TI·57·LCD

TEXAS INSTRUMENTS

TEXAS INSTRUMENTS

Last Name
Familienname
Nom
Cognome
Achternaam
Efternamn
Efternavn
Sukuniml
Ultimo nome
Appellidos

First Name
Vorname
Prénom
Nome
Voornaam
Förnamn
Fornavn
Etuniml
Primeiro nome
Nombre

Address
Adresse
Indirizzo
Adres
Gatuadress
Osoite
Endereço
Dirección

Town
Ort
Ville
Città
Stad
By
Kaupunki
Cidade/Vila
Ciudad

P.O. Code
Postleitzahl
Code Postal
Codiea Postala
Postcode
Postnr.
Postinumero
Codigo postal
D. Postal

Country
Land
Pays
Paese
Maa
Pais

Date Datum, Data,
Päivämäärä, Dato, Fecha

© 2010 Joerg Woerner
Datamath Calculator Museum

TASTEN - VERZEICHNIS

Dieses Tastenverzeichnis gibt eine kurze Beschreibung jeder Taste.

2nd	12	INV	12	R/S	44	OFF	12	ON/C	14
x=t	74	x≥t	74	SBR	91	Dsz	87	Del	71
RST	44	GTO	56	LBL	56	BST	40	SST	39
log	23	lnx	23	1/x	22	x^2	22	$\sqrt{x}$	22
DRG▸	25	P⇄R	28	gmd dd	27	π	14	n!	24
DRG	25	sin	26	cos	26	tan	26	y^x	22
C.t	32	Fix	21	lglg	24	Frac	24	|x|	24
x⇄t	32	EE	19	(	17	)	17	÷	15
Part	38								
STO	30	7	14	8	14	9	14	×	15
CM	30								
RCL	30	4	14	5	14	6	14	−	15
CP	39								
EXC	30	1	14	2	14	3	14	+	15
Pause	73								
LRN	39	0	14	•	14	+/−	14	=	15

© 2010 Joerg Woerner
Datamath Calculator Museum

ACHTUNG

Ihr TI-57 LCD hat Speicher mit "Constant Memory", somit bleiben die folgenden Informationen, selbst wenn der Rechner abgeschaltet ist, erhalten:

- die Speicheraufteilung des Rechners
- das gespeicherte Programm
- die in den Datenspeichern und den T-Registern abgelegten Werte.

Falls Sie dabei auf irgend ein Problem stoßen sollten, erlaubt die folgende Befehlsfolge die Neuinitialisierung Ihres Rechners.

ON/C ON/C 2nd Part 7 2nd CM 2nd C.t R/S ON/C

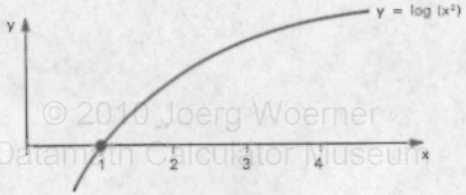
Falls es nicht möglich sein sollte den Programmier-Modus einzuschalten, muß die Speicherbereichsverteilung mit folgender Tastenfolge geändert werden : 2nd Part m.
m ist die Anzahl der Speicher die Sie benötige (m ≤ 7).

BEISPIEL

Schreiben Sie ein Programm zur Berechnung folgender Funktion.

AUFGABE

Zeichnen der Kurve $y = \log(x^2)$ für die ganzzahligen Werte von a, beginnend mit $x = 1$.



© 2010 Joerg Woerner
Datamath Calculator Museum

TASTE	ANZEIGE		BEMERKUNGEN
ON/C ON/C		0	
2nd Part 1		48,1	Den Rechner initialisieren
LRN 2nd CP	– –	St	Das Programm in den Speicher des
STO 0 x² log	31	02	Rechners eingeben
2nd Pause 1 +	85	05	
RCL 0 = RST	21	08	
LRN ON/C		0	
RST 1 R/S		0	y (0)
		0.60206	y (1)
		0.9542425	y (2)

Auf R/S drücken, um die Berechnung zu stoppen.

INHALTSVERZEICHNIS

Einführung 6

Kapitel I — Die Benutzung Ihres TI-57 LCD als wissenschaftlicher Rechner. 9
Einführung 9
Abschnitt 1 — Tastenfeld und Anzeige 11
Das Tastenfeld 11
Die Anzeige 12
Funktionszeichen 12
Meldungen 13
APD™ Automatische Stromsparschaltung. 13
Abschnitt 2 — Dateneingabetasten 14
[0] — [9] — Numerische Tasten 14
[•] — Dezimalpunkttaste 14
[+/−] — Vorzeichenwechseltaste 14
[2nd] [π] — pi Taste 14
[ON/C] — Löschtaste 14
[+], [−], [×], [÷], [=] — Arithmetische Tasten 15
AOS™ Algebraisches Operationssystem 15
[(], [)] — Klammertasten 17
Anzeigeformate 19
[EE] — Exponentialform 19
[2nd] [Fix] n — Festkommataste 21
Abschnitt 3 — Algebraische Tasten 22
[1/x], [x²], [√x] — Umkehrfunktion, Potenzierung und Radizierung . 22
[yˣ], [INV] [yˣ] — Allgemeine Potenzen und Wurzeln 22
[lnx], [log] [INV] [lnx], [INV] [log] — Logarithmen und Antilogarithmen . 23
[2nd] [n!] — Fakultät 24
[2nd] [|x|], [2nd] [Intg], [2nd] [Frac] — Teilzahltasten 24
[DRG], [INV] [DRG], [2nd] [DRG▸], [INV] [2nd] [DRG▸] — Grad, Radiant und Gon (Neugrad) 25
[sin], [cos], [tan], [INV] [sin], [INV] [cos], [INV] [tan] — Trigonometrische Tasten 26
[2nd] [DMS DD], [INV] [2nd] [DMS DD] — Umrechnung von Grad/Minuten/Sekunden in Dezimalgrad und umgekehrt 27
[2nd] [P⇄R], [INV] [2nd] [P⇄R] — Umrechnung von Polarkoordinaten in rechtwinklige Koordinaten und umgekehrt 28

© 2010 Joerg Woerner
Datamath Calculator Museum

Abschnitt 4 — Speicherfunktionen 30
2nd Part m — Taste für die Speicherbereichsverteilung 30
STO m — Abspeichern 30
RCL m — Speicherrückruf 30
EXC m — Speicheraustausch 30
2nd CM — Speicherlöschung 30
Speicherarithmetik 31
x⇄t — Austausch mit "t" 32
2nd C.t — "t" Register löschen 32

Kapitel II - Programmieren des TI-57 LCD 33
Aufbau des TI-57 LCD 35
Programmierbeispiele 41
1. Programmierbeispiel 41
2. Programmierbeispiel 51
3. Programmierbeispiel 53
4. Programmierbeispiel 56
5. Programmierbeispiel 70
6. Programmierbeispiel 77
7. Programmierbeispiel 83

© 2010 Joerg Woerner
Datamath Calculator Museum

Kapitel III — Anwendungen 102
1. Statistische Analysen 102
2. Kombinationen und Permutationen 104
3. Hyperbelfunktionen 106
4. Umgekehrte Hyperbelfunktionen (Areafunktionen) 100
5. Quadratische Gleichungen 110
6. Funktionsplotter 113
7. Grenze einer Funktion 115
8. HI-LO Spiel 117

Anhänge:
A. Fehlerbedingungen 119
B. Genauigkeit 121
C. Tastencodes in numerischer Anordnung 122
D. Zusammenfassung der Löschtasten 123
E. Wartung und Garantie 124
Im Falle von Schwierigkeiten 124
Batterieaustausch 125
Vorschläge 126
Ein Jahr Gewährleistung 127
F: Verzeichnis 128

© 2010 Joerg Woerner
Datamath Calculator Museum

Warum ein programmierbarer Rechner?

Wie programmiert man eine datenverarbeitende Maschine?

Man vertraut ihr einfach eine Arbeit an, welche sie an unserer Stelle ausführen kann.

Was muß man wissen, um dieses Ziel zu erreichen?

Man muß als erstes fähig sein, die Arbeit selber auzuführen, in der Folge den Aufbau der Maschine kennen und schließlich in der Lage sein die auszuführenden Befehle in die Sprache der Maschine zu übersetzen.
Fähig sein Arbeit auszuführen, bedeutet nicht immer, daß man weiß wie man die Durchführung dieser Aufgabe beschreibt. Der Rechner (wie ein Computer) ist SCHNELL ABER DUMM. Man muß ihm alles sagen, auch die Schritte, die uns als selbstverständlich erscheinen oder die die wir, geleitet von unserer Gewohnheit, unbewußt ausführen.
Die Kenntnis des Aufbaus der Maschine liefert das Wissen wo und wie, die Daten die ihr zu ihrer Aufgabenerfüllung notwendig sind, gespeichert werden. Benutzt man numerische Speicher, Schreib/Lesespeicher um die Programmbefehle zu speichern, Speicher die schon Funktionen enthalten...? Alle diese Informationen müssen in einer Sprache eingegeben werden die dem Rechner verständlich ist.
Das Programmieren eines Rechners bedeutet nicht nur eine neue Disziplin anzugreifen, es bedeutet auch sich mathematisch zu betätigen (schließlich kennt er nur Ziffern und ihre Bearbeitung). Aber diese Mathematik ist dynamisch. Man sieht die Entwicklung der Zahlen, man erstellt und erschafft "mathematische Wesen", man erlebt ihr Verhalten; diese Wissenschaft wird allen zugänglich.
Hat man den Spaß zu verstehen gesucht, den alle diese jungen (und weniger jungen) Leute bei der Programmiererei haben: vielleicht genießen sie eine Revanche über diese Mathematik die sie endlich verstehen, weil sie sie selbst entdecken.
Man regt derartig seine Neugierde an, man wird Forscher.
Man muß ordentlich sein um ein Programm, d.h. das "mathematische Wesen" zu erstellen. Man muß die erreichten Ergebnisse sammeln, sie orden, sie deuten, man muß seine Eingebung zur Hilfe rufen, um die sichtbaren Eigenschaften herauszustellen, die man zu bestätigen oder entkräften sucht, durch vermehrte Versuche in die oder andere Richtung.

Sehr oft erweitert die Anwendung eines einfachen Programmes die Ansprüche, die die Erweiterung der Problemstellung mit sich bringen, dies bringt Öffnungen auf Gebiete an die man zuvor nicht gedacht hätte, Anforderungen und ein gewisser Perfektionismus gegenüber der Maschine und somit an sich selbst.
Der logische Aspekt wird trotzdem nicht zur Seite geschoben, denn die Arbeitsweise des Rechners ist einer stengen logischen Mathematik untergeordnet. Jeder Irrtum auf diesem Gebiet wird von der Maschine gnadenlos geahndet. Aber es handelt sich nur um eine Maschine; sie blockiert sich, zeigt falls notwendig "ERROR" an, aber macht keinerlei Kommentar. Man fühlt sich verantwortlich aber nicht schuldig. Wie oft hat man Schüler gesehen, die sich versessen bemühen einen Fehler in ihrem Programm (also in ihrem Vorgehen) zu verbessern, und die nicht beharrt hätten wenn derselbe Fehler von einem Lehrer, in einer niedergeschriebenen Aufgabe, entdeckt worden wäre.

Was ist ein pädagogisches Werkzeug?

Sich mit der Programmierung vertraut machen, bedeutet eine Disziplin von Morgen anzupacken, aber auch die Fähigkeit zu erwerben "mathematische Wesen" zu erschaffen, sie zusammenzufügen, ihnen das Leben zu geben, ihre Reaktionen zu beobachten - das heißt ein wenig Schöpfer zu werden. Wenn die Pädagogik die Kunst ist zu Lehren, also dann ist der programmierbare Rechner zweifellos ein pädagogisches Werkzeug. Welche bessere Erkenntnis als die, die der Schöpfung und der Erfahrung entspringt? Die Mathematik erscheint zu oft als ein isoliertes Fachgebiet für die Erwählten mit dieser Sonderbegabung. Ohne Zweifel, weil sie im allgemeinen auf zu starre, zu logische Art dargestellt wird, unter Verdrängung von allem was nicht diesen Kriterien entspricht, wie die Einführung, die Kreativität, das ästhetische Gefühl usw... Ein Problem lösen, bedeutet in erster Linie es zu analysieren . Aber wenn diese Analyse nicht zur Lösung führt? Dann führt man eine Aufteilung durch und erstellt "elementare mathematische Einheiten" die man kennt, man programmiert sie, man studiert sie (durch die Beobachtung ihres Verhaltens), und sehr oft wird die Art in der sie zusammengesetzt werden müssen, um das Ziel zu erreichen selbstverständlich.

Aber was ist der Unterschied zwischen einem Computer und einem programmierbarem Rechner?

Beide beruhen auf derselben Arbeitsweise und haben denselben Aufbau.
Sie unterscheiden sich vor allem durch die Speicherkapazität und die Zahl der dem Anwender zur Verfügung stehenden Befehle.
Ihr TI-57 LCD bietet Ihnen die notwendigen und auserichenden Befehle um alle verschiedenen Arten von Programmen und Kombination zu realisieren

Linear Schleife Aussprung Alternative Unterprogramm

Es handelt sich hier also um ein ausgezeichnetes Material zur Einführung in die Programmierung. Durch seine beschränkte Kapazität, durch seine Mindestzahl von Befehlen, verpflichtet es zu Enthaltsamkeit und somit zu einer bewußten Organisierung; es wird somit zum pädagogischen Werkzeug.

© 2010 Joerg Woerner
Datamath Calculator Museum

Kapitel I — Die Benutzung Ihres TI-57 LCD als wissenschaftlicher Rechner

Einführung

Ihr TI-57 LCD ist mehr als nur ein programmierbarer Rechner. Sie können ihn als einen leistungsfähigen wissenschaftlichen Rechner gebrauchen, der für Berechnungen direkt über die Tastatur zur Verfügung steht.
Der TI-57 LCD als wissenschaftlicher Rechner bietet folgende Möglichkeiten:

- **Das Algebraische Operationssystem AOS®**
  Verständliche Dateneingabe von Zahlen, Dezimalzahlen, der pi-Funktion und Klammern. Multiplikation, Division, Addition und Subtraktion könne mit dem Algebraischen Operationssystem angewandt werden (dies erlaubt die Eingabe der meisten Berechnungen so wie sie geschrieben werden) mit bis zu vier schwebenden Operationen und fünfzehn Klammern. Ergebnisse können in bis zu sieben Datenspeichern festgehalten werden. Daten können in Standardform (mit der notwendigen Anzahl von Dezimalstellen) oder in wissenschaftlicher Notation eingegeben und angezeigt werden.

- **Mathemathische und wissenschaftliche Funktionen**
  Die algebraischen und wissenschaftlichen Tasten bieten Umkehrfunktion, Potenzierung, Radizierung, allgemeine Potenzen und Wurzeln, Fakultät, Absolutwert, Bruch- und Ganzzahlteile, dekadische und natürliche Logarithmen, und alle gebräuchlichen trigonometrischen Funktionen und ihre Inversion für die Beschreibung von Winkeln in Grad, Radiant oder Gon.

- **Eingebaute Umrechnungen:**
  Umrechnung von Polarkoordinaten in rechtwinklige Koordinaten, von Grad/Minuten/Sekunden in Dezimalgrad, und entsprechende Inversionen.

Das vorliegende Kapitel erklärt die obenstehenden Fähigkeiten und Tasten, mit der Bedeutung jeder Taste und ihrer Anwendung.

DIE BENUTZUNG IHRES TI-57 LCD ALS WISSENSCHAFTLICHER RECHNER

I

Dieses Kapitel ist in die untenstehenden Abschnitte aufgeteilt. Wenn Sie mit diesen Grundsätzen vertraut sind, können Sie sofort auf das zweite Kapitel übergehen, um mit der Programmierung zu beginnen.

Abschnitt 1 - Tastenfeld und Anzeige
Abschnitt 2 - Dateneingabetasten
Abschnitt 3 - Algebraische Tasten
Abschnitt 4 - Speicherfunktionen

© 2010 Joerg Woerner
Datamath Calculator Museum

Abschnitt 1 — Tastenfeld und Anzeige

Dieser Abschnitt enthält eine kurze Übersicht über die Grundlagen. Bitte nehmen Sie den Rechner in die Hand um jede vorgestellte Möglichkeit sofort zu probieren.
Schalten Sie Ihren TI-57 LCD mit der [ON/C] Taste ein (rechts oben auf dem Tastenfeld). Eine Null erscheint in der Anzeige.
Falls die Batterien entfernt waren oder ausgetauscht wurden, den Rechner durch Betätigung folgender Tasten in Grundstellung bringen: [ON/C], [ON/C], [2nd] [Pgm] [7], [2nd] [CM], [2nd] [Ct], [RST] und [ON/C].

DAS TASTENFELD
Wenn Sie das Tastenfeld des Rechners ansehen, stellen Sie fest, daß die Funktionen nach Aufgaben geordnet sind (Dateneingabe, Speicheroperationen, trigonometrische Funktionen, algebraische Funktionen, Programmierung). Dies erleichtert die Auffindung einer Funktion im Tastenfeld. Da der Rechner zahlreiche Möglichkeiten bietet, haben gewisse Tasten mehrere Funktionen. Die Symbole, die über die Tasten gezeichnet sind, sind Zweitfunktionen. Um eine derartige Funktion auszuführen, müssen Sie die [2nd] Taste drücken und dann die Taste der Funktion, die Sie ausführen wollen. Zum Beispiel, um die Quadratwurzel von 4,5 zu ziehen, 4.5. eingeben und [√x] drücken. Um die Fakultät von 5 zu finden, 5 eingeben und [2nd] [n!] drücken. Beachten Sie, daß einmaliger Druck auf [2nd] das 2nd Zeichen in der Anzeige setzt, und somit anzeigt, daß die nächste Taste als Zweitfunktion behandelt wird. Neuerlicher Druck auf 2nd schaltet den Zweitfunktionsmodus ab und löscht das 2nd Zeichen.

In diesem Handbuch sind Tasten für Erstfunktionen in Schwarz auf weißem Grund gedruckt. Tasten mit einem schwarzem Grund bezeichnen Zweitfunktionen.

Die Inversionstaste - [INV] - liefert ebenfalls zusätzliche Rechenfunktionen. Wenn [INV] betätigt wird, führt der Rechner die Inversion der folgenden Funktion aus.

BEMERKUNG: Falls die Anwendung beider Tasten [2nd] und [INV] notwendig ist, spielt die Reihenfolge, in der sie gedrückt werden, keine Rolle.

TASTENFELD UND ANZEIGE

DIE ANZEIGE

Die Anzeige zeigt bis zu 8 Ziffern plus eventuell zwei Ziffern als Exponent an. Bis zu 8 signifikante Ziffern können mit jeder Eingabe in den Rechner eingetippt werden.
Trotzdem benutzt das Anzeigeregister für Berechnungen intern 11 Ziffern, die in der Anzeige auf 8 Ziffern gerundet werden.
Negative Zahlen werden mit gleitendem Minuszeichen vor der Zahl angezeigt.
Das Aus-und Einschalten des Rechners (mit der [OFF] Taste) löscht die Zahl in der Anzeige und alle laufenden Berechnungen. Der Inhalt der Datenspeicher bleibt unberührt.

© 2010 Joerg Woerner
Datamath Calculator Museum

FUNKTIONSZEICHEN

Fünf Funktionszeichen geben Ihnen zusätzliche Auskünfte über den augenblicklichen Zustand Ihres Rechners.

Sonderfunktionszeichen

2nd und INV Zeichen erscheinen jedesmal wenn die [2nd] oder [INV] Tasten gedrückt werden, um daran zu erinnern, daß Sie eine umgekehrte oder zweite Funktion ansprechen. Um diesen zweiten oder umkehrten Funktionsmodus zu verlassen, muß [2nd] oder [INV] ein zweitesmal gedrückt werden.

Winkeleinheitszeichen

Jedesmal wenn der Rechner angeschaltet wird, befindet er sich im Gradmodus. Die Arbeitsweise in anderen Winkeleinheiten wird durch ein Zeichen angegeben: RAD für den Radiantmodus und Grad für den Gonmodus (Neugrad).
Die gebrauchten Winkeleinheiten können geändert werden durch [DRG], [INV] [DRG], [2nd] [DRG] und [INV] [2nd] [DRG].

Programmlaufzeichen
Das Programmlaufzeichen (RUN) gibt an, daß ein Programm läuft. Um die Ausführung anzuhalten und das Zeichen zu löschen, muß die [R/S] Taste gedrückt werden.

MELDUNGEN

- "CALC" erscheint jedesmal, wenn Ihr Rechner eine Funktion ausführt. Während des Rechenvorganges bleiben die Funktionszeichen angezeigt und die Ziffern verschwinden zugunsten der Meldung "CALC".
- "ERROR" erscheint jedesmal, wenn eine Fehlerbedingung vorliegt (zum Beispiel: Division durch Null). Das Verzeichnis dieser Fehlerbedingungen wird im Anhang A gegeben. Nach jeder Fehlermeldung muß die [ON/C] Taste gedrückt werden, um die Fehlermeldung zu löschen.

Automatische Sparschaltung (APD™-Automatic Power Down)

Durch die APD™-Funktion schaltet sich der Rechner automatisch ab, um die Batterien zu schonen, wenn 15 bis 35 Minuten lang keine Eingabe erfolgte. Wenn Sie das Gerät aber wieder einschalten, können Sie im gleichen Modus fortfahren und die Werte in den Datenspeichern oder in einem Programm weiterverwenden. Alle unvollständigen Operationen und Zwischenergebnisse sind jedoch verloren.

BEMERKUNG: Solange der TI-57 LCD ein Programm ausführt, wird die automatische Sparschaltung außer Betrieb gesetzt, um die Ausführung von sehr langwierigen Programmen zu ermöglichen.

©2011 Jeero Weerner
Datamath Calculator Museum

Abschnitt 2 - Dateneingabe-Tasten

Die folgenden Tasten werden für die Eingabe, Löschung und Bearbeitung von Daten verwendet.

[0]-[9] — Ziffern

Die Zifferntasten ermöglichen die Eingabe einer beliebigen Zahl in die Anzeige in der logischen Folge von links nach rechts.

[•] — Dezimalkomma

Der Rechner arbeitet mit einem Gleitkomma, das an jede erforderliche Stelle gesetzt werden kann. Bei Zahlen kleiner 1 ist dem Dezimalkomma eine Null vorangestellt. Nachnullen beim Dezimalteil einer Zahl werden im allgemeinen nicht ausgewiesen, es sei denn, die Anzahl der angezeigten Dezimalstellen wurde mit [2nd] [Fix] auf einen bestimmten Wert festgelegt.

[+/−] — Vorzeichenwechsel

Mit dieser Taste weist man den Rechner an, das Vorzeichen des Anzeigewertes zu wechseln. Damit wird die Verwendung von negativen Zahlen in den Berechnungen möglich.

© 2010 Jörg Woerner
Datamath Calculator Museum

[2nd] [π] — pi Taste

Die [2nd] [π] Taste gibt für die Zahl pi den 11-stelligen Wert 3.1415926536 ein. In der Anzeige erscheint der auf 8 Stellen gerundete Wert 3.1212922.

[ON/C] — Löschtaste

Die Taste [INV] dient zur Löschung von Eingaben und Operationen.
Nach Eingabe einer falschen Zahl, drückt man [ON/C] und gibt die Zahl von Neuem ein. Wenn bereits eine Operationstaste gedrückt wurde, löscht [ON/C] alle eingegebenen unvollständigen Operationen und Operanden. Bei zweimaligem Drücken von [ON/C] werden alle laufenden Operationen und Operanden gelöscht. Die Datenspeicher und Programmregister werden nicht beeinflußt.

[+], [−], [×], [÷], [=] — Arithmetik

Mit diesen 5 Tasten werden die arithmetischen Grundfunktionen Addition, Subtraktion, Multiplikation und Division durchgeführt. Die Taste [=] schließt alle unvollständigen Operationen ab und bereitet den Rechner auf eine neue Aufgabe vor.

In einem Ausdruck können mehrere Operationen kombiniert und in den Rechner in der Folge von links nach rechts wie bei der schriftlichen Formulierung eines Problems eingegeben werden. Der Rechner arbeitet im AOS (Algebraisches Operationssystem), um die Operationen zu sortieren und in der korrekten Reihenfolge durchzuführen.

AOS® Algebraisches Operationssystem

Das Algebraische Operationssystem erlaubt die Eingabe von Zahlen und kombinierten Operationen in den Rechner in der gleichen Reihenfolge, wie der mathematische Ansatz geschrieben wird. Die Problemstellungen werden nach den Standardregeln der algebraischen Hierarchie durchgeführt, die den verschiedenen mathematischen Operationen Prioritäten zuordnet. Ohne verbindliche Regeln könnten Ausdrücke mit mehreren Operationen mehrdeutig sein.

© 2010 Joerg Woerner
Datamath Calculator Museum

Der Ausdruck

$$5 + 4 \times 3 - 2$$

könnte mehrere Lösungen haben. Die Regeln der algebraischen Hierarchie besagen jedoch, daß Multiplikationen und Divisionen vor Additionen und Subtraktionen durchzuführen sind. Bei Anwendung dieser Prioritäten ermittelt der Rechner als korrekte Lösung den Wert 15.
Die obenstehende Gleichung muß bewertet werden: 5 + (4 x 3) -2.

Die vollständige algebraische Hierarchie in absteigender Priorität ist:

1. Einfache Funktionen werden unmittelbar auf den angezeigten Wert angewandt. Dies betrifft Potenzierung, Radizierung, Umkehrfunktion, trighonometrische, logarithmische und Fakultät, Absolutwert, Bruch und Ganzzahlteile, und Umrechnungen.
2. Allgemeine Potenzen und Wurzeln: [y^x] und [INV] [y^x].
3. Multiplikation und Division
4. Addition und Subtraktion
5. Die Gleichheitsanweisung [=] bewirkt die Ausführung aller schwebenden Operationen.

DATENEINGABE- TASTEN

Die Art, in der die Tasten auf der rechten Seite des Rechners angeordnet sind, wird ihnen bei der Einprägung der AOS® Hierarchie behilflich sein.

[y^x]
[÷]
[X]
[−]
[+]
[=]

Operationen mit gleicher Priorität werden von links nach rechts bewertet. Zur Veranschaulichung des Algebraischen Betriebssystems das folgende Beispiel:

4 ÷ 5 + 7 x 3 x 4 - 2 =

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Löschen der Anzeige und laufenden Operationen
4 [÷] 5	5	Division schwebend
[+]	0.8	Die Division wird ausgeführt, da die Addition eine geringere Priorität hat.
7 [X] 3	3	Die Addition und Multiplikation bleiben unvollständig
[X]	21	Die erste Mutiplikation wird ausgeführt
4 [−]	84.8	Die zweite Multiplikation und die Addition werden ausgeführt, die Subtraktion bleibt schwebend
2 [=]	82.8	Die Gleichheitsanweisung schließt alle unvollständigen Operationen ab

© 2010 Joerg Woerner
Datamath Calculator Museum

[(] [)] — Klammern

Einige Berechnungen verlangen die Festlegung der genauen Reihenfolge, in der Zahlen und Operationen zu gruppieren sind. Wenn man eine Reihe von Zahlen und Operationen in Klammern setzt, tritt die normale algebraische Hierarchie in den Hintergrund und der Klammerausdruck muß zuerst berechnet werden. Innerhalb der Klammern arbeitet der Rechner nach den Prioritätsregeln. Klammern sind auch empfehlenswert, wenn Sie Zweifel haben, wie der Rechner einen Ausdruck verarbeiten wird.

