PV</i>)
\boxed{D} $\boxed{R/S}$	592,44	(<i>PMT</i>)

Der Restrückzahlungsbetrag nach 24 Monaten ergibt sich zu:

24 \boxed{B} \boxed{E} $\boxed{R/S}$	49.749,56	(Restdarlehen (<i>FV</i>) nach 24 Monaten)
--	------------------	--

Dieser Betrag ist zu speichern; anschließend ist der Restrückzahlungsbetrag nach 12 Monaten zu berechnen:

\boxed{STO} 9		
12 \boxed{B} \boxed{E} $\boxed{R/S}$	49.883,48	(Restdarlehen (<i>FV</i>) nach 12 Monaten)

Damit ergibt sich eine Tilgung zwischen der 12. und 24. Rate von:

\boxed{RCL} 9 $\boxed{-}$	133,92
-----------------------------	---------------

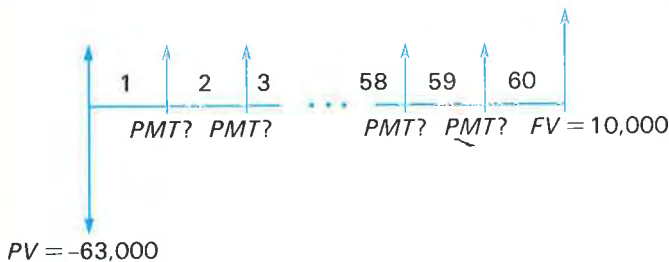
Die anteiligen Zinsen ergeben sich jetzt aus der Summe der 12 Raten minus der Tilgung:

\boxed{RCL} 4 12 $\boxed{\times}$	7.109,23	(gezahlter Gesamtbetrag)
$\boxed{x \approx y}$ $\boxed{-}$	6.975,31	(anteiliger Zinsbetrag)

Beispiel 5: Eine Leasing-Firma beabsichtigt den Erwerb eines Mini-Computers zu einem Preis von 63.000 Dollar. Man kalkuliert die Weitergabe des Rechners im Leasing-Verfahren an einen Kunden für die Dauer von fünf Jahren bei einem jährlichen Ertrag von 13%. Da das Eigentumsrecht an dem Rechner bei der Leasing-Firma verbleibt, hofft man, das Gerät nach fünf Jahren für mindestens 10.000 Dollar weiterverkaufen zu können. (Da Leasing-Zahlungen

am Anfang jeder Periode zu leisten sind, ist dies ein Vorausfälligkeitsproblem.)

Welche Monatsraten muß die Firma verlangen, um einen 13prozentigen Ertrag zu sichern.



Tastenfolge

f CLEAR REG
 13 ENTER 12 ÷ A
 R/S
 5 ENTER 12 × B
 63 000 CHS C
 10 000 E D R/S

Anzeige

1,08 (i)
 0,00 (Vorausfälligkeit)
 60,00 (n)
 -63.000,00 (PV)
 1.300,16 (PMT)

Welche Raten wären bei einem Preis von 70.000 Dollar notwendig?

70 000 CHS C
 D R/S

-70.000,00
 1.457,73 (PMT)

U-Boot-Jagd

Sie haben die Aufgabe, mit Ihrem Zerstörer ein feindliches U-Boot in einem 10×10 Gitter aufzuspüren, und es mit Wasserbomben zu vernichten.

Dazu geben Sie einen beliebigen Startwert (zwischen 0 und 1) ein, den der Rechner zur zufälligen Positionierung des U-Boots im Zentrum eines der 100 Quadrate benutzt. Der Zugriff auf die einzelnen Quadrate erfolgt über die Koordinaten (R, C) (R = row (Zeile) und C = column (Spalte)), wobei R und C jeweils zwischen 0 und 9 liegen können.

Zunächst suchen Sie die Position des Unterseeboots mit Ihrem Sonargerät. Dazu geben Sie die Position Ihres Zerstörers (R, C) ein

und drücken [B]. Wenn sich das U-Boot in einem der 8 benachbarten Quadrate (oder direkt unter Ihrem Zerstörer) befindet, zeigt der Rechner eine „1“, ansonsten eine „0“ an.

Wenn Sie die Position gefunden zu haben glauben, gehen Sie mit Ihrem Zerstörer direkt über das U-Boot (bewegen sich in das gleiche Quadrat) und werfen eine Wasserbombe. Eine blinkende „1“ deutet einen Treffer an, ein Fehlwurf wird durch eine „0“ charakterisiert. Bei einem Fehlwurf weicht das Unterseeboot auf einer Zufallsbasis in eines der vier benachbarten Quadrate der gleichen Zeile oder Spalte aus.

Um das Spiel etwas spannender zu machen, können Sie nach Eingabe des Startwerts die Taste [C] drücken. In diesem Fall bewegt sich das Unterseeboot nach jedem Sonarecho genauso wie nach einem erfolglosen Wasserbombenabwurf. Das Unterseeboot weicht dann (auf Zufallsbasis) in eines der benachbarten Quadrate der gleichen Zeile oder Spalte aus.

Eine Wasserbombe hat eine Reichweite von 0,9 Quadraten. Wenn Sie Ihren Zerstörer für einen Wasserbombenabwurf positionieren, können Sie sich zu einem beliebigen Punkt innerhalb des Gitters bewegen, nicht notwendigerweise nur zum Mittelpunkt eines Quadrats. Zum Beispiel würde eine an der Position (2,5, 6,5) abgeworfene Wasserbombe jedes U-Boot im Zentrum der Quadrate (2, 6), (2, 7), (3, 6) und (3, 7) zerstören.

Versuchen Sie zur Vernichtung des Unterseeboots nicht mehr als 10 Echolotungen und einen Wasserbombenabwurf zu verwenden. Sie können Ihren Status jederzeit durch Drücken von [D] abfragen. Der Status wird in dem Format XX.YY angezeigt, wobei

XX = Anzahl der abgeworfenen Wasserbomben

YY = Anzahl der Echolotungen

Tastenfolge	Anzeige	Tastenfolge	Anzeige
[f] CLEAR [PRGM]	000-	[g] [RTN]	005- 43 32
[f] [LBL] [C]	001-42.21.13	[f] [LBL] [E]	006-42.21.15
1	002- 1	[f] CLEAR [REG]	007- 42 34
[STO] 0	003- 44 0	[g] [CF] 0	008-43. 5. 0
[g] [SF] 0	004-43. 4. 0	[STO] [RAN#]	009- 44 36

Tastensequenz	Anzeige	Tastensequenz	Anzeige
[GSB] 9	010- 32 9	[f] [PSE]	039- 42 31
[STO] 1	011- 44 1	[g] [RTN]	040- 43 32
[GSB] 9	012- 32 9	[f] [LBL] [B]	041-42.21.12
[STO] 2	013- 44 2	1	042- 1
[f] [FIX] 0	014-42. 7. 0	[STO] [+ 8	043-44.40. 8
[g] [CL _y]	015- 43 35	[R↓]	044- 33
[g] [RTN]	016- 43 32	[f] [FIX] 0	045-42. 7. 0
[f] [LBL] [A]	017-42.21.11	[g] [CF] 1	046-43. 5. 1
1	018- 1	[GSB] 6	047- 32 6
[STO] [+ 7	019-44.40. 7	[RCL] 0	048- 45 0
[R↓]	020- 33	[STO] 5	049- 44 5
[g] [SF] 1	021-43. 4. 1	[g] [F?] 0	050-43. 6. 0
[GSB] 6	022- 32 6	[GSB] 5	051- 32 5
[g] [x≠0]	023- 43 30	[RCL] 3	052- 45 3
[GTO] 0	024- 22 0	[g] [RTN]	053- 43 32
1	025- 1	[f] [LBL] 5	054-42.21. 5
[STO] 5	026- 44 5	[GSB] 9	055- 32 9
[GSB] 5	027- 32 5	4	056- 4
[g] [RTN]	028- 43 32	[x≧y]	057- 34
[f] [LBL] 0	029-42.21. 0	[f] [x>y]	058- 42 20
9	030- 9	[GTO] 0	059- 22 0
[1/x]	031- 15	[RCL] 5	060- 45 5
[f] [FIX] 3	032-42. 7. 3	[CHS]	061- 16
[f] [PSE]	033- 42 31	[GTO] 1	062- 22 1
[f] [FIX] 5	034-42. 7. 5	[f] [LBL] 0	063-42.21. 0
[f] [PSE]	035- 42 31	[RCL] 5	064- 45 5
[f] [FIX] 7	036-42. 7. 7	[f] [LBL] 1	065-42.21. 1
[f] [PSE]	037- 42 31	[STO] 6	066- 44 6
[f] [FIX] 9	038-42. 7. 9	[GSB] 9	067- 32 9

Tastensequenz	Anzeige	Tastensequenz	Anzeige
5	068- 5	f [FIX] 0	097-42. 7. 0
f [x>y]	069- 42 20	g [RTN]	098- 43 32
[GTO] 0	070- 22 0	f [LBL] 6	099-42.21. 6
[RCL] 1	071- 45 1	[RCL] 2	100- 45 2
[GSB] 1	072- 32 1	-	101- 30
[STO] 1	073- 44 1	[x≥y]	102- 34
[GTO] 2	074- 22 2	[RCL] 1	103- 45 1
f [LBL] 0	075-42.21. 0	-	104- 30
[RCL] 2	076- 45 2	g [→P]	105- 43 26
[GSB] 1	077- 32 1	[STO] 4	106- 44 4
[STO] 2	078- 44 2	g [F?] 1	107-43. 6. 1
[GTO] 2	079- 22 2	[GTO] 0	108- 22 0
f [LBL] 1	080-42.21. 1	1	109- 1
[RCL] 6	081- 45 6	-	110- 30
+	082- 40	f [LBL] 0	111-42.21. 0
g [x<0]	083- 43 10	.	112- 48
[GTO] 0	084- 22 0	9	113- 9
9	085- 9	-	114- 30
[x≥y]	086- 34	g [x<0]	115- 43 10
f [x≤y]	087- 42 10	[GTO] 0	116- 22 0
g [RTN]	088- 43 32	0	117- 0
f [LBL] 0	089-42.21. 0	[GTO] 1	118- 22 1
[RCL] 6	090- 45 6	f [LBL] 0	119-42.21. 0
2	091- 2	1	120- 1
x	092- 20	f [LBL] 1	121-42.21. 1
-	093- 30	[STO] 3	122- 44 3
g [RTN]	094- 43 32	g [RTN]	123- 43 32
f [LBL] 2	095-42.21. 2	f [LBL] [D]	124-42.21.14
[RCL] 3	096- 45 3	f [FIX] 2	125-42. 7. 2

Tastenfolge	Anzeige	Tastenfolge	Anzeige
[RCL] 7	126- 45 7	f [LBL] 9	133-42.21. 9
[RCL] 8	127- 45 8	f [RAN#]	134- 42 36
[EEX]	128- 26	1	135- 1
2	129- 2	0	136- 0
[÷]	130- 10	x	137- 20
[+]	131- 40	g [INT]	138- 43 44
g [RTN]	132- 43 32	g [RTN]	139- 43 32

Register			R ₁ : unbelegt
R ₀ : 0, 1	R ₁ : P ₁	R ₂ : P ₂	R ₃ : Echo
R ₄ : d	R ₅ : belegt	R ₆ : belegt	R ₇ : belegt
R ₈ : unbelegt			

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
1	Eingabe des Programms			
2	User Mode wählen			
3	Eingabe Startwert (zwischen 0 und 1)	n	[E]	0
4	Bei einfachem Spiel, Forts. mit Schritt 6			
5	Verschärftes Spiel wählen (U-Boot immer beweglich)		[C]	1
6	Echolotung	R	[ENTER]	
	„0“ – kein Echo	C	[B]	0 oder 1
	„1“ – Echo empfangen			
	oder			
	Wasserbombenabwurf	R	[ENTER]	
	„0“ bedeutet Fehlwurf	C	[A]	0 oder
	Blinkende „1“			blinken
	kennzeichnet Treffer			

Schritt	Anweisung	Eingabe Daten/Einh.	Tasten	Ausgabe Daten/Einh.
7	Schritt 6 wiederholen bis U-Boot getroffen			
8	Anzeige Status (zu beliebigem Zeitpunkt) XX = Anz. d. abgew. Wasserbomben YY = Anzahl d. Echolotungen		<input type="checkbox"/> D	XX.YY
9	Bei neuem Spiel Fort- setzung mit Schritt 3			

	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9		
9											9												9
8											8												8
7											7												7
6											6												6
5											5												5
4											4												4
3											3												3
2											2												2
1											1												1
0											0												0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9		
9											9												9
8											8												8
7											7												7
6											6												6
5											5												5
4											4												4
3											3												3
2											2												2
1											1												1
0											0												0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9		
9											9												9
8											8												8
7											7												7
6											6												6
5											5												5
4											4												4
3											3												3
2											2												2
1											1												1
0											0												0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9		

„Spielbretter“ zur U-Boot-Jagd. Sie können Kopien dieser Seite für Ihre Spiele verwenden.

Beispiel 1:**Tastenfolge**

Schalten Sie auf User Mode.

□ 58 [E]

Erster Zug:

3 [ENTER] 8 [B]

Anzeige

0,

1,

Echo.

Sie wissen nun, daß sich der Feind in einem der durch „x“ gekennzeichneten Quadrate befindet.

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								X	X	X	4
3								X	X	X	3
2								X	X	X	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

Diagramm des ersten Zugs.

Zweiter Zug:

4 [ENTER] 7 [B]

0,

Kein Echo

Das U-Boot kann sich nicht in den mit „⊗“ gekennzeichneten Quadraten befinden.

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								⊗	⊗	X	4
3								⊗	⊗	X	3
2								X	X	X	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

Diagramm des zweiten Zugs.

Dritter Zug:

2 [ENTER] 9 [B]

0,

Kein Echo

Sie haben die mögliche U-Boot-Position jetzt bis auf zwei Quadrate eingengt. Diese beiden Quadrate sind durch ein „x“ ohne Kreis gekennzeichnet.

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								X	X	X	4
3								X	X	X	3
2								X	X	X	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

Diagramm des dritten Zugs.

Vierter Zug:

4 [ENTER] 9 [B]

1,

Echo

Dieser Zug eliminiert (2, 7) als die einzig mögliche Position des Unterseeboots. Sie haben den Feind damit gefunden.

	0	1	2	3	4	5	6	7	8	9	
9											9
8											8
7											7
6											6
5											5
4								X	X	X	4
3								X	X	X	3
2								X	X	X	2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

Diagramm des vierten Zugs.

Fünfter Zug:

4 [ENTER] 9 [A]

0,111

0,11111

0,1111111

0,11111111

0,111111111

Treffer!

Beispiel 2:

Tastenfolge

. 60 [E]

[C]

Anzeige

0,

1,

Das U-Boot reagiert jetzt sowohl auf Sonarechos wie auf Fehlwürfe von Wasserbomben.

Erster Zug:

7 [ENTER] 4 [B]

1,

Echo

Das Unterseeboot befindet sich in einem der durch „x“ gekennzeichneten Quadrate im linken Diagramm. Durch die mögliche Reaktion auf die Echolotung kann es in jedes der durch „x“ gekennzeichneten Quadrate im rechten Diagramm ausweichen.

	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	
9										9					X	X	X					9
8				X	X	X				8				X	X	X	X	X				8
7				X	X	X				7				X	X	X	X	X				7
6				X	X	X				6				X	X	X	X	X				6
5										5					X	X	X					5
4										4												4
3										3												3
2										2												2
1										1												1
0										0												0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	

Diagramme des ersten Zugs.

Zweiter Zug:

6 [ENTER] 4 [B]

0,

Kein Echo

Sie haben einige Positionen eliminiert (linkes Diagramm: x), aber gleichzeitig sind durch das zufällige Ausweichen des Unterseeboots neue Positionen möglich geworden.

	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	
9				X	X	X					9			X	X	X	X	X				9
8			X	X	X	X	X				8		X	X	X	X	X	X	X	X		8
7			X	X	X	X	X				7		X	X	X	X	X	X	X	X		7
6			X	X	X	X	X				6		X	X	X		X	X	X			6
5				X	X	X					5			X				X				5
4											4											4
3											3											3
2											2											2
1											1											1
0											0											0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	

Diagramm des zweiten Zugs.