Beispiel: $7 \times (3 + 4) = 49$

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Löschen der Anzeige und der unvollständigen Operationen
7 [×] [(] 3 [+] 4 [)]	7	Additionsergebnis Multiplikation unvollständig
[=]	49	Ergebnis

© 2010 Joerg Woerner

Die offene Klammer kann auch angewandt werden, um eine Zahl, die sich schon in der Anzeige befindet, in einen weiteren Rechenvorgang einzubeziehen wie in folgendem Beispiel gezeigt wird:

Beispiel: $4 - (4 + 2) = -2$

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Anzeige und unvollständige Operationen löschen
4 [−] [(] [+]	4	Eingabe der Zahl 4; die offene Klammer, gefolgt von [+], bewirkt die Wiederholung des Wertes 4 innerhalb der Klammer
2 [)] [=]	−2	Ergebnis

Die rechte Klammer ergänzt keine fehlende Zahl. Sie schließt jedoch die Operation bis zurück zur entsprechenden offenen Klammer ab. Wenn keine Klammer mehr offen ist, werden mit der rechten Klammer alle noch unvollständigen Operationen abgeschlossen.

DATENEINGABE- TASTEN

Die zulässige Anzahl der unvollständigen Operationen mit den zugeordneten Zahlen ist limitiert. Maximal 15 Klammern können gleichzeitig geöffnet und maximal 4 Operationen gleichzeitig unvollständig sein, aber nur in sehr komplexen Aufgabenstellungen nähert man sich dieser Grenze.
Die Nichtbeachtung dieser Beschränkungen bewirkt eine Fehlermeldung.

Teilweise werden Gleichungen und Ausdrücke mit Klammern geschrieben, um eine Multiplikation anzugeben: (2 + 1) (3 + 2) = 15. Der Rechner führt keine derart symbolisierte Multiplikationen durch; Sie müssen in diesem Fall das Malzeichen mit eingeben.

[(] 2 [+] 1 [)] [×] [(] 3 [+] 2 [)] [=]

Hier ein Beispiel über die Verwendung von Klammern:

Beispiel: Berechnen Sie $\frac{(8 + 9) \times -19}{(3 + 10) + 7} = -173.92308$

Bei Problemen dieser Art muß der Rechner zunächst den Wert des gesamten Zählers ermitteln, und das Ergebnis durch den Wert des Nenners dividieren. Um die korrekte Ausführung des Problems zu gewährleisten, werden Zähler und Nenner zusätzlich in Klammern gesetzt.

© 2010 Joerg Woerner
Datamath Calculator Museum

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Anzeige und unvollständige Operationen löschen
[(] [(] 8 [+] 9 [)] [×]	17	Anzeige von (8 + 9)
19 [+/−] [)] [÷]	−323	Wert des Zählers
[(] [(] 3 [+] 10 [)] [÷] 7 [)]	1.8571429	Wert des Nenners
[=]	−173.92308	Ergebnis

Anzeigeformen

Obwohl nur maximal 8 Stellen eingegeben und angezeigt werden können, speichert das interne Anzeigeregister die Ergebnisse mit 11 Stellen, um in den nachfolgenden Berechnungen eine größere Genauigkeit zu erhalten. Der angezeigte Wert wird auf acht Stellen gerundet.

Neben der 8-stelligen Standardanzeige mit Gleitkomma gibt es weitere Anzeigeformate, die die Vielseitigkeit des Rechners erhöhen.

[EE] — Exponentialform

In vielen wissenschaftlichen und technischen Berechnungen kommen sehr große oder sehr kleine Zahlen vor, die schwer zu manipulieren sind. Die Exponentialform erleichtert das Arbeiten mit solchen Werten. Jede Zahl läßt sich in der Exponentialform als das Produkt aus einem Basiswert a (Mantisse) und einer Zehnerpotenz (Exponent) ausdrücken. Der Wert, 1.050.000 wird zum Beispiel in der Exponentialform als 1.05×10^6 geschrieben. Das Vorzeichen des Exponenten (+ oder −) gibt an, wo das Dezimalkomma stehen muß, wenn die Zahl in der Standardform geschrieben wird. Bei einem positiven Exponenten wird das Dezimalkomma in der Standardform nach rechts verschoben, bei einem negativen Exponenten nach links. Der Wert des Exponenten bezieht sich auf die Anzahl der Stellen, um die das Komma nach rechts oder links rückt. Die folgenden Tabelle zeigt einige Zahlenbeispiele für Standardform und Exponentialform:

Standardform	Exponentialform
6.789	6.789×10^3
.0000000021	2.1×10^{-9}
−16389043	-1.6389043×10^7
8.775	8.775×10^0

Die Exponentialform erlaubt bei Ihrem Rechner die Arbeit mit Zahlen im Bereich $\pm\ 1 \times 10^{-99}$ bis $\pm\ 9.9999999 \times 10^{99}$. Zahlen, die kleiner sind als $\pm\ 1 \times 10^{-7}$ und größer als $\pm\ 9.9999999 \times 10^7$, müssen in der Exponentialform eingegeben werden. Wenn bei Berechnungen diese Grenzen überschritten wurden, zeigt der Rechner das Ergebnis automatisch in Exponentialform an.

DATENEINGABE- TASTEN

Um eine Zahl in Exponentialform einzugeben, tastet man zunächst die Mantisse (einschließlich [+/−] bei negativer Mantisse) ein. Dann drückt man [EE] und rechts in der Anzeige erscheint der Wert 00. Schließlich folgt die Eingabe des Exponenten und dessen Vorzeichen. Eine falsche Exponenteneingabe können Sie durch einfaches Drücken der korrekten Ziffern korrigieren. Der Rechner ersetzt die alten Werte durch die zuletzt eingegebenen Ziffern.

Beispiel: Sie wollten 6.023×10^{23} eingeben, haben aber versehentlich die Ziffern des Exponenten in umgekehrter Reihenfolge eingetastet.

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Anzeige und unvollständige Operationen löschen
6.023 [EE] 32	6.023 32	Fehler bei der Exponenteneingabe
3	6.023 23	Die neue Eingabe verschiebt die Exponenten und korrigiert den Fehler

© 2010 Joerg Woerner
Datenmuseum Joerg Woerner

Unabhängig davon, wie eine Mantisse in der Exponentialform eingegeben wird, normalisiert der Rechner die Zahl in der Form, daß links vom Dezimalkomma eine einzelne Stelle hervorgehoben wird, wenn man eine Funktions- oder Operationstaste drückt.

Nach Drücken der Taste [EE] werden alle Ergebnisse in Exponentialform angezeigt. Um die Exponentialform aufzuheben oder eine Zahl in Standardform zu bringen, drückt man [INV] [EE]. Auch durch [ON/C] oder durch kurzes Ausschalten des Rechners wird die Exponentialform aufgehoben. Liegt die angezeigte Zahl nicht im Bereich $\pm\ 1 \times 10^7$ bis $\pm\ 9.9999999 \times 10^7$, schaltet sich der Rechner erst in die Standardform, wenn das Ergebnis oder die Eingabe im Anzeigebereich liegen.

Beispiel: Geben Sie 32.5×10^4 in der Exponentialform ein, und wechseln Sie dann zu Standardform.

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Anzeige und unvollständige Operationen löschen
32.5 [EE] 4	32.5 04	Eingabe
[=]	3.25 05	Exponentialform
[INV] [EE]	325000	Standardform

DATENEINGABE- TASTEN

Für schnellere Berechnungen können Dateneingaben in Standardform und Exponentialform gemischt werden. Der Rechner wandelt die Standardzahlen um und zeigt die Ergebnisse in Exponentialform an.

Beispiel: $3.2 \times 10^3 + 12575.321 = 15775.321$

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Anzeige und unvollständige Operationen löschen
3.2 [EE] 3	3.2 03	Eingabe der ersten Zahl
[+] 12575.321	12575.321	Addition der zweiten Zahl
[=]	1.5775321 04	Ergebnis in Exponentialform
[INV] [EE]	15775.321	Ergebnis in Standardform

[2nd] [Fix] n — Festkomma

In einigen Berechnungen ist die Anzeige einer festen Anzahl von Dezimalstellen von Vorteil, gleich, ob eine Zahl in Standardform, oder Exponentialform dargestellt wird. Mit [2nd] [Fix] n wird der Rechner angewiesen, die Anzeige auf n Dezimalstellen zu runden.
Das interne Anzeigeregister hält die Zahlen nach wie vor mit 11-stelliger Genauigkeit für nachfolgende Berechnungen gespeichert.

© 2010 Vereinigung zur Erhaltung des Rechenzentrums Konrad Zuse e.V.

In Verbindung mit der Exponentialform bezieht sich die Festkomma-Einstellung [2nd] [Fix] n auf die Anzahl der Dezimalstellen in der Mantisse.

Ist bei einem gewählten Festkomma ein Rechenergebnis größer als $\pm\ 9.9999999 \times 10^7$ oder kleiner als $\pm\ 1 \times 10^{-7}$, schaltet sich die Anzeige automatisch in die Exponentialform, und die Festkomma-Einstellung wird solange ignoriert, bis die Exponentialform nicht mehr erforderlich ist.

Beispiel:

Taste	Anzeige
[ON/C] [ON/C]	0.
[2nd] [π]	3.1415927
[2nd] [Fix] 2	3.14
[2nd] [Fix] 4	3.1416
[INV] [2nd] [Fix]	3.1415927

Abschnitt 3 - Algebraische Tasten.

In diesem Abschnitt werden Sie mit den leistungsfähigen algebraischen Funktionen Ihres TI-57 LCD vertraut gemacht.

BEMERKUNG: Beschränkungen des Bereiches und der Genauigkeit dieser Funktion werden im Anhang besprochen.
Werte die die erlaubten Bereiche überschreiten, veranlassen eine Fehlermeldung.

[1/x], [x²], [√x] — Umkehrfunktion, Potenzierung und Radizierung
Diese drei Tasten bewirken die sofortige Bearbeitung der angezeigten Zahl ohne weiteren Einfluß auf laufende Operationen.
[1/x] 1 wird durch die angezeigte Zahl dividiert.
[x²] Berechnet das Quadrat der angezeigten Zahl durch Multiplizierung dieser Zahl mit sich selbst.
[√x] berechnet die Quadratwurzel der angezeigten Zahl. Die Quadratwurzel einer Zahl (x) wird mit $\sqrt{x}$ bezeichnet und hat die Eigenschaft, daß $\sqrt{x}$ mal $\sqrt{x}$ gleich x. Die angezeigte Zahl muß positiv sein.

Datamath Calculator Museum

[yˣ], [INV] [yˣ] — Allgemeine Potenzen und Wurzeln

Mit y^x wird eine beliebige positive Zahl in eine beliebige Potenz erhoben. Die Anwendung dieser Taste:

- Eingabe der Zahl, die in eine Potenz erhoben werden soll ("y")
- Taste [yˣ] drücken
- Eingabe der Potenz ("x")
- Taste [=] oder eine beliebige Operationstaste drücken

Beispiel: Berechnen Sie $3.1897^{4.7343}$

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Anzeige und unvollständige Operationen löschen
3.1897 [yˣ]	3.1897	Wert von y
4.7343	4.7343	Wert von x
[=]	242.60674	Ergebnis: y^x

[INV] [yˣ] ist die Tastenfolge zur Berechnung allgemeiner Wurzeln. Sie erlaubt die Erhebung einer beliebigen positiven Zahl in eine beliebige Wurzel($\sqrt[x]{y}$). Diese Funktion wird folgendermassen ausgeführt.

- Eingabe der Zahl y, deren Wurzel errechnet werden soll
- Taste [INV] [y^x] drücken
- Eingabe der Wurzel (x)
- Taste [=] oder eine andere Operationstaste drücken

Beispiel: Berechnen Sie $\sqrt[3.871]{21.496}$

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Anzeige und unvollständige Operationen löschen
21.496 [INV] [y^x]	21.496	Wert von y
3.871	3.871	Wert von x
[=]	2.2089685	Ergebnis: $\sqrt[x]{y}$

BEMERKUNG: Siehe Anhang B für Genauigkeitsbeschränkungen

Denken Sie daran, daß [y^x] und [INV] in der AOS Hierachie höhere Priorität haben.

© 2010 Joerg Woerner
Datamath Calculator Museum

Betrachten Sie folgendes Beispiel:
$4 + 6 \times 2^3 \div 7 = 10.857143$

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	
[4] [+]	4	
[6] [X]	6	Addition unvollständig
[2] [y^x]	2	Addition und Multiplikation unvollständig
[3] [÷]	48	[y^x] und Multiplikation ausgeführt
[7] [=]	10.857143	[=] beschließt die Berechnung

[lnx], [log], [INV] [lnx], [INV] [log] — Logarithmus und Antilogarithmus (Numerus)

Logarithmen sind mathematische Funktionen, die in einer Vielzahl von technischen und theoretischen Problemen Anwendung finden. Daneben bilden sie einen wesentlichen Teil vieler mathematischer "Modelle" von Naturerscheinungen. Die Logarithmentasten ermöglichen den unmittelbaren Zugriff auf den Logarithmus einer beliebigen Zahl, ohne in einer Tabelle nachschlagen zu müssen.

Die Taste [lnx] weist den natürlichen Logarithmus (zur Basis e = 2.7182818) der Zahl in der Anzeige aus. Diese Zahl muß positiv sein.

Die Taste [log] ermittelt den dekadischen Logarithmus (zur Basis 10) der Zahl in der Anzeige. Diese Zahl muß positiv sein.

Die Antilogarithmus-Tasten (Numerus-Tasten) [INV] [lnx] und [INV] [log] erheben e bzw. 10 in die Potenz der Zahl in der Anzeige.

Beispiel: Berechnen Sie log 15.32, ln 203.451, $e^{-.69315}$ und 10^{π}.

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Anzeige und unvollständige Operationen löschen
15.32 [log]	1.1852588	
203.451 [lnx]	5.3154252	
.69315 [+/−] [INV] [lnx]	0.4999986	
[2nd] [π] [INV] [log]	1385.4557	

© 2010 Joerg Woerner
Datamath Calculator Museum

[2nd] [n!] — **Fakultät Tastenfolge**

Die Fakultät jeder positiven Ganzzahl (n) wird n! geschrieben und ist gleich 1 x 2 x 3 ... n. 0! ist vereinbarungsgemäß 1.

[2nd] [n!] ermittelt die Fakultät der angezeigten Zahl, welche eine positive Ganzzahl kleiner als 70 sein muß.

[2nd] [|x|], [2nd] [Intg], [2nd] [Frac] — **Teilzahlen**

Mit [2nd] [|x|] wird der Absolutwert einer Zahl, d.h., die reine Größenordnung unabhängig vom Vorzeichen, berechnet und angezeigt. Das Ergebnis von [2nd] [|x|] ist also immer eine positive Zahl.

[2nd] [Intg] kürzt die Zahl im Anzeigeregister durch Abschneiden seines Dezimalteiles.

BEMERKUNG: Bei negativen Zahlen, müssen Sie 1 von dem angezeigten Ergebnis abziehen, um die mathematische Regel zu befolgen, die besagt, daß der Ganzzahlteil einer Zahl gleich der nächstliegenden kleineren oder gleichen Ganzzahl ist. Der Ganzzahlteil von -4.5 ist -5, obwohl der Rechner -4 zeigt.

[2nd] [Frac] zeigt den Bruchteil (Dezimalteil) der Zahl im Anzeigeregister an, und löscht den ganzzahligen Teil. Siehe folgende Anmerkung.

Anmerkung: [2nd] [Intg] *und* [2nd] [Frac] *arbeiten mit den 11 internen Stellen im Anzeigeregister, nicht mit dem 8-stelligen Wert in der Anzeige. Das heißt: wenn der Wert 4.9999999999 intern im Anzeigeregister gespeichert ist (der in der Anzeige selbst auf 5 gerundet wird), bleibt beim Drücken von* [2nd] [Intg] *die Zahl 4 in der Anzeige. Ähnlich ergibt sich bei der Funktion* [2nd] [Frac] *das Resultat .9999999, wenn der intern gespeicherte Wert .9999999999 für die Anzeige auf 1 gerundet wurde.*

[DRG], [INV] [DRG], [2nd] [DRG▸], [INV] [2nd] [DRG▸] — Grad, Radiant und Gon (Neugrad)

Ihr Rechner verarbeitet eine Vielzahl von Problemen mit Winkeln, zum Beispiel trigonometrische Funktionen und Umrechnungen von Polarkoordinaten in rechtwinklige Koordinaten und umgekehrt. Bei der Durchführung dieser Berechnungen wählen Sie eine der drei Winkeleinheiten.

© 2010 Joerg Woerner
Datamath Calculator Museum

Ein Grad ist 1/360 eines Kreises. Ein rechter Winkel beträgt 90°.

Ein Radiant ist 1/2π eines Kreises. Ein rechter Winkel beträgt π/2 Radiant.

Ein Gon (Neugrad) ist 1/400 eines Kreises. Ein rechter Winkel beträgt 100 Gon.

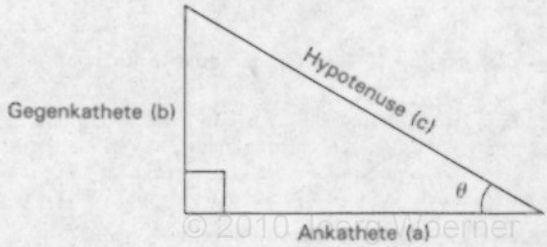
Beim Einschalten ist der Rechner immer auf Grad eingestellt (kein symbol in der Anzeige). Mit der Taste [DRG] schalten Sie auf die Winkeleinheit Radiant um (Symbol RAD in der Anzeige). Durch erneutes Drücken von [DRG] wird die Winkeleinheit Gon (Symbol GRAD in der Anzeige) gewählt. Wenn Sie [DRG] ein drittes Mal drücken, schaltet sich der Rechner wieder auf Grad.

Sie können die einzelnen Winkeleinheiten auch in umgekehrter Reihenfolge — Grad-Gon-Radiant-Grad — mit [INV] [DRG] eingeben.

Auch die Taste [2nd] [DRG▸] ändert die angezeigte Winkeleinheit, und rechnet zusätzlich die Zahl in der Anzeige auf die neuen Einheiten um. So wechselt bei 90 Grad, gefolgt von [2nd] [DRG▸] die Winkeleinheit auf Radiant, und die Anzeige auf den Wert 1.5707963 (π/2). Drückt man [2nd] [DRG▸] ein zweites Mal, schaltet der Rechner auf die Winkeleinheit Gon um und der Wert 100 wird angezeigt. Mit [INV] [2nd] [DRG▸] können Sie die einzelnen Winkeleinheiten und Umrechnungen auch in umgekehrter Reihenfolge durchgehen — Grad-Gon-Radiant-Grad.

[sin], [cos], [tan], [INV] [sin], [INV] [cos], [INV] [tan] — Trigonometrische Funktionen

Mit den Funktionstasten [sin], [cos] und [tan] berechnet man den Sinus, Kosinus und Tangens des angezeigten Winkels, wobei der Winkel in den Einheiten ausgedrückt ist, die mit den Tasten [DRG], [INV] [DRG], [2nd] [DRG•], oder [INV] [2nd] [DRG•] gewählt wurden. Die trigonometrischen Funktionen beziehen sich auf die Winkel und Seiten eines rechtwinkligen Dreiecks wie folgt:



© 2010 Joerg Woerner
Datamath Calculator Museum

$$\sin\theta = \frac{b}{c} \qquad \cos\theta = \frac{a}{c} \qquad \tan\theta = \frac{b}{a}$$

Die Umkehrfunktionen (Arkusfunktionen) ermitteln den Winkel in den jeweils gewählten Einheiten, dessen Sinus, Kosinus oder Tangens angezeigt wird. [INV] [sin] berechnet den Arkussinus ($\sin^{-1}$), [INV] [cos] den Arkuskosinus ($\cos^{-1}$), und [INV] [tan] den Arkustangens ($\tan^{-1}$).

Beachten Sie die folgenden Winkelbereiche, die aus den Arkusfunktionen resultieren.

Arkusfunktion	Bereich des resultierenden Winkels
arc sin x	0 bis 90°, π/2 Radiant, oder 100 g
arc sin (−x)	0 bis −90°, −π/2 Radiant, oder −100 g
arc cos x	0 bis 90°, π/2 Radiant, oder 100 g
arc cos (−x)	90° bis 180°, π/2 bis π Radiant, oder 100 g bis 200 g
arc tan x	0 bis 90°, π/2 Radiant, oder 100 g
arc tan (−x)	0 bis −90°, −π/2 Radiant, oder −100 g

[2nd] [DMS.DD], [INV] [2nd] [DMS.DD] — Umrechnungen Grad/Minuten/Sekunden in Dezimalgrad

Es gibt zwei Möglichkeiten, einen Winkel in Grad darzustellen. Eine Methode ist das Dezimalformat GG.dd.DD ist hier der ganzzahlige Teil des Winkels, dd der Dezimalteil. Bis zu 8 Stellen können eingegeben werden.

Die zweite Methode ist das Grad/Minuten/Sekunden-Format GG.MMSSss. Wieder bezieht sich DD auf den ganzzahligen Gradwert des Winkels. MM ist das Symbol für Minuten, und SS bedeutet Sekunden. In der Position ss können im Interesse noch größerer Genauigkeit Sekundenbruchteile eingegeben werden. Das Dezimalkomma trennt die Grad von den Minuten.

Um Grad/Minuten/Sekunden in das Dezimalformat zu bringen, geben Sie den Winkel in der Anzeige in der Form GG.MMSSss ein, und drücken [2nd] [DMS.DD]. Die Tastenfolge [INV] [2nd] [DMS.DD] kehrt das Verfahren um, und rechnet Dezimalgrad in das Grad/Minuten/Sekunden-Format um. Zwei Stellen müssen immer für Minuten und zwei für Sekunden eingegeben werden. Die Eingabe von Nachnullen ist nicht erforderlich.

© 2010 Jörg Woerner
Datamath Calculator Museum

Beispiel: Rechnen Sie 3°1'30.456" in das Dezimalgrad-Format um und umgekehrt.

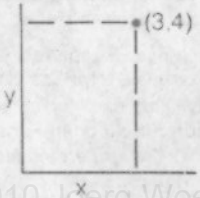
Eingabemethode:

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Anzeige und unvollständige Operationen löschen; [DRG] drücken, bis DEG eingeschaltet ist
3.0130456 [2nd] [DMS.DD]	3.0251267	Dezimalgrad
[INV] [2nd] [DMS.DD]	3.0130456	Erneute Umrechnung in Grad/Minuten/Sekunden

Dasselbe Verfahren kann man für die Umwandlung von Stunden/Minuten/Sekunden in das Dezimalformat anwenden.

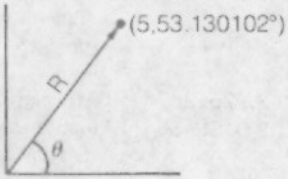
[2nd] [P→R], [INV] [2nd] [P→R] — Umrechnungen Polarkoordinaten/Rechtwinklige Koordinaten

Das rechtwinklige Koordinatensystem beschreibt mit Hilfe eines Zahlenpaares, wo sich einzelne Punkte in einem Rasterfeld befinden. Die erste Zahl, die x-Koordinate (Abszisse), gibt den Abstand des Punktes von der y-Achse (vertikale Linie) an. Die zweite Zahl bezieht sich auf die y-Koordinate (Ordinate), und sie beschreibt den Abstand des Punktes von der x-Achse (horizontale Linie). Die folgende Abbildung zeigt den Punkt (3,4) im rechtwinkligen Koordinatensystem.



© 2010 Joerg Woerner
Datamath Calculator Museum

Das Polarkoordinaten-System beschreibt einen Punkt mit Hilfe eines Richtungsstrahls, der vom Pol O ausgeht. Auch hier verwendet man ein Zahlenpaar. Die erste Zahl gibt den Abstand R an, die zweite Zahl bezieht sich auf den Richtungswinkel θ. Die folgende Abbildung zeigt den Punkt (5,53.130102°).



Die Umrechnung von Polarkoordinaten in rechtwinklige Koordinaten und umgekehrt erfordert viel Arithmetik, die der Rechner intern aber ohne Schwierigkeiten verarbeiten kann.

Um Polarkoordinaten in rechtwinklige Koordinaten umzuformen, gehen Sie wie folgt vor :

— Wert von R eingeben

Taste [x ⇄ t]

θ eingeben

Taste [2nd] [P→R]

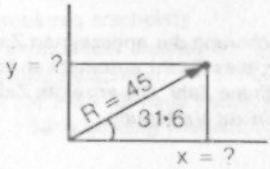
Die y- Koordinate wird angezeigt.
Auf [x⇄t] drücken, um die x- Koordinate in die Anzeige zu holen.

- Folgende Schritte zeigen, wie man rechtwinklige Koordinate in Polarkoordinate umrechnet.
  Die x- Koordinate eingeben
  [x⇄t] drücken
  Die y- Koordinate eingeben
  [INV] [2nd] [P→R] drücken.

Der θ Wert wird in der durch die [DRG] Taste gewählten Einheit angezeigt.
[x⇄t] drücken, um den R - Wert in die Anzeige zu holen.
Der Bereich von θ erstreckt sich von + 180 bis -180 Grad, pi bis -pi Radiant, und 200 bis -200 Gon (Neugrad).

Achtung: Die Folgen [2nd] [P→R] und [INV] [2nd] [P→R] löschen schwebende Operationen.

© 2010 Joerg Woerner
Datamath Calculator Museum



Beispiel: R = 45 Meter, θ = 31.6 Grad in rechtwinklige Koordinate umrechnen.

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C]	0	Löscht die Anzeige und unvollständige Operationen. [DRG] drücken, um den Gradmodus zu wählen
45 [x⇄t] 31.6 [2nd] [P→R]	23.579366	y - Koordinate
[x⇄t]	38.327712	x - Koordinate

Abschnitt 4 — Speicherfunktionen

Der TI-57 LCD kann bis zu 7, von 0 bis 6 nummerierte, Datenspeicher benutzen, die Zahlen für späteren Gebrauch speichern können (Zwischenergebnisse, Festwerte). Wegen des "Constant Memory" des Rechners, geht der Inhalt dieser Datenspeicher nicht verloren, wenn der Rechner abgeschaltet wird. Dies erlaubt, Daten in Ihrem Rechner festzuhalten.
Der folgende Abschnitt beschreibt die Speicherfunktionen.

[2nd] [Part] m — Speicherbereichsverteilung
Die [2nd] [Part] Tastenfolge teilt den Speicher des Rechners in Datenspeicher und Programmspeicher. Diese Folge muß mit der Zahl der gewünschten Datenspeicher (1 bis 7) beschlossen werden. Zum Beispiel, um über alle sieben Datenspeicher zu verfügen, muß man [2nd] [Part] 7 drücken.
Bitte lesen Sie das Kapitel II, um mehr Auskünfte über diese wichtige Fähigkeit zu erhalten.

Copyright © 2016 Rick Bensa
Hamath Calculator Museum

[STO] m — Abspeichern
Die [STO] Taste erlaubt die Speicherung der angezeigten Zahl in dem mit m angesprochenen Datenspeicher, dies löscht automatisch jede zuvor in diesem Datenspeicher festgehaltene Zahl. Die anzeigte Zahl bleibt unbeeinflußt und steht weiterhin zur Verfügung.

[RCL] m — Speicherückruf
Die [RCL] Taste ruft die im Datenspeicher m festgehaltene Zahl in die Anzeige zurück. Der Inhalt dieses Speichers bleibt unberührt. Die Zahl, die sich zuvor in der Anzeige befand, ist verloren.

[EXC] m — Speicheraustausch
Die [EXC] Taste vertauscht die Zahl in der Anzeige mit dem, der im Datenspeicher m festgehalten wird. Dies erlaubt den Rückruf des Speicherinhaltes ohne Verlust der angezeigten Zahl.

[2nd] [CM] — Speicherlöschung
Die [2nd] [CM] Tastenfolge löscht alle Datenspeicher. Die Anzeige und der Programmspeicher werden nicht beeinflußt.

Beispiel.

Taste	Anzeige	Speicher 0	Bemerkungen
[ON/C] [ON/C]	0	-	Anzeige und unvollständige Operationen löschen
[2nd] [CM]	0	0	Alle Speicher löschen
3 [STO] 0	3	3	3 in Speicher 0 legen
5	5	3	5 in die Anzeige geben
[EXC] 0	3	5	Anzeige und Speicher 0 vertauschen
[RCL] 0	5	5	Rückruf Speicher 0 in die Anzeige

BEMERKUNG: Um einen einzelnen Speicher zu löschen, drückt man auf [ON/C] [STO] m (mit m der Nummer des zu löschenden Datenspeichers).

© 2010 Joerg Woerner
Datamath Calculator Museum

SPEICHERARITHMETIK

Zusätzlich zu den Basisspeicherfunktionen, die Sie eben sahen, erlaubt der TI-57 LCD die Ausführung von Rechnungen mit den gespeicherten Zahlen, ohne daß die angezeigte Zahl und die laufende Berechnung davon gestört werden. Diese Fähigkeit wird folgendermaßen angewandt:

- Eingabe der Zahl, die als Operand für den Speicherwert dienen soll
- Taste [STO] drücken
- Eingabe der gewünschten Operation
- Eingabe der gewünschten Speichernummer

ANMERKUNG: der Dauerspeicher (Constant Memory™) bewirkt, daß die Datenspeicher nicht gelöscht werden, wenn man den Rechner ausschaltet. Drücken Sie also immer [ON/C] [STO] m, um den gewünschten Datenspeicher zu löschen, ehe Sie eine der nachstehenden Tastenfolgen anwenden. [2nd] [CM] löscht die Gesamtzahl der durch die Bereichsverteilung definierten Datenspeicher.

SPEICHERFUNKTIONEN

[STO] [+] m — Addition des Anzeigewerts zum Inhalt des Datenspeichers m
[STO] [−] m — Subtraktion des Anzeigewerts vom Inhalt des Datenspeichers m
[STO] [X] m — Multiplikation des Datenspeicherinhalts m mit dem Anzeigewert
[STO] [+] m — Division des Datenspeicherinhalts m durch den Anzeigewert
[STO] [y^x] m — Der Inhalt des Datenspeichers m wird in die Potenz in der Anzeige erhoben
[STO] [INV] [y^x] m — Berechnung der x-ten Wurzel (x = Anzeigewert) des Datenspeicherinhalts m

Beispiel:

28.3 x 7	=	198.1
173 + 16	=	189
31 − 42 + 7.8	=	−3.2
	Total	383.9

Taste	Anzeige	Bemerkungen
[ON/C] [ON/C] [STO] 0	0	Anzeige, unvollständige Operationen und Speicher 0 löschen
28.3 [X] 7 [=] [STO] [+] 0	198.1	Das Ergebnis der ersten Aufgabe wird zum Daten-Speicher 0 addiert
173 [+] 16 [=] [STO] [+] 0	189	Das ergebnis der zweiten Aufgabe wird zum Datenspeicher 0 addiert
31 [−] 42 [+] 7.8 [=] [STO] [+] 0	−3.2	Das Ergebnis der dritten Aufgabe wird zum Datenspeicher 0 addiert
[RCL] 0	383.9	Summe aller Resultate

© 2010 Joerg Woerner
Datamath Calculator Museum

[x ⇄ t] — x Austausch mit "t"
Dies ist eine Sondertaste, die die angezeigte Zahl mit dem Inhalt eines besondern Datenregisters vertauscht, welches "t" oder "Test" Register genannt wird. Das "t" Register findet seine Anwendung für Rechtwinklige-Polar Umrechnungen und für bedingte Sprünge, wie in Kapitel II beschrieben.

[2nd] [Ct] — "t" Register löschen
Die [2nd] [Ct] Tastenfolge erlaubt die Löschung des Inhaltes des "t" Registers.

BEMERKUNG: Die im "t" Register gespeicherte Zahl kann nur durch die folgenden Tasten beeinflußt werden: [x ⇄ t], [2nd] [Ct], [2nd] [P/R] und [INV] [2nd] [P/R].

Kapitel II — Programmieren des TI-57 LCD

Unter einem Programm versteht man den Ablauf von Einzeltätigkeiten, die man zuvor genau überlegt hat, so daß man von einem Anfangszustand in einen gewünschten Endzustand kommt.
Jeder Mensch absolviert während seines Tageslaufes eine Vielzahl von Programmen, z.B. Morgentoilette machen, Kaffee zubereiten, Auto starten etc.

Die hier genannten Programmbeispiele sind sehr grob beschrieben, denn zum "Kaffeekochen" gehören bestimmte Unterprogramme und dazu vielleicht weitere Unter-Unterprogramme. Es ist eine Sache der Übereinkunft und des Vorwissens, wie weit man die Einzeltätigkeiten unterteilen und zur Programmbeschreibung heranziehen will.

In der Mathematik gibt es zur Lösung der unterschiedlichsten Probleme häufig feste Programme. So könnte man jede Formel als ein mathematisch formuliertes Programm auffassen, z.B. den Lehrsatz des Pythagoras

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

© 2010 Joerg Woerner
Datamath Calculator Museum

Mit einer bestimmeten Abfolge von tastendrücken auf dem Taschenrechner läßt sich diese Aufgabe lösen.

Je nach Umfang oder Schwierigkeit einer Aufgabe werden die Tastenprogramme unterschiedlich lang. Die Aufstellung eines Tastenprogramms erfordert anspruchsvolle geistige Tätigkeit; das immer sich wiederholende Abtippen eines solchen Programms ist dagegen eine langweilige und ziemlich geistlose Tätigkeit, die man gern — wie alle stupiden Arbeiten — Maschinen überlassen möchte; Maschinen ermüden nicht, sind schneller als Menschen und machen kaum Fehler.

Der TI-57 LCD ist ein solches Maschinchen, das Tastenprogramme so oft man will fehlerlos abarbeiten kann.

PROGRAMMIEREN DES TI-57 LCD

Zur vollständigen Lösung einer Programmieraufgabe gehören folgende Einzelschritte:

1. Problem eindeutig und klar formulieren
2. Problem mathematisieren (z.B. Lösungsformel aufstellen)
3. Programmieren, d.h. Tastenprogramm aufstellen, Speicherverteilung überlegen, nach Einsparmöglichkeiten bei knappen Speicherraum suchen etc.
   Dieser Punkt 3 ist das Kernstück des gesamten Arbeitskomplexes.
4. Tastenprogramm einlesen
5. Benutzeranweisung schreiben
6. Probeläufe durchführen

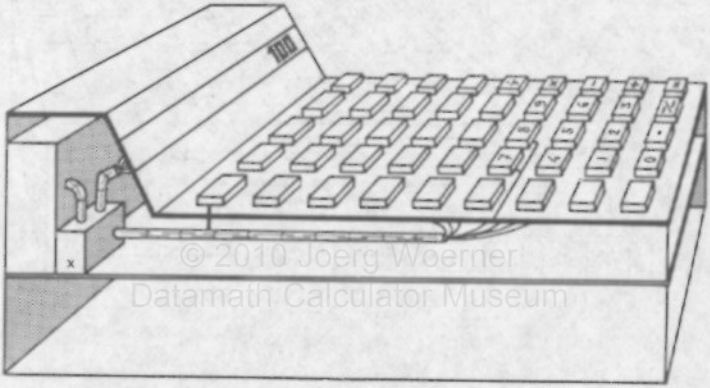
Die hier aufgeführten Einzelschritte werden in den meisten der nachfolgenden Programmierbeispiele eingehalten und deutlich gekennzeichnet.

Die Programmiertasten werden jeweils mit entsprechenden Beispielen erläutert. Am Ende des Programmierlehrgangs (Seite 99) wird für jede Taste eine Kurzerläuterung angegeben.

Datamath Calculator Museum

Aufbau des TI-57 LCD

So wie es z.B. in der Physik üblich ist, schwierige Zusammenhänge an vereinfachenden Modellen zu veranschaulichen, kann man auch den TI-57 LCD modellhaft darstellen; man muß nur wissen, daß die elektronische Wirklichkeit anders aussieht als das Modell.

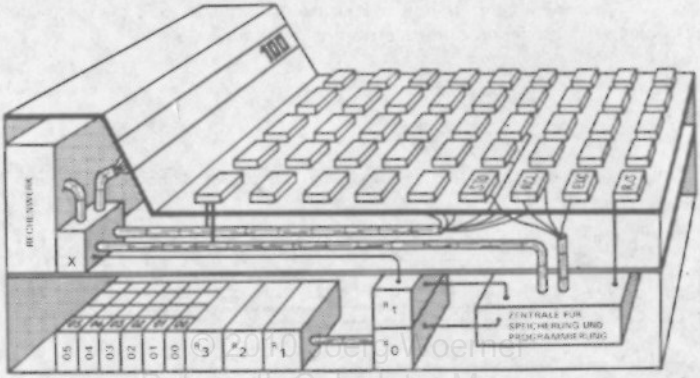


© 2010 Joerg Woerner
Datamath Calculator Museum

Abb. 1.

Der TI-57 LCD hat sozugagen zwei "Etagen", in denen mehrere Abteilungen untergebracht sind, die zwar selbständig und unabhängig voneinander arbeiten können, aber doch in bestimmter Weise untereinander in Verbindung stehen.
In der oberen Etage ist alles das, was eine "normaler" Taschenrechner auch hat: Dis Tastatur, die Leitungen, die von den einzelnen Tasten in der "Hauptleitung" gebündelt werden und ins Anzeige-Register führen, das mit "x" gekennzeichnet ist.
Vom Anzeigeregister führt eine Hauptleitung ins Rechenwerk und eine Hauptleitung in die Anzeige.
Im Rechenwerk werden sämtliche Rechnungen durchgeführt, von einer einfachen Addition bis zur Lösung einer komplizierten Gleichung. Auf dem Bild sind nicht alle Leitungen eingezeichnet, weil es sonst zu unübersichtlich geworden wäre.
Wir nennen diese "Etage" die RUN-Ebene (engl. to run = rennen, laufen), weil hier jeder Tastendruck sofort die entsprechenden Befehle durchführen läßt.

In der zweiten Ebene befinden sich die wesentlichen Teile Für Speicherung und Programmierung.



© 2010 Joerg Woerner
Datamath Calculator Museum

Abb. 2.

Man unterscheidet zwischen Datenspeichern und Programmspeichern. Die Datenspeicher werden auch "Register" genannt und mit "R" abgekürzt. Auf dem Bild sind die Register R_0, R_1, R_2, R_3, und die beiden Register mit den Bezeichnungen "x" und "t" zu sehen. R_x ist das uns schon bekannte Anzeigeregister. Es ist das einzige Register, in das man gewissermaßen hineinsehen kann, denn der jeweilige Inhalt wird in der Anzeige ausgewiesen, mit der Einschränkung, daß in R_x eine 11-stellige Zahl gespeichert sein kann, aber in der Anzeige nur 8 Stellen erscheinen. Bei der Division 1 : 7 etwa weist die Anzeige 0.1428571 aus, während in R_x 0.1428571429 gespeichert wird.
R_t hat besondere Aufgaben bei bestimmten Programmiertechniken (s.S.74) kann aber unter Berücksichtigung einiger Besonderheiten auch als Datenspeicher benutzt werden (s. Programmierbeispiel 1.4 und folgende).
R_0 ist ein gewöhnlicher Datenspeicher, kann aber für eine besondere Programmiertechnik benutzt werden

Neben R_3 beginnen die Programmspeicher, die viel kleiner sind als die Register: Je 8 Programmspeicher sind so groß wie ein Register. Man erkennt 24 Programmspeicher, die mit 00 bis 23 numeriert werden (wegen der Übersichtlichkeit sind nur die Programmspeicher 00 bis 05 numeriert). Es ist üblich, die einzelnen Programmspeicher mit "step" zu bezeichnen (engl.: step = Schritt).
Programmspeicher und Datenspeicher (Register) haben gewisse Gemeinsamkeiten, aber auch wesentliche Unterschiede:

1. In einem Register läßt eine beliebig große Zahl (gemäß der Rechnerkapazität) speichern.
   In einem Programmspeicher läßt sich nur eine einstellige Zahl oder eine Ziffer einer mehrstelligen Zahl speichern. Die Zahl 3,1416 z.B. würde 6 steps in Anspruch nehmen, denn auch der Dezimalpunkt erfordert einen step.
2. In einem Programmspeicher (Step) läßt sich ein Befehl (Tastendruck) speichern, und das ist in einem Register nicht möglich. Die Programmspeicher sind also der Ort im Rechner, in den die Tastenprogramme eingelesen werden.

© 2010 Joerg Woerner
Datamath Calculator Museum

In der Programmspeicherabteilung muß man sich noch einen Programmzeiger vorstellen, der auf einen bestimmten Tastendruck hin losläuft, von Step zu Step springt und die dort gespeicherten Tastendrücke so ausführt, als tippe sie der Anwender mit dem Finger auf der Tastatur. Eine "Zentrale" sorgt dafür, daß in der Speicherabteilung alles folgerichtig abläuft.
In diesem Kapitel "Programmieren" werden bestimmte Tastengruppen unter folgenden Bezeichnungen zusammengefaßt:

1. Zahlentasten (0 bis 9 , einschl. Dezimalpunkt)
2. Operationstasten (+ , − , × , ÷ , = und die Klammertasten ( )).
3. Funktionstasten (sin , cos , tan , x^2 , log , P→R etc)
4. Programmiertasten (R/S , RST , LBL , Pause , x=t , x≥t , GTO , SBR , Dsz)
5. Prozedurstaten (LRN , Part , SST , BST , Del , CP)
6. Speichertasten (STO , RCL , STO + , STO − , EXC , CM , x⇄t etc)
7. Einzeltasten, die sich in 1. bis 6. nicht einordnen lassen (OFF , 2nd , INV , ON/C)

Die Gruppen 1,2,3,6 und 7 sind dem Leser aus den vorhergegangenen Kapiteln bekannt; dieser Programmierteil befaßt sich im wesentlichen mit den Gruppen 4 und 5, wobei aber auch die anderen Gruppen, besonders die Gruppe der Speichertasten, einbezogen werden.

Die Programmiertasten steuern den Lauf des Programmzeigers durch die Tastenprogramme;

Die Prozedurtasten dienen der Vorbereitung für die Programmierung, der Eingabe der Tastenprogramme sowie der Kontrolle und Berichtigung. Sie können selbst nicht als Befehl in einen Step eingespeichert werden; sie haben daher auch keine Code-Nr. (Code-Nr. S.S. 122)

Die Prozedurtasten [Part] [LRN] [CP] [SST] [BST]

1. [Part]

Eine Besonderheit des TI-57 LCD ist es, daß man Datenspeicher in Programmspeicher umwandeln kann und umgekehrt. Betrachten Sie Abb. 2 : Man sieht die Register R1, R2, R3 zusammen mit den Programmspeichern Step 00 bis Step 23. In diesem Speicherblock können die Umwandlungen vorgenommen werden. R0 und Rt werden davon nicht berührt.
Man kann diesen Speicherblock maximal in 6 Register umwandeln, so daß mit R0 zusammen 7 vollwertige Datenspeicher zur Verfügung stehen (Rt und Rx werden in diesem Zusammenhang nicht mitgezählt).
Allerdings stehen bei einer solchen Speicherverteilung keine Programmspeicher mehr zur Verfügung, so daß keine Tastenprogramm eingelesen werden kann.
Andererseits kann man 48 Steps einteilen. Man hätte dann noch R0 (und Rt mit der diesem Speicher eigenen Einschränkung) als Datenspeicher zur Verfügung.
Die Speicherverteilung wird mit der Taste [2nd] [Part] m (m ist eine Zahlentaste mit $1 \leqslant m \geqslant 7$) geregelt. Wird m außerhalb des angegebenen Bereichs gewählt, erfolgt eine Fehlermeldung.

Tippe [2nd] [Part] [4] Anzeige: 24.4

Das bedeutet, daß 24 Steps und 4 Register eingerichtet sind.

Allgemein: Mit [2nd] [Part] m stehen m Register und 8 × (7 − m) Steps zur Verfügung.
Mit der Tastenfolge [INV] [2nd] [Part] ruft man die augenblicklich gültige Speicherverteilung ab.

Zur Beachtung

Bevor man mit [2nd] [Part] m die Speicherverteilung einrichtet, müssen evt. Fixkommastellen mit [INV] [2nd] [Fix] und Anzeigeformate mit Zehnerpotenzen mit [INV] [EE] herausgenommen werden.

2. [LRN]
[LRN] ist die entscheidende Taste für die Aufnahme eines Tastenprogramms in die Programmspeicher. [LRN] ist die Abkürzung von **LEARN** (engl. to learn = lernen). Diese Vokabel ist gewählt worden, um auszudrücken, daß der Rechner nach dem Drücken dieser Taste bereit ist, etwas zu lernen, nämlich Tastenprogramm "auswending" zu lernen. Mit der Taste [LRN] betritt man — bildlich gesprochen — die Programmspeicherabteilung des Rechners. Allerfings muß man vorher mit [2nd] [Part] m (m<7) dafür sorgen, daß Programmspeicher zur Verfügung stehen; andernfalls gibt es eine Fehlermeldung.

© 2010 Joerg Woerner
Datamath Calculator Museum

3. [2nd] [CP]
[CP] ist die Abkürzung von engl. clear program = lösche Programmspeicher. Diese Taste ist nützlich (nicht unbedingt notwendig!), wenn man ein Tastenprogramm einlesen will. Man könnte ein altes Tastenprogramm auch einfach mit einem Neuen überschreiben, aber man kommt dabei leicht mit alten und neuen Programmteilen durcheinander.
Die Taste [CP] veranlaßt, daß der Programmzeiger auf die Startposition vor Step 00 gesetzt wird.

Tippe: [LRN] [2nd] [CP]
In der Anzeige erscheint: St (Programmzeiger steht auf "Start")

4. [SST]
[SST] ist die Abkürzung von engl **single Step** - Einzelschritt.

Tippe	**In der Anzeige erscheint**
[LRN]	-- St
[SST]	00 00
[SST]	00 01
[SST]	00 02

Das bedeutet: Der Programmzeiger ist von der Startposition auf Step 00 danach auf Step 01, dann auf Step 02 gesprungen.

Die beiden Ziffern rechts in der Anzeige geben also Auskunft, auf welchem Step sich der Programmzeiger gerade befindet.
Die beiden Ziffern links geben die Code-Nummer der Taste an, die in den Step einprogrammiert wurde, in diesem Fall immer "00", weil keine Taste eingelesen worden ist.

5. [BST]
[BST] ist die Abkürzung von engl. **back Step** = Rückwärtsschritt.

Tippe	**In der Anzeige erscheint**
[LRN]	-- St
[SST]	00 00
[SST]	00 01
[BST]	00 00

Mit den beiden Tasten [SST] und [BST] kann man innerhalb der Programmspeicher vor und zurückwandern. Die Tasten sind nützlich, um eingelesene Tastenprogramme zu überprüfen und an der gewünschten Stelle zu berichtigen oder zu ergänzen.

Copyright © 2010 Joerg Woerner
Datamath Calculator Museum

[SST] kann noch bei einem anderen Zusammenhang benutzt werden, der im 1. Programmierbeispiel unter dem Stichwort "Probelauf" besprochen wird (s.S 45)

Tippe [LRN]
Der Rechner ist wieder auf die RUN-Ebene eingestellt, und man findet die Anzeige so vor, wie man sie verlassen hat, als das erstemal [LRN] gedrückt wurde.
Mit [LRN] kann man also wechselweise die beiden Ebenen des Rechners aufsuchen.

Programmierbeispiele
Vorbemerkung

Es ist ziemlich sicher, daß die nachfolgenden Beispiele nicht dieselben Sachverhalte treffen, die Sie programmieren wollen.
Wenn Sie aber die hier angebotenen Beispiele nachvollziehen, werden Sie die notwendigen Programmiertechniken so lernen, daß Sie Ihre eigenen Probleme gut lösen können. Dabei ist es unerheblich, ob Sie eher praktische Probleme etwa aus dem kaufmännischen Bereich oder mehr wissenschaftlich-mathematische Aufgaben lösen wollen.

Die Einführung in die Programmiertechnik erfolgt hier an Aufgaben mit einfachstem mathematischen Hintergrund, so daß auch der mathematisch nicht besonders vorgebildete Leser ohne weiteres folgen kann.
Nebenbei sei erwähnt, daß es viel leichter ist, eine Lösungsformel wie z.B. den eingangs erwähnten "Pythagoras" zu programmieren – selbst dann, wenn man den mathematischen Zusammenhang überhaupt nicht kennt –, als den Rechner zu veranlassen, von 1 bis 10 und zurückzuzählen.

© 2010 Joerg Woerner
Datenmuseum Darmstadt

Bei der Besprechung der Aufgaben werden Sie weitere Tasten kennenlernen, und Sie werden manches, was Sie bisher vielleicht nicht in allen Einzelheiten verstanden haben, begreifen.
Probieren Sie eigene Lösungswege, experimentieren Sie, so viel Sie können (Sie können den Rechner mit falschen Tastenprogrammen nicht beschädigen), Sie werden wahrscheinlich für viele Aufgaben bessere Lösungswege finden als die hier angebotenen. Die Abb. nn wird Ihnen behilflich sein, die Verarbeitungswege Ihrer Eingangszahlen und der Zwischenergebnisse zu verfolgen.

1. Programmierbeispiel

(Tasten: [LRN], [CP], [R/S], [RST])

1.1. PROBLEMSTELLUNG

Ein Einzelhandelskaufmann bekommt eine Sendung von Gegenständen unterschiedlichster Preislage. Er kalkuliert 65% des Einkaufspreises als Aufschlag für Gewinn und Unkosten; darauf sind noch 13% Mehrwertsteuer zu entrichten.
Der Verkaufspreis wird auf 10 Pf gerundet.
Wie lauten die jeweiligen Verkaufspreise?

Auszug aus der Liste der Preise:

E	3.45	9.61	23.19	134.72	54.88	17.04	82.66
V	6.40	17.90	43.20	251.20	102.30	31.80	154.10

1.2. MATHEMATISIEREN

Bezeichnungen: E: = Einkaufspreis
K: = Kalkulationsprozent/100 (hier: 0,65)
M: = Mehrwertsteuer/100 (hier: 0.13)
V: = Verkaufspreis

Lösungsformel: $V = E \cdot (1 + K) \cdot (1 + M)$
$V = E \cdot 1.65 \cdot 1.13$

1.3. PROGRAMMIEREN

Man kann das Tastenprogramm unmittelbar aus der Lösungsformel übernehmen. Dabei muß man die Eingabe der immer wechselnden Einkaufspreise gedanklich vorwegnehmen: E ist also nicht Bestandteil des Tastenprogramms!

Copyright © 2010 Rudolf Hanisch
Datamath Calculator Museum

	Tastenprogramm:	Kommentar:
1.	[X]	Multiplikationsbefehl für die Zahl in der Anzeige (E)
2.	1.65	zweiter Faktor der Multiplikation
3.	[X]	Multiplikationsbefehl
4.	1.13	dritter Faktor der Multiplikation
5.	[=]	Befehl an den Rechner, die Multiplikationen abzuschließen; Ergebnis erscheint in der Anzeige.

Das Tastenprogramm ist hier in fünf Einzelschritte unterteilt und mit Ordnungsnummern versehen. Die Anzahl der Ordnungsnummern ist nicht gleichbedeutend mit der Anzahl der Steps, die für dieses Programm erforderlich sind. Die Ordnungsnummern fassen die Programmschritte zusammen, die **sinngemäß** zusammengehören; aus technischen Gründen kann es vorkommen, daß gedanklich zusammengehörige Teile beim Einlesen des Programms auseinandergenommen werden (siehe unten).

Das Programm ist kurz, und man bräuchte eigentlich keine Register (außer R_0) einzurichten. In Voraussicht auf die folgenden Programme kann man aber schon eine Speicherverteilung vornehmen: Tippe

[2nd] [Part] [4] (24 Steps und 4 Register: $R_0/R_1/R_2/R_3$)

1.4. EINLESEN DES TASTENPROGRAMMS

Tippe	Anzeige	Kommentar:
LRN		Vorbereitungsprozedur, um das Tasten-
CP	-- St	programm einlesen zu können
X	65 00	Die beiden Ziffern links geben die Code-Nr.
1	01 01	der eingelesenen Tasten an;
•	93 02	die beiden Ziffern rechts geben
6	06 03	die Step-Nummer an, in die die Taste
5	05 04	eingelesen wurde.
X	65 05	(Code-Nummern: s.S.122)
1	01 06	
•	93 07	
1	01 08	
3	03 09	
=	95 10	

Step 00 bis Step 10 sind mit dem Tastenprogramm belegt worden; die Taste [=] ist die letzte Taste. Wenn der Programmzeiger das Programm abarbeitet, "weiß" er nicht, daß [=] die letzte Taste ist, er würde weiterlaufen und nach weiteren Befehlen suchen. Da er keine mehr findet, gibt der Rechner eine Fehlermeldung. Es ist also notwendig, das Ende des Tastenprogramms kenntlich zu machen; man liest dafür die Taste [R/S] ein. Sobald der Programmzeiger diese Taste erreicht, stoppt er.

Tippe	Anzeige	Kommentar:
R/S	13 11	[R/S] hat die Code-Nr. 13 und ist in Step 11 einprogrammiert.
LRN		Mit [LRN] verläßt man die Programmspeicherabteilung, und man kann das eingelesene Programm für die Berechnungen benutzen: Die Prozedur "Einlesen" ist beendet.

Vorbereitungen für den 1. Programmlauf

Die Taste [R/S] kann auch von der RUN-Ebene her betätigt werden. Sie setzt den Programmzeiger in Bewegung, wenn er in Ruhe ist; sie stoppt den Programmzeiger, wenn er läuft. [R/S] ist die Abkürzung von engl. to **run**/to **stop**.

Der Programmzeiger steht im Augenblick auf Step 11, denn beim Einlesen des Tastenprogramms ist er bis Step 11 mitgelaufen. Würde man jetzt [R/S] tippen, würde er von Step 11 aus starten und nach Befehlen suchen. Da keine Befehle mehr eingelesen sind, wird der Rechner eine Fehlermeldung geben.
Bevor der erste Programmlauf durchgeführt werden kann, muß der Programmzeiger auf den Anfang des Tastenprogramms gesetzt werden, hier also auf Step 00.
Das Zurücksetzen auf Step 00 besorgt die Taste [RST] (to **reset** = zurücksetzen).