Dritter Zug:

7 [ENTER] 3 [B]

1,

Echo

Durch diesen Zug werden viele der möglichen Positionen des U-Boots eliminiert, aber wiederum ist ein Ausweichen in neue Quadrate möglich.

	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	
9			X	X	X	X	X				9			X	X	X						9
8		X	X	X	X	X	X	X			8		X	X	X	X	X					8
7		X	X	X	X	X	X	X			7		X	X	X	X	X					7
6		X	X	X		X	X	X			6		X	X	X	X						6
5			X					X			5			X	X							5
4											4											4
3											3											3
2											2											2
1											1											1
0											0											0
	0	1	2	3	4	5	6	7	8	9		0	1	2	3	4	5	6	7	8	9	

Diagramm des dritten Zugs.

Vierter Zug: Sie riskieren einen Wasserbombenabwurf.

8 3

0,111

0,11111

0,1111111

0,111111111

0,111111111

Treffer!

Es hat sich gelohnt!

In diesem Beispiel lag das Unterseeboot hier.

	0	1	2	3	4	5	6	7	8	9	
9											9
8				X							8
7											7
6											6
5											5
4											4
3											3
2											2
1											1
0											0
	0	1	2	3	4	5	6	7	8	9	

Programmiertechniken

Struktur

Unsere Definition des Begriffs „Struktur“ besagt, daß selbst eine so weit vom Englischen entfernte Sprache wie die Programmiersprache des HP-11C eine gewisse Organisation enthalten kann. Wir wollen Ihnen die Erkenntnis vermitteln, daß die Programme, die Sie auf dem HP-11C realisieren, sowohl nützlich als auch „freundlich“ sein können. Aber es liegt an Ihnen, dem Programmierer, sie dazu zu machen. Um Ihre Programme effizient, lesbar und modifizierbar zu gestalten, ist lediglich ein wenig Voraussicht und Planung erforderlich. Dieser Abschnitt stellt eine mit Beispielen angereicherte Zusammenfassung der verschiedensten Programmiertechniken dar, von denen wir hoffen, daß sie Ihnen beim Schreiben solcher Programme behilflich sein können.

Das Problemstatement

Der erste Schritt beim Schreiben eines Programms sollte darin bestehen, daß man das zu lösende Problem analysiert. Dieser Schritt scheint offensichtlich zu sein, wird jedoch häufig außer acht gelassen. Der Programmierer sieht sich dann oft in der Lage, daß das Programm nicht die gewünschten Resultate produziert, weil die ursprünglichen Anliegen nicht klar genug durchdacht worden sind. Dieser erste Schritt ist immer wichtig, da durch ihn zumeist eine klare Vorstellung von der Problemstellung und ein bestimmter Weg zur Lösung vermittelt werden. Erst danach sollte mit der logischen Entwicklung des Programms begonnen werden.

Beispiel: Gesucht seien die Lösungen der Gleichung $ax^2 + bx + c = 0$, wobei a , b und c Konstanten sind. Unser Problemstatement könnte dann die Form haben: „Bei Vorgabe von a , b und c sind die beiden Lösungen der Gleichung $ax^2 + bx + c = 0$ zu berechnen.“

Ein derartiges Statement scheint offensichtlich zu sein; es enthält jedoch zwei wichtige Informationen: 1) die vorzugebenden Daten und 2) die gewünschte Ausgabe. Mit diesen beiden Grundmerkmalen und einem Lösungskonzept können wir zum zweiten Schritt, dem Entwurf des Algorithmus, übergehen.

Der Algorithmus

Ein Algorithmus ist kein Programm, sondern eine Zusammenstellung der zur Lösung eines Problems erforderlichen, logischen

Schritte. Ein solcher Algorithmus sollte am Anfang nicht zu spezifisch sein, und sich unter Weglassen der Details nur auf die logischen Einheiten beschränken. Diese Details werden später durch die Programmiersprache, die Speichermöglichkeiten und den persönlichen Programmierstil aufgefüllt. Hier in diesem Schritt sollen nur die Grundlagen der Lösung formuliert werden. In unserem Fall soll jetzt versucht werden, das Problem der quadratischen Gleichung mit Hilfe der (bekannten) algebraischen Lösung

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

zu formulieren. Damit könnte ein erster Algorithmus die folgende Gestalt haben:

1. Berechne $b^2 - 4ac$
2. Wenn die Differenz positiv ist, berechne $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$
3. Wenn die Differenz negativ ist, berechne

$$\frac{-b}{2a} \quad \text{und} \quad \frac{\pm \sqrt{|b^2 - 4ac|}}{2a}$$

Wenn der Ausdruck $b^2 - 4ac$ positiv ist, existieren zwei reelle Lösungen; bei negativer Differenz besteht die Lösung aus zwei komplexen Zahlen. Beachten Sie, daß noch keine konkreten Programmschritte formuliert worden sind. Bisher wurde nur eine Folge von Rechenschritten, mit der die Lösung ermittelt werden kann, definiert.

Durch Verfeinerungen verdichtet sich der Basisalgorithmus zu Mustern und Sequenzen, die auf die letztlich zu benutzenden, konkreten Programmschritte hindeuten. Dieser Verfeinerungsprozeß hilft Ihnen dabei, die Zielstellungen Ihrer Programmierung im Auge zu behalten. Eine derartige Verfeinerung könnte wie folgt aussehen:

Unter Verwendung der hypothetischen Register R_A , R_B und R_C sind die folgenden Operationen auszuführen:

1. Die Konstante b ist mit (-1) zu multiplizieren und zweimal durch a zu teilen.
2. Das Ergebnis ist in R_A zu speichern.
3. b ist zu quadrieren, von dem Quadrat ist viermal c zu subtrahieren, und die Differenz ist in R_B zu speichern.
4. Aus dem Betrag des Werts in R_B ist die Quadratwurzel zu ziehen und zweimal durch a zu dividieren.

5. Das Ergebnis ist in R_C zu speichern.
6. Wenn der Wert in R_B positiv oder Null ist, sind die Zahlen in R_A und R_C zu addieren, und das (reelle) Ergebnis ist anzuzeigen.
7. Wenn der Wert in R_B kleiner Null ist, sind die Inhalte der Register R_A und R_C getrennt anzuzeigen (die komplexe Nullstelle).
8. Das Vorzeichen des Werts in R_C ist umzukehren, und die vorangegangenen beiden Schritte sind zu wiederholen.

Bei dieser Verfeinerung wird die Bedeutung der Verwendung von Registern zur Speicherung von Zwischenergebnissen deutlich. Des weiteren sind die einzelnen Operationen bereits exakter definiert, und die aktuellen Ausgaben werden schon berücksichtigt.

Der Grad der Verfeinerung des letztlich resultierenden Algorithmus liegt im Ermessen des Programmierers; es gilt jedoch immer die Regel, daß je feiner der Algorithmus ist, desto weniger Modifikationen sind für einen einwandfreien Ablauf des Programms notwendig. Im folgenden wird eine befriedigende Verfeinerung unseres Algorithmus' dargestellt:

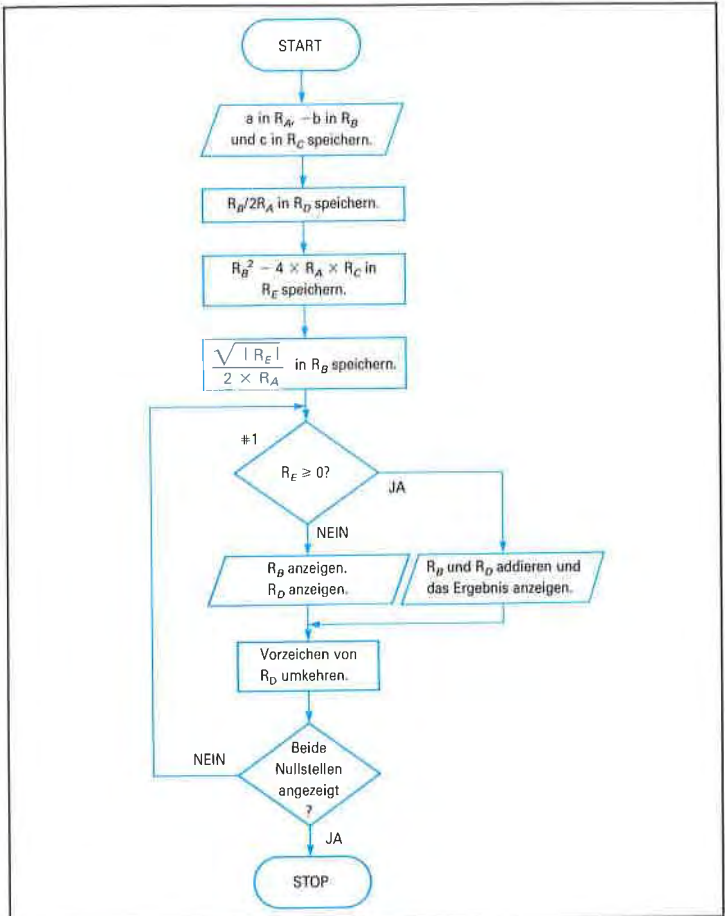
Unter Verwendung der hypothetischen Register R_A , R_B , R_C , R_D und R_E sind die folgenden Operationen auszuführen:

1. In R_A ist a , in R_C c und in R_B ist das Produkt von b und (-1) zu speichern.
2. R_B ist zweimal durch R_A zu teilen, und das Ergebnis ist in R_D zu speichern.
3. Das Produkt von R_A und R_C ist viermal vom Quadrat von R_B zu subtrahieren, und das Ergebnis ist in R_E zu speichern.
4. Aus dem Absolutwert des Inhalts von R_E ist die Quadratwurzel zu ziehen und zweimal durch R_A zu dividieren. Das Ergebnis ist in R_B zu speichern.
5. Wenn R_E positiv oder Null ist, sind R_B und R_D zu addieren, und das Ergebnis ist anzuzeigen.
6. Wenn R_E negativ ist, sind R_B und R_D getrennt anzuzeigen.
7. Das Vorzeichen von R_B ist umzukehren, und die vorangegangenen beiden Schritte sind zu wiederholen.

Dieser Algorithmus berücksichtigt die zu Beginn notwendige Speicherung der Daten und enthält eine Doppelverwendung des Registers R_B , wodurch sich die Gesamtanzahl der benötigten Speicherregister verringert.

Flußdiagramme

Ein sehr nützlicher Weg zur Veranschaulichung eines Algorithmus besteht in der Verwendung von „Flußdiagrammen“. In einem Flußdiagramm wird der Ablauf des Algorithmus aufgezeichnet, dabei die zugrundeliegende, logische Gedankenentwicklung sichtbar machend. Unser zuvor entwickelter Algorithmus könnte durch das folgende Flußdiagramm charakterisiert werden:



Auf den ersten Blick mag so ein Flußdiagramm kompliziert aussehen, doch bei näherer Betrachtung tritt eine einfache Logik zutage. Um das Flußdiagramm zu lesen, müssen Sie in der linken, oberen Ecke beginnen und immer den Pfeilen folgen. Jedes Kästchen ist mit dem nächsten durch einen Pfeil in einer Richtung verbunden. Dies besagt, daß das Diagramm in einer Richtung durchlaufen wird, in der Regel von oben nach unten.

Beachten Sie, daß bei dem mit # 1 gekennzeichneten Kästchen eine Wahlmöglichkeit bei den Ausgängen oder eine „Verzweigung“ besteht. Die Durchlaufrichtung wird an dieser Stelle durch die Antwort auf die im Kästchen gestellte Frage bestimmt, also in diesem Fall, ob der Wert in R_E größer oder gleich Null ist. Bei einer Antwort von „JA“ wird nach rechts verzweigt, während im Fall von „NEIN“ das Diagramm weiter nach unten durchlaufen wird.

Wie Sie sehen, kann ein Flußdiagramm sehr nützlich für eine klare Aufgliederung des Algorithmus sein, insbesondere im Bereich von Verzweigungen und Schleifen, wo die Durchlaufrichtung nicht eindeutig ist.

Unterprogramme

Das Erste, was Ihnen beim Betrachten des Programms zur Matrizenrechnung vielleicht auffallen wird, ist die überraschende Länge. Es ist so groß, daß es, in seiner Gesamtheit, nicht im Speicher des Rechners gehalten werden kann. Was auf den ersten Blick nicht auffällt, ist die Tatsache, daß es auch ein stark komprimiertes Programm ist. Dies liegt daran, daß die bei der Lösung des Problems auftretenden, sich wiederholenden Muster ausgenutzt werden. In der Tat wird sogar die Operation zur Bildung der Matrizeninversen wiederholt. Deshalb sind diese und weniger wichtige Funktionen in das Programm als Unterprogramme eingegliedert.

Formal (in der Sprache des Rechners) kann ein Unterprogramm aus einer Folge von beliebigen Programmbeehlen bestehen, die mit einem Label (`[LBL] n`) beginnt und mit einem Rücksprung (`[RTN]`) oder dem Programmende aufhört. Diese Grenzen ermöglichen den Einsprung in das Unterprogramm und den Rücksprung aus dem Unterprogramm. (Es ist sogar möglich, in ein Unterprogramm an beliebigen Zeilennummern einzuspringen, siehe Seite 136, Indirekte Label-Adressierung und indirekter Aufruf von Unterprogrammen.)

Der Zugriff auf ein Unterprogramm erfolgt über den Befehl `[GSB] n` (go to subroutine n). Beachten Sie, daß im Hauptteil (Zeile 000 bis 078) des Programms zur Matrizenrechnung die Anweisung `[GSB] 8`

acht mal auftritt. Jede Ausführung dieses Befehles beinhaltet einen Sprung zu der Anweisung **[LBL 8]**, die Abarbeitung der Befehle zwischen **[LBL 8]** und **[RTN]** und den Rücksprung zu der unmittelbar auf **[GSB]** folgenden Anweisung.

Die Vorteile bei der Verwendung von Unterprogrammen werden zuerst bei der Speicherplatzersparnis deutlich. Es ist wesentlich günstiger, das Unterprogramm 8 mit acht **[GSB]** Befehlen aufzurufen, als die Befehle des Unterprogramms acht mal in das Hauptprogramm einzufügen. Ein weiterer Vorteil bei der Verwendung von Unterprogrammen liegt in der verbesserten Organisation. Das Programm zur Matrizenrechnung wurde in seine Einzelkomponenten zerlegt, da jede Komponente für sich einfacher zu lesen und zu verstehen ist, als das unsegmentierte Gesamtprogramm. Wenn die Funktion einer jeden Komponente verstanden ist, wird auch das Verständnis des Programms als Ganzes erleichtert. Diese Aufteilung vereinfacht auch die Fehlerkorrektur. Die Fehler sind leichter zu isolieren, und die Gefahr, daß bei der Korrektur unerwünschte Effekte in anderen Programmteilen auftreten, wird verringert.

[ISG mit RCL (i)]

Wenn Sie Ihren Algorithmus auf sich wiederholende Muster hin durchsuchen, ist es von Vorteil, dabei auf sequentielle Speicher- und Rückrufoperationen z.B. **[STO 1, ..., STO 2, ..., STO 3]**, zu achten. Solche Sequenzen können mit Hilfe der **[STO (i)]** und **[RCL (i)]** Funktionen in Unterprogramme eingegliedert werden. Diese sehr effektive Technik wird auch in dem Programm zur Matrizenrechnung benutzt.