Tippe [RST]

Die Anzeige bleibt unverändert, trotzdem ist der Befehl "unten" in der Programmspeicherabteilung durchgeführt worden: Der Programmzeiger steht auf Step 00.
Das kann man nachprüfen, indem man nochmal [LRN] [SST] tippt: Die Anzeige weist 65 00 aus. Nun nochmal [LRN] tippen, um wieder aus der Programmspeicherabteilung herauszukommen!

Tippe nun die erste Zahl ein, die verarbeitet werden soll: 3.45 und jetzt: [R/S]

Die Anzeige weist 6.432525 aus; gerundet: 6.40 (DM)

Der Programmzeiger hat das Programm durchgearbeitet und steht nun wieder auf Step 11. Für die Verarbeitung der nächsten Zahl muß er zunächst wieder auf Step 00 zurückgesetzt werden.

Damit der Benutzer alles richtig macht, ist eine Benutzeranweisung notwendig, die Schritt für Schritt angibt, was gemacht werden muß:

© 2010 Joerg Woerner
Datamath Calculator Museum

1.5. BENUTZERANWEISUNG

1. Tippe [RST]
2. Tippe Einkaufspreis ein
3. Tippe [R/S]
4. Fahre fort mit 1.

1.6. PROBELÄUFE

Bevor man die tatsächlichen Werte verarbeitet, muß man sich überzeugen, ob das Programm auch richtig läuft. Es gibt verschiedene Möglichkeiten:

1. Man läßt das Programm mit Werten ablaufen, die man vorher manuell ausgerechnet hat.
2. Man nimmt "Grenzwerte", hier z. B. 0; das Ergebnis muß 0 sein.
3. Man gibt einen Wert ein und tippt das eingelesene Tastenprogramm sorgfältig mit dem Finger ab.
4. Man benutzt [SST] von der RUN-Ebene her. [SST] läßt den Programmzeiger nicht nur vorrücken, sondern die vorgefundenen Befehle werden tatsächlich ausgeführt.

   Tippen Sie also [RST] und dann einen Wert ein

   Tippen Sie nun [SST]/[SST]... und Sie sehen in der Anzeige genau, wie das Programm abgearbeitet wird.
5. Führen Sie die Benutzeranweisung selbst so aus, als hätten Sie sie nie zuvor gesehen!

© 2010 Joerg Woerner
Datamath Calculator Museum

Das Problem ist nunmehr völlig durchprogrammiert, und man kann die Tabelle auf Seite 42 nachvollziehen.

1. Programmänderung (Taste [RST])

   Die Programmiertaste [RST], die nach jedem Lauf manuell gedrückt werden muß, kann auch als Programmbestandteil eingelesen werden. Es erweitert sich dann die Wirkung dieser Taste: Sobald der Pgm-Zeiger sie erreicht, springt er nicht nur auf Step 00 zurück, sondern er läuft gleich weiter und arbeitet das Programm ab, bis er wieder bei [R/S] angekommen ist. Dieser Sachverhalt ist wesentlich bei der Überlegung, an welcher Stelle des Tastenprogramms [RST] eingespeichert werden muß: [RST] muß nach dem Stop-Befehl [R/S] eingespeichert werden.

Machen Sie sich klar, was bei einem Programmablauf geschieht:

1. Sie tippen einen Wert in die Anzeige, dann [R/S]; der Pgm-Zeiger arbeitet das Programm ab, das Ergebnis wird ausgewiesen, und der Pgm-Zeiger stoppt bei [R/S] (Step 11).
2. Sie tippen den nächsten Wert ein, dann [R/S]: Der Pgm-Zeiger läuft von Step 11 weiter, stößt in Step 12 auf [RST], springt infolge dessen auf Step 00 und arbeitet sich durch das Programm, bis er bei [R/S] angelangt ist.

Damit ist das lästige Zurücksetzen des Programmzeigers nach jedem Programmlauf nicht mehr dem Benutzer aufgebürdet.
Tippe [LRN]
Gene mit [SST] (oder [BST]) auf Step 11; die Anzeige weist dann 13 11 aus.
Tippe [RST] Anzeige: 21 12
Tippe [LRN]

Man könnte das Runden der Ergebnisse auch noch in die Benutzeranweisung aufnehmen, die sich nunmehr so ändern muß:

© Rechenzentrum Wuppertal
Datamath Calculator Museum

BENUTZERANWEISUNG

1. Tippe [RST]
2. Tippe [2nd] [Fix] 1
3. Tippe Einkaufspreis ein
4. Tippe [R/S] (Ergebnis erscheint in der Anzeige)
5. Fahre fort mit 3.

1.7 PROGRAMMÄNDERUNG

(Tasten [RCL] [R/S] [RST])

Die Werte für (1 + K) = 1.65 und (1 + M) = 1.13 sind feste Bestandteile des Tastenprogramms und somit nur mit einiger Mühe zu ändern. Es ist hier angebracht, Register einzusetzen, deren Inhalte man bequem von der RUN-Ebene her ändern kann.
Register sind vor Ablauf des Programms zu belegen (man vergißt das leicht!):

R_0 ← 1.65
R_1 ← 1.13

Es ist üblich, die Register, die belegt werden sollen, nach links zu schreiben und die Werte rechts daneben, so daß der Belegungspfeil nach links weist.

Unterscheiden Sie die Bezeichnungen "R" einerseits und [STO] [RCL] andererseits:

"R" bezeichnet das Register selbst;
[STO] und [RCL] sind Befehle, die sich auf "R" beziehen.

© 2010 Joerg Woerner
Datamath Calculator Museum

Das Tastenprogramm:	Kommentar:
[X]	Zahl in dem Anzeige-Register multiplizieren mit
[RCL] [0]	der Zahl, die aus R_0 abgerufen wird;
[X]	nochmal multiplizieren mit der Zahl, die aus R_1
[RCL] [1]	abgerufen wird;
[=]	Ergebnis ermitteln
[R/S]	Stop
[RST]	Pgm-Zeiger auf Step 00 zurück und neuer Programmlauf.

Einlesen des Tastenprogramms:

Tippe:		Anzeige:	
[LRN]			
[CP]			St
[X]	65		00
[RCL] [0]	71	00	01
[X]	65		02
[RCL] [1]	71	01	03
[=]	95		04
[R/S]	13		05
[RST]	21		06

Kommentar zur Anzeige:

71	00	01
71	01	03
Code für RCL	Nr. des Registers	Step-Nr.

Tippe:
[LRN] (Verlassen der Programmspeicher-Ebene)

BENUTZERANWEISUNG

1. Belege R_0 mit 1.65
2. Belege R_1 mit 1.13
3. Tippe [2nd] [Fix] [1]
4. Tippe [RST]
5. Tippe Einkaufspreis in die Anzeige
6. Tippe [R/S] (Ergebnis erscheint in der Anzeige)
7. Fahre fort mit 5.

1.8 PROGRAMMÄNDERUNG

(Tasten: [STO], [X] n, [RCL], [R/S], [RST])

Joerg Woerner ©2019 Datamath Calculator Museum

Diese Programmänderung bringt für den Anwender keine weiteren Vorteile. Sie soll lediglich zeigen, daß es auch andere Möglichkeiten gibt, das Problem zu lösen, hier mit Speichermultiplikation (s.S. 31)

Das Tastenprogramm:	Kommentar:
[STO] 2	Eingabewert wird in R_2 gespeichert
[RCL] 0	1.65 in die Anzeige
[STO] [X] 2	Speichermultiplikation : 1.65 × Eingabewert
[RCL] 1	1.13 in die Anzeige
[STO] [X] 2	Speichermultiplikation: 1.13 × (1.65 × Eingabewert)
[RCL] 2	Ergebnis aus R_2 in die Anzeige
[R/S]	
[RST]	

Die Notierung der Tastenfolge [STO] [X] [2] in einem Step ist so zu deuten:

Code für Taste [STO]	Code für Taste [X]	Nr. des Registers	Step-Nr.
61	65	02	02

Die Benutzeranweisung ändert sich nicht!

1.9 PROGRAMMÄNDERUNG

(Tasten: [x⇄t] [RCL] [R/S] [RST])

In den vorstehenden Programmierungsvorschlägen ging der Eingangswert durch Verrechnung verloren. Es wäre ganz nützlich, den Eingabewert zu erhalten, um ihn möglichst einfach mit dem Ergebnis vergleichen zu können. Hier bietet sich der Einsatz des t-Registers an. Die wesentliche Leistung dieses Registers wird hier noch nicht angesprochen (s.S. 74); R_t wird als "normaler" Datenspeicher benutzt, dazu die Möglichkeit, die Inhalte von R_t und R_x mit einem Tastendruck der Taste [x⇄t] auszutauschen (s. Abb. 2: direkte Leitung zwischen R_t und R_x).

Das Tastenprogramm:	**Kommentar:**
[X]	siehe unten nach Zeilen geordnet!
[x⇄t]	
[RCL] [0]	
[X]	
[RCL] [1]	
[=]	
[R/S]	
[RST]	

© 2010 Joerg Woerner
Datamath Calculator Museum

Kommentar:

1. Der Eingabewert im Rechenwerk zusammen mit der Operationstaste "[X]" notiert. Solange keine weitere Operationstaste ([+], [−], [X], [÷], [=]) getippt wird, bleibt dieser Zustand im Rechenwerk erhalten; man sagt, die Operation "schwebt". Man kann in diesem Zustand alle möglichen Zahlen in R_x geben, sie wieder löschen oder speichern, auch Speicherarithmetik betreiben (z.B. [STO] [+] 1 etc), der Schwebezustand wird dadurch nicht verändert. Nur durch eine **Operationstaste** wird dieser Zustand genau mit der Zahl verändert, die als letzte in R_x ist. Mit der Operationstaste [=] würde der Schwebezustand abgeschlossen. Die Veränderung des Schwebezustandes geschieht gemäß der AOS-Logik (s.S. 15)
2. Der Eingabewert wird in R_t gebracht, gleichzeitig wird irgendein Wert, der noch von vorherigen Rechnungen in R_t verblieben sein mag, in R_x und damit in die Anzeige gebracht.
3. Zahl aus R_0 (also 1.65) wird in R_x geholt; die Zahl in R_x wird überschrieben.

Hinweis: Der Kommentar kann in Verbindung mit der Abb.2 anschaulich verfolgt werden.

4. Die schwebende Multiplikation wird so verändert:
   x 1.65 wird ausgeführt, das Ergebnis wird zusammen mit der weiteren Operationstaste [X] als neuer Schwebezustand im Rechenwerk notiert.
5. Wert aus R_1 (also 1.13) wird in R_x (Anzeige) geholt; alter Wert wird überschrieben.
6. Die schwebende Multiplikation wird abgeschlossen
7. Der Programmzeiger stoppt; das Endergebnis kann abgelesen werden. Der neue Wert kann eingegeben werden.
8. Der Programmzeiger springt auf Step 00 und arbeitet mit dem neuen Eingabewert das Programm durch.

Nach Ablauf des Programms steht der jeweilige Einkaufspreis im Register R_t und der Verkaufspreis in R_x.
Mit der Taste [x⇄t] können die beiden Werte wechselweise in R_x gebracht und schnell verglichen werden.

© 2010 Joerg Woerner
Datamath Calculator Museum

2. Programmierbeispiel

(Tasten [STO] [+] n, [STO] [X] n, [STO] [−] n, [x⇄t], [RCL], [R/S], [RST])

2.1 PROBLEMSTELLUNG

In einem medizinisch-technischen Labor sind vielerlei Untersuchungen zu machen. Bei bestimmten Tests müssen prozentuale Abweichungen von gemessenen Werten zu einem Sollwert festgestellt werden.
Eine med.techn. Assistentin bekommt eine Liste von Meßwerten, deren prozentuale Abweichung von dem Sollwert S = 92.3 sie feststellen soll.

Meßwert	87.4	102.8	92.0	93.5	81.5	111.9	73.0	92.3
% Abwchg.	-5.3	11.4	-0.3	1.3	-11.7	21.2	-20.9	0.0

2.2 MATHEMATISIERUNG

Bezeichnungen: S := Sollwert (hier: 92.3)
M := Meßwert
A := Abweichung in Prozent

© 2007 Rolf Woerner
Datamath Calculator Museum

Lösungsformel: $A = \frac{M \cdot 100}{S} - 100$

2.3. PROGRAMMIERUNG

Speicherverteilung: [2nd] [Part] 3 (3 Register/32 Steps)

Registerbelegung:
R_0 ← 92.3 R_2 ← A (Speicherarithmetik)
R_1 ← 100 R_t ← M (jeweiliger Meßwert)

Tastenprogramm:	Kommentar:
[STO] [2]	Speichern des Meßwertes in R_2
[x⇄t]	Speichern des Meßwertes in R_t
[RCL] [0]	92.3 (Sollwert) in die Anzeige
[STO] [+] [2]	M ÷ S innerhalb von R_2 (Speicherdivision)
[RCL] [1]	100 in die Anzeige
[STO] [X] [2]	Inhalt von R_2 multipliziert mit 100
[STO] [−] [2]	Inhalt von R_2 minus 100
[RCL] [2]	Inhalt von R_2 in die Anzeige (Ergebnist)
[R/S]	
[RST]	

2.4. EINLESEN

Wie vorher

2.5. BENUTZERANWEISUNG

1. Belege R_0 mit 92.3 (Sollwert)
2. Belege R_1 mit 100
3. Tippe [2nd] [Fix] 1 (1 Nachkommestelle in die Anzeige)
4. Tippe [RST]
5. Tippe Meßwert ein
6. Tippe [R/S] (% - Abweichung erscheint in der Anzeige; kann mit Taste [x ⇄ t] wechselweise mit Meßwert verglichen werden)
7. Fahre fort mit 5.

2.6. PROBELÄUFE

S.S. 42 Abs. 1,6

© 2010 Joerg Woerner
Datamath Calculator Museum

3. Programmierbeispiel

(Tasten: R/S STO RCL RST)

Prozentuale Abweichungen wie in Beispiel 2 sind für jedermann interessant, wenn er einkaufen muß und Angebote vergleichen will.

3.1. PROBLEMSTELLUNG

In einer Drogerie werden 183 ml eines Hautpflegemittels für 7.55 DM angeboten. 426 ml werden für 18.50 DM verkauft.
Man erwartet bei einer größeren Menge einen relativ günstigeren Preis; ist das hier der Fall?

3.2. MATHEMATISIERUNG

Die Lösungsformel ist prinzipiell dieselbe wie im letzten Beispiel:

$$A = \frac{M \cdot 100}{S} - 100$$

© 2010 Joerg Woerner

Der "Sollwert" ist hier das Verhältnis des kleineren Preises zur kleineren Menge; der "Meßwert" ist das Verhältnis des größeren Preises zur größeren Menge:

$$M = \frac{P_2}{M_2} \qquad S = \frac{P_1}{M_1}$$

P_1 = kl. Preis P_2 = gr. Preis M_1 = kl. Menge M_2 = gr. Menge

Damit ändert sich die Lösungsformel entsprechend:

$$A = \frac{\frac{P_2}{M_2} \cdot 100}{\frac{P_2}{M_1}} - 100$$

Diese Formel sieht sehr steif aus, und man könnte sie mit Hilfe der üblichen algebraischen Methoden auf eine elegantere Form bringen. Für die Programmierung eignet , sich diese Lösungsformel viel besser als jede Umformung.

3.3. PROGRAMMIEREN

Bei den bisherigen Beispielen war jedesmal nur ein Wert einzugeben; in diesem Beispiel haben wir es mit vier Eingabewerten zu tun. Die Eingabe soll möglichst schnell und einfach organisiert werden. Es ist dabei zu bedenken, daß der Gesamtspeicherraum des Rechners begrenzt ist, der Sollwert zweckmäßigerweise in ein Register gegeben wird und daß damit für jeden Eingabewert nicht noch je ein Register bereitgestellt werden kann; bei insgesamt 5 Registern bleibt ein Programmierraum von 16 Steps, und das ist zu wenig.

Die Eingabewerte werden, soweit das möglich ist, schon mit der Eingabe verrechnet (schwebende Operationen).

Registerverteilung	Speicherverteilung
R_0 – 100	[2nd] [Part] 3
R_1 – S	
R_2 – Endergebnis	

© 2010 Joerg Woerner
Datamath Calculator Museum

Tastenprogramm:	Kommentar:
	(P_1 in die Anzeige)
[+]	P_1 [+] als schwebende Division
[R/S]	Stop für Eingabe von M_1
[=]	Die schwebende Division wird abgeschlossen
[STO] [1]	Das Ergebnis (S) wird in R_1 gespeichert
[R/S]	Stop für Eingabe von P_2
[÷]	P_2 [÷] als schwebende Division
[R/S]	Stop für Eingabe von M_2
[=]	Schwebende Division wird abgeschlossen
[÷]	das Ergebnis (M) wird dividiert
[RCL] [1]	Durch die Zahl, die aus R_1 in die Anzeige geholt wird (S)
[X]	Das Teilergebnis wird multipliziert mit
[RCL] [0]	Der Zahl, die aus R_0 (100) in die Anzeige geholt wird
[−] [RCL] [0]	Davon wird 100 subtrahiert
[=]	das Ergebnis ermittelt und
[STO] [2]	in R_2 gespeichert
[R/S]	
[RST]	

(Einlesen des Tastenprogramms wie immer!)

3.5. BENUTZERANWEISUNG

1. Belege R_0 mit 100
2. Lege gewünschte Nachkommastellen mit [2nd] [Fix] n fest
3. Tipp [RST]
4. Kleineren Preis eintippen
5. [R/S]
6. Kleinere Menge eintippen
7. [R/S]
8. Größeren Preis eintippen
9. [R/S]
10. Größere Menge eintippen
11. [R/S] (Prozentuale Abweichung der größeren Menge von der kleineren erscheint. Bei negativen Ergebnis liegt das größere Angebot unter dem kleineren, ist also günstiger).
12. fahre fort bei 4.

© 2010 Joerg Woerner
Datamath Calculator Museum

3.6. PROBELÄUFE

Geben Sie bequeme Zahlen ein, z.B. $P_1 = 10$ $P_2 = 20$
$M_1 = 100$ $M_2 = 200$

Das Ergebnis muß 0 sein!

Ergebnis der Eingangsaufgabe: A = 5.2507033
das größere Angebot liegt rd 5.3% über dem kleineren, ist also ungünstiger!

4. Programmierbeispiel

In diesem 4. Beispiel werden zwei neue Programmiertasten besprochen:

[GTO] (engl.: **go to** = gehe zu)
[LBL] (engl.: **LABEL** = Etikett/Aufschrift)

4.1. PROBLEMSTELLUNG

In einem Architekturbüro müssen die einkommenden Handwerkerrechnungen, die oft viele Einzelposten hat, nachgerechnet werden. Alle Rechnungen haben denselben Aufbau:

Menge/Stck	Einzelpreis		Betrag
3.57	25.55		91.21
18.93	4.18		79.13
326.70	6.09		1989.60
		Summe:	2159.94
		3 % Skonto:	64.80
		Nettobetrag:	2095.14
		13 % MST:	272.37
		Endbetrag:	2367.51

© 2010 Jörg Woerner
Datamath Calculator Museum

Es ist zu überlegen, daß man zwei verschiedene Programme benötigt:

1. Berechnung der Zeilenbeträge und deren Summe
2. Endabrechnung

Man muß also den Programmzeiger veranlassen, das jeweils richtige Programm aufzusuchen.
Bisher war das deshalb problemlos, weil nur ein Programm eingelesen war, das bei Step 00 begann, so daß man den Programmzeiger für einen erneuten Durchlauf nur zu Step 00 zu schicken brauchte ([RST] !).

Der TI-57 LCD besitzt die Möglichkeit, die Anfänge von Tastenprogrammen zu kennzeichnen; das geschieht mit der Tastenfolge [LBL] [0] oder [LBL] [1], [LBL] [2] ... [LBL] [9]. Es gibt demnach 10 Möglichkeiten, einen Programmanfang zu kennzeichnen oder, wie der Fachmann sagt, zu "adressieren".

Die Adressierung geschieht innerhalb der Programmspeicher, also im "LRN". Die Taste [LBL] kann nicht von der RUN-Ebene her eingesetzt werden.
In der Anzeige erscheint:

23 01 09

Code für [LBL] — Nr. des Labels hier: 1 — Step-Nr hier: 09

Die Taste [GTO] kann von der Run-Ebene wie auch im "LRN" eingesetzt werden.

Wirkung von der RUN-Ebene her: z.B. [GTO] [1] : Programmzeiger zu Lbl 1 springen und stoppen.

Wirkung als einprogrammierte Taste: z.B. [GTO] 1: Programmzeiger zu Lbl 1 springen und das nachfolgende Programm durcharbeiten.

In diesem Beispiel wird [GTO] vom RUN her eingesetzt.

© 2010 Joerg Woerner
Datamath Calculator Museum

4.2. MATHEMATISIEREN

Bezeichnungen:

M: = Menge/Stück
E: = Einzelpreis
B: = Betrag der Zeile
S: = Summe
Sk: = Skonto
Nt: = Nettobetrag
Mwst: = Mehrwertsteuer
Eb: = Endbetrag

Lösungsformeln:

1. $B = M \times E$
2. $S = \sum B_i$ (Summe aller Zeilenbeträge von der 1. bis nten Zeile)
3. $Sk = S \times 0.03$ (bei 3% Skonto)
4. $Nt = S - Sk$
5. $Ms = Nt \times 0.13$ (13% Mehrwertsteuer)
6. Eb = Endbetrag

4.3. PROGRAMMIERUNG

Speicherverteilung: [2nd] [Part] 2 (2 Register, 40 Steps)

Registerverteilung:

R_0 – summiert die Zeilenbeträge
R_1 – Faktor für Skonto-Berechnung, hier: 0.03
R_t – Zwischenwerte bei Endabrechnung

Tastenprogramm:	Kommentar:
[X]	Schwebende Multiplikation nach Eingabe des 1. Wertes
[R/S]	Stop für Eingabe 2. Wert
[=]	Schwebende Multiplikation wird abgeschlossen, Zeilenbetrag ermittelt.
[STO] [+] [0]	Zeilenbetrag wird in R_0 summiert
[R/S]	Stop für Kontrolle des Zeilenbetrages
[RST]	Nach Eingabe des nächsten Wertes: Programmzeiger auf Step 00 und bis zum ersten Stop weiterlaufen
[LBL] [1]	Adresse für das Berechnungsprogramm
[RCL] [0]	Summe in die Anzeige bringen
[R/S]	Stop zur Kontrolle der Summe
[X] [RCL] [1] [=]	Berechnung Skonto
[R/S]	Stop zur Kontrolle des Skontos
[−] [RCL] [0] [=] [+/−]	Berechnung Nettobetrag; wegen Einsparung von Steps wird zunächst nicht S - Sk, sondern Sk - S gerechnet; deswegen Vorzeichenwechsel.
[R/S]	Stop zur Kontrolle des Nettobetrages
[X]	Schwebende Multiplikation
[x=t]	Zwischenspeichern des Nettobetrages in R_t
[0] [•] [1] [3]	Faktor für Mehrwertsteuer in die Anzeige
[=]	Abschluß der schwebenden Multipl.; Mehrwertsteuer wird ermittelt
[R/S]	Stop zur Kontrolle der MwSt
[+]	Addition der MwSt +
[x=t]	Nettobetrag aus R_t wieder in die Anzeige
[=]	Ergebnis MwSt + Nettobetrag wird ermittelt (Endergebnis)
[R/S]	Stop zur Kontrolle des Endergebnisses

© 2010 Joerg Woerner
Datamath Calculator Museum

4.4. EINLESEN

Prozedur wie immer!

4.5. BENUTZERANWEISUNG

1. Für die Zeilenbeträge
   1.1 Tippe [2nd] [Fix] 2
   1.2 Belege R_1 mit Faktor für Skonto, hier: 0.03
   1.3 Belege R_0 mit 0
   1.4 Tippe [RST]
   1.5 Tippe 1. Wert in die Anzeige
   1.6 [R/S]
   1.7 Tippe 2. Wert der Zeile in die Anzeige
   1.8 [R/S] (Zeilenbetrag erscheint in der Anzeige)
   1.9 Fahre fort mit 1.5

2. Für die Endabrechnung
   2.1 Tippe [GTO] 1
   2.2 [R/S] (Summe erscheint)
   2.3 [R/S] (Skonto erscheint)
   2.4 [R/S] (Nettobetrag erscheint)
   2.5 [R/S] (Mehrwertsteuer erscheint)
   2.6 [R/S] (Endbetrag erscheint)
   2.7 Fahre fort bei 1.3 - für neue Rechnung
   2.1 bis 2.6 kann beliebig wiederholt werden.

© 2010 Joerg Woerner
Datamath Calculator Museum

4.6. PROBELÄUFE

Bequeme Zahlen für einige Zeilenberechnungen nehmen; Möglichkeit mit [SST] im "RUN" bedenken.

Eine Zwischenbilanz

Mit dem bisher erworbenen Programmierkenntnissen kann jede Aufgabe, die mit einer mathematischen Formel beschrieben wird, programmtechnisch bewältigt werden. Es macht dabei keinen Unterschied, ob man die Fläche eines Dreiecks mit

$$F = \frac{g.h}{2}$$

oder Werte der hyperbolischen Funktion

$$\sinh x = \frac{e^x - e^{-x}}{2}$$

zu berechnen hat. Auch längere Formeln etwa aus der Statik, z.B.

$$w = \frac{b.h^2 + \frac{b^2\pi}{4}(h+c)}{2a}$$

© 2010 Joerg Woerner
Datamath Calculator Museum

machen keine grundsätzlichen Schwierigkeiten.
Allenfalls ist die Speicherkapazität des Rechners zu klein, wenn zu viele Variablen in Register eingespeichert werden müssen.
Nachfolgend werden einige Aufgaben durchgeführt, die zur Festigung der Kenntnisse nachvollzogen werden können.