Im Unterprogramm „A“ (Zeile 079 bis 101) wird die Determinante der in den Registern R_5 bis R_3 gespeicherten 3×3 Matrix berechnet. Die mathematische Vorschrift zur Berechnung der Determinante lautet:

$$\begin{vmatrix} R_5 & R_4 & R_7 \\ R_8 & R_9 & R_0 \\ R_1 & R_2 & R_3 \end{vmatrix} = R_5 (R_9 \times R_3 - R_0 \times R_2) \\ - R_6 (R_8 \times R_3 - R_0 \times R_1) \\ + R_7 (R_8 \times R_2 - R_9 \times R_1)$$

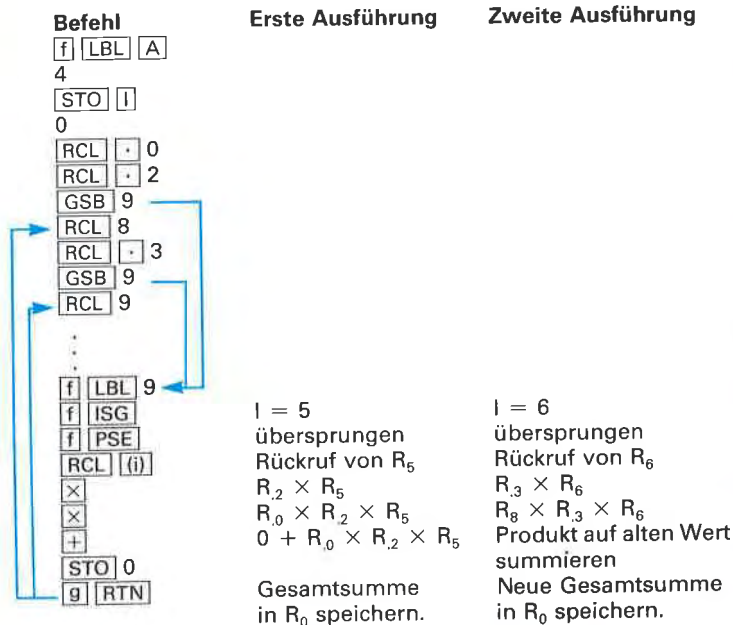
Durch Umordnen erhält man das folgende Muster:

$$-(R_0 \times R_2 \times R_5 + R_6 \times R_3 \times R_6 + R_1 \times R_9 \times R_7) \\ + R_7 \times R_2 \times R_8 + R_5 \times R_3 \times R_9 + R_6 \times R_1 \times R_0$$

Als Ergebnis erhält man nicht nur das Muster $R \times R \times R +$ sondern auch den sequentiellen Rückruf aus den Registern R_5 bis R_0 . Diese beiden Merkmale wurden zu dem Unterprogramm 9 (Zeile 001 bis 009) zusammengefaßt.

In diesem Unterprogramm wird der Befehl **ISG** (Increment and Skip if Greater) zum Erhöhen des Werts im Indexregister benutzt (siehe Seite 128). Bei jedem Aufruf des Unterprogramms wird der in I gespeicherte Wert um 1 erhöht, und der Inhalt des durch den Wert des Indexregisters adressierten Registers (**RCL (i)**) wird zurückgerufen. Da der dezimale Anteil der Zahl im Indexregister immer Null und der ganzzahlige Anteil immer größer Null ist, wird der auf die **ISG** Anweisung folgende Befehl in jedem Fall übersprungen. Aus diesem Grund wird an dieser Stelle ein Dummy-Befehl (ein „leerer“ Befehl ohne jede Wirkung) eingesetzt, in unserem Fall **PSE**.

Das schrittweise Durchlaufen des Programms illustriert die Funktion des Unterprogramms.



Die Speicherplatzersparnis ist klar. Ohne Verwendung des Unterprogramms wären vier Schritte sechsmal zu wiederholen, also insgesamt 24 Programmzeilen notwendig. Durch das Unterprogramm wird diese Zahl auf 10 reduziert (einschließlich von 4 `[STO]` Befehlen und ohne die `[STO]` 0 Befehle, die in jedem Fall notwendig gewesen wären).

Dateneingabe

Wenn Sie beim Schreiben eines Programms festgestellt haben, welche Daten das Programm benötigt, müssen Sie danach festlegen, wie diese Daten gespeichert werden sollen. Die Art der Speicherung ist abhängig von dem zur Verfügung stehenden Speicherplatz und der Anzahl der zu speichernden Daten.

In dem Programm zur Dreiecksberechnung waren jeweils drei Eingabegrößen für jeden der fünf Fälle SSS, ASA, SAA, SAS und SSA notwendig. Es ist klar, daß zur Eingabe der Daten fünf ähnliche, doch letztlich unterschiedliche Befehlsfolgen benötigt werden. Hier kann jedoch, da der HP-11C über fünf Benutzer-definierbare Tasten verfügt, der Eingabe jeder auftretenden Variablenkombination eine Taste zugeordnet werden.

Es gibt mehrere Möglichkeiten, mit Hilfe der Benutzer-definierbaren Tasten Variablen zu laden. Eine davon wäre, die bekannten Größen per Hand in den entsprechenden Registern zu speichern (z.B. „S1“ `[STO]` 1, „A1“ `[STO]` 2, usw.), und anschließend die zu dem jeweiligen Fall gehörende Routine über die zugeordnete Benutzer-definierbare Taste anzuwählen. Obwohl dieses Verfahren im Fall von wenigen Variablen oder geringem Speicherplatz als Eingaberoutine akzeptabel ist, werden in der Regel dabei die Zeit und Aufwand ersparenden Möglichkeiten des HP-11C nicht voll ausgenutzt.

Eine andere, etwas bequemere Möglichkeit, bildet das „stop-and-store“-Verfahren. Dabei ist zunächst die zu dem jeweiligen Fall gehörende Benutzer-definierbare Taste zu betätigen. Das Programm hält dann sofort an und wartet auf die Eingabe der ersten Variablen. Sie geben diese ein und drücken `[R/S]` zur Fortsetzung der Programmausführung. Das Programm speichert den Eingabewert in dem entsprechenden Register und wartet dann auf die nächste Eingabe. Dieser Prozeß wiederholt sich, bis alle Variablen eingegeben sind. Etwas ausgefeiltere Versionen des „stop-and-store“-Verfahrens beinhalten Eingabeschleifen, die die Eingabevariablen abfragen und sogar die Anzeige und Modifikation der Eingaben zulassen (siehe Unterprogramm A im Unterabschnitt „Lineare Gleichungssysteme mit 3 Unbekannten“).

Die Methode, die wir im Fall unseres Programms (zur Dreiecksberechnung) benutzt haben, besteht darin, den Stack mit den Eingabewerten (es sind hier nur drei) zu laden und anschließend die entsprechende Benutzer-Taste zu wählen. Die erste Funktion jeder der Benutzer-Tasten besteht nun darin, die drei Eingaben mittels der Sequenz $\boxed{\text{STO}} R_A, \boxed{R\downarrow}, \boxed{\text{STO}} R_B, \boxed{R\downarrow}, \boxed{\text{STO}} R_C$ in den entsprechenden Registern zu speichern. Diese Methode ist schnell und einfach und erfordert nur wenige Programmschritte. Ihr Nachteil liegt darin, daß ihre Anwendung auf Programme, die nur eine geringe Anzahl von Eingaben benötigen, begrenzt ist.

Programmschleifen

Das Programm zur Nullstellenbestimmung mit dem Newton-Verfahren behandelt das allgemeine Problem der Approximation der (unbekannten) Lösung einer Gleichung. Derartige Approximationen sind deswegen notwendig, weil die exakte Bestimmung von Nullstellen bei einigen Funktionen sehr schwierig und in vielen Fällen sogar unmöglich ist.

Bei dem Newton-Verfahren wählt man einen Punkt x , an dem die Funktion $f(x)$ erklärt ist, und von dem man hofft, daß er in der Nähe einer Nullstelle der Funktion liegt. (Im allgemeinen läßt sich der Bereich, in dem eine Nullstelle liegt, durch grobes Abschätzen der Funktionswerte einschränken.) Im nächsten Schritt wird die Näherung korrigiert, indem eine auf der Startnäherung basierende, bessere Approximation der Nullstelle berechnet wird. Diese verbesserte Näherung wird dann zur Berechnung einer neuen, noch besseren Approximation verwendet.

Wenn dieser Prozeß unendlich oft wiederholt wird, führt er auf die exakte Lösung.

Man sieht, daß ein bestimmter Teil des Programms immer und immer wieder wiederholt werden muß, bis die gesuchte Lösung gefunden ist. Ein derartiger Programmteil wird als „Programmschleife“ bezeichnet. (Die Hauptschleife in dem Programm zur Nullstellenberechnung liegt zwischen den Zeilen 026 und 052.) Da eine unendliche Anzahl von Wiederholungen einen unendlichen Zeitaufwand implizieren würde, muß eine Schranke für die Anzahl der zulässigen Iterationen gesetzt werden.

Eine Möglichkeit zum Setzen einer derartigen Schranke bestünde darin, in die Schleife einen Zähler einzubauen, so daß bei jedem Durchlaufen der Schleife ein bestimmtes Register um den Wert eins erhöht wird. Die Anzahl der Iterationen (Schleifendurchläufe) wird

dann ständig mit dem gewünschten Maximum verglichen. Das Programm verläßt die Schleife dann, wenn die beiden Werte gleich sind. (Im vorliegenden Programm wird das Maximum bei jedem Durchlauf herabgesetzt, und die Schleife wird beendet, wenn der Wert Null erreicht hat. Siehe Zeile 051). Beachten Sie, daß sich bei dieser Methode kein konstanter Genauigkeitsstandard erreichen läßt. Die Anzahl der zur Berechnung einer akzeptablen Näherung benötigten Iterationen hängt von der jeweiligen Funktion und der Güte der Startnäherung ab, und kann im voraus (a priori) nicht abgeschätzt werden. Mit anderen Worten, der Abbruch des Verfahrens nach hundert Iterationen und mehr kann eine völlig unbrauchbare Lösung liefern.

Ein geeigneteres Abbruchkriterium besteht im Testen der letzten beiden berechneten Näherungen, d.h. in der Abfrage, ob sich die beiden Werte signifikant unterscheiden. Diese mit Δx bezeichnete Differenz ist vom Benutzer festzulegen und beim Initialisieren des Programms einzugeben.

Eine derartige Schleife könnte die folgende Form haben.

1. Speichern der Differenz Δx und der Startnäherung.
2. Berechnen und Speichern der ersten, auf der Startnäherung basierenden Approximation.
3. Berechnen und Speichern einer zweiten, auf der ersten Approximation basierenden Näherung.
4. Rückruf des Werts der vorangegangenen Näherung und Subtraktion von der aktuellen Näherung.
5. Rückruf von Δx und Vergleichen dieses Werts mit dem Betrag der eben berechneten Differenz.
6. Fortsetzen der Schleife, wenn die aktuelle Differenz betragsmäßig größer als Δx ist.
7. Verlassen der Schleife, wenn die aktuelle Differenz betragsmäßig kleiner als oder gleich Δx ist.

Bei dieser Vorgehensweise ist zu beachten, daß ein Abbruch der Schleife nicht garantiert ist. Beispielsweise kann die vorgegebene Startnäherung zu weit von der gesuchten Nullstelle entfernt sein, oder die gesuchte Nullstelle existiert überhaupt nicht. Im allgemeinen wird eine Kombination beider Abbruchkriterien (wie auch in diesem Programm) benutzt.

Ein nichttriviales Phänomen ist das Auftreten sogenannter „Phantomwurzeln“ (d.h. das Verfahren liefert Nullstellen, die in Wirklichkeit überhaupt nicht existieren). Ein derartig exotisches Verhalten ist bei Funktionen zu beobachten, deren Steigung so groß ist, daß der Unterschied zweier aufeinanderfolgenden Näherungen innerhalb der Δx Differenz liegt. Das vorliegende Programm enthält eine Sicherung gegen das Auftreten von Phantomwurzeln, indem nicht nur der Schleifenzähler und die Δx Differenz geprüft, sondern auch der Wert der Funktion an der in Frage kommenden Stelle untersucht wird. Der ermittelte Funktionswert wird mit einer vom Benutzer vorgegebenden Toleranz verglichen, die festlegt, in welchem Bereich um Null die berechnete Lösung liegen muß.

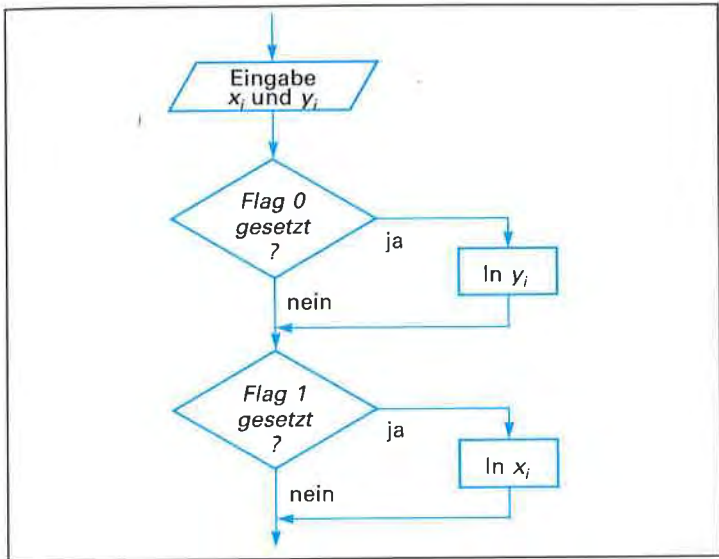
Flags

Beim Schreiben eines Programms, das unterschiedliche Problemstellungen in ähnlicher, jedoch nicht identischer Weise behandelt, können Flags zur Kontrolle der Unterschiede bei der Verarbeitung der Daten benutzt werden. Flags dienen als „Erinnerungshilfen“. Beispielsweise kann ein Flag gesetzt werden, wenn ein Programm eine optionale Initialisierungsoperation durchführt. Später im Programm kann es von Interesse sein, zu wissen, ob die Operation durchgeführt wurde oder nicht. Dazu kann dann der Status des Flags abgefragt werden, d.h. ob der Flag gesetzt ist oder nicht. Auf diese Weise stellen die Flags einen Teil der Entscheidungsmöglichkeiten des Rechners dar.

Bei dem Kurvenanpassungsprogramm können drei sehr ähnliche Prozesse, in Abhängigkeit von der bei der Regression zu verwendenden Kurve, angestoßen werden. Zur Illustration der Rolle der Flags könnte die Tabelle auf Seite 162 um zwei Spalten erweitert werden.

Regression	A	X_i	Y_i	Code	Flag 0	Flag 1
Exponentiell	$\ln a$	x_i	$\ln y_i$	1	gesetzt	nicht ges.
Logarithmisch	a	$\ln x_i$	y_i	2	nicht ges.	gesetzt
Potenz	$\ln a$	$\ln x_i$	$\ln y_i$	3	gesetzt	gesetzt

Die beiden Flags werden beim Initialisieren des Programms gesetzt, und, in Abhängigkeit von der gewählten Regressionsart, wird der eine oder andere, gemäß der obigen Tabelle, zurückgesetzt. Der Status der beiden Flags bestimmt die Behandlung der Eingabedaten (x_i und y_i) wie folgt:



Die Behandlung des Ausgabewerts (A) ist ähnlich.

In diesem Fall erweist sich die Verwendung von Flags als sehr komfortabel. Ein Problem, das ansonsten eventuell drei verschiedene Programme erfordert hätte, ist nun sehr sauber in einem gelöst worden. Die Verwendung von Flags kann ein sehr wesentliches Hilfsmittel zur Steigerung der Effizienz Ihrer Programme sein.

Zufallszahlen

Der HP-11C enthält eine sehr komfortable Funktion zur Erzeugung von Pseudo-Zufallszahlen ([RAN#]). Der Begriff „Pseudo-Zufallszahlen“ besagt, daß kein noch so großer Rechner „echte“ Zufallszahlen erzeugen kann. Die meisten Zufallszahlengeneratoren sind so aufgebaut, daß sie nach einer Eingabe eine fest vorgegebene Folge von Programmbefehlen ausführen, und damit ein vorhersehbares Resultat produzieren. So verwendet auch der Zufallszahlengenerator des HP-11C einen vom Benutzer zu speichernden „Seed“ (Startwert), um damit über eine interne Umformung ein im allgemeinen (vom Benutzer) unvorhersehbares Resultat zu produzieren. Das erzeugte Resultat ist immer eine Zahl zwischen 0 und 1 (wobei die Zahl 1 selbst nicht vorkommen kann).

Eine häufige Anwendung von Zufallszahlen liegt im Bereich vom Spielprogrammen. Dies liegt schon darin begründet, daß niemand gegen einen vollständig vorhersagbaren und damit uninteressanten Gegner spielen will. Die U-Bootjagd ist ein gutes Beispiel für ein derartiges Spiel. Dieses Spiel erfordert am Anfang eine zufällige Positionierung des Unterseeboots und danach vom Zufall gesteuerte Ausweichbewegungen.

Dabei ist zu beachten, daß das Programm zwei ganzzahlige Zufallszahlen (x - und y -Koordinaten), die zwischen 0 und 9 einschließlich liegen müssen, benötigt. Der Zufallszahlengenerator erzeugt aber nur Zahlen zwischen 0 und 1 (ausschließlich der 1). Dies bedeutet jedoch keine Einschränkung. Durch Multiplikation der vom Zufallszahlengenerator erzeugten Zahlen mit 10 erhält man Werte im Bereich von 0 bis 10 (ausschließlich der 10). Die Einschränkung auf den ganzzahligen Anteil dieser Ergebnisse liefert die gewünschten ganzen Zahlen zwischen 0 und 9 einschließlich. Das Unterprogramm mit dem Label 9 enthält die notwendigen Befehle.

Die Leichtigkeit, mit der Zufallszahlen in den verschiedensten Intervallen erzeugt werden können, soll noch an einem weiteren Beispiel illustriert werden. Angenommen, es wird eine reelle Zahl in den Grenzen 34,5 und 98,26, einschließlich der unteren und ausschließlich der oberen Grenze, gesucht. Dazu sind Zufallszahlen zwischen 0 und 63,86 ($98,26 - 34,50$) zu erzeugen, auf die dann die Konstante 34,5 addiert wird. Zufallszahlen im Intervall 0–63,86 wiederum erhält man durch Multiplikation der Werte des Zufallszahlengenerators mit 63,86.

Benutzer-definierbare Tasten

Zu den nützlichsten Funktionen des HP-11C gehören die Benutzer-definierbaren Tasten **[A]**, **[B]**, **[C]**, **[D]** und **[E]**. Diese Tasten eignen sich insbesondere für drei spezielle Anwendungsbereiche: 1) Speichern von Daten in bestimmten Registern (s. Seite 213); 2) selektive Ausführung von verschiedenen Routinen innerhalb eines Programms; und 3) selektive Ausführung verschiedener Programme innerhalb des Programmspeichers. Die ersten beiden Anwendungsbereiche werden innerhalb des Finanzprogramms benutzt.

Datenspeicherung

Das Finanzprogramm läßt fünf mögliche Eingaben zu: Zinssatz pro Periode (i), Anzahl der Perioden (n), gegenwärtiger Wert (PV),

Ratenzahlung pro Periode (*PMT*), zukünftiger Wert (*FV*). Da fünf Benutzer-definierbare Tasten verfügbar sind, kann jeder Taste ein Programm, das die Eingabedaten in bestimmte Register lädt, zugeordnet werden. Beispielsweise beginnt das der Taste **C** zugeordnete Programm mit **f** **LBL** **C**, **STO** 3, **R/S**. Bei jedem Auslösen der Taste **C** wird der Wert im Stackregister X in das Register R₃ geladen. Auf diese Weise können bis zu fünf, jeweils einer Taste zugeordnete Werte gespeichert werden. Die Reihenfolge ist dabei beliebig. Sie müssen lediglich den zu speichernden Wert eingeben und die entsprechende Benutzer-definierbare Taste drücken.

Selektion von Routinen

Es ist möglich, daß ein Programm wahlweise mehrere, unterschiedliche Resultate berechnen soll. Die Auswahl, welcher Wert jeweils aktuell zu berechnen ist, kann über die Benutzer-definierbaren Tasten erfolgen. Zum Beispiel wird im Finanzprogramm der Wert *PMT* durch Drücken der Tastenfolge **D** (*PMT* auswählen), **R/S** (Rechnerbeginn) berechnet. Durch Auslösen der Taste **D** teilt der Benutzer dem Rechner mit, daß die mit **LBL** **D** beginnende Routine ausgeführt werden soll. Bei diesem Programm wird in R₄ ein irrelevanter Wert gespeichert und der Rechner auf die zur Berechnung von *PMT* geeignete Adresse positioniert. Der in R₄ gespeicherte Wert ist deshalb ohne Bedeutung, da bei der durch **R/S** ausgelösten Berechnung von *PMT* der Inhalt von R₄ überschrieben wird.

Wie Sie gesehen haben, stellen die Benutzer-definierbaren Tasten ein sehr vielfältig anwendbares und nutzbringendes Hilfsmittel dar.

Fehlerbedingungen

Wenn in einer Berechnung eine unzulässige Operation – z. B. eine Division durch Null – enthalten ist, erscheint in der Anzeige die Meldung **Error** und eine Zahl. Jede Fehlermeldung kann durch Auslösen einer beliebigen Taste gelöscht werden.

Im folgenden sind die möglichen Fehlermeldungen und die zugehörigen Fehlerbedingungen gelistet.

Error 0: Unzulässige mathematische Operation

Unzulässiges Argument in einer der folgenden mathematischen Routinen:

$\boxed{\div}$, wo $x = 0$.

$\boxed{y^x}$, wo $y = 0$ und $x \leq 0$, oder $y < 0$ und x nicht ganzzahlig.

$\boxed{\sqrt{x}}$, wo $x < 0$.

$\boxed{1/x}$, wo $x = 0$.

$\boxed{\text{LOG}}$, wo $x \leq 0$.

$\boxed{\text{LN}}$, wo $x \leq 0$.

$\boxed{\text{SIN}^{-1}}$, wo $|x| > 1$.

$\boxed{\text{COS}^{-1}}$, wo $|x| > 1$.

$\boxed{\text{STO}} \boxed{\div}$, wo $x = 0$.

$\boxed{\Delta \%}$, wo der Wert im Y-Register Null ist.

$\boxed{\text{HYP}^{-1}} \boxed{\text{COS}}$, wo $|x| < 1$.

$\boxed{\text{HYP}^{-1}} \boxed{\text{TAN}}$, wo $|x| > 1$.

$\boxed{C_{y,x}}$, wo 1. x oder y nicht ganzzahlig; 2. x oder y kleiner Null;
3. $x > y$; 4. $x \geq 10^{10}$.

$\boxed{P_{y,x}}$, wo 1. x oder y nicht ganzzahlig; 2. x oder y kleiner Null;
3. $x > y$; 4. $x \geq 10^{10}$.

Error 1: Speicherregister-Overflow

Überlauf eines Speicherregisters (außer bei $\boxed{\Sigma +}$, $\boxed{\Sigma -}$). Die Zahl im Speicherregister hätte eine Wertigkeit größer als $9.999999999 \times 10^{99}$.

Error 2: Unzulässige statistische Operation

$\boxed{\bar{x}}$ wo $n = 0$.

\boxed{s} wo $n \leq 1$.

\boxed{r} wo $n \leq 1$.

$\boxed{\hat{y}, r}$ wo $n \leq 1$.

$\boxed{\text{L.R.}}$ wo $n \leq 1$.

Bemerkung: Die Meldung **Error 2** erscheint auch, wenn bei der Berechnung einer der folgenden Formeln eine Division durch Null oder die Quadratwurzel einer negativen Zahl auftreten würde:

$$s_x = \sqrt{\frac{M}{n(n-1)}} \quad s_y = \sqrt{\frac{N}{n(n-1)}} \quad r = \frac{P}{\sqrt{M \cdot N}}$$

$$A = \frac{P}{M} \quad B = \frac{M \sum y - P \sum x}{n \cdot M} \quad (A \text{ und } B \text{ sind die bei der Operation [L.R.] berechneten Werte, d. h. } y = Ax + B.)$$

$$\hat{y} = \frac{M \sum y + P(n \cdot x - \sum x)}{n \cdot M}$$

wobei

$$M = n \sum x^2 - (\sum x)^2$$

$$N = n \sum y^2 - (\sum y)^2$$

$$P = n \sum xy - \sum x \sum y$$

Error 3: Unzulässige Registeradresse

Das durch die angegebene Zahl adressierte Speicherregister ist momentan in Programmspeicher umgewandelt oder existiert nicht.

Error 4: Unzulässige Zeilennummer oder unzulässiger Labelaufruf

Die durch die Zeilennummer adressierte Programmzeile ist momentan nicht belegt oder existiert nicht (Zeilennummer > 203); Versuch mehr als 203 Zeilen in den Programmspeicher zu laden; das aufgerufene Label existiert nicht.

Error 5: Unterprogrammverschachtelung zu tief

Eine Unterprogrammverschachtelung enthält mehr als vier Verschachtelungsebenen.

Error 6: Unzulässiger Flag

Die hinter einem Flag-Befehl angegebene Zahl ist nicht 0 oder 1.

Error 9: Service

Siehe Seite 233, Funktionsprüfung.

Pr Error

Der Permanentenspeicher wurde wegen Stromausfall gelöscht.

Stack Lift und LAST X

Bei der Konstruktion des HP-11C wurde auf eine möglichst „natürliche“ Arbeitsweise Wert gelegt. Wie Sie beim Durcharbeiten dieses Handbuchs gemerkt haben werden, brauchen Sie nur selten über die Abläufe im automatischen Speicherstack nachzudenken – Sie lösen jedes Problem wie mit Papier und Bleistift, immer eine Operation nach der anderen.

Es kann jedoch insbesondere bei der Programmierung des HP-11C vorkommen, daß es von Interesse ist, die Auswirkungen einer bestimmten Operation auf den Stack zu kennen. Die folgenden Erläuterungen sollen Ihnen dabei helfen.

Abschluß der Zifferneingabe

Die meisten Operationen des Rechners, sei es nun bei der Ausführung einer Anweisung innerhalb eines Programms oder über das Tastenfeld, beenden die Eingabe von Ziffern. Dies besagt, daß der Rechner jede, nach Abschluß einer dieser Operationen eingegebene Ziffer als Teil einer neuen Zahl auffaßt.

Stack Lift

Die Operationen des Rechners lassen sich je nach ihrer Auswirkung auf den Stack Lift in drei Klassen einteilen. Dies sind Stack-sperrende (*Stack-disabling*) Operationen, Stack-freigebende (*Stack-enabling*) Operationen und *neutrale* Operationen.

Disabling Operationen

Der Rechner enthält vier Stack-disabling Operationen.* Bei diesen Operationen wird der Stack Lift gesperrt, so daß eine nach dieser Operation erfolgende Eingabe den momentanen Inhalt des angezeigten X-Registers überschreibt und der Stack nicht nach oben verschoben wird. Die Stack-disabling Operationen sind:

ENTER CLX $\Sigma +$ $\Sigma -$

* Siehe Fußnote auf Seite 29.

Enabling Operationen

Die meisten Operationen auf dem Tastenfeld des Rechners, einschließlich der mathematischen Funktionen einer und zweier Variabler wie x^2 und \times , sind Stack-enabling Operationen. Diese Operationen geben den Stack frei, so daß bei Eingabe einer neuen Zahl der Stackinhalt nach oben verschoben wird. Es ist zu beachten, daß das Umschalten von Program auf Run Mode und das Ein- und Ausschalten des Rechners als Stack-enabling Operationen wirken.

T →				
Z →				
Y →		4,0000	4,0000	53,1301
X →	4	4,0000	3	5,0000
Tasten →	4	ENTER Stack gesperrt	3 Kein Stack Lift	g →P Freigabe d. Stacks

T →				
Z →				
Y →	53,1301	53,1301	53,1301	53,1301
X →	0,0000	7	1,0000	9
Tasten →	g CLV Stack gesperrt	7 Kein Stack Lift	Σ+ Stack gesperrt	9 Kein Stack Lift

Neutrale Operationen

Einige Operationen, wie **CHS** und **FIX**, sind neutral, d.h. der bestehende Status des Stacks wird durch sie nicht verändert. Wenn beispielsweise der Stack Lift mit **ENTER** gesperrt worden ist, so bewirkt das Drücken von **f** **FIX** *n* und die Eingabe einer neuen Zahl ein Überschreiben des Inhalts des X-Registers und keine Verschiebung des Stacks. Im anderen Fall, bei einer Freigabe des Stacks zum Beispiel durch x^2 , bewirkt die Ausführung einer **FIX** Anweisung, gefolgt von der Eingabe einer Ziffernsequenz, eine Verschiebung des Stacks nach oben.

Die folgenden Operationen sind neutral:

FIX
SCI

GTO **·** *nnn*
BST

CLEAR **PREFIX**
CLEAR **REG**

ENG
DEG
RAD
GRAD
DSP I

SST
MEM
USER

(Im Run Mode kann durch **SST** eine Stack-freigebende Anweisung ausgeführt werden.)

CLEAR Σ
CHS*
PREFIX
R/S
PSE
P/R

LAST X

Bei den folgenden Operationen wird der alte Inhalt des X-Registers in das LAST X Register gerettet.

-
+
 \times
 \div
 \rightarrow H.MS
 \rightarrow H
ABS
 \rightarrow R
 \rightarrow D

$\Sigma+$
 $\Sigma-$
%
 $\Delta\%$
 \hat{y}, r
 $x!$
FRAC
INT
LN

e^x
LOG
 10^x
SIN
 SIN^{-1}
COS
 COS^{-1}
TAN
 TAN^{-1}

\sqrt{x}
 x^2
 $1/x$
 y^x
 \rightarrow R
 \rightarrow P
 $P_{y,x}$
 $C_{y,x}$

RND
HYP SIN
HYP COS
HYP TAN
 HYP^{-1} SIN
 HYP^{-1} COS
 HYP^{-1} TAN

* **CHS** ist neutral während der Eingabe von Ziffern einer Zahl über die Tastatur; wie bei 1, 2, 3, **CHS** zur Eingabe von -123, oder bei 1, 2, 3, **EEX** 6 **CHS** zur Eingabe von 123×10^{-6} . Ansonsten gibt **CHS** den Stack frei.

Arbeitsweise der automatischen Speicherumwandlung

Umwandlung von Speicherregistern in Programmspeicher

Die im HP-11C integrierte automatische Speicherumwandlung bietet Ihnen die Möglichkeit der an den jeweiligen Bedarf angepaßten Konvertierung von Speicherregistern in Programmzeilen. Als Voreinstellung stehen Ihnen für die Programmierung 63 Zeilen Programmspeicher und 20 Speicherregister (plus dem Indexregister, siehe Abschnitt 9) zur Verfügung. Mit 0 bis 63 Programmbefehlen im Programmspeicher wäre die folgende Speicheraufteilung gegeben:

SPICHERREGISTER

Permanent	Wandelbar	Wandelbar
R _i <input type="text"/>	R ₀ <input type="text"/> n	R ₀ <input type="text"/>
	R ₁ <input type="text"/> Σx	R ₁ <input type="text"/>
	R ₂ <input type="text"/> Σx^2	R ₂ <input type="text"/>
	R ₃ <input type="text"/> Σy	R ₃ <input type="text"/>
	R ₄ <input type="text"/> Σy^2	R ₄ <input type="text"/>
	R ₅ <input type="text"/> Σxy	R ₅ <input type="text"/>
	R ₆ <input type="text"/>	R ₆ <input type="text"/>
	R ₇ <input type="text"/>	R ₇ <input type="text"/>
	R ₈ <input type="text"/>	R ₈ <input type="text"/>
	R ₉ <input type="text"/>	R ₉ <input type="text"/>