Aufgabe 1

Man berechne Umfang und Fläche von Kreisen bei gegebenem Radius

r	3,5	0,9	1,7	16,73	0,567	19,086	4,73	104,6	12,8946
U									
F									

Lösungsformeln: $U = 2\pi r$ $F = r^2\pi$

Registerbelegung:

$R_0 \leftarrow r$
$R_1 \leftarrow U$
$R_2 \leftarrow F$

Speicherverteilung:

2nd Part 3

Tastenprogramm:

STO 0 X 2 X π
= STO 1 R/S RCL
0 x^2 X π = STO
2 R/S
RST

Benutzeranweisung:

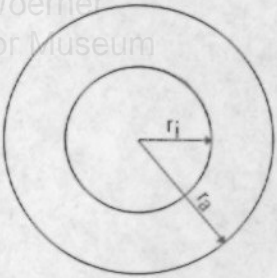
1. Tippe RST
2. Gewünscht Nachkommastellen eintippen
2nd Fix n
3. Wert für r in die Anzeige
4. R/S (Umfang erscheint)
5. R/S (Fläche erscheint)
6. Fahre fort bei 3

Aufgabe 2

Man berechne von einem Kreisring: bei gegebenen Radien r_i und r_a

1. Fläche des inneren Kreises
2. Fläche des Kreisrings
3. Umfang des inneren Kreises
4. Umfang des äußeren Kreises

© 2010 Joerg Woerner
Datamath Calculator Museum



Bezeichnungen:

A_i : = Fläche des inneren Kreises
R_g : = Fläche des Kreisrings
U_i : = Umfang des inneren Kreises
U_a : = Umfang des außeren Kreises
A_a : = Fläche des äußeren Kreises

Lösungsformeln:

$A_i = r_i^2 \cdot \pi$ $\qquad R_g = r_a^2 \cdot \pi - r_i^2 \cdot \pi$

$U_i = 2 \cdot \pi \cdot r_i$ $\qquad U_a = 2 \cdot \pi \cdot r_a$

r_i	9.56	0.95	6.79	2.5	3.09	4.17	1.99	5.55
r_a	10.56	1.00	9.09	3.5	12.65	9.11	2.05	7.99
A_i								
R_g								
U_i								
U_a								

PROGRAMMIERBEISPIELE

Registerverteilung:

$R_0 \leftarrow r_i$

$R_1 \leftarrow r_a$

$R_t \leftarrow$ Zwischenspeicher

Speicherverteilung:

[2nd] [Part] 2

Tastenprogramm:	Kommentar:
	(r_a in die Anzeige geben!)
[STO] [1]	r_a in R_1 speichern
[x^2]	Quadrat von r_a ermitteln
[X]	schwebende Multipl. r_a^2.
[R/S]	Stop für Eingabe von r_i
[STO] [0]	r_i in R_0 speichern
[π]	[π] in die Anzeige als Faktor für den Abschluß der noch schwebenden Multipl.
[=]	$r_a^2 \cdot \pi = A_a$ wird hier abgeschlossen
[−]	Einleiten der Differenz $A_a - A_i = R_g$
[(]	
[RCL] [0]	r_i in die Anzeige
[x^2]	Quadrat von r_i ermitteln
[X] [π]	mit π multiplizieren
[)] [R/S]	Klammerausdruck wird ermittelt: $r_i^2 \cdot \pi = A_i$
[=]	Abschluß der Differenz $A_a - A_i = R_g$
[R/S]	Stop für Notierung von R_g
[2] [X] [π]	2 π wird ermittelt und zwischengespeichert, weil dieser Wert nochmal gebraucht wird und auf diese Weise Steps eingespart werden.
[X]	schwebende Multipl. 2 π.
[x⇄t]	2 π zwischenspeichern in R_t
[RCL] [0]	r_i in die Anzeige
[=]	Abschluß von $2\pi \cdot r_i = U_i$
[R/S]	Stop für Notierung für U_i
[x⇄t]	2 π in die Anzeige

(→)

© 2010 Jörg Woerner
Datamath Calculator Museum

·) **Tastenprogramm:** **Kommentar:**

[X]	
[RCL] [1]	r_a in die Anzeige
[=]	U_a wird hier ermittelt
[R/S]	
[RST]	

Benutzeranweisung

1. Tippe [RST]
2. Gewünschte Nachkommastellen eintippen [2nd] [Fix] n)
3. Wert für größeren Radius R_a eintippen
4. [R/S]
5. Wert für kleineren Radius R_i eintippen
6. [R/S] (Wert für Fläche d. kleineren Kreises erscheint)
7. [R/S] (Wert für Fläche Kreisring erscheint)
8. [R/S] (Wert für Umfang d. kleinen Kreises erscheint)
9. [R/S] (Wert für Umfang d. großen Kreises erscheint)
10. Fahre fort bei 3.

© 2010 Joerg Woerner
Datamath Calculator Museum

Aufgabe 3

Man berechne in einem rechtwinkligen Dreieck eine der Seiten, wenn die beiden anderen gegeben sind.

Bezeichnungen:	**Lösungformeln:**
c: = Hypothenuse	$c = \sqrt{a^2 + b^2}$
a: = Kathete	$a = \sqrt{c^2 - b^2}$
b: = Kathete	$b = \sqrt{c^2 - a^2}$

Bei drei verschiedenen Formeln benötigt man drei verschiedene Programme, die mit Labels gekennzeichnet werden. Die Eingabe der bekannten Werte erfolgt vor Programmablauf in die entsprechenden Register.

Registerverteilung:	**Speicherverteilung:**
$R_0 \leftarrow a$	[2nd] [Part] 3
$R_1 \leftarrow b$	
$R_2 \leftarrow c$	

Tastenprogramm:	Kommentar:
LBL 1 RCL 0 x^2 + RCL 1 x^2 = $\sqrt{x}$ R/S	Adresse für das Programm, das aus gegebenen Seite a und b die Hypothenuse c berechnet.
LBL 2 RCL 2 x^2 − RCL 1 x^2 = ⊡ $\sqrt{x}$ R/S	Adresse für das Programm, das die Kathete a berechnet
LBL 3 RCL 2 x^2 − RCL 0 x^2 = ⊡ $\sqrt{x}$ R/S	Adresse für das Programm, das die Kathete b berechnet

Benutzeranweisung

1. Belege R_0 mit dem Wert für Seite a
   R_1 Seite b
   R_2 Seite c

   Das Register für die Seite, die nicht bekannt ist, braucht nicht weiter beachtet zu werden.

© 2010 Joerg Woerner
Datamath Calculator Museum

2. Falls c unbekannt ist: Tippe GTO 1 / R/S Der gesuchte Wert
   a : Tippe GTO 2 / R/S erscheint in der
   b : Tippe GTO 3 / R/S Anzeige
3. Fahre fort bei 1

Aufgabe 4

Man berechne in einem Dreieck die fehlende Seite aus den beiden anderen und dem eingeschlossenen Winkel (Kosinussatz!)

Bezeichnungen:

S: = gesuchte Seite
S_1: = bekannte Seite
S_2: = bekannte Seite
φ: = der eingeschl. Winkel
Z: = Zwischenergebnis

Lösungsformel:

$$s = \sqrt{s^2{}_1 + s^2{}_2 - 2s_1.s_2.\cos\varphi}$$

Registerverteilung:

$R_0 \leftarrow s_1$
$R_1 \leftarrow S_2$
$R_2 \leftarrow$
$R_t \leftarrow$ Zwischenspeicher

Speicherverteilung:

[2nd] [Part] 3

Tastenprogramm:	Kommentar:
[STO] [0]	s_1 in R_0 speichern
[X]	schwebenden Multiplikation s_1 ×
[R/S]	Stop für Eingabe s_2
[STO] [1]	s_2 in R_1 speichern
[X]	schwebende Multiplikation s_1 × verändern in schwebende Multiplikation s_1 × s_2 ×
[R/S]	Stop für Eingabe Winkel φ
[cos]	Kosinus des Winkels ermitteln
[X]	Änderung der schwebenden Multiplikation
[2]	Einbeziehung der Zahl 2 in die schwebende Multiplikation
[=]	Abschluß der schwebenden Multiplikation mit dem Ergebnis $z = 2.s_1.s_2.\cos\varphi$
[x⇄t]	Zwischenspeichern von z in R_t
[RCL] [0]	s_1 in die Anzeige
[x^2]	Quadrat von s_1 ermitteln
[+] [RCL] [1] [x^2]	$s^2_1 + s^2_2$ berechnen
[−]	subtrahiere
[x⇄t]	Zwischenergebnis aus R_t in die Anzeige
[=]	Ergebnis von $s^2_1 + s^2_2 - 2.s_1.s_2.\cos\varphi$
[$\sqrt{x}$]	Wurzel ziehen: Endergebnis ist ermittelt.
[R/S]	
[RST]	

© 2010 Joerg Woerner
Datamath Calculator Museum

Benutzeranweisung:

1. Tippe [RST]
2. Gewünschte Nachkommastellen einrichten [2nd] [Fix] n)
3. Wert für s_1 eingeben
4. [R/S]
5. Wert für S_2 eingeben
6. [R/S]
7. Wert für Winkel φ ergeben
8. [R/S] (Ergebnis erscheint in der Anzeige)
9. Fahre fort bei 3.

Aufgabe 5

Man erstelle eine Wertetabelle für eine Funktion 3. Grades:

$$y = ax^3 + bx^2 + cx + d$$

Die Übersetzung dieser Formel in ein Tastenprogramm ist prinzipiell sehr leicht. Da man aber die Werte für a, b, c und d und außerdem den jeweils aktuellen x-Wert speichern muß, sind 5 Register einzurichten, so daß nur noch 16 Steps zur Verfügung stünden; das ist zu wenig! Man muß hier wieder R_t einbeziehen.

© 2011 Joerg Woerner
Datamath Calculator Museum

Weiterhin kann man die Potenztaste [y^x] für negative Basiswerte nicht benutzen.

$(-2)^3$ ist demzufolge so zu lösen:

[−] [2] eintippen
[x^2]
[X]
[−] [2]
[=] (- 8)

Registerverteilung:

R_0 — x-Wert
R_1 — a
R_2 — b
R_3 — c
R_t — d

Speicherverteilung:

[2nd] [Part] 4

Tastenprogramm:	Kommentar:
	(x-Wert eingeben)
[STO] [0]	x-Wert in R_0 speichern)
[x^2] [X] [RCL] [0]	Berechnung von ax^3
[X] [RCL] [1]	
[+]	
[RCL] [0] [x^2] [X] [RCL] [2]	Berechnung von bx^2
[+]	
[RCL] [0] [X] [RCL] [3]	Berechnung von cx
[=]	Zwischenergebnis: $ax^3 + bx^2 + cx = z$
[$x \rightleftarrows t$]	z zwischenspeichern in R_t, gleichzeitig d in die Anzeige
[+]	Addition (d + z)
[$x \rightleftarrows t$]	Zwischenwert wieder in die Anzeige, d zurück in R_t
[=]	y-Wert wird ermittelt: y = d + z
[R/S]	
[RST]	

© 2010 Joerg Woerner
Datamath Calculator Museum

Benutzeranweisung:

1. Tippe [RST]
2. Gewünschte Nachkommastellen eingeben ([2nd] [Fix] n)
3. Belege R_1 mit a Es ist notwendig, **alle** Register zu belegen,
   R_2 mit b evt.mit 0 oder 1; Beispiel:
   R_3 mit c $x^3 - x^2 + 4 = y$
   R_t mit d a = 1 / b = − 1 / c = 0 / d = 4
4. Tippe x-Wert ein
5. [R/S] (y-Wert erscheint in der Anzeige)
6. Fahre fort mit 4.

Bemerkung: Man kann dieses Programm auch für alle anderen Funktionen 2. Grades und 1. Grades verwenden, indem man die entsprechenden Koeffizienten (a,b,c) bzw. das x-freie Glied d gleich null setzt; z.B.:

$$y = 0 \cdot x^3 = bx^2 + cx + d \Leftrightarrow y = bx^2 + cx + d$$

Aufgabe 6

Man ermittle die Lösungen der quadratischen Gleichung

$$x^2 + px + q = 0$$

Lösungsformel:

$$x_1 = -\frac{p}{2} + \sqrt{\frac{p^2}{4} - q}$$

$$x_2 = -\frac{p}{2} - \sqrt{\frac{p^2}{4} - q}$$

Registerverteilung:

$R_0 \leftarrow p$
$R_1 \leftarrow q$
$R_t \leftarrow \sqrt{D}$

Speicherverteilung: [2nd] [Part] 2

Tastenprogramm:	Kommentar:
	(Eingabe p)
[STO] [0]	p wird in R_0 gespeichert
[x^2]	Quadrat von p ermitteln
[÷] [4] [−]	Schwebende Rechenoperation $\frac{p^2}{4} -$
[R/S]	Stop für Eingabe von q
[STO] [1]	q wird in R_1 gespeichert
[=]	Ergebnis von $\frac{p^2}{4} - q = D$
[R/S]	Stop zur Kontrolle von D: ist D<0, gibt es keine Lösung.
[$\sqrt{x}$]	Wurzel aus D ermitteln
[x ⇄ t]	$\sqrt{D}$ in R_t speichern
[RCL] [0] [÷] [2] [=] [+/−]	Berechnung von $-\frac{p}{2}$
[STO] [1]	$\frac{p}{2}$ wird in R_1 gespeichert, damit geht q verloren; das ist möglich, weil q zu weiteren Berechnungen nicht mehr gebraucht wird.
[x ⇄ t]	$\sqrt{D}$ in R_x, gleichzeitig $-\frac{p}{2}$ in R_t
[+]	
[x ⇄ t]	wieder Austausch von $\sqrt{D}$ und $-\frac{p}{2}$
[=]	Abschluß der Rechnung: $+\sqrt{D} + (-\frac{p}{2}) = x_1$
[R/S]	Stop zur Notierung von x_1
[RCL] [1]	$-\frac{p}{2}$ in die Anzeige

(→)

© 2010 Joerg Woerner
Datamath Calculator Museum

(→) **Tasten programm:**

[−]

[x⇄t]

[=]

[R/S]

[RST]

Kommentar:

D in die Anzeige

Abschluß der Rechnung: $-\frac{p}{2} - \sqrt{D} = x_2$

Benutzeranweisung:

1. Tippe [RST]
2. Gewünschte Nachkommastellen eingeben ([2nd] [Fix] n)
3. Wert für p eingeben
4. [R/S]
5. Wert für q eingeben
6. [R/S] (Wert für D erscheint in der Anzeige; bei negativem D hat die Gleichung keine Lösung: Tippe [RST] / Fahre fort bei 3.)
7. [R/S] (x_1 erscheint in der Anzeige)
8. [R/S] (x_2 erscheint in der Anzeige)
9. Fahre fort bei 3. (neue Gleichung)

© Joerg Woerner
Datamath Calculator Museum

Schleifen

Sie lernen in diesem Kapitel neue Programmiertasten und damit neue Programmiertechniken kennen.

Zunächst noch etwas Grundsätzliches zu den Programmiertasten: Programmiertasten steuern, wie Sie an den Beispielen gesehen haben, den Programmzeiger.
Es gibt Programmiertasten, die man nur innerhalb der Programmspeicher benutzen kann, die man also nicht von der RUN-Ebene aus betätigen kann, z.B. [LBL].
Es gibt Programmiertasten, die man sowohl in die Programmspeicher einlesen als auch von der RUN-Ebene aus bedienen kann, z.B. [R/S], [RST], [GTO].
In den meisten Fällen bewirken eingespeicherte Programmiertasten etwas anderes, als wenn man sie vom "RUN" aus betätigt, z.B. [RST] vom "RUN" aus: Programmzeiger zurück auf Step 00;
[RST] im "LRN": Pgm-Zeiger zurück auf Step 00 und Tastenprogramm durcharbeiten.

Der Begriff "Schleife" wird gebraucht, wenn der Programmzeiger mehrmals ohne Stop ein Programm durchläuft. Das ist dann notwendig, wenn man das Ergebnis des vorangegangenen Laufes für die Ermittlung des nächsten Ergebnisses benötigt.

5. Programmierbeispiel

(Tasten: [RCL] [R/S] [RST])

Ein Vater legt bei der Geburt seiner Tochter ein Sparkonto an. Er zahlt 300 DM ein und fügt bei jedem Geburtstag 50 DM hinzu. Die Sparkasse gewährt eine Verzinsung von 5.75 %.
Wieviel kann die Tochter bei ihrem 18. Geburtstag abholen?

Mathematisierung:

nach dem 1. Jahr: 300 . 1.0575 + 50 = 367.25
nach dem 2. Jahr: 367.25 . 1.0575 + 50 = 438.37
nach dem 3. Jahr: 438.35 . 1.0575 + 50 = 513.57

usw.

Allgemein: $K_n = K_{n-1} \cdot q + z$

K_0 ist Anfangskapital
K_n ist das Kapital nach n Jahren
q ist der Zinsfaktor
z ist die jährliche Zahlung

Programmierung

© 2010 Joerg Woerner
Datamath Calculator Museum

Registerverteilung:

R_0 ← 1 (K_0) (hier: 300)
R_1 ← Zinsfaktor (hier: 1.0575)
R_2 ← jährl. Zahlung (hier: 50)

Speicherverteilung:

[2nd] [Part] 4

Tastenprogramm:	**Kommentar:**
	(Eingabe: 300)
[X] [RCL] [1] [+] [RCL] [2] [=] [R/S] [RST]	Um zum Endergebnis zu kommen, muß man 18 mal [R/S] tippen. Kann man sicher sein, wirklich genau 18 mal getippt zu haben?

Es ist sicher vernünftig, ein "Zählwerk" an das Berechnungsprogramm zu setzen;

Programmänderung: (Tasten: [RCL] [STO] [STO] [+] n [R/S] [RST])

Tastenprogramm:	Kommentar:
RCL 0 × RCL 1 + RCL 2 = STO 0	Das jeweilige Jahresergebnis muß gespeichert werden, weil durch die nachfolgende "1" für das Zählprogramm die Zahl in der Anzeige überschrieben wird. Mit der Speicherung des neuen Jahresergebnisses geht das alte Ergebnis verloren; das muß so sein, denn man benötigt ja immer den neuesten Wert für den nächsten.
R/S	Stop für Kontrolle des Jahresergebnisses
1 STO + 3 RCL 3	Zählprogramm
R/S	Stop für Kontrolle der Anzahl der Durchläufe.
RST	**Achtung:** Vor Ablauf des Programms muß der Anfangswert in R_0 eingegeben werden!

Programmänderung: (Taste: Del)

Der Stop-Befehl bei 8. soll herausgenommen werden. Da es viel Mühe macht, wegen eines einzigen Befehls das gesamte Programm neu einzulesen, gibt es vorteilhafte Prozedurtasten zur Berichtigung:

© 2010 Joerg Woerner
Datamath Calculator Museum

Tippe LRN

Gehe je nach Stand des Pgm-Zeigers mit SST oder RST auf Step 07
Anzeige: 13 07

Tippe 2nd Del * Anzeige: 61.00 06

Das bedeutet: Die Taste R/S (Code 13) ist aus Step of herausgenommen. Alle anderen Tasten sind innerhalb der Steps dadurch um 1 Stelle nach unten gerückt. Da hinter R/S die "1" stand ist diese nun auf Step 07 gerutscht (SST - Anzeige 01 07)

Anschaulich: Wenn man aus einem Turm von Klötzen einen herausnimmt, rutschen alle, die oberhalb lagen, um die Klotzbreite herunter.

* Del von engl.: to delete = ausradieren/löschen

PROGRAMMIERBEISPIELE

Wenn man umgekehrt eine Taste einfügen will, geht man zu der entsprechenden Stelle und tippt die gewünschte(n) Taste(n).
Damit drückt man alle eingelesenen Tasten einschließlich der in der Anzeige ausgewiesenen Taste um eine (oder mehrere) Stelle(n) nach "oben". Dabei muß man bedenken, daß man die vorhandenen Programmspeicherplätze nicht überschreitet, so daß die Tasten, die an den letzten Stellen des Tastenprogramms stehen, verloren gehen. Anschaulich: Wenn man eine Flüssigkeit in einem Schlauch nach oben drückt, werden die oberen Flüssigkeitsteilchen herausfließen, wenn das Schlauchende erreicht ist.

Tippe [LRN]

Der Rechner ist wieder auf die RUN-Ebene eingestellt, und das Programm kann gestartet werden. Zuvor: Eingangswert neu eingeben und Zählspeicher R_3 auf 0 setzen.
Man kann die jeweiligen Jahresbeträge nicht mehr kontrollieren, aber die Anzahl der Durchläufe wird bei jedem Stop angegeben. Mit [RCL] [0] kann das Ergebnis nach Ablauf der 18 Runden in die Anzeige geholt werden.

Programmänderung: (taste: [Del])

© 2010 Joerg Woerner
Datamath Calculator Museum

Der letzte Stop-Befehl soll aus dem Programm genommen werden. Er steht in Step 10:

Tippe [LRN]

Gehe mit [SST] (oder [BST]) bis Step 10;

Anzeige: 13 10

Tippe nun [2nd] [Del]; Anzeige: 71.03 09

Taste [R/S] (Code 13) ist heraus und Taste [RST] (Code 21) ist auf Step 10 heruntergerutscht.

Tippe [LRN] Eingangswert neu eingeben, R_3 auf 0 setzen und mit [R/S] starten.

Der Pgm-Zeiger arbeitet sich durch das Programm, alles wird richtig berechnet, R_3 wird zählen, der Pgm-Zeiger springt auf Step 00 und beginnt von vorn — endlos immer wieder, bis der Anwender mit [R/S] den Pgm-Zeiger stoppt.
Man spricht hier von einer "Endlos-Schleife".
Man sieht natürlich nicht, wo der Programmzeiger gerade ist, und man weiß auch nicht, wie oft er die Schleife durchlaufen hat.

Programmänderung: (Tasten: [Pause])

Die Programmiertaste [Pause] soll in Step 10 eingefügt werden. Sobald der Pgm-Zeiger an die Taste [Pause] kommt, wird das Programm einen Augenblick lang unterbrochen und die Zahl, die gerade in R_X ist, wird angezeigt. Deswegen ist es logisch, [Pause] nach "RCL 3" einzulesen.

Tippe [LRN]

Gehe mit [SST] (oder [BST]) bis Step 09; Anzeige: 71.03

Tippe [2nd] [Pause]; Anzeige: 96 10
Die Taste [Pause] (Code 96) ist in Step 10 eingelesen
Die Taste [RST] (code 21) ist nach obengerückt.

Tippe [LRN]
Das Tastenprogramm hat nun folgende Notierung:

Tastenprogramm:	Kommentar:
[RCL] [0]	K_n in die Anzeige
[X]	
[RCL] [1]	
[+]	
[RCL] [2]	
[=]	K_{n+1} wird ermittelt
[STO] [0]	speichern in R_0 (K_n wird überschrieben?)
[1]	1 in die Anzeige
[STO] [+] [3]	1 wird in R_3 addiert (zählen!)
[RCL] [3]	Inhalt von R_3 in die Anzeige
[Pause]	Kurzunterbrechung; Zahl der Umläufe wird angezeigt
[RST]	Sprung nach Step 00 und neuer Durchlauf

Es ist üblich, bei Sprüngen Pfeile zu zeichnen, damit man besser überblickt, von wo nach wo der Pgm-Zeiger springt.

© 2010 Joerg Woerner
Datamath Calculator Museum

Benutzeranweisung:

1. Belege R_0 mit Anfangskapital K_0 (hier: 300)
2. R_1 mit Zinsfaktor q (hier: 1.0575)
3. R_2 mit jährl. Zahlung (hier: 50)
4. R_3 mit 0 (wichtig für das richtige Zählen!)
5. Tippe [RST]
6. [R/S]
7. Beobachte die Anzeige und tippe schnell [R/S], wenn gewünschte Umlaufzahl erreicht ist.
8. Tippe [RCL] [0] : Endbetrag erscheint in der Anzeige.
9. Fahre fort bei 1.

Bemerkung: es ist sehr unbefriedigend, immer auf die Anzeige starren zu müssen, um im richtigen Augenblick - den man sicher leicht verpaßt - den Programmzeiger zu stoppen.
Im nächsten Kapitel wird die Programmiertechnik besprochen, mit der der Pgm-Zeiger veranlaßt werden kann, die Schleife bei der gewünschten Anzahl der Durchläufe zu verlassen.

© 2010 Joerg Woerner
Datamath Calculator Museum

Abfragen

Mit Hilfe des t-Registers (engl. test = Abfrage/Prüfung) läßt sich feststellen, ob zwischen der Zahl, die sich im t-Register befindet, und der Zahl in der Anzeige bestimmte Beziehungen herrschen oder nicht.
Man kann vier verschiedene Beziehungen durch entsprechende Fragen abtesten:

1. ist die Zahl in R_x genau so groß wie in R_t? kurz: $x = t$?
2. Ist die Zahl in R_x nicht so groß wie in R_t? kurz: $x \neq t$?
3. Ist die Zahl in R_x größer oder gleich der in R_t? kurz: $x \geq t$?
4. Ist die Zahl in R_x kleiner als die Zahl in R_t? kurz: $x < t$?

Diese vier Fragen werden in die "Tastensprache" übersetzt:

1. [2nd] [x=t]
2. [INV] [2nd] [x=t]
3. [2nd] [x≥t]
4. [INV] [2nd] [x≥t]

Die Frage-Tasten sind Programmiertasten, die nur im "LRN" Wirkung haben.

Die Fragen werden im Rechner intern beantwortet und je nachdem, ob die Antwort ja oder nein ist, nimmt der Programmzeiger einen "Ja-Weg" oder einen "Nein-Weg".
Der Ja-Weg beginnt immer an dem Step, der der Abfrage unmittelbar folgt. Bei "nein" überspringt der Programmzeiger diesen Step und setzt seinen Weg beim übernächsten Step fort.

Programmänderung (Tasten: [RCL] [STO] [STO] [+] n [Prog] [RST] [R/S] [INV] [x≥t])

Für das Zinseszinsproblem ergibt sich folgender Tastenplan:

Tastenprogramm:	Kommentar:
[RCL] [0]	bis 11.: wie vorher
[X]	
[RCL] [1]	
[+]	
[RCL] [2]	
[=]	
[STO] [0]	
[1]	
[STO] [+] [3]	
[RCL] [3]	
[Prog] [INV] [x≥t]	Die Abfrage lautet: Ist die Zahl in R_x kleiner als die Zahl in R_t? (kurz: $x<t$)
[RST] [2nd]	Hier beginnt der "Ja-Weg". Da aber nur ein Step zur Verfügung steht, setzt man hier einen Sprungbefehl ein, der den Programmzeiger in das weitere "Ja-Programm" bringt. Hier ist der Sprungbefehl [RST] einprogrammiert; das "Ja-Programm" ist die Wiederholung der Berechnung für K_n.
[RCL] [0]	Hier beginnt das "Nein-programm"; es ist sehr kurz [RCL] [0] [R/S]
[R/S]	Ende des "Nein-Programs", gleichzeitig Ende des gesamten Programm.

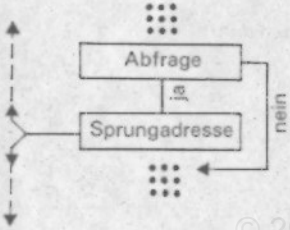
Vor Ablauf des Programms muß R_t mit der Zahl belegt werden, die angibt, wie oft die Schleife durchlaufen werden soll; hier also mit 18.
Die Tastenfolge dafür lautet: [ON/C] 18 [x↕t].

© 2010 Joerg Woerner
Datamath Calculator Museum

Der Sprung des Programmzeigers zum Programmpunkt 1. mit der Taste [RST] wird von einer Bedingung abhängig gemacht, hier von der Bedingung, daß die Zahl in R_X kleiner ist als die Zahl in R_t.
Man spricht deshalb von einem "bedingten Sprung".

Wenn man einen Sprung des Programmzeiger nicht von einer Bedingung abhängig macht - wie z.B. im tastenprogramm auf Seite 73, spricht man von einem "unbedingten Sprung".

Die Programmierstruktur bei einem bedingten Sprung ist immer dieselbe:



Sprünge — bedingt oder unbedingt — müssen nicht in eine Schleife führen; das nächste Beispiel zeigt eine andere Möglichkeit.

© 2010 Joerg Woerner
Datamath Calculator Museum

Die Einrichtung von bedingten Sprüngen verlangt vom Programmierer folgende Überlegungen:
1. An welcher Stelle im Tastenprogramm ist die Abfrage einzurichten? Dabei ist es notwendig, den Zustand von R_X (Anzeige) im ganzen Verlauf des Programms im Auge zu behalten, denn jede Abfrage ist mit R_X verbunden.
2. Welche der vier möglichen Fragen ist angebracht?
3. Das "Ja-Programm" muß eine Adresse bekommen [LBL]! Nur wenn das Ja-Programm bei Step 00 beginnt, kann auf eine Adresse verzichtet werden, weil die Taste [RST] eingesetzt werden kann.
4. Das t-Register muß vor Ablauf des Programms richtig belegt sein. Es kann für andere Zwecke nur noch mit Einschränkungen benutzt werden.

6. Programmierbeispiel

(Tasten: GTO LBL RCL RST R/S x ≥ t

Wir knüpfen an das Beispiel 1 auf Seite 42 an.