PROGRAMMSPEICHER

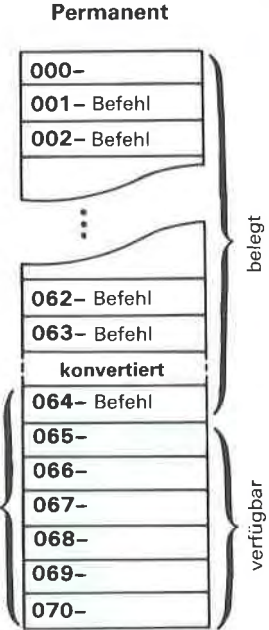
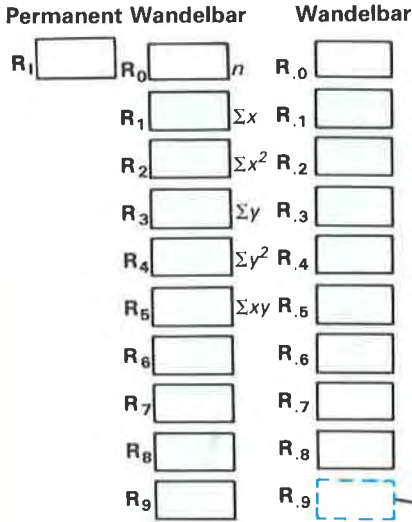
Permanent
000-
001-
002-
⋮
062-
063-

keine konvertierten Zeilen

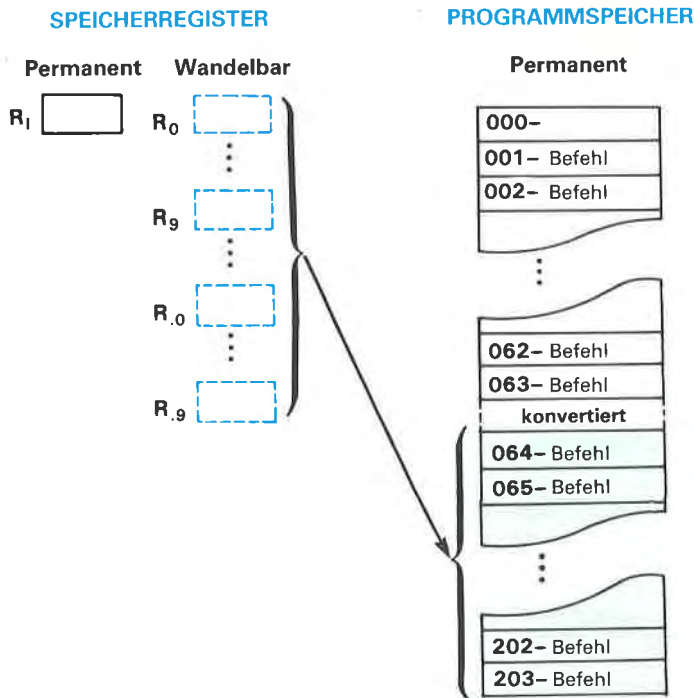
Bei der Eingabe einer 64. Programmzeile wird das Speicherregister R_9 automatisch in 7 zusätzliche Zeilen Programmspeicher umgewandelt. Damit ergibt sich die Speicheraufteilung:

SPEICHERREGISTER

PROGRAMMSPEICHER



Bei einer vollständigen Ausnutzung der 203 Zeilen des Programmspeichers, erhält man schließlich die folgende Speicheraufteilung:



Dabei ist zu beachten, daß anstelle der ursprünglich 21 Speicherregister (R₀ bis R₉, R₀ bis R₉ und R₁) jetzt nur noch das nicht konvertierbare Indexregister verfügbar ist. Die Speicherregister R₀ bis R₉ und R₀ bis R₉ sind in Programmspeicher umgewandelt worden, und zwar jeweils 7 Zeilen pro Register. Die Tabelle auf Seite 76 enthält die Zuordnungen der Programmzeilen und der entsprechenden Speicherregister.

Bei der Umwandlung wird wie folgt vorgegangen. Jedesmal, wenn der aktuell verfügbare Programmspeicher belegt ist, wird durch die Eingabe eines weiteren Programmbefehls das nächste Speicherregister in sieben zusätzliche Zeilen Programmspeicher umgewan-

delt. Es sei beispielsweise unterstellt, daß im Programmspeicher 70 Zeilen belegt sind. Dann wird bei der Eingabe eines Befehls in Zeile 71 das Speicherregister R_8 in sieben weitere Programmzeilen (Zeile 71 bis 77) konvertiert. Dieser Prozeß kann solange wiederholt werden, bis alle 20 konvertierbaren Speicherregister umgewandelt sind.

Bemerkung: Der HP-11C benutzt bei der Konvertierung von Speicherregistern in Programmspeicher die umgekehrte numerische Reihenfolge, d. h. zunächst R_9 bis R_0 und R_9 bis R_0 . Aus diesem Grund ist es zu empfehlen, bei der Programmierung von $\boxed{\text{STO}}$ und $\boxed{\text{RCL}}$ Operationen die Register in der umgekehrten Reihenfolge zu verwenden, d. h. mit Register R_0 zu beginnen. Bei dieser Vorgehensweise wird die Gefahr verringert, mit $\boxed{\text{STO}}$ oder $\boxed{\text{RCL}}$ Befehlen auf Register zuzugreifen, die zwischenzeitlich in Programmspeicher umgewandelt worden sind. Es ist des Weiteren zu beachten, daß bei der Konvertierung die in den Speicherregistern befindlichen Daten verloren gehen.

Umwandlung von Programmspeicher in Speicherregister

Durch Drücken von $\boxed{\text{f}}$ CLEAR $\boxed{\text{PRGM}}$ im Program Mode wird der gesamte konvertierte Programmspeicher (Zeile 064 bis 203) wieder in die Register R_0 bis R_9 zurückgewandelt. Es besteht jedoch auch die Möglichkeit, durch Löschen einzelner Zeilen Teile des Programmspeichers wieder in Speicherregister zurückzuwandeln, ohne den gesamten Programmspeicher löschen zu müssen. (Siehe Seite 105, Löschen von Programmzeilen.)

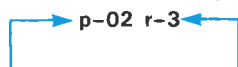
Verwendung von $\boxed{\text{MEM}}$

Die $\boxed{\text{MEM}}$ (*memory*) Funktion des Rechners dient zur Anzeige der momentanen Speicheraufteilung. Diese Funktion arbeitet sowohl Program als auch im Run Mode. Es sei beispielsweise unterstellt, daß 44 Zeilen des Programmspeichers belegt sind. Dann bewirkt das Auslösen von $\boxed{\text{g}}$ $\boxed{\text{MEM}}$ die folgende Anzeige:

Noch verfügbare Zeilen vor der nächsten automatischen Konvertierung eines Speicherregisters.

Nummer des nächsten zu konvertierenden Speicherregisters.

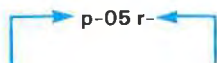
Bei einer Belegung von 173 Programmzeilen bewirkt $\boxed{9}$ $\boxed{\text{MEM}}$ die Anzeige:



Noch verfügbare Zeilen vor der nächsten automatischen Konvertierung eines Speicherregisters.

Nummer des nächsten zu konvertierenden Speicherregisters.

Wenn 198 Zeilen des Programmspeichers belegt sind, erscheint die Anzeige:



Noch verfügbare Zeilen bevor der gesamte Programmspeicher belegt ist.

Kein weiteres Speicherregister kann in Programmspeicher konvertiert werden.

Es ist zu beachten, daß das Indexregister auf Grund seiner Auslegung als permanentes Speicherregister mit Sonderfunktionen von der $\boxed{\text{MEM}}$ Operation nicht erfaßt wird.

Bemerkung: Bei der Ausführung der statistischen Funktionen werden die Register R_0 bis R_6 benötigt. Die Umwandlung eines dieser Register in Programmspeicher bewirkt bei der Ausführung einer statistischen Funktion die Ausgabe der Fehlermeldung **Error 3**.

Wartung und Pflege

Batterien

Die Stromversorgung des HP-11C erfolgt über drei Batterien. Bei „normalem“ Gebrauch des Rechners wird für einen Satz Alkalibatterien eine Betriebszeit von ungefähr sechs Monaten erreicht. Zusammen mit dem Rechner wird ein Satz Alkalibatterien geliefert; es können jedoch auch Silberoxidbatterien (die mindestens doppelt solange halten sollten) verwendet werden.

Drei frische Alkalibatterien liefern eine reine Programmrechenzeit von mindestens 80 Stunden (wobei anzumerken ist, daß die Programmausführung die Form des Rechnerbetriebs mit dem höchsten Stromverbrauch ist*). Bei der Verwendung von Silberoxidbatterien wird eine reine Programmrechenzeit von mindestens 180 Stunden erreicht. Diese Werte beziehen sich, wie bereits erwähnt, auf die aufwendigste Betriebsart, nämlich die Programmausführung; bei allen anderen Betriebsarten ist der Stromverbrauch weitaus geringer. Wenn nur das Display eingeschaltet ist, d. h. es werden weder die Tasten gedrückt noch Programme ausgeführt, ist der Stromverbrauch minimal.

Solange der Rechner ausgeschaltet ist, bleibt der Inhalt des PermanentSpeichers über einen Zeitraum erhalten, der dem der Haltbarkeit der Batterien außerhalb des Rechners entspricht – mindestens anderthalb Jahre bei Alkalibatterien bzw. mindestens zwei Jahre bei Silberoxidbatterien.

Die aktuelle Lebensdauer eines Batteriensatzes hängt davon ab, wie oft Sie den Rechner benutzen, ob Sie ihn mehr zur Ausführung von Programmen oder für manuelle Berechnungen verwenden, und mit welchen Funktionen Sie arbeiten.*

Warnung

Versuchen Sie nicht, die Batterien nachzuladen, lagern Sie sie nicht in der Nähe großer Wärmeeinstrahlung, und werfen Sie sie nicht in offenes Feuer. Die Batterien können in diesen Fällen auslaufen bzw. explodieren.

* Der Stromverbrauch des HP-11C hängt von der jeweiligen Betriebsart ab: ausgeschaltet (unter Erhaltung des PermanentSpeichers); betriebsbereit (nur das Display ist eingeschaltet); rechnend (ein Programm läuft ab, eine Berechnung wird ausgeführt, oder eine Taste ist gedrückt). Solange der Rechner eingeschaltet ist, besteht die typische Auslastung aus einer Mischung der beiden Betriebsarten „betriebsbereit“ und „rechnend“. Daher hängt die aktuelle Lebensdauer der Batterien davon ab, wie lange der Rechner in jeder Betriebsart benutzt wurde.

Sowohl die zusammen mit dem Rechner gelieferten Batterien, als auch die im folgenden empfohlenen Ersatzbatterien können *nicht* nachgeladen werden.

Die folgenden Batterien werden als Ersatzbatterien für Ihren HP-11C empfohlen:

Alkaline Batterien

Eveready A 76
UCAR A 76
RAY-O-VAC RW82
National oder Panasonic LR44

Silberoxyd-Batterien

Eveready 357
UCAR 357
RAY-O-VAC RS76 oder RW42
Duracell MS76

Anzeichen von Spannungsabfall

Bei nachlassender Batteriespannung ist die Funktionsweise des Display's des Rechners beeinträchtigt; einzelne Zahlen werden u. U. nicht mehr korrekt angezeigt. Es kann beispielsweise vorkommen, daß einzelne Ziffern nur noch schwach oder gar nicht mehr angezeigt werden, bzw. daß einzelne Segmente in der Darstellung der Ziffer fehlen. Bei Auftreten dieser Symptome ist der Rechner auszuschalten, und die Batterien sind wie im folgenden beschrieben auszuwechseln; der Inhalt des PermanentSpeichers bleibt noch für einige Tage erhalten.

Einbau neuer Batterien

Der Inhalt des PermanentSpeichers bleibt noch für einige Sekunden erhalten, wenn keine Batterien mehr im Rechner sind. Dies sollte normalerweise genügen, um die alten Batterien durch neue zu ersetzen. Wenn längere Zeit keine Batterien im Rechner sind, geht der Inhalt des PermanentSpeichers verloren.

Um neue Batterien einzubauen, verfahren Sie wie folgt:

1. Überzeugen Sie sich davon, daß der Rechner ausgeschaltet ist.
2. Halten Sie den Rechner wie hier gezeigt, drücken Sie auf den Batteriefachdeckel, bis er sich leicht öffnen läßt.



3. Greifen Sie den Batteriefachdeckel an der Außenkante und ziehen ihn aus dem Rechner heraus.



Bemerkung: Achten Sie während der beiden nächsten Schritte darauf, daß keine Tasten gedrückt werden, solange sich keine Batterien im Rechner befinden. Wenn dies geschieht, kann der Inhalt des PermanentSpeichers verloren gehen.



4. Drehen Sie den Rechner um und schütteln ihn leicht, damit Ihnen die Batterien in die Hand fallen.



Warnung

Erneuern Sie im nächsten Schritt *alle drei* Batterien. Wenn Sie nur eine oder zwei Batterien austauschen, besteht die Gefahr, daß eine der alten Batterien ausläuft. Achten Sie des weiteren darauf, daß die Batterien nicht verkehrt herum eingesetzt werden. In diesem Fall kann der Inhalt des Perma-nentspeichers verloren gehen.

5. Setzen Sie drei neue Batterien in das Batteriefach ein. Sie müssen so eingelegt werden, daß die mit + gekennzeichnete Seite zum nächsten Gummifuß zeigt (s. Illustration auf dem Rechnergehäuse).



6. Schieben Sie den Batteriefachdeckel in die Aussparung des Rechnergehäuses.



8. Schieben Sie den Batteriedeckel so weit ein, bis er mit dem Gehäuse abschließt und drücken Sie anschließend auf den Deckel, bis er leicht einrastet.

9. Drücken Sie **ON**, um den Rechner einzuschalten. Wenn der Permanentspeicher aus irgend einem Grund gelöscht wurde (der Inhalt also verloren ist), erscheint im Display die Meldung **Pr Error**. Diese Meldung kann durch Drücken einer beliebigen Taste gelöscht werden.



Funktionsprüfung

Wenn der Rechner nicht eingeschaltet werden kann oder anderweitig nicht korrekt zu arbeiten scheint, so verfahren Sie wie folgt:

1. Wenn der Rechner nicht angeschaltet werden kann oder nicht auf Tastenfolgen reagiert, bauen Sie neue Batterien ein. Achten Sie dabei darauf, daß die Batterien richtig in das Batteriefach eingelegt werden: Die mit + markierten Seiten müssen alle zum nächsten Gummifuß zeigen. Wenn der Rechner mit den neuen Batterien immer noch nicht eingeschaltet werden kann, muß er zur Reparatur eingesandt werden.

2. Halten Sie bei ausgeschaltetem Rechner die $\boxed{\times}$ Taste gedrückt und drücken Sie zusätzlich die $\boxed{\text{ON}}$ Taste.
3. Lassen Sie die $\boxed{\text{ON}}$ Taste und danach die $\boxed{\times}$ Taste wieder los. Durch diese Tastenfolge wird eine vollständige Überprüfung der elektronischen Schaltkreise des Rechners ausgelöst. Bei korrekter Arbeitsweise des Rechners sollte nach einer Zeitspanne von 15 Sekunden (in der das Wort *running* im Display blinkt) die Anzeige **-8,8,8,8,8,8,8,8,8,8**, erscheinen. Zusätzlich sollten sämtliche Statusanzeigen an sein.* Wenn das Display leer bleibt, die Meldung **Error 9** oder ein sonstiges (inkorrektes) Resultat anzeigt, deutet dies auf einen Gerätefehler hin.**

Bemerkung: Überprüfungen der Rechnerelektronik werden auch durch Drücken der $\boxed{\text{ON}}$ Taste bei gleichzeitig gedrückter $\boxed{+}$ oder $\boxed{-}$ Taste ausgelöst.*** Diese Tests sind im Rechner integriert, um während der Herstellung und Reparatur die korrekte Arbeitsweise des Geräts kontrollieren zu können.

Wenn sich Ihr Verdacht auf fehlerhafte Arbeitsweise des Rechners aufgrund einer korrekten Anzeige in Schritt 3 nicht bestätigt hat, so besteht die Möglichkeit, daß Sie bei der Bedienung des Rechners einen Fehler gemacht haben. In diesen Fällen empfehlen wir, den auf Ihr spezielles Problem zutreffenden Abschnitt dieses Handbuchs noch einmal durchzuarbeiten. Wenn Sie danach immer noch Schwierigkeiten feststellen, benachrichtigen Sie Hewlett-Packard unter einer der im Abschnitt „Service“ (Seite 237) gelisteten Adressen oder Telefonnummern.

* Unter den Statusanzeigen, die am Ende dieses Tests eingeschaltet werden, sind einige Sonderanzeigen, die der HP-11C normalerweise nicht benutzt.

** Wenn der Rechner als Ergebnis des $\boxed{\times}/\boxed{\text{ON}}$ Tests oder des $\boxed{+}/\boxed{\text{ON}}$ Tests die Meldung **Error 9** anzeigt, und Sie den Rechner trotzdem weiterhin benutzen wollen, empfiehlt es sich, den Permanentenspeicher wie Seite 20 beschreiben, zu löschen.

*** Durch die $\boxed{+}/\boxed{\text{ON}}$ Kombination wird eine dem zuvor beschriebenen Test ähnliche Überprüfung ausgelöst, die jedoch nicht von selbst abbricht. Der Test wird innerhalb von 15 Sekunden nach Auslösen einer beliebigen Taste beendet. Durch die $\boxed{-}/\boxed{\text{ON}}$ Kombination wird eine Überprüfung der Tastatur und des Display's ausgelöst. Nach dem Loslassen der $\boxed{\text{ON}}$ Taste werden bestimmte Segmente im Display aufleuchten. Zur Fortsetzung des Tests müssen alle Tasten des Tastenfelds gedrückt werden, und zwar von links nach rechts, beginnend mit der ersten Reihe. Beim Drücken der einzelnen Tasten werden jeweils einzelne Segmente des Display's aufleuchten. Wenn der Rechner korrekt arbeitet (und alle Tasten in der richtigen Reihenfolge gedrückt werden), wird nach dem Drücken der letzten Taste die Zahl **11** angezeigt.

Gewährleistung

Hewlett-Packard gewährleistet, daß der Rechner frei von Material- und Verarbeitungsfehlern ist, und verpflichtet sich, etwaige fehlerhafte Teile kostenlos instandzusetzen oder auszutauschen, wenn der Rechner – direkt oder über einen autorisierten Hewlett-Packard-Vertragshändler – an Hewlett-Packard eingeschickt wird. Die Gewährleistungsfrist beträgt 12 Monate ab Verkaufsdatum.

Weitergehende Ansprüche, insbesondere auf Ersatz von Folgeschäden, können nicht geltend gemacht werden. Schäden, die durch unsachgemäße Veränderungen des Rechners durch Dritte zurückzuführen sind, werden von dieser Gewährleistung nicht umfaßt.

Die Gewährleistung gilt nur in Verbindung mit entweder

- a) dem von einem Hewlett-Packard-Vertragshändler ausgestellten Kaufbeleg und der vollständig ausgefüllten, von diesem Hewlett-Packard-Vertragshändler unterschriebenen Servicekarte oder
- b) der Originalrechnung von Hewlett-Packard.

Die Ansprüche des Käufers aus dem Kaufvertrag bleiben von dieser Gewährleistungsregelung unberührt.

(Die **ENTER** Taste sollte sowohl zusammen mit den Tasten der dritten Reihe als auch mit den Tasten der vierten Reihe gedrückt werden.) Wenn der Rechner nicht fehlerfrei arbeitet (oder eine Taste außerhalb der Reihenfolge gedrückt wurde), erscheint im Display die Meldung **Error 9**. Beachten Sie dabei, daß der Rechner nicht notwendig reparaturbedürftig ist, wenn diese Fehlermeldung durch das Drücken einer Taste in der falschen Reihenfolge ausgelöst wird. Dieser Test kann durch Drücken einer beliebigen Taste außerhalb der Reihenfolge (wodurch natürlich eine **Error 9** Meldung induziert wird) beendet werden. Sowohl die Anzeige **Error 9** als auch die Anzeige **11** kann durch Drücken einer beliebigen Taste gelöscht werden.

Änderungsverpflichtung

Geräte werden in der zum Zeitpunkt der Herstellung gültigen Version gebaut. Hewlett-Packard übernimmt keine Verpflichtung, früher verkaufte Geräte zu verändern oder an eine neue Version anzupassen.

Gewährleistungsinformation

Wenn Sie Fragen zur Gewährleistung haben, wenden Sie sich bitte

- In USA an:

Hewlett-Packard
1000 N.E. Circle Blvd.
Corvallis, OR 97330
Tel.: (503) 758-1010

- In Europa an:

Ihren Vertragshändler oder
die nächste HP-Verkaufsniederlassung

- In anderen Ländern:

Hewlett-Packard International
3495 Deer Creek Rd.
Palo Alto, California 94304
USA
Tel.: (415) 856-1501

Hinweis: Senden Sie bitte an diese Adresse keine Geräte zur Reparatur.

Service

Hewlett-Packard unterhält in den meisten Ländern der Welt Service Center. Sie können Ihr Gerät, wenn nötig, jederzeit bei einem Hewlett-Packard Service Center zur Reparatur geben, gleichgültig, ob das Gerät sich noch in der Garantiezeit befindet oder nicht. Nach der Garantiezeit von einem Jahr werden Reparaturkosten berechnet.

Hewlett-Packard Rechner werden normalerweise innerhalb von 5 Arbeitstagen nach Eingang bei einem Service Center instandgesetzt und zurückgeschickt. Dies ist ein Durchschnittswert und kann je nach Jahreszeit und Arbeitsbelastung des Service Center unterschiedlich sein. Die Gesamtzeit, in der Sie auf Ihr Gerät verzichten müssen, hängt in erster Linie von der Versanddauer ab.

Reparatur in den USA

Das Hewlett-Packard Service Center für Taschenrechner und Tischcomputer befindet sich in Corvallis, Oregon:

Hewlett-Packard Company
Corvallis Division Service Department
P.O. Box 999/1000 N.E. Circle Blvd.
Corvallis, Oregon 97330, USA
Tel. (503) 757-2000

Reparatur in Europa

Service-Niederlassungen werden in folgenden Ländern unterhalten. In nicht aufgeführten Ländern wenden Sie sich bitte an den Händler, bei dem Sie das Gerät gekauft haben.

BELGIEN

HEWLETT-PACKARD BELGIEN SA/NU
Boulevard de la Woluwe 100
Woluwelaan
B-1200 Brüssel
Tel.: (2) 762 32 00

DÄNEMARK

HEWLETT-PACKARD A/S
Datavej 52
DK-3460 Birkerød (Kopenhagen)
Tel.: (02) 81 66 40

DEUTSCHLAND

HEWLETT-PACKARD GmbH
Kleinrechner-Service
Vertriebszentrale
Berner Straße 117
Postfach 560 140
D-6000 Frankfurt 56
Tel.: (0611) 5004-1

FINNLAND

HEWLETT-PACKARD OY
Revontulentie 7
02100 Espoo 10 (Helsinki)
Tel.: (90) 455 02 11

FRANKREICH

HEWLETT-PACKARD FRANCE
Division Informatique Personnelle
S.A.V. Calculateurs de poche
F-91947 Les Ulis Cedex
Tel.: (6) 9 07 78 25

GROSSBRITANNIEN

HEWLETT-PACKARD Ltd
King Street Lane
GB-Winnersh, Wokingham
Berkshire RG11 5AR
Tel. (734) 78 47 74

HOLLAND

HEWLETT-PACKARD NEDERLAND B.V.,
Van Heuven Goedhartlaan 121
1181 KK Amstelveen (Amsterdam)
P.O. Box 667
Tel.: (020) 47 20 21

ITALIEN

HEWLETT-PACKARD S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
I-20063 Cernusco Sul Naviglio (Milan)
Tel.: (2) 90 36 91

NORWEGEN

HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Oesterndalen 18
N-1345 Oesteraas (Oslo)
Tel.: (2) 17 11 80

ÖSTERREICH

HEWLETT-PACKARD GmbH
Kleinrechner-Service
Wagramerstr.-Lieblgasse
A-1220 Wien
Tel.: (222) 35 16 20

OST-EUROPA

Schreiben Sie an die in Österreich angegebene Adresse

SCHWEDEN

HEWLETT-PACKARD SVERIGE AB
Enighetsvägen 3
Box 205 02
S-161 Bromma 20 (Stockholm)
Tel.: (8) 730 05 50

SCHWEIZ

HEWLETT-PACKARD (SCHWEIZ) AG
Kleinrechner-Service
Allmend 2
CH-8967 Widen
Tel.: (057) 50111

SPANIEN

HEWLETT-PACKARD ESPAÑOLA S.A.
Calle Jerez 3
E-Madrid 16
Tel.: (1) 458 2600

Internationaler Service

Nicht alle Hewlett-Packard Service Center bieten für alle HP-Rechnermodelle den Service an. Wenn Sie aber Ihr Gerät von einem Hewlett-Packard Vertragshändler gekauft haben, können Sie sicher sein, daß in dem betreffenden Land der Service durchgeführt wird.

Außerhalb des Landes, in dem Sie Ihren Rechner gekauft haben, können Sie von dem ansässigen Hewlett-Packard Service Center erfahren, ob Ihr Gerät dort repariert werden kann. Falls nicht, senden Sie bitte das Gerät an die oben genannte Adresse des Service Center in USA. Eine Liste der Service Center in anderen Ländern können Sie bei der gleichen Adresse anfordern.

Versand- und Verzollungskosten sind von Ihnen zu übernehmen.

Reparaturkosten

Für Reparaturen nach der Garantiezeit wird eine Reparaturkostenpauschale erhoben. Diese schließt sämtliche Arbeiten und Materialien ein. In USA wird auf die Pauschale die entsprechende Umsatzsteuer des Kunden berechnet. In Europa unterliegt die Pauschale der Mehrwertsteuer oder ähnlichen Steuern. Sämtliche Steuern werden auf der Rechnung getrennt ausgewiesen.

Die Reparaturkostenpauschale deckt nicht die Reparatur von Geräten, die durch Gewalteinwirkung oder Fehlbedienung zerstört wurden. In diesem Fall werden die Reparaturkosten individuell nach Zeit- und Materialbedarf festgesetzt.

Garantie bei Reparaturen

Für Reparaturen außerhalb der Garantiezeit leistet Hewlett-Packard eine Garantie von 90 Tagen ab Reparaturdatum bezüglich Material- und Bearbeitungsfehlern.

Versandanweisungen

Wenn Ihr Gerät repariert werden muß, senden Sie es bitte mit folgenden Unterlagen ein:

- Vollständig ausgefüllte Servicekarte einschließlich Beschreibung des Fehlers.
- Rechnung oder anderer Kaufbeleg, wenn die Ein-Jahres-Garantiezeit noch nicht abgelaufen ist.

Das Gerät, die Servicekarte und eine kurze Beschreibung des Fehlers sowie (wenn nötig) der Kaufnachweis sollten in der Originalverpackung oder einer adäquaten Schutzverpackung versandt werden, um Transportschäden zu vermeiden. Solche Transportschäden werden durch die Einjahresgarantie nicht abgedeckt; der Versand zum Service Center erfolgt auf Ihre Gefahr. Das verpackte Gerät sollte zur nächsten Hewlett-Packard Verkaufsniederlassung oder zum nächsten Service Center geschickt werden. Ihr Händler wird Ihnen dabei helfen. Wenn Sie sich außerhalb des Landes befinden, in dem Sie Ihr Gerät ursprünglich gekauft haben, so verfahren Sie entsprechend dem obigen Abschnitt: „Internationaler Service“.

Versandkosten bis zur Auslieferung beim Hewlett-Packard Service Center gehen zu Ihren Lasten, gleichgültig, ob sich das Gerät noch in der Garanzzeit befindet oder nicht.

Nach Ausführung von Garantiearbeiten sendet das Service Center das Gerät frachtfrei an Sie zurück. Bei Reparaturen nach der Garanzzeit in USA oder anderen Ländern wird das Gerät unter Berechnung der Versandkosten zurückgeschickt.

Sonstiges

Serviceverträge werden nicht angeboten. Ausführung und Design des Rechners sind geistiges Eigentum von Hewlett-Packard; Servicehandbücher werden an Kunden nicht abgegeben.

Sollten weitere servicebezogene Fragen auftreten, so rufen Sie das nächstgelegene Hewlett-Packard Service Center an.

Benutzerberatung

Sollten beim Einsatz Ihres Rechners in bestimmten Anwendungsfällen Fragen auftauchen, so rufen Sie einfach unsere Kundenberatung an (siehe Verzeichnis der Niederlassungen) oder schreiben direkt an:

HEWLETT-PACKARD GmbH

Vertriebszentrale Frankfurt
Berner Straße 117
Postfach 560 140
D-6000 Frankfurt 56

Bestellinformationen

Für Informationen über das Bestellen von Rechnern oder Zubehör – einschließlich Händlerverzeichnis, Preisen, Auslieferung usw. – rufen Sie bitte (0611) 5004-1 an.

Temperaturbereich

- Betrieb: 0° bis 55°C (32° bis 131°F)
- Lagerung: -40° bis 65°C (-40° bis 149°F)

Index Programmiertechniken

Die folgenden Seiten enthalten eine kurze Zusammenstellung der wichtigsten Funktionen und Praktiken aus Teil III, Programmierte Problemlösungen, dieses Handbuchs. Der Index ist so aufgebaut, daß hinter jedem Eintrag die Seitennummern (in **Fettbuchstaben**) und die Nummern der entsprechenden Programmzeilen (in Normalschrift), die die jeweilige Technik illustrieren, stehen.

Vergleichsabfragen: **156**, 037; **157**, 060, 065; **160**, 055; **165**, 026; **171**, 075; **188**, 010; **196**, 023, 058; **197**, 069, 083, 087, 115.

[DSE]: **150**, 042, **156**, 051.

Flags: **143**, 065, 071; **144**, 087, 095; **164**, 010, 011; **165**, 014, 017, 020, 023, 040, 054, 057; **183**, 008, 012, 016, 041; **188**, 007, 009, 011; **189**, 072; **195**, 004, 008; **196**, 021, 046, 050; **197**, 107.

[GTO] **[I]**: **165**, 012.

Eingaberoutinen: **150**, 008–016; **156**, 001–016; **159**, 001–003, 004–008; **165**, 020–030; **170–171**, 001–006, 022–027, 038–043, 055–060, 087–092; **196**, 17–20, 41–40; **188**, 001–003, 013–015, 031–033, 039–041, 052–054; **178**, 001–004, 023–025.

[ISG]: **142**, 002, **144**, 083; **150**, 015; **151**, 081, 086; **172**, 131.

Laufschleifen: **150**, 008–016; **156–157**, 026–067; **159–160**, 009–028; **165**, 018–030; **172**, 128–132; **178–179**, 005–022, 035–084; **183**, 018–021, 025–039; **188**, 004–012.

[Σ +], **[L.R.]** und verwandte Funktionen: **165**, 029, 036, 039, 045, 056; **178**, 003, 024, 033, 042; **183**, 023.

Ausgaberoutinen: **150–151**, 055–063; **165**, 038–048, **172**, 122–132; **178**, 005–022; **179**, 081–083.

[RAN#]: **195**, 009; 198, 134.

[RCL] **[(i)]**: **142**, 004; **144**, 080; **150**, 012, 044; **172**, 129.

[STO] **[(i)]**: **150**, 014.

[STO] **[+]** *n*: **160**, 016, 018, 024, 026, 049; **183**, 010, 014; **196**, 019, 043.

[STO] **[\times]** *n*: **156**, 019.

Unterprogramme: **142–143**, 001–009, 010–068, 057–068, 069–078
144, 079–085; **150**, 020–048, 041–048, 064–076, 077–091;
156–157, 054–066, 068; **160**, 046–050, 051–059; **171**, 093–114;
172, 117–121; **189**, 063–087, 088–092; **196–198**, 029–040,
054–098, 080–094, 099–123, 133–139.

Benutzer-definierbare Tasten: **142**, 010; **143**, 079; **144**, 086; **150**,
001, 049; **156** 001, 017; 068; **159**, 001, 011; 021, 029, 033; **164**, 001;
165, 038, 049, 053; **170**, 001, 022, 038, 055; **171**, 087; **178**, 001, 005,
023, 035; **183**, 001, 018, 022, 025, 040; **188**, 001, 013, 031, 039, 052;
195, 001, 006; **196**, 017, 041; **197**, 124.

$x \geq (i)$, **151**, 079, 084, 089.

Index Funktionstasten

ON Ein- und Ausschalten der Rechneranzeige (Seite 16).

Anzeige-Kontrolle

FIX Die Anzeige wird auf Festkommaformat geschaltet (Seite 67).

SCI Die Anzeige wird auf wissenschaftliche Notation geschaltet (Seite 68).

ENG Die Anzeige wird auf technische Notation geschaltet (Seite 70).

Mantisse. Durch **f** **CLEAR** **PREFIX** werden alle zehn signifikanten Stellen der Zahl im X-Register angezeigt (solange die **PREFIX** Taste gedrückt ist); Löschen aller unvollständigen Eingaben siehe **CLEAR** **PREFIX** (Seite 72).

Hyperbolische Funktionen

HYP **SIN**,
HYP **COS**,
HYP **TAN**.