Der Einzelhandelskaufmann kalkuliert die billigeren Artikel anders als die teueren: Größere Artikel werden schlechter verkauft, sie verursachen mehr Lager- und Wartungskosten; deshalb:

Artikel unter 50.- DM Einkaufspreis bekommen einen Aufschlag von 65%, Artikel ab 50.- DM bekommen einen Aufschlag von 73%.

In beiden Fällen kommen 13% Mehrwertsteuer hinzu.

Wie lauten die Verkaufspreise?

Registerverteilung:

R_0 – 1.65
R_1 – 1.73
R_t – 50

Speicherverteilung:

2nd Part 2

© 2010 Joerg Woerner
Datamath Calculator Museum

Tastenprogramm:	Kommentar:
2nd x ≥ t	Abfrage an den Anfang, denn von der Größe des Einkaufspreises hängt der Aufschlag ab.
GTO 1	Sprungadresse mit dem GTO-Befehl: Programmzeiger springt bei "ja" nach Lbl 1 und arbeitet das darunter stehende Programm ab.
X	Beginn des "Nein-Programms"; der Eingangswert ist noch in der Anzeige, wird multipliziert mit dem kleineren Faktor aus R_0 usw.
RCL 0	
=	Ergebnis wird ermittelt
R/S	Ende des Nein-Programms
RST	mit neuem Wert von vorn beginnen
LBL 1	Beginn des Ja-programms mit Lbl 1
X	Eingangswert wird mit dem größeren Faktor aus R_1 multipliziert usw.
RCL 1	
X 1 • 1 3	
=	Ergebnis wird ermittelt
R/S	Ende des Ja-Programms
RST	Mit neuem Wert von vorn beginnen.

Benutzeranweisung:

1. Belege R_0 mit 1.65
2. Belege R_1 mit 1.73
3. Belege R_t mit 50
4. [RST]
5. Gib Einkaufspreis ein
6. [R/S] (Verkaufspreis erscheint)
7. Fahre fort bei 5.

Erweiterung der Aufgabe (Tasten: [x⇄t] [GTO] [RCL] [x≥t] [RST] [R/S])

Problem: Der Kaufmann möchte die Artikel ab 100.- DM mit 81% Aufschlag, die Artikel ab 50.- DM mit 73% Aufschlag und die Artikel unter 50.- DM mit 65% Aufschlag kalkulieren.

Vorüberlegungen:
1. Es müssen zwei Abfragen eingerichtet werden, und das bedeutet, daß das t-Register innerhalb des Tastenprogramms wechselweise mit den beiden Zahlen 50 und 100 belegt werden muß. Da die Taste [x⇄t] einen Austausch der Zahlen zwischen R_x und R_t veranlaßt, muß man aufpassen, daß dabei keine Daten verloren gehen, die man noch braucht.

2. Weil das Programm voraussichtlich sehr lang wird, können außer R_0 keine weiteren Register eingerichtet werden; es stehen dann 48 Steps zur Verfügung.

3. Das Tastenprogramm wird zunächst so geschrieben, als gäbe es genügend Steps. Der logische Ablauf wird damit deutlich. Es werden aber mehr als 48 Steps verbraucht.
Es gibt bestimmte Programmteile, die sich wiederholen. Diese werden dann zusammengefaßt, sodaß die 48 Steps schließlich doch ausreichen.

Speicherverteilung: [2nd] [Part] [1] ; Register $R_0 \rightarrow 1.13$, $R_t \rightarrow 100$

Datamath Calculator Museum

Tastenprogramm:	Kommentar:
2nd x≥t	Abfrage, ob Eingabewert größer/gleich 100.- R_t muß vorher mit 100 belegt worden sein.
GTO 1	Sprungadresse bei "ja"
x⇄t	das "Nein-Pgm" muß mit Abfrage beginnen, ob der Eingabewert größer/gleich 50 ist. Zuvor muß R_t mit 50 belegt werden, aber der Eingabewert darf nicht verloren gehen. Deshalb hier: Eingabewert in R_t zwischenspeichern.
5 0	50 in die Anzeige
x⇄t	50 in R_t und Eingabewert wieder in die Anzeige
2nd x≥t	Abfrage, ob Eingabewert größer/gleich 50
GTO 2	Sprungadresse bei "ja"
X 1 • 6 5 X	Beginn des Nein-Pgms: Nach den zwei verneinten Abfragen ist der
RCL 0 =	Eingabewert kleiner als 50.- DM. Verkaufspreis mit 65% Aufschlag wird berechnet.
x⇄t	Verkaufspreis in R_t zwischenspeichern
1 0 0	100 in die Anzeige
x⇄t	100 für neuen Durchlauf in R_t bringen, Verkaufspreis aus R_t wieder in die Anzeige.
R/S	Stop für Entnahme des Verkaufspreises; Neuer Einkaufspreis kann eingetippt werden
RST	Pgm-Zeiger auf Step 00 und von vorn beginnen
LBL 1	Programmadresse, wenn Eingabewert größer/gleich 100.
X	
1 • 8 3	
X	
RCL 0	
=	Berechnung des Verkaufspreises mit 83% Aufschlag
x⇄t 1 0 0 x⇄t R/S	
RST	
LBL 2	Programmadresse, wenn Eingabewert zwischen 50.- und 100.- DM ist.
X 1 • 7 3 X	
RCL 0 =	Berechnung des Verkaufspreises mit 73% Aufschlag
x⇄t 1 0 0 x⇄t R/S	
RST	

© 2001 Jörg Woerner
Datamath Calculator Museum

Bemerkung: Die Anzahl der Steps ging über die Kapazität des Rechners (48 Steps) hinaus. Da die nachstehende Tastenfolge 3 × im Programm vertreten ist fassen wir diese unter einem neuen Label (37) zusammen.

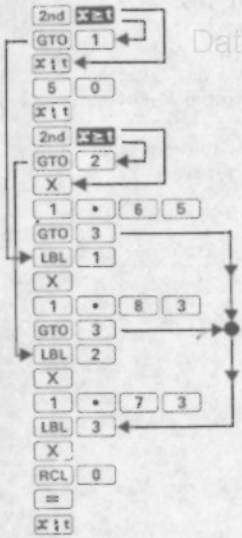
Im Gesamtprogramm ist folgende Tastenfolge dreimal vertreten:

X	siehe 10. bis 16.
RCL 0	
=	21. bis 28.
x⇄t	
1 0 0	32. bis 39.
x⇄t	
R/S	
RST	

Wir schicken den Programmzeiger aus den jeweiligen Programm mit einem unbedingten Sprung an diese Adresse.

Das Tastenprogramm hat dann die folgende endgültige Form:

© 2010 Joerg Woerner
Datamath Calculator Museum



Benutzeranweisung:

1. Belege R_0 mit 1.13
2. Belege R_t mit 100
3. RST
4. Einkaufswert eingeben
5. RST (Verkaufspreis erscheint)
6. Fahre fort bei 4.

(→)

(−) [1] [0] [0]
[x ⇄ t]
[R/S]
[RST]

Übungsaufgaben

Die nachfolgenden Zählprogramme haben kaum praktische Bedeutung, sie sind aber geeignet, die Programmiertechnik mit Abfragen zu üben.

1. Der Rechner soll von 1 bis 10 zählen und dann stoppen.
2. Der Rechner soll von 1 bis 10 zählen; wenn er bei 10 angekommen ist, soll er von vorn beginnen (Endlosschleife).
3. Der Rechner soll von 1 bis 10 zählen und dann zurück bis 1.
4. Der Rechner soll von 1 bis 10 zählen und dann zurück bis 1, wieder bis 10 usw. (Endlosschleife)
5. Der Rechner soll eine beliebige 1 × 1-Reihe "aufsagen". Es soll so programmiert werden, daß man die gewünschte 1 × 1-Zahl in die Anzeige tippt, und der Rechner gibt die Reihe bis 10 × n nacheinander in die Anzeige, dann stop.

© 2010 Joerg Woerner
Datamath Calculator Museum

Tastenprogramm (1):

[+]
[1]
[=]
[Pause]
[INV] [2nd] [x ≥ t]
[RST]
[R/S]

Benutzeranweisung:

1. Belege R_t mit 10
2. Tippe [RST]
3. Tippe 0 ein
4. [R/S] (Programm läuft; bei 10 stop)
5. Fahre fort bei 2.

Tastenprogramm (2):

[+]
[1]
[=]
[Pause]
[INV] [2nd] [x ≥ t]
[RST]
[0]
[RST]

Benutzeranweisung:

1. Belege R_t mit 10
2. Tippe [RST]
3. Tippe 0 ein
4. [R/S] (Programm läuft endlos)
5. Programmstop mit [R/S]

Tastenprogramm (3):

```
+
1
=
Pause
INV 2nd x≥t
RST
x⇄t
1
x⇄t
LBL 1
−
1
=
Pause
2nd x≥t
GTO 1
R/S
```

Benutzeranweisung:

1. Belege R_t mit 10
2. Tippe RST
3. Tippe 0 ein
4. R/S (Programm läuft)
5. Fahre fort bei 2.

© 2010 Joerg Woerner
Datamath Calculator Museum

Tastenprogramm (4):

```
+
1
= Pause
INV 2nd x≥t
RST
x⇄t
1
x⇄t
LBL 1
−
1
=
Pause
2nd x≥t
GTO 1
x⇄t
1 0
x⇄t
RST
```

Benutzeranweisung:

1. Belege R_t mit 10
2. Tippe RST
3. Tippe 0 ein
4. R/S (Programm (läuft) endlos)
5. Programstop mit R/S

Tastenprogramm (5):	Benutzeranweisung:
STO 0	1. Belege R_t mit 10
LBL 1	2. Belege R_1 mit 0
RCL 0	3. Belege R_2 mit 0
STO + 1	4. Tippe RST
RCL 1	5. 1 × 1-Zahl eintippen
Pause	6. R/S (Programme läuft)
1	7. Fahre fort bei 5
STO + 2	
RCL 2	
INV 2nd x≥t	
GTO 1	
0	
STO 2	
EXC 1	
R/S	
RST	

© 2010 Joerg Woerner
Datamath Calculator Museum

7. Programmierbeispiel

(Tasten: x↕t x≥t Intg RCL STO + EXC RST R/S)

7.1. PROBLEMSTELLUNG

Eine Zahl soll aus dem Dezimalsystem in ein anderes System übertragen werden, z.B.

13 → 1101 im 2-System
13 → 111 im 3-System
13 → 23 im 5-System

7.2. MATHEMATISIEREN

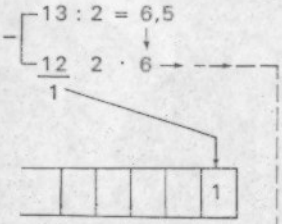
Es gibr für dieses Problem keine abgeschlossene Formel, es gibt aber ein "Rezept", nach dem man Schritt für Schritt zum Ziel kommt. In der Mathematik spricht man in diesem Zusammenhang von einem "Algorithmus".

Beispiel:

Die Dezimalzahl 13 ist in das 2-System zu übertragen.

Der Algorithmus ("Rezept") besteht aus 7 Schritten, die für dieses Beispiel 5 mal durchlaufen werden müssen.

Ausführung:

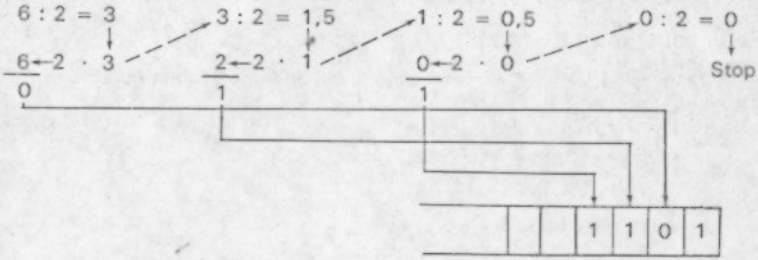


(gesuchte Zahl mit abgeteilten Ziffernstellen)

Algorithmus im Wortlaut:

1. Teile die Dezimalzahl (hier: 13) durch die Systemzahl (hier: 2)
2. Schneide vom Ergebnis evt. Nachkommastellen ab.
3. Multipliziere die so erhaltene Vorkommazahl (hier: 6) mit der Systemzahl.
4. Subtrahiere das Ergebnis (hier: 12) von der Dezimalzahl.
5. Setzt das Ergebnis (hier: 1) als Ziffer in die letztmögliche Stelle der gesuchten Zahl (die Zahl wird ziffernweise von rechts aufgebaut).
6. Setze die Vorkommazahl (hier: 6) an die Stelle der ursprünglichen Dezimalzahl und gebinne von vorn.
7. Wenn die Division (Schritt 1.) null ergibt, stoppe.

© 2010 Jörg Woerner
Datamath Calculator Museum



Ein besonderes Problem für den Rechner ergibt sich aus Punkt 5. des Algorithmus, denn es gibt keine Taste "setze an die letztmögliche Stelle". Man löst das Problem, indem man die jeweilige Ziffer mit 10^0, dann mit 10^1, 10^2 usw. multipliziert und dann summiert. Für dieses Beispiel ergibt sich:

$$
\begin{array}{rr}
1 \cdot 10^0 = & 1 \\
0 \cdot 10^1 = & 00 \\
1 \cdot 10^2 = & 100 \\
1 \cdot 10^3 = & 1000 \\
\hline
 & 1101
\end{array}
$$

7.3. PROGRAMMIEREN:

Man braucht ein Register, das die gewünschte Systemzahl aufnimmt; ein Register muß die Umläufe zählen, damit die Hochzahl für 10 sichergestellt ist; ein Register summiert.
Das t-Register muß mit 0 belegt werden, um das Ende der Umläufe herbeizuführen (Algorithmus Punkt 7.) R_t wird außerdem als Zwischenspeicher benutzt.

© 2010 Joerg Woerner
Datamath Calculator Museum

Registerverteilung:

R_1 – zählt die Umläufe
R_1 – summiert
R_2 – speichert die Systemzahl
R_t – Zwischenspeicher **und** Abfrage gegen null

Speicherverteilung:

[2nd] [Part] [3]

Tastenprogramm:	Kommentar:
[+]	1. Dezimalzahl dividieren (schwebt!)
[x⇄t]	2. Dezimalzahl in R_t zwischenspeichern
[RCL] [2]	3. Systemzahl in die Anzeige
[=]	4. Abschluß Division Dezimalzahl: Systemzahl
[Intg]	5. Abschneiden der Nachkommastellen
[x⇄t]	6. Vorkommazahl in R_t speichern, gleichzeitig Dezimalzahl wieder in die Anzeige
[−] [x⇄t] [X] [x⇄t] [RCL] [2]	7. Die Algorithmus-Punkte 3. und 4. werden
[=]	8. hier erledigt. Bei 12. ist die Differenz
	9. in der Anzeige und die Vorkommazahl des
	10. Quotienten ist in R_t gespeichert. Die
	11,12, ursprüngliche Dezimalzahl ist verloren.
[X] [1] [0] [y^x] [RCL] [0]	13. Algorithmus Punkt 5. wird hier erledigt. Die
[=]	14. Hochzahl für 10 kommt aus R_0 -(das erste

STO + 1	15,16,17. Mal null!)
1 STO + 0	18. Ergebnis von 13. bis 17. wird in R_1 aufsummiert.
0	19,20. R_0 wird um 1 heraufgesetzt
x⇄t	21. Null in die Anzeige
INV x=t RST	22. Null in R_t und gleichzeitig die Vorkommazahl des Quotienten in die Anzeige.
	23. Die Vorkommazahl wird gegen Null abge-
	24. fragt, und zwar, ob sie **ungleich** null ist. Wenn ja, wird der Programmzeiger nach Step 00 geschickt und der Algorithmus nochmal durchlaufen; die in der Anzeige stehende
STO 0	Vorkommazahl des Quotienten wird nach Algorithmusvorschrift 6. als "Dezimalzahl" aufgefaßt.
	25. Bei "nein" steht in der Anzeige eine Null.
EXC 1	Die wird in R_0 gespeichert, denn für einen ganz neuen Ablauf muß der Zählspeicher bei null anfangen.
	26. Die Null steht immer noch in der Anzeige; diese wird mit R_1, in dem das Endergebnis
R/S	aufsummiert wurde, ausgetauscht. Somit steht das Endergebnis jetzt in der Anzeige und
RST	gleichzeitig ist R_1 für einen neuen Durchgang vorbereitet.
	27. Pgm-Zeiger stoppt; neue Dezimalzahl kann eingegeben werden.
	28. Pgm-Zeiger beginnt mit neuer Dezimalzahl von vorn.

© 2010 Georg Woerner
Datamath Calculator Museum

7.4. EINLESEN DES PROGRAMMS

Prozedur wie immer

7.5. BENUTZERANWEISUNG

1. Belege R_0 mit 0 (0 STO 0)
2. Belege R_1 mit 0 (0 STO 1)
3. Belege R_2 mit der gewünschten Systemzahl (Systemzahl von 2 bis 10 einschließlich!)
4. Tippe RST
5. Tippe umzuwandelnde Dezimalzahl ein
6. R/S (Ergebnis erscheint)
7. Fahre fort bei 5.
   (evt.neue Systemzahl in R_2 geben!)

7.6. PROBELÄUFE

0 eingeben: Ergebnis: 0
1 eingeben: Ergebnis: 1 (in jedem System!)
13 eingeben: Ergebnis: siehe oben: 1101

© 2010 Joerg Woerner
Datamath Calculator Museum

Die Dsz-Taste

Die Dsz-Taste ist eine Abfragetaste, die dann eingesetzt wird, wenn man bei einer Schleife vorher weiß, wie oft sie durchlaufen werden soll.
Die Dsz-Taste kann nur in Verbindung mit R_0 eingesetzt werden. Dieses Register fällt dann für andere Aufgaben aus.
Dsz ist die Abkürzung von engl. **dekrement and skip on zero**. Zu deutsch: Zähle zurück und springe bei null.
Wenn der Programmzeiger die Taste Dsz erreicht, wird in R_0 um 1 zurückgezählt, dann erfolgt rechnerintern die Frage: Ist die Zahl in R_0 ungleich 0?
Wird die Frage mit "ja" beantwortet, nimmt der Programmzeiger den üblichen Ja-Weg wie bei den Abfragen mit dem t-Register auch: Er geht zum nächsten Step weiter, wo ihn eine Sprungadresse weiterschickt.
Bei "nein" (überspringt er diesen Step und arbeitet das "Nein-Programm" ab. (bei "nein" steht R_0 auf null!).

Beispiel (1)

Der Rechner soll bis 10 zählen und dann stoppen

Tastenprogramm (1):	Benutzeranweisung:
[+]	1. Belege R_0 mit 10
[1]	2. 0 in die Anzeige
[=]	3. [RST]
[Pause]	4. [R/S] (Programm läuft)
[Dsz]	5. Fahre fort bei 1. (evt. andere Zahl in R_0)
[RST]	
[R/S]	

R_0 kann auch mit einer negativen Zahl belegt werden, dann wird in R_0 1 hinzugezählt, bis null erreicht ist. Der Punkt 1. der Benutzeranweisung könnte demnach auch heißen: 1. Belege R_0 mit -10.

Beispiel (2) (vergl Programmierbeispiel 5)

Ein Vater legt bei der Geburt seiner Tochter ein Sparkonto an. Er zahlt 250.- DM ein bei jedem Geburtstag kommen 80.- DM hinzu. Die Bank gewährt einen Zinssatz von 6.75 %. Wieviel kann die Tochter bei ihren 18. Geburtstag abholen?

Datamath Calculator Museum

Registerverteilung:

R_0 – Dsz-Register
R_1 – Anfangskapital (hier: 250)
R_2 – Zinsfaktor (hier: 1.0675
R_3 – jährliche Zahlung (hier: 80)

Speicherverteilung:

[2nd] [Part] [4]

Tastenplan: (vergl. P 3.3.)	Kommentar:
[RCL] [1]	Kapitalwert in die Anzeige
[X] [RCL] [2] [+] [RCL] [3] [=] [STO] [1]	Kapitalwert wird mit Zinsfaktor multipliziert und jährl. Zahlung wird addiert. Das Ergebnis wird in R_1 gespeichert; der vorherige Wert geht dabei verloren.
[Dsz]	R_0 wird durch den Dsz-Befehl um 1 verringert, dann die Frage: ist Zahl in R_0 ungleich 0?
[RST]	Bei "ja" wird Berechnung wiederholt
[RCL] [1]	Beginn des "Nein-Programms), (wenn R_0 also auf null steht) Ergebnis in die Anzeige.
[R/S]	stop

Benutzeranweisung:

1. Belege R_0 mit 18 (Laufzeit)
2. Belege R_1 mit 250 (Anfangskapital
3. Belege R_2 mit 1.0765 (Zinsfaktor)
4. Belege R_3 mit 80 (jährl. Zahlung)
5. [RST]
6. Lege Fixkomma fest
7. R/S (Programm läuft, Ergebnis wird angezeigt.)
8. Fahre fort bei 1.

Die [Dsz]-Taste läßt auch mit dem [INV]-Befehl kombinieren:
[INV] [2nd] [Dsz] bedeutet: Zähle den Wert in R_0 um 1 zurück (bei negativer Zahl in R_0: Zähle um 1 herauf) und dann die Frage: Ist der Wert in R_0 gleich 0?

Der "Ja-"Weg und "Nein"-Weg wird vom Programmzeiger wie immer eingehalten. Wegen der Umkehrung der Frage überspringt der Programmzeiger den nächsten Step nach [Dsz], wenn R_0 **nicht** null ist.

Beispiel:

Von einer langen Datenkette sollen jeweils 10 zusammengefaßt und deren Mittelwert gebildet werden:

© 2018 Joerg Woerner
Datamath Calculator Museum

123/323/87/45/72/108/49/51/164/49/93/99/65/81/132/150/190/76/76/12/65...

$\bar{x}$ bilden | $\bar{x}$ bilden

Registerverteilung:

R_0 – Dsz-Register: 10
R_1 – Zwischenspeicher für Eingabewerte
R_2 – Summiert jeweils 10 Werte

Speicherverteilung:

[2nd] [Part] [3]

Tastenplan:

[STO] [1]
[STO] [+] [2]
[INV] [Dsz]
[GTO] [0]
[RCL] [1]

Kommentar:

Eingabewert wird in R_1 gespeichert
Eingabewert wird in R_2 aufsummiert
R_0 um 1 zurückzählen, dann Abfrage: Ist Wert in R_0 gleich 0? Wenn ja:
Springe nach [LBL] 0
Beginn des "Nein"-Programms: Eingabewert zur Kontrolle in die Anzeige, stop. Neuer Wert kann eingegeben werden.

R/S	
RST	Auf Step 00 springen und neuer Durchlauf
LBL 0	Beginn des "Ja"-Programms : Adresse LBL 0
RCL 2	Der summierte Wert in die Anzeige
÷ 1 0 =	Mittelwert wird errechnet
R/S	Stop zur Entnahme des Mittelwertes. Neuer Einzelwert wird eingegeben
STO 2	Der Einzelwert wird in R2 zwischengespeichert (Wert in R2 wird nicht mehr benötigt, kann also überschrieben werden).
1 0	10 in die Anzeige
STO 0	10 in R0 speichern für Dsz-Umläufe
0	0 in die Anzeige
EXC 2	Austauschen: 0 in R2, weil die neue Summierung bei 0 beginnen muß
	Der neue Eingabewert wieder in die Anzeige, mit ihm beginnt die neue Serie.
RST	Sprung auf Step 00 und neuer Programmdurchlauf.

© 2010 Joerg Woerner
Datamath Calculator Museum

Benutzeranweisung:

1. Belege R0 mit 10
2. Belege R2 mit 0
3. RST
4. Wert eingeben
5. R/S (Eingabewert wird verrechnet und erscheint zur Kontrolle nochmals in der Anzeige.
6. Fahre fort bei 4. Nach 10. Eingabe erscheint Mittelwert!

UNTERPROGRAMME

Wenn zur Lösung eines Problems bestimmte Tastenfolgen merhfach zu durchlaufen sind, ist es zur Einsparung von Steps oft zweckmäßig, diese Tastenfolgen gesondert zu speichern und mit einer Adresse zu versehen, an die man den Programmspeicher schicken kann.
Das ist im Programmbeispiel 6 (Erweiterung) schon erläutert.
Im Programmbeispiel 6. ist dieser Programmteil gleichzeitig das Ende der Berechnungen, sodaß nach dem Durchlauf keine weiteren Befehle notwendig sind.
Nun ist es aber häufig erforderlich, daß nach Durchlauf eines solchen ausgesonderten Programmteils im "Hauptprogramm" weiterzurechnen ist. Der Programmzeiger muß also aus dem "Unterprogramm" wieder an die richtige Stelle im Hauptprogramm geschickt werden. Mit den beiden Programmiertasten [SBR] und [INV] [SBR] kann dieses Programmierproblem gelöst werden.

[SBR] (von engl.: **subroutine** = Unterprogramm) in Verbindung mit einer Labeladresse, etwa [SBR], bedeutet: Programmzeiger zum Label 1 springen, und das darunterstehende Programm abarbeiten. Gleichzeitig mit [SBR] 1 wird in einem internen Register (Unterprogrammrücksprungregister) die Stepnummer notiert, die unmittelbar nach dem SBR-Befehl steht.
[INV] [SBR] muß am Ende des Unterprogramms stehen, es bedeutet: Programmzeiger zurück auf die Step-Nummer, die beim "Absprung" ins Unterprogramm im Rücksprungregister gespeichert wurde.

Schematischer Verlauf einer Unterprogramm-routine:

1. . . .
2. . . .
3. . . .
4. . . .
5. . . .
6. SBR 1
7. . . .
8. . . .
9. . . .
10. . . .
11. SBR 1
12. . . .
13. . . .
14. . . .
15. R/S
16. LBL 1
17. . . .
18. . . .
19. . . .
20. . . .
21. INV SBR

Der Programmzeiger nimmt folgenden Weg:

1.-5. Abarbeiten des Tastenprogramms im Hauptprogramm
6. Absprung zum Unterprogramm nach Label 1; Step hinter SBR wird notiert
16.-20. Unterprogramm wird abgearbeitet
21. Befehl zum Rücksprung ins Hauptprogramm; hier Ordnungs-Nr. 7.
7.-10. Weiterbearbeitung des Hauptprogramms
11. Nochmaliger Absprung ins Unterprogramm; Step hinter SBR wird notiert, der alte Wert wird gelöscht
16.-20. Unterprogramm wird abgearbeitet
21. Rücksprung ins Hauptprogramm, hier Ordnungsnummer 12.
12.-14. Hauptprogramm wird zu Ende geführt;
15. Ende des Hauptprogramms.

© 2012 Joerg Woerner
Datamath Calculator Museum

Man kann beliebig oft vom Hauptprogramm ins Unterprogramm springen;

Von einem Unterprogramm kann kein anderes Unterprogramm aufgerufen werden. Man kann aber vom Hauptprogramm in ein zweites oder, soweit der Speicherraum reicht, in ein drittes Unterprogramm springen.