Berechnung des Sinus hyperbolicus, des Cosinus hyperbolicus bzw. des Tangens hyperbolicus. Als Argument wird in allen Fällen der Inhalt des angezeigten X-Registers benutzt (Seite 48).

HYP⁻¹ **SIN**,
HYP⁻¹ **COS**,
HYP⁻¹ **TAN**.

Berechnung des inversen Sinus hyperbolicus, des inversen Cosinus hyperbolicus bzw. des Tangens hyperbolicus. Als Argument wird der Inhalt des angezeigten X-Registers benutzt (Seite 48).

Indexregister-Kontrolle

I Indexregister R_i . Kann mit **STO** und **RCL** als einfaches Datenspeicher-Register benutzt werden. Dient zusätzlich zur Speicherung eines Kontrollwerts bei Inkrement/Dekrement-Operationen und zur indirekten Steuerung der Anzeige und der Programmausführung (Seite 127).

X \geq I Vertauschen der Inhalte des angezeigten X-Registers und des Indexregisters (Seite 127).

(i) Befehl für indirekte Operationen. Kann zusammen mit **STO** und **RCL** zur indirekten Speicherung bzw. Rückruf und zur Speicherregister-

arithmetik verwendet werden (Seite 130).

X \geq (i) Vertauschen der Inhalte des angezeigten X-Registers und des durch den Wert des Indexregisters adressierten Speicherregisters (Seite 130).

DSE Dekrement und springe wenn kleiner oder gleich. Der Zählerwert wird um den spezifizierten Dekrementwert herabgesetzt. Die auf den **DSE** Befehl folgende Programmzeile wird übersprungen, wenn der Wert des Zählers kleiner oder gleich dem spezifizierten Testwert ist (Seite 128).

ISG Inkrement und springe wenn größer. Der Zählerwert wird um den spezifizierten Inkrementwert erhöht. Die auf den **ISG** Befehl folgende Programmzeile wird übersprungen, wenn der neue Wert des Zählers größer als der spezifizierte Testwert ist (Seite 128).

Konvertierungen

→ R Transformation der im X- und Y-Register gespeicherten Po-

larkoordinaten r und θ in die entsprechenden Rechteckskordinaten x und y (**Seite 52**).

➔ **P** Transformation der im X- und Y-Register gespeicherten Rechteckskordinaten x und y in die entsprechenden Polarkoordinaten r und θ (**Seite 51**).

➔ **H.MS** Konvertierung von dezimalen Stunden (oder Grad) in Stunden, Minuten und Sekunden (oder Grad, Minuten, Sekunden) (**Seite 46**).

➔ **H** Konvertierung von Stunden, Minuten, Sekunden (oder Grad, Minuten, Sekunden) in dezimale Stunden (oder Grad) (**Seite 47**).

➔ **RAD** Konvertierung von Grad in Radiant (Bogenmaß) (**Seite 47**).

➔ **DEG** Konvertierung von Radiant in Grad (**Seite 47**).

Logarithmische und Exponentialfunktionen

LN Berechnung des natürlichen Logarithmus der im angezeigten X-Register gespeicherten Zahl (**Seite 48**).

e^x Natürliche Exponentialfunktion. Berechnung der Potenz zur Basis e der im angezeigten X-Register ge-

speicherten Zahl (**Seite 48**).

LOG Berechnung des dekadischen Logarithmus der im angezeigten X-Register gespeicherten Zahl (**Seite 48**).

10^x Dekadische Exponentialfunktion. Berechnung der Potenz zur Basis 10 der im angezeigten X-Register gespeicherten Zahl (**Seite 48**).

y^x Berechnung der Potenz der im X-Register gespeicherten Zahl zu der im Y-Register gespeicherten Basis (**Seite 49**).

Mathematische Operationen

$-$ $+$ \times \div Arithmetische Operatoren (**Seite 22**).

\sqrt{x} Berechnung der Quadratwurzel der im angezeigten X-Register gespeicherten Zahl (**Seite 44**).

x^2 Berechnung des Quadrats der im angezeigten X-Register gespeicherten Zahl (**Seite 44**).

$x!$ Berechnung der Fakultät $x!$ oder des Funktionswerts der Gammafunktion $\Gamma(1+x)$ (**Seite 43**).

$1/x$ Berechnung des Reziprokwerts der im angezeigten X-Regi-

ster gespeicherten Zahl (**Seite 43**).

π Laden der Zahl π (3.141592654) in das angezeigte X-Register (**Seite 42**).

Prozentrechnung

% Berechnung des prozentualen Anteils der im Y-Register gespeicherten Zahl (**Seite 49**).

$\Delta\%$ Berechnung der prozentualen Differenz zwischen dem Wert des Y-Registers und dem Wert des angezeigten X-Registers (**Seite 50**).

Speicherfunktionen

STO Speicherung der angezeigten Zahl in dem Speicherregister, das durch die nach der

STO Anweisung anzugebende Adresse (0 bis 9, .0 bis .9, **I** oder **(i)**) spezifiziert wird. Kann zusätzlich zur Speicherregisterarithmetik benutzt werden (**Seite 37**).

RCL Rückruf des Inhalts des Speicherregisters, das durch die nach der **RCL** Anweisung anzugebende Adresse (0 bis 9, .0 bis .9, **I** oder **(i)**) spezifiziert wird, in das angezeigte X-Register (**Seite 38**).

CLEAR **REG** Löschen (d. h. auf Null setzen)

der Inhalte sämtlicher Speicherregister (Seite 38).

LST \bar{x} Rückruf der vor der letzten Operation angezeigten Zahl in das angezeigte X-Register (Seite 28).

Stackmanipulation

$x \rightleftharpoons y$ Vertauschen der Inhalte des X- und Y-Stackregisters (Seite 28).

R \downarrow Verschieben der Stackinhalte nach unten (Seite 28).

R \uparrow Verschieben der Stackinhalte nach oben (Seite 28).

CL \bar{y} Löschen (d. h. auf Null setzen) des Inhalts des angezeigten X-Registers (Seite 17).

Statistikfunktionen

CLEAR Σ Löschen der Statistikregister (R₀ bis R₅) und der Stackregister (Seite 55).

$\Sigma +$ Akkumulation der Zahlen im X- und Y-Register zu den Statistiksummen in den Speicherregistern R₀ bis R₅ (Seite 55).

$\Sigma -$ Entfernen der Zahlen im X- und Y-Register aus den Statistiksummen in den Speicherregistern R₀ bis R₅ (Seite 58).

\bar{x} Berechnung des Mittelwerts der mit

$\Sigma +$ akkumulierten x- und y-Werte (Seite 60).

s Berechnung der Standardabweichungen der aus den mit **$\Sigma +$** akkumulierten x- und y-Werten bestehenden Stichprobe (Seite 61).

\bar{y}, r Linearer Schätzwert und Korrelationskoeffizient. Berechnung eines geschätzten y-Werts (\bar{y}) für einen gegebenen x-Wert nach der Methode der kleinsten Quadrate und Speicherung des Resultats im angezeigten X-Register. Zusätzlich Berechnung des Korrelationskoeffizienten (r) der linearen Schätzung, der angibt, wie gut die vorgegebenen Datenpaare durch eine Gerade approximiert werden können. Speicherung des Ergebnisses im Y-Register (Seite 64).

L.R. Lineare Regression. Berechnung des y-Achsenabschnitts (B) und der Steigung (A) der linearen Funktion $y = Ax + B$, die die mit **$\Sigma +$** akkumulierten x- und y-Werte am besten approximiert. Speicherung des Werts des y-Achsenabschnitts im X-Register, des Werts der Steigung im Y-Register (Seite 63).

RAN # Erzeugen einer Pseudo-Zufallszahl

und Speicherung des Werts im angezeigten X-Register; zur Berechnung der Zufallszahlenfolge wird ein vom Benutzer vorzuzugewählter Startwert benutzt, der mit **STO** **RAN #** zu speichern ist (Seite 54).

Trigonometrische Funktionen

DEG Setzen des trigonometrischen Modes auf (dezimale) Alt-Grad; wird durch Fehlen der Statusanzeigen **GRAD** oder **RAD** gekennzeichnet (Seite 45).

RAD Setzen des trigonometrischen Modes auf Radiant (Bogenmaß), wird durch die Statusanzeige **RAD** gekennzeichnet (Seite 45).

GRD Setzen des trigonometrischen Modes auf Neugrad; wird durch die Statusanzeige **GRAD** gekennzeichnet (Seite 45).

SIN, **COS**, **TAN** Berechnung des Sinus, des Cosinus bzw. des Tangens der Zahl im angezeigten X-Register (Seite 45).

SIN⁻¹, **COS⁻¹**, **TAN⁻¹** Berechnung des Arcus sinus, des Arcus cosinus bzw. des Arcus tangens der Zahl im

angezeigten X-Register (Seite 45).

Vorwahltasten

f Auslösen von **f** vor einer Funktionstaste bewirkt, daß die in goldfarbenen Buchstaben oberhalb dieser Taste aufgedruckte Funktion gewählt wird (Seite 21).

g Auslösen von **g** vor einer Funktionstaste bewirkt, daß die in blauen Buchstaben auf der Vorderseite der Funktionstaste aufgedruckte Funktion gewählt wird (Seite 21).

CLEAR **PRÉFIX** Löschen von **f** und **g** Vorwahlangaben und von unvollständigen Eingaben wie **f** **SCI** oder **g** **HYP⁻¹**. Zusätzlich Anzeige der 10stelligen Mantisse der Zahl im angezeigten X-Register (Seite 21).

Wahrscheinlichkeitsrechnung

P_{v, x} Permutation. Berechnung der An-

zahl aller Möglichkeiten y verschiedene Elemente zu Mengen mit jeweils x Elementen unter Beachtung der Reihenfolge und ohne Wiederholungen zusammenzufassen (Seite 52).

C_{v, x} Kombination. Berechnung der Anzahl aller Möglichkeiten y verschiedene Elemente zu Mengen mit jeweils x Elementen ohne Beachtung der Reihenfolge und ohne Wiederholungen zusammenzufassen (Seite 53).

Zahlenmanipulation

RND Rundung der 10stelligen Mantisse der Zahl im X-Register auf das aktuelle Anzeigeformat (Seite 42).

ABS Berechnung des Absolutwerts der im X-Register gespeicherten Zahl (Seite 42).

INT Berechnung des ganzzahligen Anteils der im X-Register gespeicherten Zahl durch

Abschneiden des dezimalen Anteils (Seite 42).

FRAC Berechnung des dezimalen Anteils der im X-Register gespeicherten Zahl durch Abschneiden des ganzzahligen Anteils (Seite 42).

Zifferneingabe

ENTER Der Inhalt des angezeigten X-Registers wird in das Y-Register kopiert; dient zur Trennung von aufeinanderfolgenden Zahleneingaben (Seite 27).

CHS Das Vorzeichen der Mantisse oder des Exponenten der im angezeigten X-Register gespeicherten Zahl wird umgekehrt (Seite 17).

EEX Eingabe des Exponenten; die nächsten eingetasteten Ziffern werden als Exponent zur Basis 10 interpretiert (Seite 71).

0 bis **9** Zahlentasten, **.** Dezimalpunkt (-komma). 0 bis 9 Zifferntasten,

Index Programmtasten

[P/R] Program/Run Mode. Durch Auslösen dieser Funktion wird der Rechner wechselseitig auf Program Mode (PRGM Statusanzeige an) oder Run Mode (PRGM Statusanzeige aus) geschaltet (**Seite 78**).

[MEM] Anzeige des momentanen Status der Programmspeicher/Speicherregister-Aufteilung (**Seite 77**).

[A], [B], [C], [D], [E] Benutzer-definierbare Programmtasten für Programmabels und zur Steuerung der Programmausführung (**Seite 79**).

[USER] Wechselweises Ein- und Ausschalten des User-Modes. Im User-Mode werden die primären Tastenfunktionen **[\sqrt{x}], **[e^x], **[10^x], **[y^x] und **[$1/x$] und die Benutzer-definierbaren Tastenfunktionen **[A] bis **[E] vertauscht (**Seite 79**).**************

0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Numerische Labels; dienen zur Definition des Anfangs einer Routine, wenn eine **[LBL]** Anweisung vorangestellt ist (**Seite 78**).

[LBL] Label. Dient zusammen mit **[A] bis **[E]****

oder 0 bis 9 zur Kennzeichnung des Beginns eines Programms oder Unterprogramms (**Seite 78**).

[GTO] Sprunganweisung; wird zusammen mit **[A] bis **[E]**, 0 bis 9 oder **[I]** benutzt. Bei Verwendung über das Tastenfeld durchsucht der Rechner den Programmspeicher nach unten nach dem bezeichneten Label und hält dann gegebenenfalls an. In einem ablaufenden Programm wird die Programmausführung an dem bezeichneten Label fortgesetzt. (**Seite 112**).**

[GSB] Aufruf eines Unterprogramms; wird zusammen mit **[A] bis **[E]**, 0 bis 9 oder **[I]** benutzt. Bei Verwendung über das Tastenfeld durchsucht der Rechner den Programmspeicher (nach unten) nach dem bezeichneten Label und beginnt dort mit der Programmausführung. In einem ablaufenden Programm wird zu dem bezeichneten Label gesprungen und dort die Programmausführung fortgesetzt, bis eine **[RTN]** Anweisung ge-**

funden wird. Danach wird die Ausführung mit dem auf die **[GSB]**-Anweisung folgenden Befehl fortgesetzt (**Seite 119**).

[GTO] [.] *nnn* Sprung zu einer Zeilennummer; der Rechner wird auf die durch *nnn* spezifizierte (belegte) Zeile positioniert (**Seite 78**).

[BST] Rückschritt; der Rechner wird im Programmspeicher um eine Zeile zurückpositioniert (**Seite 97**).

[SST] Einzelschritt; der Rechner wird im Programmspeicher um eine Zeile vorwärtspositioniert (**Seite 97**).

[←] Löschen; im Program Mode wird die angezeigte Anweisung im Programmspeicher gelöscht. Alle nachfolgenden Befehle werden um eine Zeile nach oben verschoben. Im Run Mode werden einzelne Ziffern einer Zahl bzw. die ganze Zahl im X-Register gelöscht (**Seite 17**).

CLEAR **[PRGM]** Im Program Mode werden alle Befehle im Programmspeicher gelöscht, und der Rechner wird auf Zeile 000 positioniert. Im Run Mode erfolgt nur die

Positionierung auf Zeile 000 (**Seite 78**).

PSE Pause; der Befehl unterbricht die Programmausführung für etwa eine Sekunde, um den Inhalt des X-Registers anzuzeigen (**Seite 79**).

R/S Run/Stop; wechselweises Starten und Anhalten der Programmausführung. Die Programmausführung beginnt immer mit der Zeile, auf die der Rechner momentan positioniert ist (**Seite 79**).

RTN Rücksprung; der Rechner springt entweder von einer beliebigen Zeile im Pro-

grammspeicher nach Zeile 000, oder von einem Unterprogramm in die auf den Unterprogrammaufruf folgende Zeile (**Seite 79**).

SF Setzen des spezifizierten Flags (0 oder 1) (**Seite 111**).

CF Löschen (Zurücksetzen) des spezifizierten Flags (0 oder 1) (**Seite 111**).

F? Abfrage, ob der spezifizierte Flag (0 oder 1) gesetzt ist oder nicht. Wenn der Flag gesetzt ist (TRUE), setzt der Rechner die Programmausführung mit der nächsten Zeile im Programmspeicher fort. Ist der Flag nicht

gesetzt (FALSE), wird bei der Ausführung die nächste Programmzeile übersprungen (**Seite 111**).

$X \leq Y$	$X > Y$	$X \neq Y$
$X = Y$	$X < 0$	$X > 0$
$X \neq 0$	$X = 0$	

Vergleichsabfragen; bei jeder dieser Operationen wird der Wert im X-Register in der angegebenen Weise mit 0 oder dem Wert im Y-Register verglichen. Ist das Ergebnis TRUE, wird die Programmausführung mit der nächsten Zeile im Programmspeicher fortgesetzt; bei FALSE wird die nächste Programmzeile übersprungen (**Seite 110**).