Schematischer Ablauf bei zwei Unterprogrammen:	Der Programmzeiger nimmt folgenden Weg:
1. . . .	1.-3. Tasten im Hauptprogramm
2. . . .	4. Programmzeiger wird zum Unterprogramm
3. . . .	unter Label 1 geschickt
4. SBR 1	22.-25. Unterprogramm 1 wird abgearbeitet
5. . . .	26. Rücksprungbefehl für den Programmzeiger
6. . . .	hier nach Ordnungsnummer 5.
7. . . .	5.-8. Weiter im Hauptprogramm
8. . . .	9. Sprungbefehl zum Unterprogramm 2.
9. SBR 2	27.-31. Abarbeiten des Unterprogramms 2
10. . . .	32. Rücksprungbefehl für den Programmzeiger
11. . . .	hier nach Ordnungsnummer 10.
12. . . .	10.-13. Weiter im Hauptprogramm
13. . . .	14. Sprungbefehl zu Unterprogramm 1.
14. SBR 1	22.-25. Unterprogramm 1 wird abgearbeitet.
15. SBR 2	26. Rücksprungbefehl für den Programmzeiger
16. . . .	hier nach Ordnungsnummer 15.
17. . . .	15. Hier bekommt der Zeiger sofort den Befehl
18. . . .	zum Unterprogramm 2 zu springen
19. . . .	27.-31. Abarbeiten des Unterprogramms 2
20. . . .	32. Rücksprungbefehl für den Programmzeiger
21. R/S	hier nach Ordnungsnummer 16.
22. LBL 1	16.-20. Weiter im Hauptprogramm.
23. . . .	21. Ende des Hauptprogramms, stop.
24. . . .	
25. . . .	
26. INV SBR	
27. LBL 2	
28. . . .	
29. . . .	
30. . . .	
31. . . .	
32. INV SBR	

Die Reihenfolge der Befehle SBR 1 und SBR 2 ist hier willkürlich gewählt; jede andere Reihenfolge - je nach Problem - kann gewählt werden.

© 2000 Jörg Woerner
Datamath Calculator Museum

8. Programmierbeispiel

8.1. PROBLEMSTELLUNG

Ein Auto verbraucht bei einer bestimmten Geschwindigkeit eine bestimmte Kraftstoffmenge. Erhöht man die Geschwindigkeit um einen gewissen Prozentsatz, dann erhöht sich der Kraftstoffverbrauch nicht um denselben Prozentsatz, sondern er steigt steiler an.

Bei einem Autotyp wird der Kraftstoffverbrauch zwischen 60 km/h und 160 km/h nach folgender Formel berechnet:

$$k = 0{,}001\ g^2 + \frac{80}{g}$$

g: = Geschwindigkeit
k: = Kraftstoffverbrauch (l pro 100 km)

Es soll eine Tabelle aufgestellt werden, aus der hervorgeht, wie sich der prozentuale Kraftstoffverbrauch gegenüber der prozentualen Geschwindigkeitserhöhung verhält, und zwar soll eine 10% ige und eine 15% ige Geschwindigkeitserhöhung fest vorgegeben werden.

Datamath Calculator Museum

Geschw.	Kraftstoff-erhöhung bei 10%	Kraftstoff-erhöhung bei 15%
60 km/h	12,9 %	20,0 %
65 km/h	14,2 %	22,0 %
70 km/h	15,3 %	23,7 %
75 km/h	16,2 %	25.0 %

8.2. MATHEMATISIERUNG

1. Kraftstoffverbrauch für Ausgangswert (z.B. 60 km/h) nach der angegeben Formel berechnen:

$$k = 0{,}001 \cdot 60^2 + \frac{80}{60} = 4{,}933 \text{ (l pro 100 km)}$$

2. Kraftstoffverbrauch für den um 10% erhöhten Ausgangswert:

$$k = 0{,}001 \cdot (1{,}1 \cdot 60)^2 + \frac{80}{1{,}1 \cdot 60} = 5{,}568 \text{ (l/100 km)}$$

3. Prozentuale Erhöhung des Kraftstoffverbrauchs:

$$p_1 = \left(\frac{k_1}{k} - 1\right) \cdot 100$$

Um Steps einzusparen, wird im Programm nur

$p_1 = \frac{k_1}{k}$ gerechnet. Die prozentuale Erhöhung ergibt sich aus den Nachkommastellen:

$$p_1 = \frac{5{,}568}{4{,}933} = 1{,}129 \text{ (im Kopf: } (1{,}129 - 1) \cdot 100 = 12{,}9)$$

4. Kraftstoffverbrauch für den um 15 % erhöhten Ausgangswert:

$$k_2 = 0{,}001 \cdot (1{,}15 \cdot 60)^2 + \frac{80}{1{,}15 \cdot 60} = 5{,}920 \text{ (l/100 km)}$$

5. Prozentuale Erhöhung des Kraftstoffverbrauchs:

$$p_2 = \frac{k_2}{k} = \frac{5{,}920}{4{,}933} = 1{,}200 \text{ (entspricht 20% Erhöhung)}$$

© 2010 Joerg Woerner
Datamath Calculator Museum

8.3. PROGRAMMIERUNG

Registerverteilung:

R_0 – Ausgangswert u. erhöhte Ausgangswerte

R_t – Zwischenspeicher

Speicherverteilung:

[2nd] [Part] [1]

Tastenprogramm:	Kommentar:
1. STO 0	1. Hauptprogramm beginnt: Ausgangswert
2. SBR 1	wird in R_0 gespeichert
3. x⇄t	2. Hauptprogramm wird unterbrochen und der
4. 1 • 1	Programmzeiger nach Label 1 geschickt
5. STO X 0	15. Adresse für Unterprogramm 1
6. SBR 1	16. → 24. Kraftstoffverbrauch für den
7. SBR 2	Ausgangswert wird nach Formel berechnet.
8. 1 • 1	23. Programmzeiger springt zurück hier auf
9. STO ÷ 0	Ordnungsnummer 3. im Hauptprogramm
10. 1 • 1 5	3. Der im Unterprogramm ermittelte Kraftstoff-
11. STO X 0	verbrauch wird als "Grundwert" für die
12. SBR 1	prozentuale Erhöhung des Kraftstoffverbrauchs
13. SBR 2	in R_t gespeichert.
14. R/S	4. 1,1 in die Anzeige
15. LBL 1	5. Wert in R_0 wird mit 1,1 multipliziert d.h.,
16. • 0 0 1	der Ausgangswert wird um 10% erhöht.
17. X	6. Programmzeiger springt ins Unter-
18. RCL 0	programm 1
19. x^2	16. ≋ 24. Kraftstoffverbrauch für den um 10%
20. +	erhöhten Ausgangswert wird nach Formel
21. 8 0	errechnet
22. ÷	25. Programmzeiger springt zurück ins
23. RCL 0	Hauptprogramm auf Ordnungsnummer 7.
24. =	7. Programmzeiger springt ins Unter-
25. INV SBR	programm 2.
26. LBL 2	26. Label-Adresse für Unterprogramm 2.
27. x⇄t	27. Kraftstoffverbrauch für Ausgangswert in
28. ÷	die Anzeige ("Grundwert").
29. x⇄t	28. wird dividiert durch
30. =	29. erhöhten Kraftstoffverbrauch;
31. 1/x	"Grundwert" wird wieder in R_t gespeichert und
32. R/S	so für die zweite Berechnung "gerettet"
33. INV SBR	

© 2020 Jörg Woerner
Datamath Calculator Museum

Tastenprogramm:	Kommentar:
	30. Ergebnis der Division wird ermittelt
	31. Um den "Grundwert" zu retten, mußte Grundwert durch erhöhten Wert dividiert werden, deswegen muß das Ergebnis umgekehrt werden. (siehe 'Mathematisierung', 3. Rechnung).
	32. Stop für die Notierung der prozentualen Erhöhung des Kraftstoffverbrauchs bei einer 10% igen Erhöhung der Geschwindigkeit
	33. Programmzeiger springt zurück ins Hauptprogramm Ordnungsnummer 8.
	8. 1,1 in die Anzeige
	9. Wert in R_0 wird durch 1,1 dividiert, die Erhöhung um 10% wird damit zurückgenommen.
	10. 1,15 in die Anzeige
	11. Wert in R_0 wird mit 1,15 multipliziert, d.h. um 15% erhöht.
	12. Programmzeiger springt ins Unterprogramm 1
	16. ≷ 24. Kraftstoffverbrauch für 15% ige Geschwindigkeitserhöhung wird nach Formel berechnet.
	25. Programmzeiger springt zurück ins Hauptprogramm hier auf Ordnungsnummer 13.
	13. Programmzeiger wird ins Unterprogramm 2 geschickt
	27. → 31. Prozentuale Erhöhung des Kraftstoffverbrauchs bei einer 15% igen Erhöhung der Geschwindigkeit wird errechnet.
	32. Stop für Notierung des Ergebnisses
	33. Programmzeiger springt zurück ins Hauptprogramm hier auf Ordnungsnummer 14.
	14. Stop des Programms

© 2011 Joerg Woerner
Datamath Calculator Museum

8.4. BENUTZERANWEISUNG

1. Tippe Ausgangswert (Geschwindigkeit) in die Anzeige
2. Tippe [RST]
3. [R/S] (Verbrauchserhöhung in % bei 10% iger Geschwindigkeitserhöhung erscheint)
4. R/S (Verbrauchserhöhung in % bei 15% iger Geschwindigkeitserhöhung erscheint)
5. Fahre fort bei 1.

Bemerkung: an Stelle 5. könnte man nochmals [R/S] drücken, der letzte Wert bleibt in der Anzeige, der Programmzeiger wurde aus dem Unterprogramm ins Hauptprogramm springen und dort auch nur den Befehl 'stop' bekommen.

Besonderheiten für Unterprogramme

1. Wenn im Hauptprogramm Rechnungen 'schweben', darf im Unterprogramm die Taste [=] nicht benutzt werden, denn diese Taste schließt alle schwebenden Rechnungen ab, also auch die im Hauptprogramm.
   Zwischenergebnisse, die im Unterprogramm ermittelt werden und im Hauptprogramm weiter verarbeitet werden sollen, können mit Klammern eingeleitet und abgeschlossen werden.
2. Wenn Unterprogramme eingerichtet wurden, so sollte man die Taste [RST] mit Bedacht einsetzen, denn diese Taste setzt nicht nur den Programmzeiger auf Step 00, sondern löscht auch das Unterprogramm-Rücksprungregister.
   Wenn man [RST] einsetzt, muß man sicher ein, daß **nach** dieser Taste kein Rücksprungbefehl aus einem Unterprogramm mehr erfolgt.

© 2010 Joerg Woerner

9. Zusammenfassung der Programmiertasten und ihre Wirkung

1. [R/S] (von to run / to stop)

Von der RUN-Ebene aus:	Stoppt den Programmzeiger, wenn er läuft; startet den Programmzeiger, wenn er steht.
Einprogrammiert:	Stoppt den Programmzeiger

2. [RST] (von to reset = zurücksetzen)

Von der RUN-Ebene aus: Setzt den Programmzeiger auf Step 00, wo er stehen bleibt; löscht gleichzeitig das Unterprogramm-Rücksprungregister

Einprogrammiert: Setzt den Programmzeiger auf Step 00; Pgm-Zeiger läuft weiter.
löscht Rücksprungregister

3. [GTO] n (von go to = gehe zu)

Von der RUN-Ebene aus: Programmzeiger wird an die Labeladresse geschickt, die die Zahl n ($0 \leqslant n \leqslant 9$) angibt; Pgm-Zeiger bleibt dort stehen.

Einprogrammiert: Programmzeiger wird an die Labeladresse n geschickt und läuft weiter.

4. [SBR] n (von subroutine = Unterprogramm)

Von der RUN-Ebene aus: keine Wirkung

Einprogrammiert: Programmzeiger wird an die Labeladresse geschickt, die die Zahl n ($0 \leqslant n \leqslant 9$) angibt; Pgm-Zeiger läuft weiter.
Gleichzeitig wird in einem internen Register (Unterprogramm-Rücksprungregister) die Step-Nummer gespeichert, die dem SBR-befehl folgt.

© 2010 Joerg Woerner
Datamath Calculator Museum

5. [INV] [SBR]

Von der RUN-Ebene aus: Keine Wirkung

Einprogrammiert: Programmzeiger springt auf die Step-Nr., die im Rücksprungregister gespeichert wurde. Ist das Rücksprungregister gelöscht, so stoppt der Pgm-Zeiger bei diesem Befehl.
[INV] [SBR] in einem Hauptprogramm ist also dem [R/S]-Befehl gleich.

6. [Pause]

Von der RUN-Ebene aus: Keine Wirkung

Einprogrammiert: Programmzeiger stoppt für einen Augenblick seinen Lauf; es wird die Zahl in der Anzeige ausgewiesen, die gerade in R_x ist.

7. [LBL] n (von label = Etikett/Aufschrift)

Von der RUN-Ebene aus: Keine Wirkung

Einprogrammiert: Ist kein Befehl, sondern markiert eine Adresse für den Programmzeiger. ($0 \leqslant n \leqslant 9$) Pgm-Zeiger nimmt die Lbl-Adresse nur zur Kenntnis, wenn er mit [GTO] n oder [SBR] n gezielt hingeschickt wurde; andernfalls 'überläuft' er eine Lbl-Adresse.

8. [x=t]

Von der RUN-Ebene aus: Keine Wirkung

Einprogrammiert: Abfrage: Ist die Zahl in R_X gleich der Zahl in R_t? Bei 'ja' geht der Programmzeiger weiter zu dem Step, der unmittelbar folgt. Bei 'nein' überspringt der Pgm-Zeiger den nachfolgenden Step und beginnt vom übernächsten Step mit der weiteren Verarbeitung.

9. [INV] [x=t]

© 2010 Joerg Woerner
Datamath Calculator Museum

Von der RUN-Ebene aus: Keine Wirkung

Einprogrammiert: Abfrage: Ist die Zahl in R_X ungleich der Zahl in R_t? Bei 'ja' bzw. 'nein': siehe [x=t].

10 [x≥t]

Von der RUN-Ebene aus: Keine Wirkung

Einprogrammiert: Abfrage: Ist die Zahl in R_X größer oder gleich die Zahl in R_t?
Bei 'ja' bzw. 'nein': siehe [x=t]

11. [INV] [x≥t] Von der RUN-Ebene aus: kleine Wirkung einprogrammiert :: Abfrage: Ist die Zahl in R_X kleiner als die Zahl in R_t! Bei 'ja' bzw. 'nein': siehe [x=t].

12. Dsz (von dekrement und skip on zero = zähle zurück u. springe bei null)

Von der RUN-Ebene aus: Keine Wirkung

Einprogrammiert: Die Zahl, die in R_0 ist, wird um 1 zurückgezählt. Danach erfolgt die Abfrage: Ist die Zahl in R_0 ungleich 0? Bei 'ja' bzw. 'nein': siehe X=t

Wenn die Zahl in R_0 negativ ist, wird 1 hinzugezählt und dann die Abfrage: Ist die Zahl in R_0 ungleich 0?

Bei nicht ganzen Zahlen verwandelt der Rechner diese in die nächst größere und verfährt dann wie oben beschrieben.
Aus 7,8 wird 8
Aus 7,1 wird 8
Aus −7,8 wird −8
Aus −7,1 wird −8

13. INV 2nd Dsz (dekrement and skip if not zero = zähle zurück und springe Falls nichtnull). - Der Rechner bearbeitet bei dieser Tastenfolge wie unter 12 beschrieben mit der Ausnahme daß das Folgende Programmschritt übersprungen wird wenn R_0 ungleich Null ist.

© 2010 Joerg Woerner
Datamath Calculator Museum

KAPITEL III — ANWENDUNGEN

Abschnitt 1 — Statistische Analyse

- Dieses Programm berechnet den Mittelwert und die Standardabweichung eines Datensatzes.

- **Gleichungen:**

Mittelwert: $\bar{x} = \sum \frac{x_i}{N}$

Standardabweichung: $\sigma n = \sqrt{\sum \frac{(x_i - \bar{x})^2}{N}}$

- **Speicherbelegung:**

M_0	M_1	M_2	
$\sum x$	$\sum x^2$	N	Programmschritte
			31 — 00

© 2010 Joerg Woerner
Datamath Calculator Museum

- **Programmlauf:**

- [2nd] [CM] drücken, dann [RST]
- Dateneingabe xi [R/S]
- Berechnung der statistischen Ergebnisse [GTO] [0], [R/S] [R/S]

- **Beispiel:**

Taste	Anzeige	Bemerkungen
[ON/C]	0	
[2nd] [CM]	0	
[RST]	0	
1.75 [R/S]	1	Anzeige der Zahl der eingegebenen Werte
1.62 [R/S]	2	
1.84 [R/S]	3	
1.72 [R/S]	4	
1.68 [R/S]	5	
[GTO] 0 [R/S]	1.722	$\bar{x}$
[R/S]	0.0733212	σn

- **Programmprotokoll:**

Tasten	Anzeige	Schritt Nr.	Bemerkungen
2nd Part 3			3 Speicher reservieren
LRN 2nd CP	--	St	
STO + 0	61.85.00	00	
x^2	34	01	Dateneingabe
STO + 1	61.85.01	02	
1	01	03	
STO + 2	61.85.02	04	
RCL 2	71.02	05	
R/S	13	06	
RST	21	07	
LBL 0	23.00	08	Mittelwertberechnung
RCL 0	71.00	09	
+	55	10	
RCL 2	71.02	11	
=	95	12	
R/S	13	13	
RCL 1	71.01	14	Berechnung der Standardabweichung
+	55	15	
RCL 2	71.02	16	
−	75	17	
(	53	18	
RCL 0	71.00	19	
+	55	20	
RCL 2	71.02	21	
)	54	22	
x^2	34	23	
=	95	24	
$\sqrt{x}$	35	25	
R/S	13	26	
LRN			

© 2011 Joerg Woerner
Datamath Calculator Museum

Abschnitt 2 — Kombinationen und Permutationen

- Dieses Programm berechnet die möglichen Kombinationen (C_n^p) oder Permutationen (A_n^p) von p Elementen n-ter Ordnung.

- **Gleichungen:** $C_n^p = \frac{n!}{p!\,(n-p)!}$

  $A_n^p = \frac{n!}{(n-p)!}$

- **Speicherbelegung:**

M_0	M_1	
n	p	Programmschritte
		39 — 00

© 2010 Joerg Woerner
Datamath Calculator Museum

- **Programmlauf**

Zur Berechnung von C_n^p: n eingeben
[GTO] [0] [R/S] drücken
p eingeben
[R/S] drücken

Zur Berechnung von A_n^p: n eingeben
[GTO] 1 [R/S] drücken
p eingeben
[R/S] drücken

- **Beispiel:**

Taste	Anzeige	Bemerkungen	
C_{12}^3	12 [GTO] 0 [R/S]	12	
	3 [R/S]	220	$C_{12}^3 = 220$
A_8^5	8 [GTO] 1 [R/S]	8	
	5 [R/S]	6720	$A_8^5 = 6720$

- **Programmprotokoll:**

Tasten	Anzeige	Schritt Nr.	Bemerkungen
2nd Part 2			2 Speicher reservieren
LRN 2nd CP	— —	St	
LBL 0	23.00	00	C_n^p Berechnung
2nd SBR 2	28.02	01	
RCL 0	71.00	02	
2nd n!	40	03	
+	55	04	
RCL 1	71.01	05	
2nd n!	40	06	
+	55	07	
2nd SBR 3	28.03	08	
=	95	09	
R/S	13	10	
LBL 1	23.01	11	A_n^p Berechnung
2nd SBR 2	28.02	12	
RCL 0	71.00	13	
2nd n!	40	14	
+	55	15	
2nd SBR 3	28.03	16	
=	95	17	
R/S	13	18	
LBL 2	23.02	19	Unterprogramm für Eingabe
STO 0	61.00	20	
R/S	13	21	
STO 1	61.01	22	
INV 2nd SBR	−28	23	
LBL 3	23.03	24	Rechenroutine
(	53	25	
RCL 0	71.00	26	
−	75	27	
RCL 1	71.01	28	
)	54	29	
2nd n!	40	30	
INV 2nd SBR	−28	31	
LRN			

2010 Joerg Woerner
Datamath Calculator Museum

Abschnitt 3 — Hyperbel Funktionen

• Dieses Programm ermittelt die hyperbolischen Sinus, Kosinus und Tangens.

Gleichungen:

$$\sinh x = \frac{e^x - e^{-x}}{2}$$

$$\cosh x = \frac{e^x + e^{-x}}{2}$$

$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

• **Speicherbelegung:**

M_0	M_1	
x	cosh x	Programmschritte
		39 — 00

© 2000 Jörg Woerner
Datamath Calculator Museum

• **Programmlauf:**

Zur Ermittlung von cosh x: x eingeben
[GTO] [0] [R/S] drücken → cosh x
sinh x: x eingeben
[GTO] [1] [R/S] drücken → sinh x
tanh x: x eingeben
[GTO] [2] [R/S] drücken → tanh x

• **Beispiele:**

		Taste	Anzeige
cosh (3.2)	:	3.2 GTO 0 R/S	12.286646
sinh (8.1)	:	8.1 GTO 1 R/S	1647.2339
tanh (0.5)	:	.5 GTO 2 R/S	0.4621172

HYPERBEL FUNKTIONEN

• **Programmprotokoll:**

Tasten	Anzeige	Schritt Nr.	Bemerkungen
2nd Part 2			
LRN 2nd CP	--	St	
LBL 0	23.00	00	cosh x ermitteln
STO 0	61.00	01	
INV ln x	-32	02	
+	85	03	
RCL 0	71.00	04	
+/-	94	05	
INV ln x	-32	06	
=	95	07	
+	55	08	
2	02	09	
=	95	10	
INV 2nd SBR	-28	11	
LBL 1	23.01	12	sinh x ermitteln
STO 0	61.00	13	
INV ln x	-32	14	
−	75	15	
RCL 0	71.00	16	
+/-	94	17	
INV ln x	-32	18	
=	95	19	
+	55	20	
2	02	21	
=	95	22	
INV 2nd SBR	-28	23	
LBL 2	23.02	24	tanh x ermitteln
2nd SBR 0	28.00	25	
STO 1	61.01	26	
RCL 0	71.00	27	
2nd SBR 1	28.01	28	
+	55	29	
RCL 1	71.01	30	
=	95	31	
R/S	13	32	
LRN			

2010 Joerg Woerner
Datamath Calculator Museum

Abschnitt 4 — Areafunktionen

- Dieses Programm ermittelt folgende Funktionen: arsinh x, arcosh x, artanh x.

- **Gleichungen:**

$$\sin h^{-1} x = \mathrm{Ln}\,(x + \sqrt{x^2 + 1})$$

$$\cos h^{-1} x = \mathrm{Ln}\,(x + \sqrt{x^2 - 1}), \quad x \geqslant 1$$

$$\tan h^{-1} x = \frac{1}{2}\,\mathrm{Ln}\,(\frac{1 + x}{1 - x}), \quad -1 < x < 1$$

- **Speicherbelegung:**

M_0	
x	Programmschritte
	47 — 00

- **Programmlauf**

x eingeben [GTO] 0 [R/S] drücken, um arsinh x zu ermitteln
oder [GTO] 1 [R/S] drücken, um arcosh x zu ermitteln
oder [GTO] 2 [R/S] drücken, um artanh x zu ermitteln

© 2010 Joerg Woerner
Datamath Calculator Museum

Beispiele:

		Taste	Anzeige
$\cosh^{-1}(6.2)$	:	6.2 [GTO] 0 [R/S]	2.5111285
$\sinh^{-1}(3.7)$	:	3.7 [GTO] 1 [R/S]	2.0192607
$\tanh^{-1}(0.5)$	:	.5 [GTO] 2 [R/S]	0.5493061

- **Programmprotokoll:**

Tasten	Anzeige	Schritt Nr.	Bemerkungen
[2nd] [Part] [1]	48.1		
[LRN] [2nd] [CP]	--	St	
[LBL] [1]	23.01	00	arsinh x ermitteln
[STO] [0]	61.00	01	

(→)

AREAFUNKTIONEN

(−) Tasten	Anzeige	Schritt Nr.	Bemerkungen
[x²]	34	02	
[+]	85	03	
[1]	01	04	
[=]	95	05	
[√x]	35	06	
[+]	85	07	
[RCL] [0]	71.00	08	
[=]	95	09	
[lnx]	32	10	
[R/S]	13	11	
[LBL] [0]	23.00	12	arcosh x ermitteln
[STO] [0]	61.00	13	
[x²]	34	14	
[−]	75	15	
[1]	01	16	
[=]	95	17	
[√x]	35	18	
[+]	85	19	
[RCL] [0]	71.00	20	
[=]	95	21	
[lnx]	32	22	
[R/S]	13	23	
[LBL] [2]	23.02	24	artanh x ermitteln
[STO] [0]	61.00	25	
[+]	85	26	
[1]	01	27	
[=]	95	28	
[+]	55	29	
[(]	53	30	
[1]	01	31	
[−]	75	32	
[RCL] [0]	71.00	33	
[)]	54	34	
[=]	95	35	
[lnx]	32	36	
[+]	55	37	
[2]	02	38	
[=]	95	39	
[R/S]	13	40	
[LRN]	48.1		

© 2010 Joerg Woerner
Datamath Calculator Museum

Abschnitt 5 — Quadratische Gleichungen

- Dieses Programm löst die Gleichung:

Gleichungen:

$\Delta = b^2 - 4ac$

1. $\Delta > 0$ $\quad x1 = \frac{-b - \sqrt{\Delta}}{2a}$

   $x2 = \frac{-b + \sqrt{\Delta}}{2a}$

2. $\Delta = 0$ $\quad x1 = x2 = \frac{-b}{2a}$

3. $\Delta < 0$

   $z1 = \alpha + j\,\beta$

   $z2 = \alpha - j\,\beta$

   $\alpha = \frac{-b}{2a} \qquad \beta = \frac{-\sqrt{-\Delta}}{2a}$

© 2010 Rechenberg Wien
Datamath Calculator Museum

- **Speicherbelegung:**

M_0	M_1	
a	b	Programm- Schritte
		39 00

QUADRATISCHE GLEICHUNGEN

- Programmprotokoll:

Tasten	Anzeige	Schritt Nr	Bemerkungen
2nd Part 2			
LRN 2nd CP	--	St	
2nd C.t	56	00	
STO 1	61.01	01	b Wert speichern
x^2	34	02	
−	75	03	
R/S	13	04	c Wert holen
×	65	05	
4	04	06	
×	65	07	
RCL 0	71.00	08	
=	95	09	
R/S	13	10	Δ anzeigen
INV 2nd x≥t	−27	11	Δ Signal testen
GTO 1	22.01	12	
$\sqrt{x}$	35	13	.Δ ≥ 0
+	85	14	
RCL 1	71.01	15	
=	95	16	
2nd SBR 0	28.00	17	
R/S	13	18	x1 anzeigen
+/−	94	19	
−	75	20	
RCL 1	71.01	21	Bemerkung: $x2 + x1 = -\frac{b}{a}$
+	55	22	
RCL 0	71.00	23	
=	95	24	
R/S	13	25	x2 anzeigen
LBL 1	23.01	26	komplexe Wurzel berechnen
+/−	94	27	
$\sqrt{x}$	35	28	.Δ<0
2nd SBR 0	28.00	29	
R/S	13	30	β anzeigen
RCL 1	71.01	31	
LBL 0	23.00	32	
+	55	33	
2	02	34	(−)

© 2010 Joerg Woerner
Datamath Calculator Museum

QUADRATISCHE GLEICHUNGEN

Tasten	Anzeige	Schritt Nr.	Bemerkungen
[+]	55	35	
[RCL] [0]	71.00	36	
[=]	95	37	
[+/−]	94	38	
[INV] [2nd] [SBR] [LRN]	−28	39	α anzeigen

• **Programmlauf:**

a eingeben, dann [STO] [0] drücken
b eingeben, dann [RST] [R/S] drücken
c eingeben, dann [R/S] drücken → Δ
1 Wenn Δ ≥ 0 [R/S] drücken → x^1
[R/S] drücken → x^2
2 Wenn Δ < 0 [R/S] [R/S] [R/S] [R/S] drücken → β
[R/S] drücken → α

Beispiel:
$x^2 - 2x + 5 = 0$

© 2010 Joerg Woerner
Datamath Calculator Museum

Taste	Anzeige	Bemerkungen
[1] [STO] [0]	1	
[2] [+/−] [RST] [R/S]	4	
[5] [R/S]	−16	Δ ist 0, z1 und z2 sind komplexe Zahlen.
[R/S]	−2	β z1 = 1 − 2j
[R/S]	1	α z2 = 1 + 2j

Taste	Anzeige	Bemerkungen
[1] [STO] [0]	1	
[1] [RST] [R/S]	1	
[6] [+/−] [R/S]	25	Δ ist 0, x1 und x2 sind reelle Zahlen
[R/S]	−3	x1 = −3
[R/S]	2	x2 = 2

Abschnitt 6 — Funktionsplotter

- Dieses Programm ermittelt die Werte einer Funktion in einem gegebenen Intervall. Die Funktion, die geplottet werden soll, muß als ein Programm unter dem Label 0 eingegeben werden und muß von [INV] [2nd] [SBR] gefolgt sein.