Sachindex

Die Seitenangaben in **Fettschrift** beziehen sich auf die Hauptreferenzen; die Seitenangaben in Normalschrift sind Hinweise auf weitere Referenzen.

A

- Abgekürzte Tastenfolgen, **78**, 127
- Adresse
 - indirekte (*siehe* indirekte Adressierung)
 - Label-, **78**, 94
 - Speicherregister-, **37–39**
- Algorithmus, **206–207**
- Alpha-Labels, **79**
- alternative Tastenfunktionen, **20–21**, 79
- Anhalten eines Programms, **79**, 89–90, 93
- Anzeige (*siehe* Display)
- Aufteilung, momentane Speicher-, **77**
- Aufteilung, Speicher- **8**, **75**
- Ausführen eines Programms, **86–88**
- Ausführungszeiten, **53**
- Ausgaberoutinen, **241**

B

- Batterien, **230–233**
 - Austausch, **231**
 - Einbau, **231–233**
- Beenden der Zifferneingabe, 17, 18, 222
- belegter Programmspeicher, **77**, 93, 10
- Benutzer-definierbare Tasten, 213, 218–219, 242

C

- Chiquadrat-Test, **182–185**
- Cramer'sche Regel, **149**

D

- Datenspeicher-Register, **8**, **37–38**
 - Löschen, **38**
- Dekrement, Schleifen-, **128–129**, 214, 241

Determinante

lineare Gleichungssysteme, **149**Matrizenrechnung, **140–148**Dezimaltrennzeichen, **16**Differenz Δx , **214**disabling, Stack-, 29, **222**

Display

autom. Umschalten **[FIX]-[SCI]** Format, **67**Format, **67**leeres –, **233–234**löschen, **17, 29**Rundung, 67, **68, 69, 72**Wahl des Formats, **67**X-Register, **26**Doseninhalt, Programmbeispiel, **89–90**Dreiecksberechnungen, **169–176**Dateneingabe, **213****[DSE]**, Einschränkungen, **134****E**Eingabe von Daten, **213–214, 218**Eingaberoutinen, **241**enabling, Stack-, 29, **222**Ende, Programm-, **84**Exponent, **71**Vorzeichen, **71**Exponentialkurven, 162, **166–167****F**Fälligkeiten (*siehe* Finanzen)

Fehler

Flag, 221

-Korrektur, **33**löschen, **20**-Meldung, 19, 93, 96, **220–221**Permanentspeicher, **20, 233**Rundungs-, **72, 155**Service, 221, **236–240**Simpson-Regel, **159–162**Statistikregister, **220–221, 229**Unterprogramme, **119, 210**

- fehlerhafte Rechnerfunktion, **233–234**
- Finanzen, **185–194**
- Flag, 19, 96, **111**, 116–118, 216–217, 241
- Flußdiagramm, 116, **209–210**, 217
- Funktionen einer Variablen, **21**, 24, 43–49
- Funktionen zweier Variabler, **22**, 24, 29, 49–54
- Funktionsprüfung des Rechners, **233**
- Funktionstasten, Index, **243–246**

G

- Genauigkeit (*siehe* Rundungsfehler)
- Gewährleistungsinformation, **235–236**
- go to
 - indirekt, **130–131**, 136–137, 241
 - Label, **112**
 - Unterprogramm, **119–120**, 210
 - Unterprogramm (Matrizenrechnung), **210**
 - Zeile (Korrektur), **97–98**, 103
- Güte einer Regression, **163**

I

- Index Register, 8, 75, 80, **127 ff.**
 - direkte Funktionen, **127–129**
 - Dekrement, **128–129**
 - Inhalte vertauschen, **127**
 - Inkrement, **128**
 - Schleifenkontrollwert, **128**
 - Speichern und Rückruf, **127**
- Indirekte Adressierung, 127, **130**, 211–212
 - Speichern und Rückruf, **130**, 135, 241
 - Speicherregister-Arithmetik, **130**, 135
 - Tabelle, **131**
 - Unterprogramme, **131**, 136
 - Vertauschen, **130**, 135, 242
 - Verzweigungen, **130–131**, 136
 - Zeilennummer, 130, **131**, 137
- Initialisierung, **96**
- Initialisieren des Permanentenspeichers, **20**, 38, 45, 54, 75, 221, 233
- Inkrement, Schleifen-, **128–129**, 132–134, 214, 241
 - Einschränkungen, **134**
 - Verwendung bei indirekter Adressierung, **211–212**
- interne Zifferndarstellung, **67**, 72

K

-
- Kettenrechnung, **31–32**
 - Konstantenarithmetik, **34**, 35–36
 - Kontrollwert, 127, **128 ff.**
 - konvertierter Programmspeicher (*siehe* Programmspeicher)
 - konvertierte Speicherregister (*siehe* Speicherregister)
 - korrekte Rechnerfunktion, **233–234**
 - Korrelationskoeffizient, **64**, 163
 - Kreisfläche, Programmbeispiel, **83 ff.**
 - Kurvenanpassung, **162–168**

L

-
- Label, **78**, 84, 94
 - Adressen (*siehe* Adresse)
 - suchen nach einem –, **87**
 - Laden eines Programms, **85–86**
 - LAST X, 8, 28, **33–34**, 224
 - Konstantenarithmetik, **36**
 - Löschen, **38–39**
 - prozentualer Anteil, **50**
 - Statistiksummen, **65**
 - Wahrscheinlichkeitsrechnung, **53**
 - $\boxed{\Sigma+}$, **56**
 - $\boxed{\bar{y}, r}$, **65–66**
 - lineare Gleichungssysteme, Matrizenrechnung, **140–148**
 - lineare Gleichungssysteme mit 3 Unbekannten, **149–153**
 - lineare Regression, **63**
 - Löschen
 - Datenspeicher-Register, **38–39**
 - Display, **17–18**
 - Fehlermeldungen, **19**
 - Permanentspeicher (*siehe* Initialisieren des Permanentspeichers)
 - Programmbefehle, **105–107**
 - Programmspeicher, **20**, 78
 - Programmzeilen, **98**
 - Stack, **55**
 - logarithmische Kurven, **162**, 167–168

M

- Mantisse, 21, 67, 68, 71, 72
- Matrizenrechnung, 140–148
 - indirekte Adressierung, 211
 - Inkrement, 211
 - Unterprogramme, 210
- Methode der kleinsten Quadrate, 63, 164
- Mode
 - Display-, 67–71
 - trigonometrischer –, 45

N

- negative Zahlen, 17
- neutrale Operationen, 223–224
- Newton-Verfahren, 154–158
 - Schleifen, 214
- nichtprogrammierbare Funktionen, 88
- Numerierung von Programmbefehlen, 102
- numerische Integration, 159–162
- numerische Labels, 79

O

-  ON, 10, 16, 233
- Overflow, 18, 97, 220
-  $\Sigma+$, 55

P

- pending returns, 19, 119–120
- permanente Speicherregister, 8, 76, 225
- permanenter Programmspeicher, 76, 224–225 ff.
- Permanentspeicher, 19, 75, 96
 - Batterienaustausch, 20, 231–232
 - Display, 67
 - Fehler, 20, 221, 233
 - Flags, 19, 111
 - Initialisieren (*siehe* Initialisieren des Permanentspeichers)
 - Löschen (*siehe* Initialisieren des Permanentspeichers)
 - Löschen durch Stromausfall, 231, 232
 - Startwert Zufallszahlen, 54
 - trigonometrische Modi, 45

- Potenzkurven, **162**, 168
- primäre Tastenfunktioen, **20**, 79
- PRGM** Statusanzeige, 20, **78**, 80, 85, 88
- Problem-Statement, **206**
- Programm
 - Ausführung, **87**, 115, 118
 - Befehle einfügen, **102**, 105
 - Befehle löschen, 18, **98**, 105–107
 - Fehler, **96**
 - Kontrolltasten, **247–248**
 - Labels, **94**
 - Mode, **20**
 - Modifizieren, Beispiel, 101–106
 - Pausen, **91–92**
 - Planung, **206–210**
 - selbstinitialisierendes –, **96**
 - Stops, **88–91**, 93
 - Struktur, **206–208**
- Programmanfang, **84**
- Programmende, **84**
- Programmiertechniken, Index, **241–242**
- Programmspeicher, 8, 78, **80–81**
 - Anfang des –, **78**
 - belegter –, **77**
 - Ende des –, **84**
 - löschen, **78**
 - nach unten durchlaufen, **80**, 87, 125
- Pseudozufallszahlen (*siehe* Zufallszahlen)
- Pythagoras, Satz von, Programmbeispiel, **98 ff.**

Q

- quadratische Gleichungen, Algorithmus, **207–210**

R

- Reihenfolge der Eingabe, **22**
- roll-down, **28**
- roll-up, **28**
- Rückschritt, **97**, 101, 104
- Rückruf von Zahlen, **38**
 - Stack Lift, **38**
 - $\Sigma+$ Akkumulation, **57**

- Rücksprünge, nicht aufgelöste (siehe pending returns)
- Rückumwandlung von Programmspeicher, **74–77**, 228–229
- Run Mode, **20**
 - running**, **18**, 53, 88
 - Taste, **78**
- Rundungsfehler, **72**, 155, 169
 - in Statistiksummen, **59**

S

- Schleifen, **113–114**, 214–215, 241
 - Abbruchkriterium Newton-Verfahren, **155**
 - bedingte Kontrolle, **114–115**
 - Differenz Δx , **214–215**
 - Gleichungen, Matrizenrechnung, **148**
 - Kontrollwert, **128**
 - Stromkreisspannungen, Programmbeispiel, **147**
 - Verlassen einer Schleife, **113–115**, 127, 132–134, 214–215
 - Zähler, 127, **128 ff.**, 132–134, 214
- selbstinitialisierende Programme, **96**
- Service, 233, **237–240**
 - Gewährleistung, **239**
 - Versandanweisungen, **239**
- Simpson-Regel, **159**
- single-step, **97**, 100–101
- Sparguthaben (siehe Finanzen)
- Spannungsabfall, **231**
- Speicher
 - Aufteilung, **77**
 - Konvertierung, **74–77**, 224–227
 - Konvertierung in Speicherregister, **227–228**
 - MEM** Funktion, **77**, 227–228
 - Rückumwandlung, automatische, **74–77**, 224–227
- Speicheranfang, Kennung, **80**
- Speichern von Zahlen, **37**
- Speicherregister, 8, **75–77**
 - Adressen, 8, **37–39**
 - Arithmetik, **39**
 - konvertierbare –, **75–76**
 - löschen, **38–39**
 - statistische –, **55–56**

Sprünge

bedingte –, **112–113**unbedingte –, **112**Stack, **8**automatischer Speicher-, **26–27**Drop, **26–27**, 30–32– freigeben, **29**, 223-Lift, **26–27**, 29, 31, 32, 38, 222– löschen, **38**, 55, 56-Manipulation, Funktionen zur, **27–28**-neutrale Operationen, 221, **223–224**prozentualer Anteil, **50**– sperren, 27, **222** $\boxed{\text{L.R.}}$, **63** $\boxed{\text{RCL}}$ $\boxed{\Sigma+}$, **57** $\boxed{\text{s}}$, **62** $\boxed{\bar{x}}$, 60, **61** $\boxed{\bar{y}_r}$, **65–66** $\boxed{\Sigma+}$, **56**Starten von Programmen über Labels, **79**Startwert von Zufallszahlen, **54–55**, 194, 217

statistische Funktionen

Fehler, **220**, 229Genauigkeit, **56–57**Korrektur von Summen, **58–59**Löschen der Statistikregister, 55, **58**Register, **56**, 229Speicherregister, **55–56**Statusanzeigen, **16****f**, **21****g**, **21****PRGM**, 20, **78**, 80, 85, 88**USER**, **79**, 88Steigung einer Funktion, Programmbeispiel, **121 ff.**stop-and-store, **213**Stromausfall, 45, 54, **221**Stromverbrauch, **230****T**T-Register, Konstantenarithmetik, **34–36**T-Test, **176–181**Tastencodes, **77–78**, 81–83

Temperaturspezifikationen, **240**
Testen der Rechnerelektronik, **234**
Testen von Programmen, **96**
Time-out, **16**
Trapezregel, **159–162**
trigonometrische Modi, **19, 45, 169**
TRUE/FALSE Abfragen, **110–111**

U

U-Boot-Jagd, **194–205**
 Programm, Zufallszahlen, **217**
unbedingte Sprünge, **112**
Underflow, **18**
Unterprogramme, **119 ff.**, 210–211, 242
 Einschränkungen, **120–121**
 nicht aufgelöste Rücksprünge, **19, 119**
 Übergabe der Kontrolle, **119–120, 122 ff.**
 Verschachtelungen, **120–121**
Unterstützung, technische, **240**
User Mode, **79, 88**
USER Statusanzeige, **79, 88**

V

Variableneingabe, **213–214**
Vergleichsoperationen, **110–115, 241**
Vorwahl
 – löschen, **21**
 – Tasten, **21, 78, 79, 127**
Vorzeichen
 – des Exponenten, **68, 71**
 – einer Zahl, **17, 68**
Vorzeichenaustausch, **17, 42, 223**
 – des Exponenten, **71**

W

Wärmeverlust, Programmbeispiel, **12, 81 ff.**

Z

Zahlenmanipulation, Funktionen zur, **42**
Zähler, Schleifen-, **214**

Zeile 000 (siehe Kennung Speicheranfang)

Zeilennummer, **77–78**

zeilenweise Programmausführung, **99–100**

Zifferneingabe

– einer neuen Zahl, **17, 24**

Beendigung der –, **17, 18, 222**

Zifferntrennzeichen, **16**

Zufallszahlen, **54–55**, 217, 241

Zwischenresultate, **23–25**, 26, 31

LOTTO-PROGRAMM

Zufallsgener.

1 aus 45

ON

g-P/R

f-LBL A-B-C-D-E

f-FIX 4

x ≠ y

f-RAN #

Enter

44 x

g-INT

f-FIX 0

1 +

g-RTN

g-P/R

Anwendung

Zahl $0 \leq n \leq 1$

STO

f-RAN #

Enter

f-USER

0..0..0...

martin egli 1986

RER.- SERVICE:

SESCO-HANDELS AG
NORDSTR. 15
4665 OFTRINGEN 062-97 48 48

MARTIN EGLI
GRÜZEFELDSTR. 12
8400 WINTERTHUR

052/29'36'35

HEWLETT-PACKARD - VERKAUF

SCHWAMENDINGERSTR.10
8050 ZUERICH
01-315'81'81

HEWLETT-PACKARD - SCHWEIZ

ALLMEND 2
8967- WIDEN
057-31'21'11

BEDIENUNGSFRAGEN: (PC)

031-34 34 41
HR.TUTLI

SONST:

PAUL WIGENSTOSS
WILHELMSTRASSE 01-832 66 66

HP-USER-CLUB

MI 07.00 - 12.00
14.00 - 17.00

031/40 11 71

156-83-73 09-12 h
MO-DO

RUNDUNGS-PROGRAMM

auf 1/20 b. 1/100
0.075 = 0,1000

Beisp. 18,3165

ON	
f-FIX	
4	
g-P/R	
f-clear- PRGM	
f-LBL A-B-C-D-E	
Enter	18,3165
1000x	18316.50
g-INT	18316.00
100 ÷	183.16
Enter	183.16
f-FRAC	0.16
100x	16.00
0.02x	0.32
0.5+	0.82
x ≥ y	183.16
10x	1831.60
g-INT	1831.00
+	1831.82
g-INT	1831.00
10 ÷	183.10
Enter	183.10
f-FRAC	0.10
10x	1.00
1.30+	2.30
0.24x	0.55
g-INT	0.00
0.50x	0.00
x ≥ y	183.10
g-INT	183.00
+	183.00
10 ÷	18.30
g-RTN	
g-P/R	

Anwendung

Zahl f-E
oder
f-USER
Zahl E

martin egli 1986

- 1** Erste Schritte
- 2** Speicherstack, LAST X, und Datenspeicherung
- 3** Numerische Funktionen
- 4** Kontrolle der Anzeige
- 5** Grundlagen der Programmierung
- 6** Programmkorrektur
- 7** Programmverzweigungen
- 8** Unterprogramme
- 9** Das Indexregister
- 10** Anwendungsprogramme
- 11** Programmiertechniken