- **Gleichungen:**

F(x) wird berechnet, x ändert sich von a bis b mit Schritt c

- **Speicherbelegung:**

M_0	M_1		
a	c	f(x) Definition	Programm
		39 19	18 00

© 2010 Joerg Woerner
Datamath Calculator Museum

- **Ausführung dieses Programmes:**

Nach Eingabe von F(x):
- a eingeben, [RST], [R/S] drücken
- b eingeben, [R/S] drücken
- c eingeben, [R/S] drücken

- **Beispiel:**

- Den Rechner mit der [DRG] Taste in Gradmodus schalten
- $f(x) = e^{\sin x}$ für x von 0 bis 180 Grad in Schritten von 10 Grad berechnen.

	Taste	Anzeige
0 [RST] [R/S]	0	a
180 [R/S]	180	b
10 [R/S]	0	x
	1	f(x)
	10	x
	1.189637	f(x)
	20	x
	1.4077887	f(x)

- **Programmprotokoll:**

Tasten	Anzeige	Schritt Nr.	Bemerkungen
2nd Part 2			
LRN 2nd CP	– –	St	
STO 0	61.00	00	
R/S	13	01	
x⇄t	51	02	
ON/C	15	03	
R/S	13	04	
STO 1	61.01	05	
RCL 0	71.00	06	
LBL 1	23.01	07	
2nd Pause	96	08	
SBR 0	28.00	09	Aufruf des Unterprogrammes zur Berechnung von f(x)
2nd Pause	96	10	
RCL 1	71.01	11	
STO + 0	61.85.00	12	
RCL 0	71.00	13	
INV 2nd x≥t	–27	14	
GTO 1	22.01	15	
2nd x=t	26	16	
GTO 1	22.01	17	
R/S	13	18	
LBL 0	23.00	19	Definition der Funktion
sin	42	20	(kann bis zu 21 Schritte gebrauchen)
INV ln x	–32	21	In diesem Beispiel : $f(x) = e^{\sin x}$
INV 2nd SBR	–28	22	
LRN			

© 2010 Joerg Woerner
Datamath Calculator Museum

Abschnitt 7 — Grenze einer Funktion

- Dieses Programm ermittelt das Verhalten einer Funktion, (f(x)) wenn die Veränderliche sich einem bestimmten Wert nähert. In manchen Fällen kann die Funktion bei Erreichen des Endpunktes abweichen, ins Unendliche gehen oder auf einen spezifischen Grenzwert konvergieren.
- **Gleichungen:**

Berechne lim fx
$x \rightarrow x_0$

F(x) wird für die Werte von x ermittelt, die sich von x1 nach x2 entwickeln, so daß,
somit wird der Abstand zwischen x und x0 bei jedem Durchlauf halbiert.

- **Speicherbelegung:**

M_0	M_1		
x	x_0	Gleichung	Programm
		39 15	14 .00

- **Programmlauf:**

© 2010 Joerg Woerner
Datamath Calculator Museum

Nach Eingabe von F(x) in den Programmspeicher:

x0 eingeben, [RST], [R/S] drücken
x1 eingeben, [R/S] drücken
Dieses Programm läuft unendlich. Um es anzuhalten, muß man [R/S] drücken.

- **Beispiel:**

Die Grenzen von f(x) = in Annäherung auf 1 ermitteln.

Taste	Anzeige	Bemerkungen
1 [RST] [R/S]	1	x0
2 [R/S]	2	x1
	0.6931472	f(x1)
	1.5	x2
	0.8109302	f(x2)
	1.25	x3
	0.8925742	f(x3)
u.s.w.		etc.

Bei der Beobachtung der Anzeige stellt man fest, daß mit Annäherung von x an 1, sich der Grenzpunkt f(x) auf 1 zubewegt.

HI-LO SPIEL

- **Programmprotokoll:**

Tasten	Anzeige	Schritt Nr.	Bemerkungen
2nd Part 2			
LRN 2nd CP	--	st	
STO 1	61.01	00	Grenzpunkt
R/S	13	01	Ausgangspunkt (x_1) holen
LBL 1	23.01	02	
STO 0	61.00	03	
2nd SBR 0	28.00	04	Rechenroutine f(x) aufrufen
2nd Pause	96	05	F(x) anzeigen
RCL 0	71.00	06	
+	85	07	
RCL 1	71.01	08	
=	95	09	
+	55	10	
2	02	11	
=	95	12	Nächsten Wert von x (x_n) ermitteln
2nd Pause	96	13	x anzeigen
GTO 1	22.01	14	Wiederholungsrücksprung
LBL 0	23.00	15	Unterprogramm Berechnung F(x)
ln x	32	16	
+	55	17	(kann bis zu Schritt 39
(	53	18	benutzen)
RCL 0	71.00	19	$f(x) = \frac{Lnx}{x - 1}$
−	75	20	
1	01	21	
)	54	22	
=	95	23	
INV 2nd SBR	−28	24	
LRN			

© 2010 Joerg Woerner
Datamath Calculator Museum

Abschnitt 8 — Hi-Lo Spiel

Wie oft müssen Sie raten, um die geheime Zahl zwischen 0 und 1000 zu finden? Nach jedem Versuch sagt der Rechner, ob Sie zu "hoch" oder zu "tief" geraten haben.

- **Programmprotokoll:**

Tasten	Anzeige	Schritt Nr.	Bemerkungen
2nd Part 2			
LRN 2nd CP	– –	St	
2nd CM	76	00	
+	85	01	
2nd π	49	02	
=	95	03	
y^x	45	04	
8	08	05	
=	95	06	
2nd Frac	59	07	Generiert eine Pseudozufallszahl y
X	65	08	
3	03	09	= INT (10³ x FRAC [(π + x)⁸])
INV log	– 31	10	
=	95	11	
2nd Intg	58	12	
x ⇄ t	51	13	x ist die eingegebene Zahl.
ON/C	15	14	
LBL 0	23.00	15	
=	95	16	
R/S	13	17	Versuch eingeben
STO 0	61.00	18	
1	01	19	
STO + 1	61.85.01	20	Anzahl der Versuche erhöhen
RCL 0	71.00	21	
2nd x=t	26	22	Gut geraten?
GTO 3	22.03	23	
INV 2nd x≥t	– 27	24	
GTO 2	22.02	25	nein - zu hoch
1	01	26	

©2010 Joerg Woerner
Datamath Calculator Museum

HI-LO SPIEL

Tasten	Anzeige	Schritt Nr.	Bemerkungen
GTO 0	22.00	27	
LBL 2	23.02	28	.nein, zu tief
1	01	29	
+/−	94	30	
GTO 0	22.00	31	
LBL 3	23.03	32	.ja, stimmt
RCL 0	71.00	33	Aufruf der richtigen Zahl
2nd Pause	96	34	
RCL 1	71.01	35	Aufruf der Zahl der Versuche
2nd Pause	96	36	
GTO 3	22.03	37	
LRN			

- **Um dieses Spiel zu spielen:**

- [•] drücken, dann die Uhrzeit und das Datum eingeben (oder irgendeine positive Zahl geringer als 1).

© 2010 Joerg Woerner
Datamath Calculator Museum

- [RST], [R/S] drücken, der Rechner zeigt "0", um zu sagen, daß er spielbereit ist.

- Jetzt geben Sie Ihren Versuch ein und drücken [RST]
  - wenn die Eingabe zu hoch ist, wird "1" angezeigt
  - wenn die Eingabe zu tief ist, wird "-1" angezeigt
  - wenn die Eingabe stimmt, zeigt der Rechner abwechselnd die richtige Zahl und die Zahl der Versuche, die Sie dazu gebraucht haben.

- Um nach Entdeckung der geheimen Zahl von Neuem zu spielen, muß man [R/S] und [RST] drücken und eine neue Anfangszahl eingeben.

Beispiel:

Taste	Anzeige
.222312 [RST]	0.222312
[R/S]	0
500 [R/S]	1
300 [R/S]	−1
400 [R/S]	−1
450 [R/S]	1
425 [R/S]	−1
434 [R/S]	434
	6

Anhang A: Fehlerbedingungen

Im Folgenden sind die Umstände verzeichnet, die in der Anzeige "ERROR" verursachen. In diesem Fall wird keine Eingabe vom Tastenfeld mehr akzeptiert, solange nicht auf [OFF] und [ON/C] gedrückt wird. Der Druck auf [ON/C] löscht die Fehlerbedingung und alle schwebenden Operationen. Wenn ein Fehler vorliegt, weist der Programmzeiger auf den Befehl , der den Fehler verursachte, ausgenommen bei Sprungfehlern, die den Programmzeiger zurück auf die Anfangsposition des Programmes bringen. Die Analyse des Fehlers zeigt Ihnen die geeigneten Korrekturmaßnahmen.

Abschnitt 1 - Allgemeine Fehlerbedingungen

1. Zahleneingabe oder Rechenergebnis (einschließlich Speicher) liegen außerhalb des Bereichs $\pm 1 \times 10^{-99}$ bis $\pm 9.9999999 \times 10^{99}$.
2. Multiplikation einer Zahl größer 1×10^{99} mit einer anderen Zahl kann zu einer Fehlerbedingung führen.
3. Division einer Zahl durch 0.
4. [log], [lnx], oder [1/x] von null, oder nullte Wurzel einer Zahl.
5. [log], [lnx], Potenz oder Wurzel einer negativen Zahl.
6. Arkussinus oder Arkuskosinus, wenn der Absolutwert der Zahl in der Anzeige größer als 1 ist.
7. Tangens von 90° oder 270°, π/2 Radiant oder 3π/2 Radiant, 100 oder 300 Gon, oder von den entsprechenden Vielfachen, zum Beispiel 450°.
8. Mehr als 15 offene Klammerebenen, oder mehr als 4 unvollständige Operationen gleichzeitig.
9. Fakultät einer Zahl, die nicht positiv oder größer oder gleich 70 ist.
10. Versuch bei der Speicheraufteilung 0, 8 oder 9 Datenspeicher zu reservieren.
11. Anwendung eines Arguments, das nicht im Bereich der logarithmischen und trigonometrischen Funktionen liegt (siehe "Informationen zur Genauigkeit").

©2010 Joerg Woerner
Datamath Calculator Museum

12. Druck auf [LRN], [SST], [R/S], obwohl kein Programmschritt vorhanden ist.
    (Rechneraufteilung mit 7 Datenspeichern).
13. Anwendung von [GTO] n oder [2nd] [SBR] n zu einem nicht vorhandenen Label n.
14. Anwendung eines Argumentes, das die gegebene Genauigkeit überschreitet. (Siehe Anhang B) für logarithmische und trigonometrische Funktionen.)

© 2010 Joerg Woerner
Datamath Calculator Museum

ANHANG B: Genauigkeitsangabe

Der gerundete Wert von π limitiert die Genauigkeit an diesen Punkten. Im allgemeinen nimmt außerhalb dieses Bereichs die Genauigkeit um eine Stelle je Dekade ab.

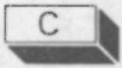
Die folgende Tabelle gibt die Grenzwerte in der Anzeige für die Berechnung bestimmter Funktionen an.

Funktion	Grenze
$\sin^{-1} x$, $\cos^{-1} x$	$-1 \leq \lvert x \rvert \leq 1$
$\ln x$, $\log x$	$1 \times 10^{-99} \leq x < 1 \times 10^{100}$
e^x	$-227.95592 \leq x \leq 230.25850$
10^x	$-99 \leq x < 100$
$x!$	$0 \leq x \leq 69$ wobei x eine ganze Zahl ist

© 2010 Joerg Woerner
Datamath Calculator Museum

Die folgende Tabelle gibt den Bereich der Ergebnisse von Arkusfunktionen an.

Arkusfunktion	Bereich des resultierenden Winkels
arc sin x	0 bis 90°, $\pi/2$ Radiant, oder 100 g
arc sin −x	0 bis −90°, $-\pi/2$ Radiant, oder −100 g
arc cos x	0 bis 90°, $\pi/2$ Radiant, oder 100 g
arc cos −x	90° bis 180°, $\pi/2$ Radiant bis π Radiant, oder 100 g bis 200 g
arc tan x	0 bis 90°, $\pi/2$ Radiant, oder 100 g
arc tan −x	0 bis −90°, $-\pi/2$ Radiant, oder − 100 g



ANHANG C - Tasten in numerischer Ordnung

00	0	53	(
01	1	54	)
02	2	55	+
03	3	56	2nd C.t
04	4	57.0n	2nd Fix n
05	5	58	2nd Intg
06	6	59	2nd Frac
07	7	61.0n	STO n
08	8	61.45.0n	STO y^x n
09	9	61.55.0n	STO + n
13	R/S	61.65.0n	STO × n
15	ON/C	61.75.0n	STO − n
21	RST	61.85.0n	STO + n
22.0n	GTO n	65	×
23.0n	LBL n	71.0n	RCL n
26	2nd x=t	75	−
27	2nd x≥t	76	2nd CM
28.0n	2nd SBR n	81.0n	EXC n
29	2nd D.ms	85	+
31	log	93	.
32	ln x	94	+/−
33	1/x	95	=
34	x^2	96	2nd Pause
35	$\sqrt{x}$		
40	2nd n!		
41	DRG		
42	sin		
43	cos		
44	tan		
45	y^x		
46	2nd DRG▸		
47	2nd P▸R		
48	2nd DMS.DD		
49	2nd π		
50	2nd |x|		
51	x ⇄ t		
52	EE		

© 2010 Joerg Woerner
Datamath Calculator Museum

ANHANG D: Zusammenfassung der Löschtasten

Gedrückte Taste	angezeigte Zahl	unvollständige Operationen	Fest-Komma	Datenspeicher	Programmspeicher	t-Register
ON/C	gelöscht	nein	nein	nein	nein	nein
ON/C ON/C	gelöscht	gelöscht	nein	nein	nein	nein
OFF ON/C	gelöscht	gelöscht	gelöscht	nein	nein	nein
2nd CM	nein	nein	nein	gelöscht	nein	nein
2nd CP	nein	nein	nein	nein	gelöscht	nein
2nd C.t	nein	nein	nein	nein	nein	gelöscht

© 2010 Joerg Woerner
Datamath Calculator Museum

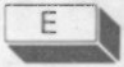
Im Falle von Schwierigkeiten

Wenn Schwierigkeiten auftreten, sollen Ihnen die folgenden Anweisungen bei der Analyse des Problems helfen. Sie können dann die Störung selbst beheben, ohne den Rechner an eine der Service-Abteilungen einzusenden.

Störung	Abhilfe
1. Der Rechner zeigt unrichtige Ergebnisse oder "Error"	Überprüfen Sie die Fehler- und Genauigkeitsbedingungen, sowie die Anweisungen für die Operationen, die Sie durchgeführt haben. Löschen Sie den Rechner wie folgt: ON/C, ON/C, 2nd Pmt 7 2nd CM 2nd Ct, RST ON/C.
2. Keine Anzeige, unsinnige Zahlen, Anzeige wird undeutlich	Die Batterien können entladen sein. Setzen Sie entsprechend den Batteriewechselanweisungen neue Batterien ein.

© 2010 Joerg Woerner
Datamath Calculator Museum

Sind die vorgeschlagenen Abhilfen unwirksam, wenden Sie sich schriftlich oder telephonisch an Ihren Verkäufer.

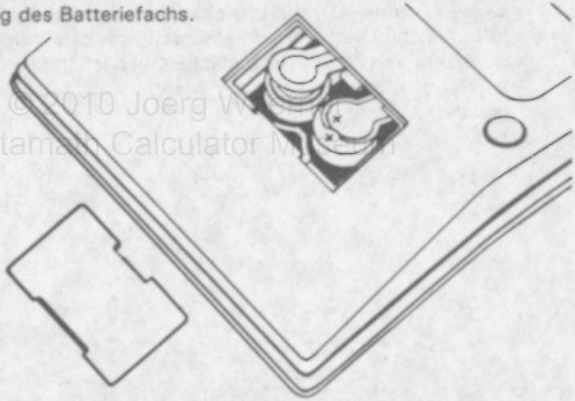


Batteriewechsel

Beachten Sie: Der Rechner verliert die in den Datenspeichern und Modus-Registern gespeicherten Daten, wenn die Batterien entladen sind oder entfernt werden.

Der Rechner benötigt für 750 Betriebsstunden jeweils zwei der folgenden Batterien: Panasonic LR-44, Ray-O-Vac RW-82, Union Carbide (Eveready) A-76, oder entsprechende Typen. Für Betriebszeiten bis zu 2000 Stunden verwenden Sie bitte Mallory 10114, Union Carbide (Eveready) 357, Panasonic WL-14, Toshiba G-13, Ray-O-Vac RW-42 oder entsprechende Typen.

1. Schalten Sie den Rechner aus. Öffnen Sie mit Hilfe eines kleinen Schraubenziehers, einer Büroklammer oder einem ähnlichen Hilfsmittel die Abdeckung des Batteriefachs.



© 2010 Joerg W
Datamath Calculator M

2. Entfernen Sie die entladenen Batterien und setzen Sie die neuen wie dargestellt ein. Berühren Sie die Folienkontakte während des Einsetzens der Batterien nicht; durch Hautfett könnten sie verschmiert werden. Achten Sie darauf, daß die Folienkontakte korrekt auf den Batterien nach dem Einsetzen aufliegen.
3. Setzen Sie die Batterieabdeckung zuerst oben ein, drücken Sie dann auf das obere Ende, bis sie mit einem Schnappen einrastet.
4. Drücken Sie [ON/C], [ON/C], [2nd] [Part] 7, [2nd] [CM] [2nd] [C.T] [RST] und [ON/C]; nun zeigt die Anzeige 0 und der Rechner ist betriebsbereit.

ACHTUNG: Werfen Sie die alten Batterien nicht ins Feuer.

Vorschläge

Weil von vielen Seiten Vorschläge mit alten und neuen Ideen herangetragen werden, kann Texas Instruments nur die Anregungen berücksichtigen, die unverbindlich und unentgeltlich zur Verfügung gestellt werden. Texas Instruments lehnt grundsätzlich den Empfang vertraulich zu behandelnder Vorschläge ab. Wenn Sie also Texas Instruments Ihre Anregungen vermitteln, oder eine selbst entwickelte Programmfolge zur Prüfung vorlegen wollen, fügen Sie bitte folgende Erklärung Ihrem Schreiben bei.

> "Alle hiermit übermittelten Informationen und/oder Unterlagen werden Texas Instruments auf nichtvertraulicher und unverbindlicher Basis zur Verfügung gestellt; mit dieser Vorlage werden keine Rechtsbeziehungen zu Texas Instruments, weder ausdrücklich noch stillschweigend, weder vertraulicher noch anderer Art, begründet. Texas Instruments kann entschädigungslos frei über diese Informationen verfügen, d.h., sie insbesondere urheberrechtlich schützen, verteilen, veröffentlichen, vervielfältigen oder anderweitig verwenden, ohne daß von mir irgendwelche Ausgleichsansprüche geltend gemacht werden".

© 2011 Joerg Woerner
Datamath Calculator Museum

EIN JAHR GEWÄHRLEISTUNG

Wenn das Gerät ausfällt oder beschädigt wird, wenden Sie sich bitte an Ihren Texas Instruments-Händler.

Texas Instruments gewährleistet nur dem Endverbraucher (Erstkäufer), daß dieser elektronische Rechner von Texas Instruments bei sachgemäßer Wartung und sachgemäßem Gebrauch für die Dauer von einem (1) Jahr ab Kaufdatum frei ist von Herstellungs- und Materialfehlern. Die Gewährleistung von Texas Instruments deckt keine Schäden ab, die durch ausgelaufene Batterien entstanden sind.

Der Gewährleistungsanspruch besteht nur, wenn : 1) der Rechner nicht durch das Auslaufen von Batterien einschließlich deren Lebensdauer, durch Unfall, unsachgemäße Behandlung, Nachlässigkeit, unsachgemäße Wartung oder andere Ursachen, die nicht auf Material- oder Herstellungsfehler zurückzuführen sind, beschädigt wurde ; 2) der Nachweis über das Kaufdatum vom Endverbraucher erbracht ist. FEHLT DIESER NACHWEIS, WIRD DER ELEKTRONISCHE RECHNER ZU DEN ZUR ZEIT DER REPARATUR GÜLTIGEN SERVICE-PREISEN REPARIERT.

Tritt während der Garantiezeit ein Fehler auf, so ist der Rechner unbedingt an den Texas Instruments-Händler zurückzugeben (bitte keine Direktsendung an Texas Instruments!). Er testet den Rechner und veranlaßt, daß das defekte Gerät nach Wahl von Texas Instruments kostenlos repariert oder durch einen nachgebesserten Rechner oder Teile jeweils entsprechender Qualität und Güte ersetzt wird. Bei berechtigten Gewährleistungsansprüchen erstattet Texas Instruments die Versandkosten.

Im Falle der Ersatzlieferung unterliegt der nachgebesserte Austauschrechner bis zum Ablauf der ursprünglichen Gewährleistungsfrist, mindestens jedoch für 90 Tage, den vorstehenden Gewährleistungsbedingungen.

Weitere Ansprüche, insbesondere Ansprüche auf Ersatz von Schäden, die nicht an dem Rechner selbst entstanden sind, sind ausgeschlossen. Es sei denn, Texas Instruments trifft der Vorwurf zurechenbaren vorsätzlich oder grob fahrlässigen Verhaltens.

Die Hersteller-Garantie besteht neben den gesetzlichen Gewährleistungsansprüchen des Endverbrauchers an seinen unmittelbaren Vertragspartner und berührt diese nicht.

VERZEICHNIS

Das folgendende Verzeichnis muß dem Tastenverzeichnis im Einbanddeckel zugefügt werden.

Absolutwert 24
Additionstaste 15
Algebraische Hierarchie 15
Algebraische Tasten 22
Algebraisches Operationssystem .. 15
Änderung eines Programm-
schrittes 47
Antilogarithmus-Tasten 23
Anwenderprogramme 41
Anwenderspeicher 30
Anzeige 12
Anzeigeformate 12
Anzeige Indikation 12
Arccosinustaste 26
Arcsinustaste 26
Arctangenstaste 26
Arithmetische Tasten 15
Aufruf von Unterroutinen 91
Austausch von x mit t 74
Ausführungsordnung der
Operationen 15
Automatische Stromsparschaltung 13

Batterieaustausch 125
Bedingte Sprünge 87

CALC Meldung 13
Cosinustaste 26

Dateneingabetasten 14
Datenspeicher 30
Datenspeichertaste 30
Dezimalpunkttaste 14

Editieren eines Programmes 47
Einzelschritt-Taste 39
Erstellen eines Programmes 41

Fakultätstaste 24
Fehlerbedingungen 119
Fehlermeldung 13
Festkommataste 21
Funktionsplotter 113

Ganzzahltaste 24
Garantie 127
Gonzeichen 12
GOTO Taste 56
Grad/Minuten/Sekunden in
Dezimal 27
Grad, Radiant, Gon 27
Grenze einer Funktion 115

HI LO Spiel 117
Hyperbelfunktionen 106

Im Falle von Schwierigkeiten 124
Informationen zur Genauigkeit 121
Inhaltsverzeichnis 3
Initialisierungstasten 1
INV-Indikator 12

Klammertasten 17
Kombinationen 104

Label Taste 56
Lauf/Stop-Taste 44
Laufzeichen 13
Logarithmustaste 23
LRN-Taste 39
Löschtaste 14

Minustaste 15
Multiplikationstaste 15

Nachkomma 121

Pausentaste 21
Permutationen 104
Pi-Taste 14
Plustaste 15
Polar/Rechtwinkel-Taste 20
Potenzierungstaste 22
Programmaufbereitung 41
Programmbeginn 41
Programmprüfung 40
Programmspeicher 36
Programmzähler 37
Programmspeicher-Löschtaste 39

Joerg Woerner
Datamath Calculator

VERZEICHNIS

Quadratische Gleichungen 110
Quadratwurzeltaste 22

Rad-Zeichen 12
Rechtwinkel/Polar-Taste..... 28
Reziproktaste 22
Rückschritt-Taste 40
Rücksprung von Unterroutinen..... 91

Schleifen bilden..... 69
Schrittweises Erstellen eines
Programmes 39
Sinus-Taste 26
Speicherarithmetik 31
Speicheraustauschtaste 32
Speicherbereichsverteilung 30
Speicherlöschtaste 32
Speicheroperation 31
Speicherprogramm 36
Speicherrückruftaste 30
Speicherteilung 38
Speicherteilungstasten..... 38
Sprungbefehle 87
Statistische Analyse 102
Subtraktionstaste 15

Tangente-Taste..... 26
Tastatur..... 11
Tasten in numerischer Reihenfolge 122
Tastenverzeichnis 1
Taste "Wissenschaftliche
Notierung" 19
t oder Testregister 74
"t"-Register Löschtaste 32
Trigonometrische Tasten 26

Umrechnungstasten Grad/Minuten/
Sekunden in Dezimalgraad Polar-
Rechwinkelige Koordinaten 28
Universelle Potenzierungs- und
Radizierungs- Tasten..... 22
Unterroutinen 91

Vermindern um 1 und Sprung bei
Null 87
Vermindern um 1 und Sprung bei
nicht Null..... 87
Vorzeichenaustausch..... 14

Wartungs- und Garantie-
Informationen 124
Winkel-Moduszeichen 12
Winkel-Wahltaste 27

x gleich t 74
x größer oder gleich t..... 74

Zifferntasten 14
Zusammenfassung der Lösch-
tasten.....123
Zweitfunktionstaste 12
Zweitfunktionszeichen 12

© 2010 Joerg Woerner
Datamath Calculator Museum

© 2010 Joerg Woerner
Datamath Calculator Museum

Texas Instruments behält sich das Recht vor, ohne vorherige Ankündigung, Änderungen des Rechners und der Gebrauchsanweisung durchzuführen.

© TEXAS INSTRUMENTS, 1982

D

© 2010 Joerg Woerner
Datamath Calculator Museum

TEXAS INSTRUMENTS

© TEXAS INSTRUMENTS, 1982

DNOm624C2582 BP

110 3851-0001